БЪЛГАРСКА АКАДЕМИЯ НА НАУКИТЕ

ИНСТИТУТ ПО ПАРАЛЕЛНА ОБРАБОТКА НА ИНФОРМАЦИЯТА

Галя Младенова Ангелова

Концептуални структури между изкуствения интелект и компютърната лингвистика

Дисертация за присъждане на научната степен
„доктор на математическите науки"
по специалността 01.01.12 „Информатика"

София, юни 2009

Концептуални структури между изкуствения интелект и компютърната лингвистика

Съдържание

УВОД

Представянето на знания е една от ключовите дисциплини на изкуствения интелект. Тъй като 'интелигентният' компютър извършва умозаключения чрез логически операции, необходимо е знанията за света да се кодират като предикатно-аргументни структури в системите на изкуствения интелект. Повечето езици за декларативно представяне на знанията изграждат концептуални описания над атомарни единици, съответстващи на *понятия* и *отношения/връзки* между тях. При превода към предикатно-аргументни структури понятията се превръщат в аргументи, а отношенията – в предикати. Затова понякога се казва, че фреймовете и семантичните мрежи са синтактични варианти на предикатното смятане от първи ред. Тази постановка съответства и на постулатите на логическото програмиране. Например, прологовите клаузи

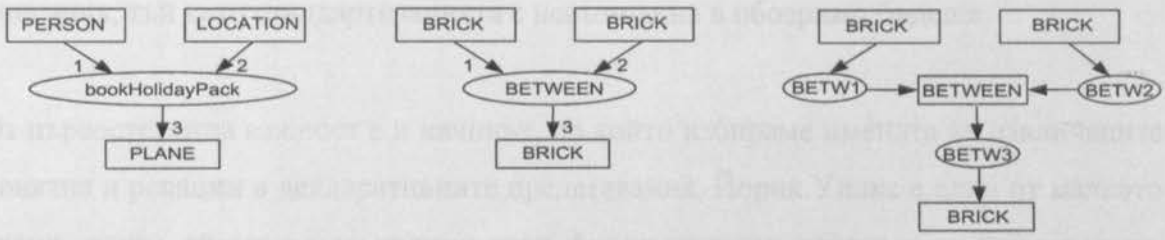
```
дете(петър,анна).
обича(петър,ирина).
ще_донесе(иван_асен,елена,гердан,средец)[1].
```

отразяват традиционното схващане, че при декларативно представяне физическите обекти се кодират като (имена на) аргументи, а разнообразните връзки между тях – като (имена на) предикати. Подходът се прилага и днес, тъй като езиците за декларативно представяне на знанията са основани върху идеите от 70-те години на миналия век. Продължават и усилията за ръчно и полу-автоматично извличане на концептуални ресурси от знание, в духа на аристотеловия метод за идентификация на типовете чрез техните надтипове (*genus*) и набора от уникални характеристики, които ги отличават от останалите типове в концептуалните йерархии (*differentiae*). Оказва се обаче, че има значителни трудности в процеса на извличането и декларативното представяне на езиково-независимо знание, и то по логически-непротиворечив начин. Тези трудности произтичат както от самия характер на знанието, така и от невъзможността то да се описва експлицитно без използване на естествения език като носител и посредник.

---

[1] Пример от книгата на И. Держански и И. Ненова '*Пролог за лингвисти*', София, 1997.

Джон Сова, бащата на концептуалните графи – един вид семантични мрежи, предложени през 1984 година в [Sow84] – въвежда в [Sow00] метафората *knowledge soup*, за да опише неструктурираното и динамично-променящо се съдържание на човешкото знание. Тъй като в този труд ще става дума за концептуални структури в смисъл на изкуствения интелект, преди всичко трябва да изясним какви са практиките за извличане на явни и формализирани описания на такива структури, декларирани в термините на *понятия* и *релации*. Понятията обозначават обектите, атрибутите, събитията, състоянията и действията в света, а *n*-мерните концептуални релации представят взаимо-връзките между понятията. Но дори и повърхностен преглед на литературата ни показва колко неопределени и неангажиращи са препоръките за вътрешно фрагментиране на концептуалните структури. Нека разгледаме Фигура I и показаните там три концептуални графа, които са изобразени като насочени двуделни графи с понятия-правоъгълници и релации-елипси. По принцип събитията често се представят като понятия, но няма когнитивни ограничения, които да възпрепятстват тяхното деклариране като релации с произволна размерност (Фиг. Ia). За всеки набор от обекти можем да дефинираме релация `BETWEEN` със съответната размерност, за да изразим факта, че един обект е между *n* други (Фиг. Iб). Но има и алтернативна възможност: отношението `BETWEEN` да се разглежда като понятие-атрибут, свързан с двумерни релации към обкръжаващите го понятия (Фиг. Iв). По този начин инженерите на знанията имат неограничени възможности да използват различни градивни елементи при експлицитно деклариране на знания за света. С други думи, съдържанието на семантичната супа на Сова (Фиг. II) няма предварително зададена грануларност или фрагментация на атомарни конструктивни елементи (например



а. `bookHolidayPack` събитие като тримерна концепт. релация [CrCo04]

б. Отношение `BETWEEN` като тримерна концептуална релация [Sow84]

в. Атрибут `BETWEEN` като понятие и три специални двумерни релации

Фигура I. Представяне на знания за света като семантични мрежи от понятия и релации

Фигура II. Семантичната супа [Sow00] и формати за декларативното й представяне

понятия/релации), както и 'вградена' вътрешна структура. Сегментацията на света на типове и индивиди е наложена от думите на езика и свързаните с тях семантични модели. Обикновено физическите обекти са наименовани със съществителни и затова ги виждаме представени в декларативните описания предимно като понятия. Останалите категории могат да се кодират по произволен начин в зависимост от целите на приложението. Има много алтернативни концептуални модели, които са вградени в различни софтуерни системи. В контекста на метафората за семантичната супа бихме казали, че супата може да се разлива в различни по форма съдове, както е показано на Фиг. II (освен това *изгребването* й не означава *изчерпване*, тъй като досега в декларативен вид е представен съвсем ограничен фрагмент от човешкото знание). Разнообразието на изразните средства затруднява неимоверно унификацията при представяне на знанията; дори усилията за съгласуване на най-горните класификационни типове остават безплодни в работни групи като SUO[2]. Налага се заключението, че трябва да се създават технологии за обработка на множество алтернативни концептуални описания, тъй като стандартизацията е невъзможна в обозримо бъдеще.

От първостепенна важност е и начинът, по който избираме имената на извличаните понятия и релации в декларативните представяния. Йорик Уилкс е един от малкото учени, които обсъждат от години този фундаментален аспект на представянето. Например в [Wi09] той твърди, че 'етикетите' на концептуалните структури се

[2] Standard Upper Ontology, вж. http://suo.ieee.org/, последно посещение на 13 април 2009.

пишат на някакъв идеализиран английски език, макар че претенцията винаги е да се постигне езиково-независимо логическо представяне. Употребата на думи за наименоване на концептуалните единици в декларативните представяния е неизкоренима практика просто защото човекът няма друга; убеждаваме се в това, като погледнем графите на Фиг. I, които са разбираеми единствено понеже имената на понятията и релациите указват цитираното значение. Но прикритото, неявно влагане на естествен език в декларативните описания внася в тях и свойствата на езика: неопределеност и многозначност. Например в един диалог на естествен език са възможни изреченията: *Колата на Петър е Форд. .... Форд е фирма за производство на коли*, но човек не би направил оттук заключението, че *колата на Петър е фирма за производство на коли*. Поради неопределеността на естествения език, едни и същи думи функционират като наименования на типове или индивиди, а хората лесно избират правилното значение в контекста на употреба. Обаче при декларативните представяния на концептуални структури няма контекст на употреба и е необходим систематичен подход за третиране на това явление (засега няма добри практики в тази насока). Друг проблем е многозначността и дори незнанието на английски език от чужденци, които наименоват понятията и релациите, без да знаят всички значения на съответните английски думи. Освен това езикът не може да остане 'в замразено състояние' дълго време; например след 50-100 години вътрешните етикети на много големи онтологии от типа на CyC, които са изградени чрез слепване на английски думи, ще носят друго значение на читателите. Според Уилкс, редно е специалистите по изкуствен интелект да осъзнават тези факти - без да се тревожат особено, тъй като реалността не може да се промени, да ги отчитат при разработката на приложни системи и да приемат последствията. Впрочем, и на естествен език хората разполагат с практически безкрайни възможности да разкажат някое събитие (подобно на описаната по-горе свобода за представяне при извличане на декларативни описания на концептуални структури) и този факт е още едно важно указание за неструктурирания характер на семантичната супа. Изглежда природата на декларативните (концептуални) описания е да съществуват в различни алтернативни варианти: изградени от различни по грануларност концептуални елементи, които са наименовани с 'ключови думи' на естествен език (най-често 'контролиран' английски).

При така очертаните проблеми, които опират до фундаментални когнитивни и философски въпроси, в каква посока следва да насочим своите усилия като информатици и изследователи на машинния интелект? Един възможен отговор е:

- да създаваме и изследваме нови технологии за обработка на концептуална информация, които отчитат нейната специфика. Според Уилкс натрупаният опит в областта на изкуствения интелект показва, че ефективните и работещи интелигентни системи рядко постигат своя успех благодарение на логически усъвършенствания във вътрешното представяне на знанията – т.е. технологиите за обработка и архитектурата също са много важни [Wi09];
- да се фокусираме върху концептуално моделиране на значими (но ограничени) проблемни области, където термините имат сравнително фиксирано и неизменно значение;
- да търсим ниши за прилагане на изследователските резултати по начин, който осигурява получаването на полезни софтуерни продукти.

От тук произтичат **целите и задачите**, които авторът си поставя:

- Изследване на проблема за ефективна обработка на концептуална информация и конструиране на алгоритми за бързо търсене;
- Изследване на концептуалната компонента на терминологични колекции в определени области и създаване на декларативни описания за представяне на знанията в тези области;
- Конструиране на научни прототипи и лабораторни версии на системи, базирани върху знания.

Настоящият труд представя резултатите на автора, свързани с изброените задачи, в една област на изкуствения интелект, която се намира на границата между компютърната лингвистика и декларативното представяне на знанията чрез концептуални структури. Глава 1 съдържа кратък обзор на състоянието на изследванията в областта. Като технологична иновация в Глава 2 е предложен двуфазов подход за семантично търсене на шаблони в концептуална информация чрез предварително кодиране на обобщения в минимален краен автомат с маркери на

заключителните състояния. Проведени са експерименти с тестови данни, които показват значителна степен на компресия на концептуалния архив, представен чрез краен автомат на предварителната фаза на обработка. Кодирането позволява извънредно бързо намиране на концептуални шаблони в декларативно-представено знание по време на изпълнение на конкретна потребителска заявка. В Глава 3 са описани резултати на автора при приложение на концептуалните структури в прототипи за автоматична обработка на естествения език. В Глава 4 са обсъдени достиженията при проектирането на концептуалния ресурс на средата за обучение STyLE, която е базирана върху концептуални графи. Глава 5 съдържа заключение и насоки за развитие на досегашните резултати. В авторската справка се изброяват приносите на автора. Даден е списък на 31 авторски публикации, които представят описаните в дисертацията резултати. В момента те са цитирани поне 95 пъти.

БЛАГОДАРНОСТИ

Задължена съм на многобройните си сътрудници през последните 15 години и тук бих искала да спомена някои от тях. На първо място, това са студенти магистърска програма от специалността 'Изкуствен интелект' на Факултета по математика и информатика на СУ 'Св. Кл. Охридски', които работиха по проектите DB-MAT и DBR-MAT през 1993-1998 година като дипломанти:

- Калина Бончева, сега в Университета на Шефийлд след докторантура там,
- Кристина Тутанова, сега в MicroSoft-Research, Сиатъл след докторантура в Станфордския университет;
- Светлана Дамянова (Светлана Хенсман), сега в Университета на Дъблин след докторантура там,
- Невелин Бойнов, днес в Unisys Corporation (Англия).

Друга група студенти и дипломанти разработи редица от решенията и модулите на средата за електронно обучение STyLE в проекта Ларфласт през 1999-2003:

- Ани Ненкова, сега в Университета на Пенсилвания след докторантура в Колумбийския университет;

- Преслав Наков, в момента пост-докторант в Националния университет на Сингапур, след докторантура в Университета на Калифорния в Бъркли;
- Павлин Добрев, технически мениджър на ПроСист-България след докторантура в ИПОИ-БАН;
- Огнян Калайджиев, главен експерт в Би-системс след докторантура в ИПОИ-БАН;
- Албена Струпчанска, понастоящем в Монреал - Канада, докторант на самостоятелна подготовка в ИПОИ-БАН.

Всички те допринесоха за реализацията на изброените проекти със стремежа си за навлизане в изследователската проблематика и със своя ентусиазиран и неуморен програмистки труд, който осигури изработката на качествени научни прототипи и демонстратори – а като следствие, и статии в престижни международни издания. (До голяма степен именно тези публикации помогнаха на много от изброените студенти да получат пълни докторантски стипендии в реномирани университети.)

Дължа специална благодарност на ст.н.с. д-р Стоян Михов за крайно стимулиращия обмен на идеи през последните 2-3 години, както и на ст.н.с. д-р Елена Паскалева и доц. д-р Светла Бойчева за дългогодишното ползотворно сътрудничество. Проф. д-р Валтер фон Хан от Хамбургския университет оказа неоценима помощ за израстването ми като специалист по компютърна лингвистика през 90-те години със стартирането на проектите DB(R)-MAT и с цялостната си подкрепа в многобройни съвместни начинания. Дружелюбната атмосфера в Секцията по лингвистично моделиране на ИПОИ беше съществен елемент от колегиалния контекст на сътрудничество, което е от извънредно голямо значение в една експериментална област.

На края, но не на последно място, съм благодарна на семейството си за помощта и разбирането. Всеотдайната подкрепа на родителите ми Йорданка и Младен Насвади ми позволи да се справя с многобройни затруднения, задължения и предизвикателства през изминалите години. За съжаление работата ми отне голяма част от времето, което бих искала да прекарвам със Съби и Деница.

ГЛАВА 1:

Обзор на съвременното състояние на изследванията в областта и въвеждащи понятия

Разработката на семантично-базирани приложения е приоритет на изкуствения интелект от самото му създаване. Напоследък интересът към тях се засилва, тъй като – с нарастването на хардуерния потенциал – възникват нови глобални системи от голямо значение (Семантичния интернет, дигиталните библиотеки, ресурсни банки за електронно обучение, електронната наука e-science и др.). Глобалните приложения предполагат обмен на данни чрез споделени онтологии, които представят знанията за предметната област. Тези онтологии се очертават като динамични структури, създадени от различни потребители, и интерфейсите на отделните приложения следва да осигуряват хармонизиране и съвместна функционалност на концептуалните ресурси. Затова през последните 2-3 години интензивно се изследват проблемите за обработка на много големи масиви от декларативно знание, включително и за скоростта на обработката. Най-активно се работи по този въпрос в областта на Семантичния интернет; дефинирани са езиците RDF и OWL, които осигуряват съхранение, обновяване, формулиране на заявки за търсене и процедури за извод над големи архиви, кодиращи знание за света в декларативен вид. Целта е да се намерят адекватни техники за ефективна обработка на извънредно големи концептуални ресурси, с милиарди екземпляри на различни онтологии. Също така е важно да се идентифицират 'добри примери' за ефективна обработка, които да се използват като база за сравнение на новопоявяващите се алгоритми и приложения (т.нар. benchmarking). Изработката на пилотни експериментални тестови множества данни е от първостепенно значение.

Така на преден план излиза и друг основен (постоянно появяващ се) въпрос: *откъде всъщност да се вземе знанието, представено в декларативен вид*, което ще осигури успеха на възникващите глобални приложения и в крайна сметка ще осмисли многогодишните усилия на изкуствения интелект. Трудно и скъпо е да го

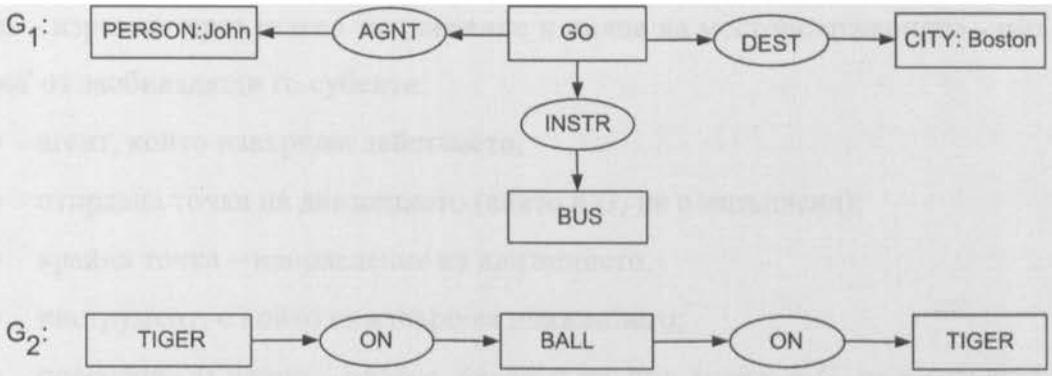
извличаме ръчно – това е установено опитно след дългогодишна практика. Търси се начин знанието да бъде извлечено автоматично от документи на естествен език, чрез интелигентни програми за разбиране на текста. Настоящият труд представя постиженията на автора в областта на извличане и обработка на масиви от декларативно-представени знания, а също и приложни резултати, получени при създаване на прототипи в изкуствения интелект и компютърната лингвистика.

1.1. Концептуални графи

Концептуалните Графи (КГ) са вид семантични мрежи, предложени в [Sow84] като език за представяне на знанията, основан едновременно върху логиката и теорията на графите. С течение на времето много изследователи са допринесли за развитието и теоретичното изясняване на формалните постановки на езика; например фундаменталното понятие 'опора' е дефинирано в [ChMu92], а прецизирането на алгоритъма за превод на КГ във формули на предикатно смятане от първи ред е направено в [Wer95]. КГ са предложени като надстройка над екзистенциалните графи на Ч. Пърс и Сова изтъква ролята им като графичен интерфейс към логиката [Sow08]. От семантична гледна точка, всеки КГ е твърдение (факт) в модела на предметната област и кодира знание за света. Зад – или под – всеки КГ стои един свързан, краен, двуделен (мулти)граф с два вида върхове:

- *c-върхове* или типове понятия, кодиращи нещата, обектите, атрибутите, събитията и състоянията в света, и
- *r-върхове* или типове концептуални релации, кодиращи *n*-мерните отношения между *c-върховете,* които показват как понятията са свързани помежду си.

Всеки връх е наименован с името на съответен тип от предметната област. Двуделният граф визуализира връзките между понятията, релациите и екземплярите в твърдението за предметната област. Два примерни графа са дадени на Фиг. 1.1. Типовете PERSON, GO, CITY, BUS, TIGER и BALL са понятия; AGeNT, DESTination, INSTrument и ON са релации, а John и Boston са екземпляри.



Фигура 1.1. Концептуални графи от понятия и релации, кодиращи знания за света

Етикетът PERSON:John в G_1 означава, че John е екземпляр на понятието PERSON; съответно Boston е екземпляр на CITY. Елементите на графите се конструират по начин, силно повлиян от изказвания на естествен език. Сова твърди, че семантичните мрежи са разбираеми както за хората, така и за компютрите – хората разбират визуалното представяне, а формалният вътрешен запис позволява логическа обработка на твърденията [Sow91]. Поради това можем да изкажем двата графа на български език като две твърдения:

G_1: *Джон отива в Бостън с автобус* и

G_2: *Съществува един тигър, който е върху една топка, която е върху (друг) тигър.*

При изписване на графи с малко върхове често се използва следната нотация: всеки правоъгълник е заменен с квадратни скоби, всяка концептуална релация – с кръгли скоби, а отместването показва към кое понятие са свързани релациите, които по необходимост са пренесени на нов ред. Например G_1 се записва като:

[PERSON:John] ← (AGNT) ← [GO] → (DEST) → [CITY:Boston]
→ (INSTR) → [BUS]

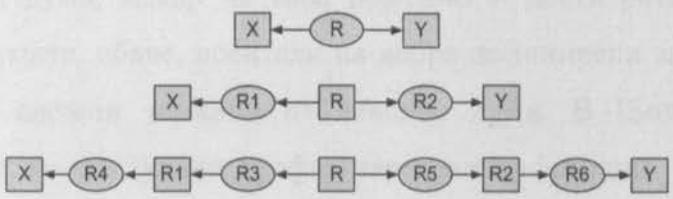
За нас са важни принципите, приети при извличане и наименоване на концептуалните релации. Например и трите релации в G_1 от Фиг. 1.1 не присъстват в явен вид като думи от изречението, чрез което изказваме твърдението на български или английски език. Но тези три релации са свързани със събитието GO,

което – изразено чрез глагол за движение и смяна на местоположението - има своя 'рамка' от заобикалящи го субекти:

- агент, който извършва действието;
- отправна точка на движението (която в G_1 не е запълнена);
- крайна точка – направление на движението;
- инструмент, с който се извършва движението;
- пояснение за начин – *удобно*, *бързо*, и др. под. (което в G_1 не е запълнено);
- пояснение за време – напр. *днес* и др. под. (което в G_1 не е запълнено).

Съгласно общоприетите лингвистични теории, глаголът - изразяващ събитие или състояние в изречение на естествен език - има свои 'валенции', които се запълват с думите на изречението и така всъщност се формира самото изречение. Ще наричаме тези 'валенции' *тематични роли* и ще предполагаме, че всяко събитие-глагол е зададено заедно със списъка на своите явно-наименовани тематични роли. Имената на тези роли (в лингвистичната литература не повече от 30-40 на брой, включително показаните на Фиг. 1.1 AGNT, DEST и INSTR) са отправна точка за дефиниране на имена на концептуални релации в семантичните мрежи.

Тук е редно да отбележим, че в големи онтологии като CyC има хиляди ръчно-извлечени релации, например над 100 вида различни под-релации на AGENT [Leh96]. Както е показано на Фиг.1.2, няма принципни ограничения за дълбочината и грануларността на концептуалните релации (още повече, че много от тях са неявни). Отношенията могат да се проследяват до произволно ниво на детайлност, но хората не са свикнали да го правят. Затова в повечето концептуални ресурси се използват релации от семейството на традиционните тематичните роли, релации-предлози или релации наименовани с етикети като: *has-part*, *has-property* и т.н.



Фигура 1.2. Нива на детайлност при деклариране на концептуални релации [Leh96]

Размерността на релациите е друг съществен въпрос. В [Sow84] релациите са *n*-мерни; по дефиниция те имат по *n-1* наименовани входни аргумента и по един изходен. Релации с произволна размерност са възприети също в дескриптивните логики, където се правят изводи над *n*-мерни предикати. Знаем освен това от базите данни, че съществуват таблици от *n* стълба, представящи *n* взимно-обусловени признака, които е невъзможно да се разделят по двойки. Но в практическия подход за семантично инженерство, предложен от създателите на съвременния Семантичен интернет, се приема, че *n*-мерните релации могат да се представят чрез двумерни [W3C06]. Затова езикът RDF се гради над тройки, т.е. всяко твърдение в RDF е множество от тройки <subject, predicate, object>, а езикът OWL ползва синтаксиса на RDF [W3C-Lang]. Унификацията на размерността (*one size fits all*) позволява сравняване на предикати в две различни бази от знания, тъй като всички отношения са двумерни. Дискусията относно размерността на релациите продължава и досега, например [Zar09] мотивира необходимостта от използване на *n*-мерни релации чрез примерни изречения на естествен език. От друга страна, подходите на компютърната лингвистика за формално превеждане на изречението до логическа формула обикновено се стремят да представят връзките между глагола и обектите, запълващи *n*-те му тематични роли, като тройки от вида <глагол, тематична-роля, обект>. По този начин на преден план излизат имената на семантичните отношения и става възможно тяхното сравняване в две съседни изречения [Allen95]. Така на практика *n*-мерните предикати-глаголи се описват чрез двумерни предикати, съответстващи на тематичните роли на глаголите. В този труд ще използваме двумерни концептуални релации.

Както се вижда от Фиг. 1.1, на обектите-съществителни съответства по едно понятие. Интуитивно, естествената грануларност е да се съпостави по едно понятие на всяка значеща дума, макар че това решение е 'доста разточително' според [Hob85]. Освен думите, обаче, носители на добре дефинирани значения са редица фрази, например сложни термини от няколко думи. В [Sow84] са въведени специални видове концептуални графи, наречени дефиниции на типове, които

позволяват задаването на типове с по-голяма грануларност, например тип CIRCUS-ELEPHANT както е показано по-долу:

type CIRCUS-ELEPHANT (*x*) **is**
[ELEPHANT: *x] ← (AGNT) ← [PERFORM] → (LOC) → [CIRCUS].

След въвеждане на дефиниции на типовете става възможно да се прилагат операции за *разширяване на сложен тип* (чрез заместването на типа с тялото на дефиницията) или за *свиване на сложен тип* (чрез заместването на тялото на дефиницията с етикета на типа).

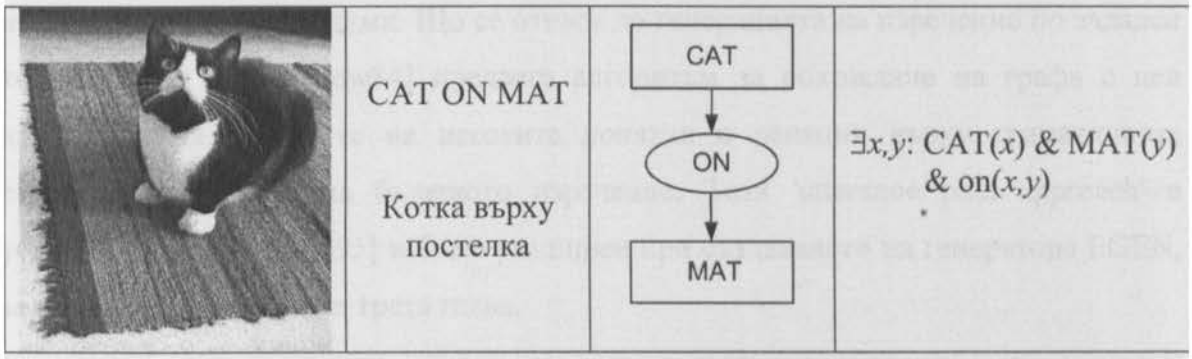
Екземплярите John и Boston на Фиг. 1.1 са показани като *референти* на съответните понятия PERSON и CITY. В книгата [Sow84] са въведени различни видове референти: множества, квантори, мерки и т.н. В този труд ние ще използваме като референти на дадено понятие два вида множества:

- от явно-изброени екземпляри – например, [PERSON:{John, Mary}], който описва определените индивиди John и Mary от предметната област, и
- от неспецифицирани екземпляри – например, [OIL_PARTICLES:{*}], който обозначава много неопределени екземпляри и се изписва още като [OIL_PARTICLES:*] и [OIL_PARTICLES:plural].

Концептуалните графи се развиват 25 години и съдържат различни конструкции, с помощта на които се моделират контексти, логически оператори, линии за идентичност (указващи еднаквост на екземплярите на различни типове) и т.н. Днес КГ са фамилия от езици за представяне на знанията. Някои учени ги изследват като теоретичен формализъм, а други ги прилагат за най-различни цели: описание на софтуерни спецификации, обработка на естествен език, извличане на информация и добиване на знания, правене на изводи и т.н. По този повод Фритьоф Дау казва, че 'Концептуалните графи не са фиксирани, а са open-minded. ... Поради разнообразието на диалектите и сложността им, не е възможно и може би дори не е желателно да разглеждаме цялата система като графичен подход към формалната логика' [Dau09]. В семейството на КГ е отделено едно добре-дефинирано ядро,

наречено Common Logic, което наскоро беше прието като ISO-стандарт за представяне на знанията [CL-ISO07].

Една от обективните причини за относителното дълголетие на концептуалните графи, в сравнение с някои други формализми за представяне на знанията, е възможността за визуализация. Идеята за показване на концептуална информация пред крайния потребител като някаква диаграма е толкова устойчива, че и днес наблюдаваме развитие на модерни софтуерни среди, които изобразяват например RDF-графи пред потребители неспециалисти. Визуализацията се разглежда като предимство при търсене [Hea09] и е желано качество на дружелюбните интерфейси. Но графичното представяне е неразделна част от концептуалните графи още от 1984 г. В [Sow84] се твърди, че измежду различните начини за предаване на някаква информация: картинка, изречение на естествен език,



Фигура 1.3. Различни начини за изказване на факт от реалността

концептуален граф или формула, най-удобен за компютърна и човешка обработка е концептуалният граф (вж. Фиг. 1.3). Съществена част от информацията в графа е интуитивно-разбираема за човека, а КГ лесно се превежда в предикатно-аргументна структура и може да се обработи алгоритмично. Това твърдение се потвърждава и от наблюдения на реакциите на обикновени хора, на които се показват концептуални графи. Изследване в тази насока е направено при подготовката на дисертационния труд на В. Димитрова [Dim01]. Оказва се, че малки по обем концептуални графи в графичен формат са разбираеми след кратък инструктаж за значението на стрелките и посоката на концептуалните релации.

Това превръща графичният формат в алтернатива за създаване на интерфейси на системи, които трябва да съобщят някой динамично-получен факт на потребителя (друга възможност е генерацията на естествен език, но тя не е елементарна).

През 90-те години на миналия век концептуалните графи са използвани активно за анализ и генерация на естествен език. Обикновено се анализира само по едно изречение или по даден граф се генерира също едно изречение. Една от най-ранните статии за анализ е [FLDC86]; в нея чрез операцията join (съединение) се обединяват семантичните примитиви на отделните думи на входното изречение и така се елиминират невъзможните интерпретации. На абстрактно ниво, операцията съединение се държи като композицията в днешните логически граматики (вж. част 1.2.1). Друга много успешна система за анализ на естествен език е DANTE [VPG88], която разполага с голям речник от 'канонични графи' и разбира изречения над речник от стотици думи. Що се отнася до генерацията на изречение по зададен концептуален граф, [Sow84] предлага алгоритъм за обхождане на графа с цел налагането на етикетите на неговите понятия и релации върху граматичната структура и думите на бъдещото изречение. Този 'utterance path approach' е усъвършенстван в [NMR95] и беше разширен при създаването на генератора EGEN, за който ще стане дума в трета глава.

Една от целите на настоящия труд е да изследва задачата за ефективна обработка на концептуална информация. Търсенето в света на концептуалните графи се извършва по два начина:

- чрез представяне на твърденията във вид на логически формули - при което обикновено се работи с процедури за извод на Пролог, или
- чрез подходи от теорията на графите - при което алгоритмите за търсене на подграфи се адаптират към задачата за търсене на концептуални (под)графи.

Намирането на концептуален подграф е NP-пълна задача в общия случай [Sow84, MuCh92, Mug95]. Във втора глава ние ще предложим алтернативно решение чрез алгоритъм за двуфазово изчисление, при който се избягват NP-пълните пресмятания по време на изпълнение на заявката. Там ще дефинираме формално и

разглеждания от нас клас КГ – т. нар. прости концептуални графи, които на семантично ниво се покриват с RDF-графите [Bag05] и съответстват на положителните, екзистенциално-квантувани конюнктивни формули с двоични предикати.

В трета глава ще разгледаме подходи за извличане на декларативни модели на знания чрез концептуални графи и ще предложим алгоритми за автоматично добиване на концептуални графи от текст. В четвърта глава ще се спрем на концептуалното моделиране в терминологични колекции чрез концептуални графи. Получените резултати се основават върху изложените тук постановки за избор на грануларност на онтологичните примитиви – понятията съответстват на думи или фрази на естествен език, а релациите – на семантични отношения между тях.

В този дисертационен труд ще различаваме следните видове ресурси от декларативно-представени концептуални описания:

- ***Таксономията*** е класификационна схема на типове неща или понятия, организирана като дърво или решетка. При конструиране на класификацията *тип-подтипове* обикновено се предполага, че подтипът има същите характеристики като своя надтип, както и допълнителни свойства;
- Под ***база от знания*** ще разбираме формален модел на знанията за света, в който се описват множество от понятия, техни свойства и връзки между тях. Съществуват различни екземпляри на понятията, както и класове от понятия. Задават се логически изрази, които определят условията едно твърдение за обектите да се приема за истина в разглеждания свят. Множество от аксиоми фиксира свойствата на понятията и отношенията между тях. Освен декларативно-представеното знание, базата обикновено включва и правила за извод (често от вида if-then-else), които дефинират процедури за правене на умозаключения. Въвеждат се и операции за извличане, тъй като в базата от знания – по аналогия с базата от данни – се предлага по-цялостно решение за управление за концептуалния ресурс;

- ***Онтология*** в информатиката е 'формална и явна спецификация на споделена концептуализация' [Gru93]. Онтологията е фокусирана върху декларативното описание на определена предметна област. Използват се т. нар. ***плитки онтологии*** (shallow, light-weight ontologies), в които се декларират само формализираната таксономия на понятията и техни основни свойства. Терминът ***дълбока онтология*** обозначава модел на света, който по своята детайлност и степен на формализация съответства до голяма степен на концептуалния модел в база от знания;
- ***Фолксономията*** е динамична таксономия от свободно-избрани ключови думи, която се създава спонтанно от експертите и потребителите на колаборативни среди и социални системи. Ключовите думи се натрупват от участниците в социалната група с цел анотиране, категоризиране и управление на дигитални обекти. Терминът се появява през 2004 [VW07];
- ***Номенклатурата*** (***класификационна система*** или ***терминологично-базирана онтология***) е таксономия на обекти в дадена предметна област, създадена и поддържана от специализирана институция. Такива са йерархиите на *International Classification of Diseases* ICD (поддържана от Световната здравна организация, вж. българския вариант [МКБ-10]) или *PRODCOM* (унифициран списък на продуктите, произвеждани в Европа, поддържан от Евростат за статистически цели – вж. българския вариант [НКПИД03]). Обикновено номенклатурите съдържат буквено-цифров идентификатор и изчерпателно текстово описание на всяко понятие.

1.2. Автоматична обработка на естествен език в компютърната лингвистика

Ще разгледаме накратко методите за анализ и генерация с използване на правила, които са от значение за проследяване на изложението в следващите глави.

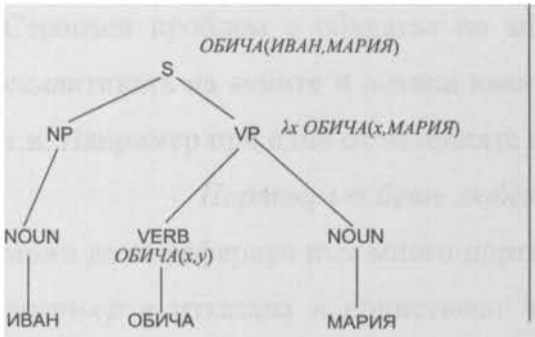
1.2.1. Разбиране и генерация на естествен език

Фиг. 1.4 илюстрира процеса на анализа - ниво по ниво се конструират вътрешни представяния и структури от граматически категории, в чиито термини да се постигне пълно разпознаване на входните единици. Лингвистичните данни се описват в специални езикови ресурси. Разбирането на текста започва със стартирането на програма - *морфологичен анализатор*, която сравнява въведените низове с предварително подготвен речник и разпознава отделните *словоформи* на думите, споменати в конкретния входен текст. Създаването на морфологичен речник е задължителна стъпка при флективни езици с много словоформи за всяка дума [Пас07, БаЛинк]. Морфологичният анализ се извършва в термините на грама-

Фигура 1.4. Обработка на естествения език при подходи, използващи правила [Анг08а]

тични характеристики, които се присвояват на отделните думи в текста. След разпознаване на основната форма на думите (т.нар. леми) започва анализ на ниво изречение, при който се строи модел на структурата на изречението във вид на дърво. Широко известна процедура за синтактичен анализ е разборът с безконтекстни правила в конституентните граматики на Чомски, които позволяват анализ в реално време поради сравнително простия алгоритъм за проверка дали дадено изречение е граматически правилно (тоест изводимо от правилата на граматиката). В процеса на проверката се композира дървото на извода за всяко изречение. Предимство на композиционния подход е, че при съставянето на синтактичните конституенти може да се формира по унифициран начин и една предикатно-аргументна структура, която се нарича *'логическа форма'* на изречението. За нас семантична интерпретация е от особено значение и поради това ще се спрем по-подробно на композицията на логически изрази в най-простите случаи на съществително, прилагателно и глагол.

Всяко прилагателно или съществително от входния текст се превръща в едноместен предикат, наименован със самата дума. Например, ако в едно изречение се срещне думата *ЧОВЕК*, програмата за композиране на логическа форма ще произведе $\exists x$ *ЧОВЕК*(x) и ще разглежда получения логически израз като семантика на думата *ЧОВЕК*. При срещане на *УМЕН ЧОВЕК*, с просто правило за конюнкция се получава $\exists x$ *ЧОВЕК*(x) & *УМЕН*(x). Глаголите се превръщат в n-местни предикати, като n е броят на задължителните за запълване семантични валенции на глагола. Например, семантиката на глагола *обичам* в изречението *Иван обича Мария* се изразява чрез двуместния предикат *ОБИЧА*(x,y), където x е агентът, а y - обектът. Констуирането на логическа форма на това изречение е илюстрирано на Фиг. 1.5а чрез дървото на синтактичния анализ. Двуместният предикат *ОБИЧА*(x,y) е частично запълнен във върха VP (Verb Phrase), понеже изречението е анализирано само отчасти, и се е превърнал в λ-израз, който 'чака' да бъде вложен в по-висока синтактична конституента с интегриране на подходящ втори аргумент. Преминаването от думи към (вътрешни за системата) логически форми позволява да се контролира логическата коректност на входните изречения. Например,



Фигура 1.5а. Композиране на логическа форма

Фигура 1.5б. Затворен свят с аксиоми, дефиниращи допустими композиции от думи

изречението *Зелените идеи яростно спят*[3] е недопустима комбинация от думи в света, показан на Фиг. 1.5б. В него само материалните обекти имат цвят – а 'идея' е абстрактен обект, и 'яростни' могат да бъдат само процесите – а 'спя' е състояние. Така, следвайки нивата от Фиг. 1.4, алгоритмите за анализ тръгват от неструктурирания входен текст и произвеждат множества от логически форми, които се обработват с техниките на изкуствения интелект. Този процес, наречен семантичен анализ, само по себе си не е елементарен на практика. Освен това преходът към предикатно-аргументно представяне не решава задачата за разбиране на естествения език, тъй като няма големи, публично-достъпни концептуални ресурси от декларативно знание като показаните на Фиг. 1.5б, с чиято помощ да се извършват умозаключения (а и техниките за извод също имат своите ограничения).

На 'по-дълбоките' езикови нива от Фиг. 1.4 се сблъскваме с лингвистични явления, които не са добре изучени и няма психолингвистични и когнитивни теории за тяхното цялостно обяснение и моделиране. Такова явление е прагматиката, която изучава значенията и тяхното функциониране в зависимост от контекста. Няма теория за компютърно моделиране на контекста по начин, който да осигури алгоритмичното разпознаване на прагматичните отсенки в смисъла на изреченията. С други думи, техниките за умозаключения на изкуствения интелект не разрешават прагматичната многозначност, дори да имахме бази знания от милиарди факти.

[3] Известен пример на Чомски за безсмислено изречение с коректна синтактична структура.

Сериозен проблем е обхватът на значенията на думите, който се преплита със семантиката на явните и неявни квантори, отрицанието, с темпоралните наречия и т.н. Например при едно от четенията на многозначното изречение

Портиерът беше любезен във всеки хотел (1)

може да се реферира към много портиери, по един във всеки хотел, но самата дума *портиер* е изказана в единствено число и е извън синтактичната фраза '*всеки хотел*'. Тогава системата трябва да изчисли от знанието за света, че всеки хотел си има портиер. Поради това кванторът за съществуване на променливата, свързана с думата *портиер*, трябва да се вложи в обхвата на квантора за всеобщност на променливата, свързана с думата *хотел*. Естествено тези фини логически трансформации са невъзможни на практика в голям мащаб. Всъщност автоматичното композиране на логически форми произвежда предикатите, свързани с конкретните думи и фрази от изречението, но остава много работа по прецизно наместване на скобите и разрешаване на обхвата на значенията. Прототипите за разбиране на естествен език генерират всички възможни форми, измежду които най-често се избира една в зависимост от потенциала на конкретната система. Този етап е онагледен с трион на Фиг. 1.4.

Алгоритмично-нерешен проблем е и автоматичната обработка на референцията. По принцип естественият език функционира като последователност от линейно-наредени клаузи, тъй като не можем да изкажем всичко наведнъж. Една система за разбиране на естествения език трябва да може да разпознава различните референции към един и същи обект, които обикновено се изказват по различен начин с цел избягване на повторенията и добавяне на нова информация. На теория, системата трябва да разпознае всички референции, за да се постигне разбиране на естествения език от компютъра. На практика обаче това е невъзможно и се работи само за местоименната референция, чиято обработка е задължителна при машинния превод (понеже при превод на местоименията в единствено число често е необходимо да се смени рода – например, английското *it* за неодушевен обект може да се преведе на български като *той*, *тя* или *то*). Най-добрите алгоритми за разрешаване на местоименната референция на английски език работят с успех под

80%. Нека отбележим, че човекът няма проблеми при разбиране и генериране на естествен език, че прави контекстна интерпретация без съзнателни усилия и т.н. Фактически човекът е най-добър на нивата, с които компютърът не се справя. Но повечето хора не си дават сметка за несъзнателно използваните от тях морфологични и синтактични категории, което пък е компютърният подход за анализ на думите и изреченията. Поради голяма сложност на разбирането на естествения език, практически се използва подходът Information Extraction [Cun99], при който системата разбира само по едно събитие в текста, например описание на терористични актове в полицейски доклади[4]. Системите първо разпознават наименованите единици, тъй като имената на лица, географски обекти, фирми и т.н. са важни указатели за описания на случки от даден вид. Постига се точност над 95% за английския език. След това се анализират само изреченията с *'важни думи'*, сигнализиращи търсеното събитие. Автоматично се разпознават половината кореференции, а точността на разрешените е около 70%. След това се конструира т.нар. сценарий за намереното събитие (template). Разпознават се до 70-80% от текстовите фрагменти, в които се говори за търсеното събитие, а човек постига точност 93%. Автоматичното запълване на сценария, еквивалентно на семантичен анализ, се извършва с точност до 56%. Хората извършват тази задача с точност 81%.

На Фиг. 1.4 е показано, че генерацията на естествен език работи в посока, противоположна на разбирането (отдолу нагоре). Принципен проблем при генерацията е да се построи план на текста, т.е. да се реши в каква последователност ще се изкажат подбраните фрагменти от знания. Но построяването на дискурсен модел не е лесно поради (неявните) връзки, съществуващи между клаузите на кохерентния дискурс [Анг08б]. Нека анализираме наредбата на клаузите в следния текст:

> *(2.1) Статията разглежда клас мобилни роботи, оборудвани със сензорни системи за перцепция, навигация и управление. (2.2) Наречени условно емоционални, тези роботи имитират в известен смисъл емоционално поведение. (2.3) Е-моционален се интерпретира като електронно-управляван и същевременно притежаващ реакции на комплекс от сензорни стимули.*[5]

---

[4] На английски език има два термина, които често се смесват при превод: *Information Retrieval*, т.е. изваждане на цели документи от архив или интернет, нещо като *търсене на документи*; и *Information Extraction*, с превод *извличане на информация*.

[5] Лаков, Д. *Мобилен сензорен робот*. Списание „Автоматика и Информатика", 4/2007, стр. 16-19.

Изречение (2.1) въвежда темата – специални *'мобилни роботи'*. Изречение (2.2) развива (2.1), като разширява темата с допълнителни сведения. Изречение (2.3) дефинира понятие, въведено в (2.2). Виждаме сложната референция между обектите, но освен нея текстът изразява и връзки между клаузите. Една илюстрация на неявните *риторични връзки* между трите изречения е дадена на Фиг. 1.6. Използваната днес теория за риторичната структура [MaTh88] изброява около 30 дискурсни релации, които моделират отношенията между клаузите в текста. Все още не са предложени алгоритмични модели за автоматично разпознаване на тези връзки, освен за най-прости случаи в ограничена предметна

Фигура 1.6. Риторични връзки между изречения в примерен свързан текст

област. Така че областта на генерацията е в процес на развитие. Нека отбележим, че човекът-читател анализира естествения език несъзнателно и има 'вграден' механизъм за разпознаване на структурните отношения между фрагментите на текста и на референциите.

Друг интересен лингвистичен феномен е начинът за смяна на обекта-фокус или темата, за която се говори в по-дълъг дискурс. Изучаването на добре построени текстове показва, че е за предпочитане първо да се изчерпи дадена тема и тогава да се мине към следваща. Нека разгледаме следните примерни изречения:

(3.1) Иван е първокурсник в Техническия университет в София.
(3.2) Първокурсниците обикновено имат труден първи семестър.
(3.3) Иван има лекции по програмиране, графика и хардуер.
(3.4) Най-интересен е курсът по графика.
(3.5) Иван живее в Студентския град.

Изречение (3.1) въвежда темата (глобален фокус) на целия параграф – студентът Иван. Изречение (3.2) съдържа обобщаващо твърдение за първокурсниците и естествено следва (3.1), понеже именно в (3.1) 'първокурсник' е потенциален фокус. След това се преминава към подтемата 'учебна програма' и (3.3) и (3.4), при

което водещо е (3.3) за глобалния фокус Иван. В (3.5) се преминава към следващата подтема – жилище и т.н. Така фокусът на отделните клаузи се мести към нови подтеми; след въвеждащо изречение идва доразвиване на темата. Наблюдения от този род подсказват алгоритми за планиране на наредбата на клаузите.

Автоматичната генерация на естествен език цели произвеждане на кохерентен дискурс по дадено знание във вътрешно представяне. Системата за генерация решава редица нетривиални задачи:

- подбира знанията, които са релевантни като отговор на заявката на потребителя (content selection),
- подрежда фактите един след друг в кохерентни параграфи (text planning),
- избира от речник думите, с които да се разкажат фактите (lexical choice), и
- оформя синтактичната и морфологична структура на изреченията (surface verbalisation), като генерира и (местоименни) референции между клаузите.

Компютърната генерация е силно ограничена поради голямата сложност на задачата и липсата на цялостни модели за функционирането на естествения език, които да отразяват употребите в контекта на комуникация. На практика в софтуерните системи се развиват сценарии за генериране чрез предварително зададени фиксирани планове от дискурсни релации, които управляват подредбата на клаузите в бъдещия текст.

Такива планове се конструират след наблюдения на документи със специфичен стил. Много текстове са създадени по характерен дискурсен шаблон, например[6]:

> РОБОТ: Автоматизиран програмно управляем манипулатор с възможност за сложни пространствени движения; частично или пълно може да изпълнява някои функции на човека при взаимодействие с околната среда. С помощта на датчици възприема сигнали от околната среда и въз основа на тях чрез изпълнителни механизми извършва сложни работни операции. Роботът има висока ефективност и може да работи във вредни условия и в труднодостъпни за човек места.
>
> ДИСТАНЦИОННО УПРАВЛЕНИЕ: Управление на обекти от разстояние чрез изпращане на управляващи електрически сигнали. Приложение - при централизирано управление или когато присъствието на оператора близо до обекта е опасно.

[6] http://www.znam.bg/, Енциклопедии, Техн. кибернетика, последно посещение 19 март 2009.

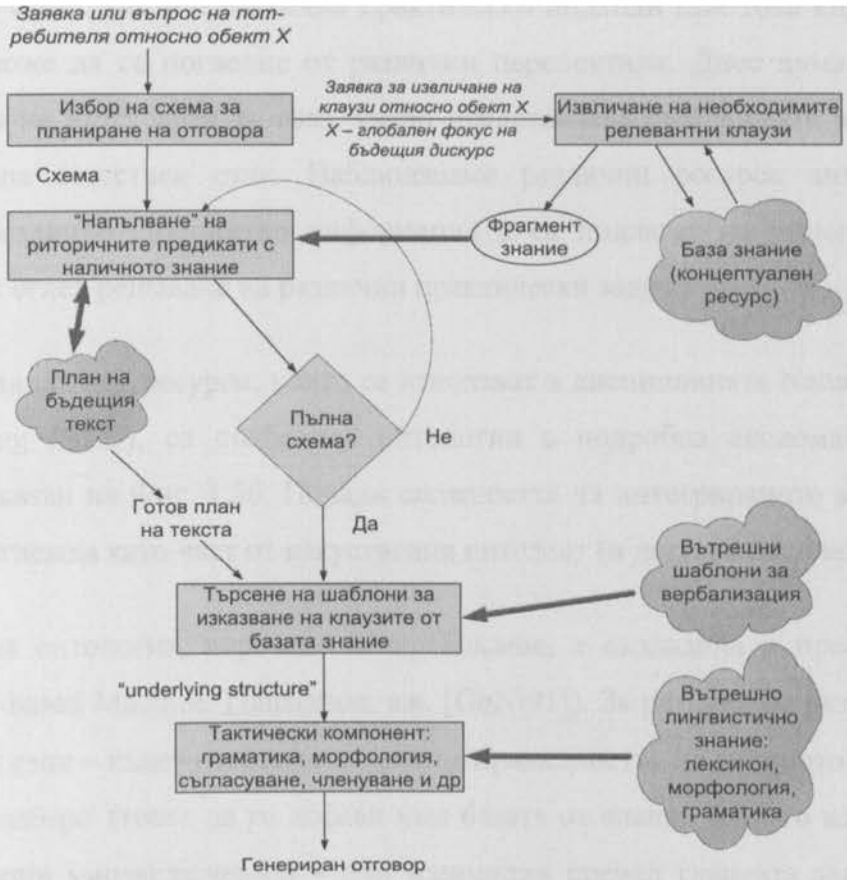
Следвайки понятието за *риторична схема* [McK85], лесно разпознаваме дискурсните планове за построяване на горните текстове. По-долу в Таблица 1.1 е дадена примерна схема, по която тези обяснения са генерирани от човек. Използват се означенията: '{}' - възможни изборни елементи, '/' - алтернативи, '+' - елемент, който се среща от 1 до *n* пъти, и '*' - елемент, който се среща от 0 до *n* пъти. Тези символи осигуряват известна гъвкавост на схемите.

Дискурсен план за описание на технически обект в енциклопедия
Риторични предикати: Идентификация на обекта (дефиниция, {атрибут}$^+$, функция) {Развитие на темата / Подобекти / Атрибути / Описание на приложението }* {Пример }*

Таблица 1.1. Примерен дискурсен план за генерация на обяснения в техническа област

Текстовете за РОБОТ и ДИСТАНЦИОННО УПРАВЛЕНИЕ са различни, тъй като вербализират различни знания, но все пак те си приличат поради повтарящата се дискурсна структура на описанията. Изреченията се нареждат в последователността, зададена чрез схемата. При наличие на повече факти се организират няколко клаузи, като вътре в тях фокусът се мести по определен начин. Някои клаузи могат да се пропуснат при липса на запълващи факти.

Автоматичната генерация на текст се извършва чрез предварително зададено множество схеми за различни комуникативни ситуации. Когато потребителят постави заявка от даден тип, системата избира схема-план на отговора. Фигура 1.7 показва процеса на запълване на всяка схема; илюстрирана е най-важната част от процеса – как схемата филтрира клаузи от знания при подбора на съдържанието (тоест, самата схема налага ограничения за избор на знанието, което да се разкаже). Така схемата контролира *колко* и *какво* знание да бъде извлечено от концептуалния ресурс на системата и *кога* да бъде изказано в текста. При напълване на схемата се избягват повторения на факти, които са вече обяснени в рамките на текущия сеанс. Един алгоритъм за извличане на факти при генерация на технически обяснения ще бъде предложен в трета глава.



Фигура 1.7. Автоматично генериране на текст чрез схеми (система TEXT [McK85])

В настоящия труд ще говорим за *езикови технологии* – софтуерни модули за обработка на естествен език, които използват алгоритми и методи на компютърната лингвистика и са достатъчно стабилни и надеждни, за да функционират като обособени компоненти на по-голяма софтуерна система. Езиковите технологии се нуждаят от данни за езика, наречени *лингвистични ресурси* - множества от специално разработени лингвистични данни, най-често в строго определен формат.

1.2.2. Концептуални ресурси, използвани в компютърната лингвистика

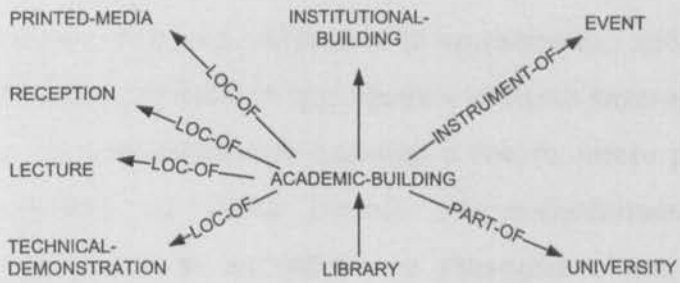
За да имаме програми, които разбират, генерират и превеждат естествен език, трябва да можем да кодираме в системите както значението на думите, така и

знанието за света. Но има различни практически подходи към това кодиране и на проблема може да се погледне от различни перспективи. Днес няма стандарт и унификация на възгледите за това, какво представлява онтологията в система за обработка на естествен език. Наблюдаваме различни ресурси знание, които съдържат различно количество информация и се подлагат на различни по вид обработки с оглед решаване на различни практически задачи.

Най-формалният вид ресурси, които се използват в дисциплината Natural Language Understanding (NLU), са дълбоките онтологии с подробна аксиоматика. Такъв ресурс е показан на Фиг. 1.5б. Поради сложността на интегрираното знание, NLU често се разглежда като част от изкуствения интелект (и дори се нарича AI-NLP).

Една плитка онтология, наречена МикроКосмос, е създадена в проекта KBMT (Knowledge-based Machine Translation, вж. [GoNi91]). За разлика от разбирането на естествения език – където системата доказва правилността на входното изречение с цел да го 'разбере' (тоест да го добави към базата от знания и да го използва като факт в бъдещи умозаключения) – при машинния превод главната задача е да се реши как думите от входния текст се 'заменят' с думи в изходния текст, за да се получи добър превод с максимално запазване на значението на входа. При машинния превод се извършва пълен морфологичен и синтактичен анализ, но няма проверка на семантичната коректност на входа. За сметка на това трябва да се намерят правилните преводни еквиваленти на входните думи в конкретния контекст на употреба. Например, англ. EAT за животно трябва да се преведе с немското FRESSEN, а не с глагола ESSEN. Проектът KBMT има за цел да построи система, при която знанията за значенията на думите се пазят в онтология (а не в двуезичен речник). Онтологията МикроКосмос е извлечена ръчно. Тя съдържа около 5000 понятия и е най-големият концептуален ресурс, интегриран в система с детайлен синтактичен анализ. Както е показано на Фиг. 1.8, МикроКосмос е типична семантична мрежа. Лексикалната семантика на всяка дума се кодира в речника на системата чрез понятията на онтологията. МикроКосмос е извлечена именно с цел да служи като набор от семантични примитиви, в термините на които се дефинират значенията на думите от двата езика. С оглед размера на

МикроКосмос можем да приемем, че множеството от концептуални релации е сравнително стабилно, тоест натрупана е критична маса отношения. Разработчиците на KBMT нямат претенции, че множеството концептуални релации е универсално, но твърдят, че около 30 релации са достатъчни за добро поведение на системата [NMB96]. За жалост трябва да признаем, че KBMT като базиран върху знания прототип не е по-добър от днешния статистически машинен превод, който с натрупването на учебни данни изпреварва всички системи, работещи чрез правила.



Фигура 1.8. Понятието ACADEMIC-BUILDING и неговата околност в МикроКосмос

Най-плоски като онтологии са ресурсите, които се интегрират в статистически подходи за извличане на информация. Плосък ресурс е WordNet [WNet], която се състои само от думи, организирани в мрежа чрез четири вида връзки: хипонимия (is-a), меронимия (part-whole), омонимия и синонимия.

При обработката на естествен език се създават концептуални ресурси, в които по премълчаване грануларността на понятията съответства на думите [Ang00]. В по-общ план, обаче, можем да подходим към грануларността по произволен начин. Например и двата израза

$$\forall p\ (\text{MaleHuman}(p)\ \&\ \exists c\ \text{Person}(c)\ \&\ \text{Parent}(p,c))\ \&\ \exists\ s\ \text{Building}(s)\ \&\ \&\ \text{UsedForTeaching}(s)\ \&\ \text{MoveByCar}(p,s) \quad (4.1.)$$

$$\forall p\ \text{Father}(p)\ \&\ \exists s\ \text{School}(s)\ \&\ \text{drive}(p,s) \quad (4.2)$$

кодират изречението "All fathers drive to school", но (4.1) е композирано от примитиви, които не съответстват на думи и това би затруднило една система за разбиране на естествения език (понеже тя се ръководи от грануларността на входните думи и превръща думите в предикати на логическата форма) [Allen 95].

Изложението в тази част показва, че базираната на знания обработка на два и повече езика е истинско предизвикателство, поради нетривиалните проблеми на препокриващите се значения на думите-преводни съответствия.

1.2.3. Автоматично извличане на знания от текст и принципни ограничения

В [Ang05] е направен обзор на подходите за автоматично добиване на знания от текст, като извличаните единици са групирани в няколко категории. Първата от тях са екземплярите; т.е. наименованите единици в текста, чието разпознаване днес е задължителен елемент от всяка голяма текстообработваща система. След разпознаването им, което за английски се извършва с над 96% точност при неструктуриран входен текст, системата класифицира индивидите в предварително-зададени категории: имена на хора, на географски обекти, на фирми, на мерки, дати, време и т.н. Една от най-добрите системи за разпознаване на наименовани единици в произволен английски текст е KIM, създадена в българската фирма OntoText Lab [PKOMK04]. Срещат се различни таксономии за класификация на наименованите единици и индивидите: тази в KIM е различна от категориите, използвани например в OntoMat Annotizer [HaSt03].

Много подходи атакуват проблема за извличане на хипонимични връзки (т.е. *is-a*). Най-лесно това се прави от структуриран текст, например от речници:

> **Deal:** A transaction on a stock exchange by a broker or institution.
> **Dealer**: A person or company that trades financial instruments and takes positions in them on their own account[7].

Модул за обработка на естествен език лесно би извлякъл от тези определения, че:

DEAL *IS-A* **TRANSACTION**
DEALER *IS-A* **PERSON**
DEALER *IS-A* **COMPANY**

Така се откриват много таксономични отношения, но все пак публично-достъпните полу-структурирани текстове са сравнително редки. Според [WSG97], речниците са бази от знания в текстов вид и от тях могат да се извлекат редица факти.

---

[7] http://www.finance-glossary.com/terms, последно посещение 15 април 2009

Един от най-ранните опити за извличане на таксономични връзки чрез търсене на лингвистични шаблони в неструктуриран текст е представен в [Hea92]. Шаблони като '*NP*0 *such as NP*1, *NP*2, … *and/or* NP_n' и '*NP*1 *is a kind of NP*2', където *NP* е група на съществителното, дават много добри резултати. Този подход е разширен в [SJA04], където са предложени алгоритми за откриване на шаблони с използване на значенията в WordNet. Така се откриват шаблоните '*NP*1, *a NP*2', '*NP*1 *called NP*2' и др. Чрез натрупване на няколко езикови технологии и съответни ресурси - тествани шаблони, екстрактори на термини и др. под. - се появява възможност за автоматично генериране на цялостни таксономии от голям корпус.

Автоматичното разпознаване на термините в свободен текст е важна стъпка към концептуализацията на дадена предметна област, тъй като термините по дефиниция са езиковото проявление на понятията. Напоследък има доста добри статистически екстрактори на терминология, например публичният TermExtractor [TExtr]. Тези системи извличат устойчиви словосъчетания – фразови термини като напр. *credit card*, *local tourist information office*, *board of directors*. След емпирично установяване на филтри за подбор на кандидатите, става възможно да се отсее множество от термини, които да се предложат на инженера на знанията за по-нататъшен избор. Термини се търсят и с анализ, базиран върху правила. Следва да отбележим, че разпознаването на термините в произволен текст е трудна задача дори когато търсените термини са известни предварително. Например продуктът FastCode®, разработен от фирмата Language and Computing, разпознава автоматично медицински термини в текст с цел извличане на информация за направените манипулации и автоматично осчетоводяване на стойността на лечение [NLPLC04]. Автоматично се разпознават 67,6% от термините, които означават медицинска услуга, а след намиране на допълнителна информация и/или преформулиране на текста успеваемостта на автоматичното разпознаване на термините достига 90%. Така че списъците от автоматично-извлечени значими термини се коригират с ръчна човешка намеса.

Правят се опити за намиране на понятия в текста и с разбор на изреченията. Чрез наблюдения на синтактично-анализиран текст става възможно да се отделят множества от думи, за които може да се предполага, че имат сходна семантика понеже заемат подобни синтактични позиции. Статиите [ReSp04] и [RSPD04] представят един типичен подход на онтологично инженерство. След анализиране на медицински текст на английски език с плитък синтактичен анализатор, авторите наблюдават синтактични шаблони [*Main Verb - Nominal String*], където групата на съществителното се състои от прилагателно и съществително. След клъстеризиране спрямо глаголите се получават групи като

{ *liver transplantation, transplantation, orthotopic liver transplantation* }

за които се предполага, че се състоят от подобни думи и формират понятия. С усложняване на шаблоните за наблюдаване на фрази се получават клъстери като

```
[dose, injection, vaccination] of
                    [hepatitis B vaccine, HBV vaccine, vaccine]
[use] of [face mask, mask, glove, protective eyewear]
[vaccination,vaccine] against [disease,virus, virus type].
```

За жалост, точността на разпознатите понятия е ниска – под 17%. Не е висока и 'покриваемостта' (recall) - около 33%, т.е. 2/3 от срещанията на тези понятия в текста не са разпознати чрез скицирания подход. Добра система от този вид е ASIUM, която работи над изхода на дълбок синтактичен анализатор [FaNé98]. Извлечените клъстери са задоволително точни, още повече че ASIUM е демонстрирана за тясната област на готварските рецепти. Човек-експерт трябва да наименова намереното понятие с явно име, тъй като клъстерът съдържа само думи-низове.

Като резюме подчертаваме, че всички споменати подходи за извличане на знания от неструктуриран текст добиват информация само в рамките на едно изречение. Това означава, че допълнителни факти, изказани в следващо изречение с местоименна референция към обекта в първото изречение, няма да бъдат разпознати и извлечени. В [GPMM05] е даден списък на езикови технологии за извличане на знания от текст и съответни софтуерни среди; там също се цитират подходи, които извличат концептуални представяния посредством обработка на отделните изречения в текста. В трета глава ще предложим прототипен модул за извличане на КГ от текст; той също работи в рамките на входното изречение.

1.2.4. Компресия на морфологични речници чрез крайни автомати и дефиниции на основни понятия в теорията на крайните автомати

Крайните Автомати (КА) се използват в компютърната лингвистика за представяне на синтактично знание от 1970 г., когато Уудс предлага т. нар. 'мрежи на преходите' за анализ на английски изречения [Woo70]. Днес КА се използват за кодиране на лингвистично знание в редица системи за обработка на текст и реч [RoSch97]. Тук ще се спрем по-подробно на едно приложение, което е от значение за по-нататъшните ни разглеждания. През 1993 Макс Зилберщайн предлага формат на морфологичен речник, в който всички словоформи са експлицитно кодирани като варианти на основната им лема с помощта на минимален ацикличен краен автомат с маркери на заключителните състояния [Sil93]. Нека разгледаме малък морфологичен речник от цели словоформи, зададен в лексикографска наредба:

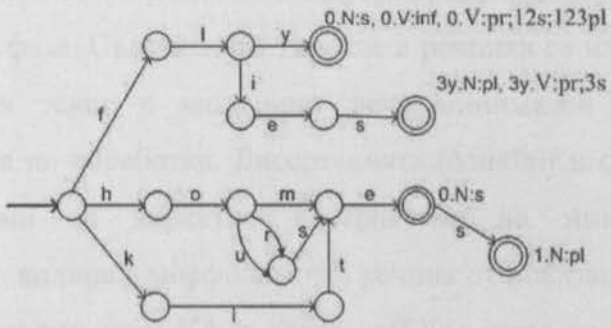
fly,fly.N:s
fly,fly.V:inf
fly,fly.V:pr;12s;123pl
flies,fly.N:pl
flies,fly.V:pr;3s
home,home.N:s
homes,home.N:pl
horse,horse.N:s
horses,horse.N:pl
house,house.N:s
houses,house.N:pl
kite,kite.N:s
kites,kite.N:pl

След всяка словоформа има разделител ',' и след него е зададена основната форма (лексема) на съответната словоформа. Друг разделител '.' отделя думите от граматичната им категория, зададена с главна буква. След разделителя ':', който следва категориите N(oun) или V(erb), явно са изброени характеристиките на словоформата: *s* за единствено число и *pl* за множествено число, *inf* за инфинитивни форми на глаголите, *pr* за лична глаголна форма в сег. време, и *12s*, *3s*, *123pl* за лице и число на личните глаголни форми. Ако при анализ на текст се използва такъв речник с цели форми, не е необходимо да се *изчислява* всеки път коя е основната форма на дадена дума, напр. че низът *flies* идва от *fly* и е или словоформа за множ. число на съществителното *fly*, или лична форма на глагола *fly* в сег. време, 3^{то} лице ед. число. Анализаторът трябва само бързо да намери информацията в речника, когато на входа му се подаде низът *flies*. Предлага се речникът да се трансформира в по-ефективно представяне чрез проста замяна на символи. Нека **Xu** означава следната трансформация: словоформата w_1 се получава

от друга словоформа w_2 чрез изтриване на **X** букви от края на w_2 и залепяне на низа **u** на тяхно място в края на w_2. Сега вместо са задаваме явно лексемата, можем да покажем как тя се получава от конкретната словоформа:

fly,0.N:s	horse,0.N:s
fly,0.V:inf	horses,1.N:pl
fly,0.V:pr;12s;123pl	house,0.N:s
flies,3y.N:pl	houses,1.N:pl
flies,3y.V:pr;3s	kite,0.N:s
home,0.N:s	kites,1.N:pl
homes,1.N:pl	

Този запис показва, че много думи се получават от основната си форма по подобни флективни шаблони. Например, 4 форми на съществителни в множ. число са получени чрез добавяне на *s* в края на лексемата (*homes*, *horses*, *houses* и *kites*). Затова етикетът <1.N:pl> се появява 4 пъти като морфологична характеристика, съчетаваща граматична информация и указания как лексемата да бъде изчислена от подадената на входа словоформа (а именно, чрез изтриване на *s* от края й). Тези наблюдения ни водят към кодиране на речника от думи като минимален краен автомат с маркери на заключителните състояния, показан на Фиг. 1.9. Съгласно приетите означения, състоянията на автомата са изобразени като кръгове, а преходите – като насочени дъги, които имат за етикети символите на азбуката. Началното състояние е маркирано със стрелка. Четирите заключителни състояния са изобразени с двойни кръгове. Граматичната информация е зададена като маркер на всяко заключително състояние, като се допуска изброяване на повече от един маркер (например, *fly* има 7 възможни анализа: същ. ед. ч-ло, глагол инфинитив и глаголна форма за сег. време в 1^во и 2^ро л. ед. ч-ло и 1^во, 2^ро и 3^то л. мн. ч-ло).



Фигура 1.9. Морфологичен речник, кодиран като минимален ацикличен КА.

Всички маркери се присвояват на думата, чрез която автоматът достига от началното до съответното заключително състояние, следвайки пътя образуван от символите на думата. Ако на входа на КА постъпи низа *houses,* след трасиране на автомата се достига до заключително състояние – тоест *houses* принадлежи на регулярния език, разпознаван от автомата. След успешното достигане на заключително състояние, словоформата *houses* получава маркер <1.N:pl>, който ни показва коя е лексемата (тя се получава след изтриване на последния символ) и кои са граматическите характеристики на *houses* (съществително, мн. ч-ло). Автоматът на Фиг. 1.9 е детерминиран и минимален, което позволява трасирането да се извършва в линейно време O(*n*) спрямо дължината *n* на входната дума. Така граматичната информация за *houses* се извлича за 6 стъпки и тази скорост не зависи от големината на речника. Забелязваме, че еднаквите префикси и суфикси на думите се кодират само веднъж, което позволява значителна компресия на речника, тъй като много думи на естествен език имат общо начало или еднакви окончания.

През 1997 г. този подход за предварително (off-line) кодиране беше приложен от Стоян Михов към български морфологичен речник с 60000 основни форми, които генерират 893313 словоформи. Минималният ацикличен краен автомат с маркери на заключителните състояния има 47536 състояния, 110105 прехода и 6244 маркера (т.е. 893313 думи принадлежат към 6244 граматически класа). Автоматът заема около 400 килобайта, съхранява се в оперативната памет и осигурява анализ на български текст със скорост около 100000 думи в секунда на стандартен персонален компютър [Мих00]. Обикновено изброяването по метода на грубата сила не се счита за елегантно решение в информатиката, но тук то се прилага само на предварителната фаза. Същинското търсене в речника се извършва по време на анализа на зададен текст и наличният вече минимален автомат радикално подобрява скоростта на обработка. Дисертацията [Мих00] и статията [DMWW00] предлагат алгоритми за директно построяване на минимален ацикличен дървовиден автомат, кодиращ морфологичен речник от показания тук вид. Идеята е речникът да се поддържа като КА в режим off-line, като всички обновявания се правят без да се пречи на изпълнение на конкретни заявки за търсене. Така

търсенето на дума в речника се свежда до трасиране на КА, който е построен специално за целта – да осигури максимално ефективно намиране на граматичната информация към всяка предварително-известна дума.

В глава 2 се предлага едно нетривиално разширение на скицирания по-горе подход, за да се кодират като минимален КА положителните конюнктивни формули с двоични предикати, които могат да се интерпретират като твърдения при представяне на декларативно знание за света. Следвайки [HMU83] и българската терминология от [Ман03], тук даваме дефинициите на основни понятия от теорията на крайните автомати, които са необходими за изложението в глава 2.

Дефиниция 1.1. Крайният автомат (КА) A е наредена 5-орка $A = \langle \Sigma, Q, q_0, F, \Delta \rangle$, където Σ е крайна азбука, Q е крайно множество от състояния, $q_0 \in Q$ е начално състояние, $F \subseteq Q$ е множество от заключителни състояния, и $\Delta \subseteq Q \times \Sigma \times Q$ е отношение на преходите. Преходът $\langle q, a, p \rangle \in \Delta$ започва в състояние q, свършва в състояние p и има етикет a. **Детерминираният краен автомат** $A = \langle \Sigma, Q, q_0, F, \Delta \rangle$ е такъв КА, за който Δ е функция $\Delta: Q \times \Sigma \to Q$. □

Дефиниция 1.2. Нека $A = \langle \Sigma, Q, q_0, F, \Delta \rangle$ е КА. **Път c в A** е крайна поредица от $k > 0$ прехода:

$c = \langle q_1, a_1, q_2 \rangle \langle q_2, a_2, q_3 \rangle \ldots \langle q_k, a_k, q_{k+1} \rangle$, където $\langle q_i, a_i, q_{i+1} \rangle \in \Delta$ за $i = 1, \ldots, k$.

Цялото число k се нарича *дължина* на c. Състоянието q_1 се нарича *начало* на c и q_{k+1} се нарича *край* на c. Низът $w = a_1 a_2 \ldots a_k$ се нарича *етикет* на c. *Нулевият път* на $q \in Q$ е 0_q, той започва и завършва в q с етикет ε, където ε е празният символ. Път, който започва в q_0 и завършва в заключително състояние, се нарича *успешен*. □

Дефиниция 1.3. Нека $A = \langle \Sigma, Q, q_0, F, \Delta \rangle$ е КА. Нека Σ^* е множеството от всички низове с крайна дължина върху азбуката Σ, включително празния символ ε. **Обобщеното отношение на преходите** Δ^* е най-малкото подмножество на $Q \times \Sigma^* \times Q$, което е затворено спрямо следните свойства:

- За всяко $q \in Q$ е изпълнено $\langle q, \varepsilon, q \rangle \in \Delta^*$;
- За всеки $q_1, q_2, q_3 \in Q$ и $w \in \Sigma^*$, $a \in \Sigma$ е изпълнено следното: ако $\langle q_1, w, q_2 \rangle \in \Delta^*$ и $\langle q_2, a, q_3 \rangle \in \Delta$, то $\langle q_1, w \cdot a, q_3 \rangle \in \Delta^*$. □

Дефиниция 1.4. Формалният език $L(A)$, разпознаван от даден КА $A = \langle \Sigma, Q, q_0, F, \Delta \rangle$ е множеството на всички низове, които са етикети на успешни пътища в A:

$$L(A) := \{ w \in \Sigma^* \mid \exists\, q \in F : \langle q_0, w, q \rangle \in \Delta^* \}.$$

Тези низове се наричат **думи** от езика $L(A)$. □

Всяко крайно множество от думи над дадена (крайна) азбука от символи е **регулярен** език. Крайните автомати разпознават регулярни езици. Съществуват алгоритми, които построяват детерминиран КА, разпознаващ даден краен регулярен език L.

Теоремата на Myhill-Nerode (цитирана в [HMU83]) гласи, че измежду всички детерминирани автомати, които разпознават даден регулярен език L, съществува единствен автомат (с точност до изоморфизъм), който има най-малък брой състояния. Този автомат се нарича **минимален** КА, разпознаващ L. В [Hop71] се предлага алгоритъм за минимизация на детерминиран КА с m състояния, със сложност $k \times m \times \log(m)$, където k е константа, която зависи линейно от размера на входната азбука. За сравнение, директното построяване на минимален ацикличен КА A по зададен списък от сортирани думи на краен регулярен език L е $O(n \log(m))$, където n в общият брой на символите във входния списък на думите от езика L и m е броят на състоянията в A [Мих00, DMWW00].

Дефиниция 1.5. Нека $A = \langle \Sigma, Q, q_0, F, \Delta \rangle$ е КА. Нека Σ^+ е множеството от всички низове w с крайна дължина върху азбуката Σ, където $|w| \geq 1$. Автоматът A се нарича **ацикличен** тогава и само тогава, когато за всяко $q \in Q$ не съществува низ $w \in \Sigma^+$ такъв, че да е изпълнено $\langle q, w, q \rangle \in \Delta^*$. □

Дефиниция 1.6. Детерминираният краен автомат с маркери на заключителните състояния A е наредена 7-морка $A = \langle \Sigma, Q, q_0, F, \Delta, E, \mu \rangle$, където Σ е крайна азбука от символи, Q е крайно множество от състояния, $q_0 \in Q$ е начално състояние, $F \subseteq Q$ е множество от заключителни състояния, $\Delta: Q \times \Sigma \to Q$ е функция на преходите, E е крайно множество от маркери и $\mu: F \to E$ е функция, присвояваща маркер от E на всяко заключително състояние. □

Да отбележим, че детерминираният краен автомат с маркери на заключителните състояния присвоява маркер на всяка разпозната дума. Това свойство ще бъде много полезно за конструкцията, предложена в глава 2. Ще използваме без доказателство и следните твърдения:

Твърдение 1.1. Нека A е детерминиран автомат, в който всеки път може да се разшири до успешен. A е ацикличен автомат тогава и само тогава, когато A разпознава крайно множество от думи. □

Твърдение 1.2. Нека L е краен регулярен език. Съществуват алгоритми за конструиране на ацикличен детерминиран автомат, който разпознава L. □

Твърдение 1.3. Нека $A = \langle \Sigma, Q, q_0, F, \Delta \rangle$ е детерминиран КА и $w = a_1 \ldots a_n$ е дума, съставена от символи на Σ. Сложността на трасиране на A с w е $O(n)$. □

1.3. Приложения на концептуалните структури в семантично-базирани системи

Ще се ограничим до кратки бележки върху отделни по-важни за нас аспекти на областта. В този труд се интересуваме от системи за обучение, базирани върху знание. Както е посочено в [ДиД95], интелигентни системи за обучение наричаме компютърни програми, които използват методи и средства на изкуствения интелект, за да помогнат на човек да учи. Обзорът [Bru98] определя интелигентността на системата за обучение като способност да се решават поне следните две задачи:

- Персонализиране на процеса на обучение,
- Възможност на системата да извършва умозаключения относно целите на обучението; да оценява знанията на обучаемия; да предлага стимули за мотивиране на студента и да разпознава различни стилове на преподаване.

Обикновено наличието на знание е предпоставка за интелигентността на системата ([Bru04], [BrMi07]), тъй като тя ползва концептуалните единици при стратегиите за персонализация, адаптивност и моделиране на студента.

Нашето внимание е насочено и към системите за изучаване на втори език, които интегрират езикови технологии. Обзорите [Ner02] и [GaKn02] анализират и систематизират видовете модули за обработка на естествения език, които се влагат в системи за обучение. Тъй като изучаването на чужд език е една от най-застъпените дисциплини в училище, разработката на системи-помощници на учителя е много важна задача. Този вид образователен софтуер се разглежда като технологична иновация, която намира приложение в различни дейности на езиковата педагогика като слушане, говорене, четене и писане. Най-разпространените системи подпомагат изучаване на морфология и синтактиса на чуждия език.

Разгледаните в [Ner02] и [GaKn02] модули като правило не са интегрирани в цялостни интелигентни среди за обучение. Обикновено приложението на езикови технологии в образователния софтуер е трудна задача, понеже човешкият начин на изучаване на естествен език се различава от използваните в компютърната лингвистика формални модели. Поради това е спорно дали образователните програми трябва да показват пред студента вътрешно-системното представяне на граматическата информация [Ner02].

Концептуалните ресурси привличат изключително внимание след появата и активното рекламиране на т.нар. Семантичен интернет [BLHL01]. Онтологиите се разглеждат като семантични скелети, към чиито елементи са свързани всички html-страници по света. Като цяло идеите за глобален Семантичен интернет се

редуцираха до конкретни приложения през последните 4-5 години и се наложи виждането, че дигиталните архиви се строят и управляват посредством метаданни и всеки отделен дигитален обект има нужда от анотация. Анотацията може да бъде изградена само чрез използване на думи и термини на естествен език. В тази връзка на преден план излиза проблемът за построяване на терминологични колекции. Преди няколко години специалисти по компютърна лингвистика са изказали препоръки към правителството на САЩ да се обърне внимание на интеграцията на държавните архиви чрез ключови думи и стандартизиран подход към терминологията [CCH02]. Според автора, задачата за изграждане и динамично управление на онтологии и фолксономии от ключови думи е една от най-важните научни теми в съвременните семантични системи.

1.4. Състояние на изследванията по темата на дисертацията в България

Интересът на автора към концептуалното моделиране възниква през 90-те години на миналия век и е свързан с участие в Работната група РГ-18 'Представяне на знанията' на КНВИТ[8] [АБВСБХ84]. Първото системно изложение на тази проблематика у нас намираме в книгата по изкуствен интелект [ПоДа90]. След десетилетие на обясним спад в научно-изследователските резултати, днес наблюдаваме значителна активност в рамките на различни проекти. По-новите разработки по концептуално моделиране и семантични системи обикновено са свързани с конкретна приложна дейност за конструиране на онтология в дадена предметна област, с оглед нуждите на определено приложение. В [SDPPDS07], [SSD08], [PDPD07], [PRL07] и [ДКНП08] се представят резултати от концептуално моделиране за целите на търсене на обекти в дигитални архиви, а в [StTo09] е представено създаване на онтология в областта на компютърните науки. Активно се разработват прототипи на семантични системи, предимно среди за обучение с определен обем учебно съдържание или системи за достъп до дигитални архиви в областта на културно-историческото наследство: такива са [AgDo08], [PaNP06],

[8] Комисия по Научните Въпроси на Изчислителната Техника (КНВИТ) на страните от СИВ.

[NPP06], [LVKSECM07] и [LSOMM08]. Изследват се начините за изграждане на колекции от дигитални обекти, базирани върху знания, което позволява налагане на метаданните на обектите върху концептуалния модел, вж. [SVPM08] и [GSP08]. Прототипи на интелигентни системи със способности за навигация в дигитално съдържание посредством онтологии ще бъдат създадени и през 2009-2011 в проектите *Умна книга* [УК09] и *Семантични технологии за интернет услуги и технологично-поддържано обучение* [СемТех09].

Като форум на по-теоретично ориентирани изследователи, в последните години функционира *Семинар по динамични онтологии* с ръководители М. Хаджийски и В. Петров. Представени на семинара доклади на български и чуждестранни учени бяха издадени наскоро в първата книга по приложни онтологии у нас [НаРе08].

Специално трябва да отбележим може би най-успешната източно-европейска фирма в областта на семантичните технологии – ОнтоТекст Лаб, създадена като част от групата СИРМА. Отделът за фирмени научно-изследователски разработки на ОнтоТекст произвежда впечатляваща по обем продукция, която се представя както на научни форуми, така и на индустриални конференции [ОТекст09]. По различни проекти са разработени редица платформи за услуги в Семантичния интернет. Във връзка с проекта SEKT, ОнтоТекст Лаб създава OWLIM, най-бързата и скалируема база от данни за RDF-графи, над които се правят изводи с OWL.

Основите на компютърната лингвистика у нас са поставени през 1964 год. със създаването на Групата по машинен превод на проф. Александър Людсканов в Института по математика на БАН. Днес българската компютърна лингвистика е изненадващо продуктивна и понастоящем десетки научни групи и фирми създават езикови технологии и ресурси за обработка на българския език. Най-активните научно-изследователски групи в областта се намират в БАН (в ИПОИ, ИМИ, ИБЕ и ИИТ), в Пловдивския университет и в Софийския университет. Напоследък се оформиха научно-изследователски групи в Нов български университет и в Търновския университет. В тази област работят информатици, лингвисти и логици.

Активни индустриално-ориентирани частни организации са ОнтоТекст Лаб, ПроЛангс (разработчиците на [Бултра]), Сиела, АПИС, BMG (от ACT Soft), dir.bg, netinfo и БАКЛ (Българската асоциация по компютърна лингвистика). Създадени са 7-8 много големи морфологични речника на българския език и съответни анализатори към тях, на пазара има няколко програми за корекции на правописни грешки, съществуват поне три прототипни разработки на синтактични анализатори на български изречения и две системи за машинен превод. Непрекъснато се подобрява търсенето в архиви от документи на български език. БАКЛ предлага синтезатор на българска реч по зададен входен текст като продукт, ориентиран към граждани с нарушено зрение. Наскоро беше демонстрирана и програма за разбиране на реч на български, чиято разработка е финансирана от Сиела. Налице е и впечатляващо количество лингвистични ресурси от различен вид, създадени главно в академичните среди. Измежду тях ще спомена ресурса BulTreeBank, една от 5-те най-големи банки от синтактично-анотирани дървета в света [БДБ04] и нейната частична граматика за българския език [ОсСи07].

Автоматичната генерация на естествен език не е много популярна в България, където традициите на компютърната лингвистика са свързани предимно с анализа на текст и дълги години бяха концентрирани предимно върху морфологията и синтаксиса. Първата прототипна система за генерация на текстове е създадена в Института по математика и информатика на БАН и генерира обяснения на геометрични обекти ([MiSi89], [Mit90a] и [Mit90b]). По проекта AGILE 'Автоматично генериране на технически ръководства на езици от Източна Европа' в Института по информационни технологии на БАН е адаптирана за български език средата за генерация Комет-Пенман ([StDo00] и [DoSt01]).

Средите за електронно обучение са вече част от обичайното информационно осигуряване на учебния процес във висшето образование. В практиката навлизат изцяло дистанционни форми на обучение. През 90-те години на миналия век има отделни изолирани разработки за електронизация на висшето образование (вж. напр. [NiNi96]), но днес е създадено образователното пространство на Българския

виртуален университет [БВУ]. Разглеждат се подходи за създаване на архиви от виртуални обекти, които са една алтернатива на традиционните по-дълги и неразделяеми учебни материали във вид на книги [ChTo03]. Може да се очаква, че необходимостта от стандартизация и обмен на учебния материал скоро ще катализира създаването на платформено-независим, унифициран подход към структурирането на дигитализирано учебно съдържание. Създават се прототипи на адаптивни среди за обучение или персонализиран достъп до дигитализирани обекти, напр. [SVPM08]. В [VaBo06] е представено решение за адаптивната навигация чрез динамично присвояване на тежести на отделните страници с учебен материал. Персонализацията в системите за обучение се обсъжда в [PaPa06]. Един подход за моделиране на потребителя е представен в [Pan06].

Разработките на продукти за електронното обучение са все още тясно свързани с образователни институции (например университети) и държавно финансиране по правителствени проекти. Не се наблюдават индустриални приложения, които предлагат специфични образователни решения за по-тесни области и се разработват от софтуерни фирми. Отчасти това се дължи на ниския стандарт, който затруднява купуването на софтуер от частни лица, на относително малкият пазар в страната, както и на нерешените проблеми със защита на интелектуалните права при разпространението на софтуер. Така че опитите за създаване на интелигентни системи и системи, подпомагащи изучаването на чужд език се свеждат до (научни) прототипи, най-често свързани с определени проекти.

Активно се работи в областта на компютърно-подпомогнато изучаване на български като втори или чужд език. От 20 години се правят опити за влагане на езикови технологии в обучението на български език, вж. например:

- система за граматически тестове, изградена над морфологичен речник на българския език [SAP99],
- написан на Java прототип, който служи за създаването и интерактивното използване на упражнения и тестове за езиково обучение [Дас07],
- интегрирана компютърна среда за обучение по български език [Кру07],

- система за автоматично генериране на тестове по зададен електронен учебник на български език [Nik08].

Секцията за лингвистично моделиране на ИПОИ-БАН многократно участва в международни проекти за интегриране на езиковите технологии в системи за изучаване на чужд език. Такива проекти се финансират постоянно от Европейската комисия, което показва голямото внимание към темата: проект *ГЛОСЪР* (за подпомагане изучаването на чужд език чрез извличане на примери от двуезични текстове [NKPPR96], [PaMi98]); проект *LARFLAST*, разгледан в четвърта глава на предложения труд; проект LT4eL (Language Technologies for eLearning) [MLS06] и проект LTfLL (Language Technologies for Life-long Learning). Във всеки от тях са интегрирани различни по вид езикови технологии, които подпомагат един или друг аспект в обучението и са ориентирани към различни образователни активности. В Пловдивския университет също се разработват проекти за влагане на езикови технологии в среди за обучение, вж. [БГ-19].

Методите на изкуствения интелект се прилагат в областта на културното наследство от много години (вж. напр. [KoDo94]). Проекти на регионално ниво проправят пътя на дигитализацията, напр. [Сто08]. Същевременно обаче има значително изоставане в масовата практика за създаване на дигитални архиви. У нас все още не са натрупани големи колекции от дигитализирани обекти – както в образованието, така и в областта на културното наследство. Процесът е затруднен от липсата на централизирана политика за интегриране на усилията в областта на дигитализацията. Поради това няма систематичен подход към вътрешната организация на информацията; в много случаи музейните работници създават колекции с подръчни средства и решения ad hoc. Например:

- Страниците на Художествената галерия 'Владимир Димитров – Майстора' в Кюстендил[9] са организирани в около 20 'екрана' в 6 страници, които са 'закачени' към 6-те бутона отляво. При избор на бутон се отваря нова страница, която се разлиства с '*още...*' при нужда. Всички вътрешни

[9] На http://www.artgallery-themaster.com/bg/index.htm, последно посещение на 5 май 2009.

страници се цитират една друга, без да имат свой собствен адрес в интернет. Така за Гугъл цялата папка от 6 страници е един документ. Липсата на адекватна фрагментация и адресация в по-дребни атомарни единици не позволява по-фино търсене;

- Екраните на колекцията фотографии за Перперикон[10] са свързани последователно. Всяка фотография има текстова анотация, но е невъзможно да се търси директно в нея с ключови думи, тъй като текстът не е обособен в отделни полета. Фотографиите се 'обхождат' задължително една след друга.

Така че у нас културните институции все още не са направили първата сериозна крачка към проектиране и разработване на модерни дигитални архиви, които да се натрупват като устойчиво решение за многогодишна употреба и да осигурят многократно използване на веднъж създаден обект. За жалост не са добре познати и европейските стандарти за създаването на такива колекции. Наблюдава се активност в тази област, но главно от страна на отделни колективи от информатици; те създават прототипни системи, които се демонстрират над ограничен обем дигитализирано съдържание.

[10] На http://www.perperikon.bg/galeria.php, последно посещение на 5 май 2009.

ГЛАВА 2:
Ефективно търсене на концептуални шаблони

Търсенето на концептуални шаблони е задача, поставена в [Sow84] с дефинирането на т. нар. *проекция*[11] – операция, която по даден въпрос-шаблон намира негови отговори-специализации в базата от знания. Хората често правят специализации и не е трудно да намерим примери за този вид семантична обработка в ежедневието.

Пример 2.1: Да потърсим в архив от новини отговори на въпроса: *Кои са политическите събития в света от миналата седмица*? Възможни отговори са:

1. На среща на финансовите министри на страните от ЕС, проведена миналия вторник в Брюксел, беше обсъдена цената на суровия петрол;

2. Миналата сряда в София президентът Георги Първанов призова за обновяване на политическата коалиция и т.н.

В този случай въпросът се състои от следния концептуален шаблон:

[ПОЛИТИЧЕСКО_СЪБИТИЕ] → (ВРЕМЕ) → [МИНАЛАТА_СЕДМИЦА]
→ (МЯСТО) → [ПО_СВЕТА]

Отговорите 1 и 2 са получени чрез *специализация*, понеже всяко понятие е заменено с по-специфично такова и са запазени релациите за ВРЕМЕ и МЯСТО на шаблона, както и тяхната структурна конфигурация:

- Понятието ПОЛИТИЧЕСКО_СЪБИТИЕ е специализирано до
  - *Среща на финансовите министри ... за обсъждане цената на петрола* и
  - *Президентът ... призова за обновяване на политическата коалиция*
- МИНАЛАТА_СЕДМИЦА е *миналия вторник* и *миналата сряда*, а
- Понятието ПО_СВЕТА е специализирано до *Брюксел* и *София*.

Така отговори 1 и 2 са концептуално-подобни, понеже те са специализации на един и същи въпрос-шаблон в архива от новини. □

Операцията проекция е формализирана в [Sow84], където се изтъква и важността ѝ при обработка на явно-декларирано знание за света. Но тя има експоненциална

[11] Самият термин *проекция* е въведен по аналогия с едноименната операция в релационните бази от данни; по това време Сова въвежда и операцията *съединение* (*join*), която се извършва над знания.

сложност, освен в случаите когато шаблоните съответстват на графи-дървета [MuCh92, ChMu92]. На практика това означава, че времето за отговор е много дълго при големи онтологични модели с хиляди типове в йерархиите от понятия и релации. Тук можем да вметнем освен това, че редица традиционни алгоритми за извод в изкуствения интелект трудно се адаптират към съвременните изисквания за обем и скорост, тъй като не са пригодени към ефективно използване на външна памет и липсват съответни техники за индексиране на междинните резултати. Практическата неефективност налага цялостно преосмисляне на инструментариума за обработка на големи концептуални ресурси. В тази глава е даден оригинален отговор на изброените предизвикателства за случая на пресмятане на инективна проекция, чрез предложение за двуфазова обработка на концептуални структури:

- *Предварителна (off-line) обработка*: всички възможни инективни проекции в конкретния концептуален свят се пресмятат за експоненциално време и след това се кодират като компактен ацикличен краен автомат с маркери на заключителните състояния, и
- *Извличане на отговор по време на изпълнение*: зададен от потребителя въпрос се проектира върху концептуалния архив-автомат чрез обхождане на път в детерминирания краен автомат, за линейно време спрямо дължината на въпроса. По този начин времето за изчисление на проекцията не зависи от големината на базата от знания.

Идеята е алтернатива на традиционния подход – а именно, всички пресмятания да се извършват по време на изпълнение (в run-time), едва когато системата получи заявката на потребителя. Така една NP-пълна задача се разделя на два компонента, като всички обемни пресмятания са изнесени в предварителната фаза. В реално време се използват само резултатите от предварителната (off-line) обработка. При нарастващия потенциал на евтината външна памет, скоростта за изпълнение на заявката се превръща в най-важния параметър за оптимизация. Сценарий за двуфазова обработка виждаме например и при търсенето на документи в Интернет с Гугъл; всеки ползва Гугъл, без да се интересува колко обемно и тромаво е предварителното индексиране на html-страниците по метода на 'грубата сила'.

2.1. Обекти в речници vs. обекти в концептуални ресурси

Предлаганата тук идея за двуфазово изчисление на проекцията е вдъхновена от подхода за кодиране на морфологични речници чрез минимални ациклични крайни автомати с маркери на заключителните състояния (вж. част 1.2.4). От структурна гледна точка, каква е разликата между словоформите от Фиг. 1.9 и предикатно-аргументните формули, които искаме да кодираме като думи на регулярен език, с цел превръщането им в концептуален архив – минимален автомат? Да разгледаме думата `houses`, която се разпознава от автомата-речник на Фиг. 1.9. И при двете си срещания в `houses`, както и при срещания в други словоформи в примерния речник, буквата `s` има едно и също значение – тя е символ от крайната азбука на автомата. Така че на етапа на предварителното конструиране на автомата-речник, двете срещания на `s` в `houses` трябва само да бъдат запомнени в подходяща последователност като етикети на две различни дъги на прехода. Това обуславя и еднозначното им разпознаване в езика на автомата. По време на анализ на конкретен текст, при подаване на буквения низ `houses` като вход към автомата, `houses` получава маркер `1.N:pl` след 6 стъпки, тъй като входната дума е низ в същата азбука, над която е конструиран речникът-автомат. Така словоформите (низове над латинската или друга азбука) лесно се превръщат в регулярен език.

За разлика от тях, графите и съответните им логически формули очевидно имат нелинейна структура. Още в увода на Фиг. Iб виждаме и специфичните дублиращи се етикети в концептуалните графи: представени са 3 различни тухли (това е интуитивно ясно поради 3-те върха за понятието BRICK). При Фиг. Iв забелязваме, че имаме нужда от вътрешни индекси за кодиране на топологичната структура на графа – понеже една тухла е свързана с релацията BETW1, друга е свързана с релацията BETW2 и трета - с релацията BETW3. Следователно етикетът BRICK на Фиг. Iв има три различни значения, понеже той обозначава три различни екземпляра на понятието ТУХЛА. Чрез индекси се изграждат и логическите формули, към които се свеждат концептуалните графи. Но въвеждането на вътрешна индексация, специфична за отделните графи в базата от знания, ще

направи невъзможно налагането на произволен входен въпрос върху концептуалния архив в линейно време, понеже символите на въпроса трябва да се съпоставят на индексирани символи от конкретните графи. Освен това, съхранението на индекси ще породи различни етикети за различните графи в базата от знания. Бихме загубили компактността и елегантността на речника-автомат от Фиг. 1.9, в който еднаквите начала на думите са кодирани заедно и краищата на 'подобните' думи са събрани в общи заключителни състояния. Следователно при концептуалните обекти имаме нужда от специални 'трикове', за да запазим преимуществата на кодиране чрез краен автомат и същевременно да се справим със следните задачи:

- да сведем графите (или съответните логически формули) до думи на регулярен език, като думата – линеен низ – ще осигури трасиране чрез дъгите на преходите в детерминиран краен автомат;
- да поддържаме крайна азбука, осигуряваща кодирането както на архива на фазата на предварителната обработка, така и на въпроса-шаблон, който ще се подаде на системата по време на изпълнение на потребителска заявка.

2.2. Основни понятия

Ще въведем формално разглеждания тук подклас концептуални графи, като използваме дефиниции и примери от [Sow84, MuCh92, ChMu92, Mug95 и BaMu02].

Концептуалните графи не се задават като изолирани декларативни твърдения; те съществуват в контекст, който фиксира общата онтологична рамка за моделиране на предметната област – т.нар. опора (support), въведена формално в [ChMu92]. Ще дефинираме опората само за двумерни концептуални релации.

Дефиниция 2.1. Опората на даден концептуален модел на знанията за света е наредена четворка $S = (T_C, T_R, I, \tau)$, където:

- T_C е крайно, частично-наредено множество от различни типове-понятия. Частичната наредба дефинира йерархията на типовете понятия: за $x,y \in T_C$, $x \leq y$ означава, че x е под-тип на y. Тогава x е *специализация* на y и y е

обобщение на x; казваме още, че *у включва* или *съдържа* (subsumes) x. Универсалният тип $\top$ (top) съдържа всеки тип $x \in T_C$. Всички типове в T_C съдържат типа-абсурд $\bot$ (bottom);

- T_R е крайно, частично-наредено множество от различни типове-релации. Частичната наредба дефинира йерархията на типовете-релации. $T_C \cap T_R = \varnothing$. Всяка $R \in T_R$ е двумерна релация и е в сила между екземплярите на два различни типа-понятия $x,y \in T_C$ или два различни екземпляра на един тип $x \in T_C$. На всеки тип релация $R \in T_R$ се съпоставя двойка $(c1_{maxR}, c2_{maxR}) \in T_C \times T_C$; тя дефинира най-общите типове понятия, които могат да бъдат свързани чрез R като първи и втори аргумент. Множеството от всички двойки $(c1_{maxR}, c2_{maxR})$ се нарича *базис* на опората. R е в сила между екземплярите на понятията $x,y \in T_C$ ако $x \le c1_{maxR}$ и $y \le c2_{maxR}$. Ако $R_1, R_2 \in T_R$ и $R_1 \le R_2$, то $c1_{maxR1} \le c1_{maxR2}$ и $c2_{maxR1} \le c2_{maxR2}$. Решетката на типовете релации също има за връх универсалния тип $\top$ и за дъно абсурдния тип $\bot$;
- I е крайно множество от различни индивидуални маркери, които означават различни определени екземпляри на типовете понятия. $T_C \cap I = \varnothing$ и $T_R \cap I = \varnothing$. *Обобщеният* (*generic*) маркер $*$, където $* \notin (T_C \cup T_R \cup I)$, обозначава неопределен екземпляр от даден тип $x \in T_C$. Така понятията имат екземпляри, за разлика от релациите. Елементите на I не са наредени, но $i \le *$ за всяко $i \in I$;
- τ е изображение от I към T_C и определя принадлежност на индивидите към типовете понятия. Ако $\tau(i) = x_1, x_2, \ldots, x_n$ за $i \in I$ и $x_1, x_2, \ldots, x_n \in T_C$, тогава съществува най-специализиран тип понятие, към който i принадлежи (напр. x_1) и $x_1 \le x_j$ за $2 \le j \le n$. С други думи, всеки екземпляр принадлежи на даден тип и негови надтипове в един клон на йерархията понятия. Така τ дефинира съответствието и принадлежността (*conformity*) на екземплярите към типовете понятия. Ако $\tau(i) = x$ за $i \in I$ и $x \in T_C$, записваме го като $x{:}i$. Тогава неспецифицираният обобщен екземпляр $x{:}*$ от типа x обобщава $x{:}i$. □

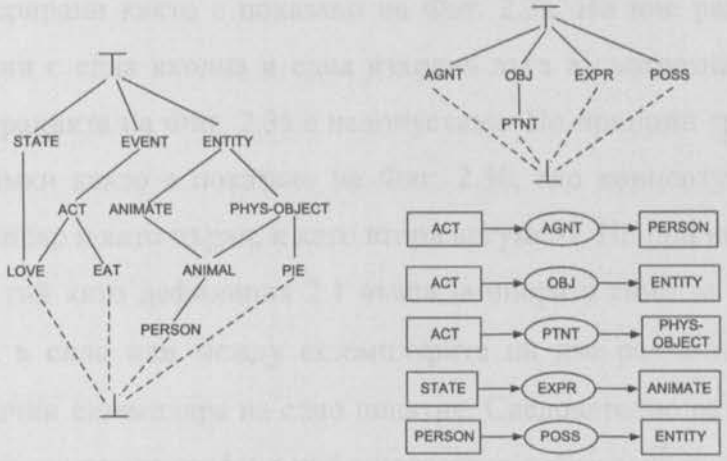
Дефиниция 2.2. **Прост концептуален граф** (ПКГ) G, дефиниран над опора S, е краен, свързан двуделен граф $(V = V_C \cup V_R, E, \lambda)$, където:

- Върховете V са дефинирани чрез V_C – множеството от върхове, съответстващи на типовете понятия или *c*-върхове и V_R – множеството от върхове, съответстващи на типовете релации или *r*-върхове. $V_C \neq \emptyset$, т.е. всеки ПКГ съдържа поне един *c*-връх. За $x \in V$, *type*(x) означава етикета $x \in T_C \cup T_R$;
- Ребрата E са крайно множество от наредени двойки (x,r) или (r,y), където $x,y \in V_C$ и $r \in V_R$. Така ребрата са насочени или от някой *c*-връх към *r*-връх – напр. (x,r), или от *r*-връх към *c*-връх – напр. (r,y). Реброто (x,r) се нарича *входна дъга* в *r*-върха r, а реброто (r,y) се нарича *изходна дъга* от *r*-върха r;
- Изображението λ дефинира съответствия между елементите на опората S и върховете на G. То присвоява етикети на елементите на $V_C \cup V_R$. На всеки *c*-връх $c \in V_C$ се съпоставя етикет *type(c):marker*(c), където *type(c)* $\in T_C$ и *marker*(c) $\in I \cup \{*\}$. Всеки *c*-връх с обобщен маркер се нарича *обобщен връх*, той обозначава неспецифициран екземпляр от указания тип понятие. Всеки *c*-връх с индивидуален маркер се нарича *индивидуален връх*, той обозначава конкретен екземпляр от указания тип понятие. Всеки *r*-връх $r \in V_R$ има за етикет тип релация $R \in T_R$. Първият аргумент на R се изобразява в *c*-върха, който е начало на входната дъга в r, а вторият аргумент на R се изобразява в *c*-върха, който е край на изходната дъга на r. За всеки *r*-връх $r \in V_R$ има точно една входна и една изходна дъга, инцидентни с r. □

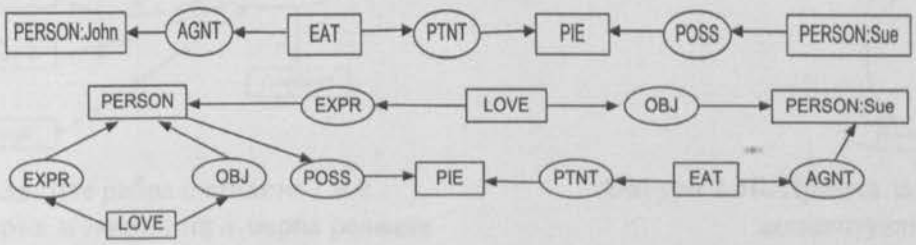
Пример 2.2. Фигура 2.1 илюстрира йерархиите и базиса в една примерна опора:

- T_C = {STATE, EVENT, ENTITY, ACT, ANIMATE, PHYS-OBJECT, LOVE, EAT, ANIMAL, PIE, PERSON} със съответната частична наредба;
- T_R = {AGNT, OBJ, EXPR, POSS, PTNT} с частична наредба и базис, показани на Фиг. 2.1. Наименованията на релациите са съкращения от AGeNT, OBJect, EXPeRiencer, POSSesor и PaTieNT. Тъй като PTNT≤OBJ, аргументите на PTNT в базиса са специализация на аргументите на OBJ, т.е. ACT≤ACT и PHYS-OBJECT≤ENTITY;
- I = {John, Sue};

- τ(John) = PERSON, τ(Sue) = PERSON. (По принцип тези два екземпляра могат да принадлежат и към ANIMAL, ANIMATE, PHYS-OBJECT и ENTITY, но тук ограничаваме обобщенията с цел опростяване на примера.)



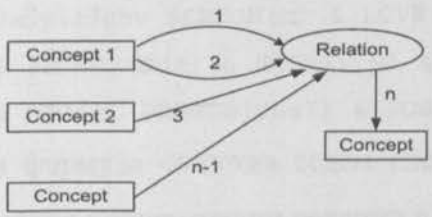
Фигура 2.1. Йерархии на типовете и базис за 5-те концептуални релации от пример 2.2



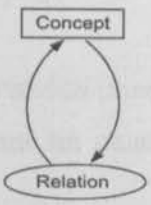
Фигура 2.2. Графично представяне на примерни прости концептуални графи G_1 и G_2

Фиг. 2.2 представя база от знания (БЗ) от два ПКГ G_1 и G_2, дефинирани над опората от Фиг. 2.1. Тяхното значение може да се изкаже на естествен език както следва: G_1 *'Джон яде пая на Сю'* и G_2 *'Съществува човек, който обича себе си и Сю, която му яде пая'*. По премълчаване обобщеният маркер може да се пропуска, напр. обобщеният екземпляр (EAT,*) в G_1 е изобразен с етикет EAT. В G_1 има индивидуални *c*-върхове – напр. PERSON:John. Графът G_2 е *цикличен* концептуален граф, тъй като съответният му двуделен граф е цикличен. □

Съгласно дефинициите в [Sow84 и ChMu92], ПКГ са мулти-графи и могат да имат по няколко ребра между даден *r*-връх и негов съседен *c*-връх. Това се случва при *n*-мерни концептуални релации, когато ребрата между *r*-върховете и съседните *c*-върхове са номерирани както е показано на Фиг. 2.3а. Но ние разглеждаме само двумерни релации с една входна и една изходна дъга в съответните *r*-върховете, така че конфигурацията на Фиг. 2.3а е недопустима. По принцип графите могат да съдържат и примки както е показано на Фиг. 2.3б, ако концептуалните релации имат един екземпляр и като първи, и като втори аргумент. Но при нас и този случай не е възможен, тъй като дефиниция 2.1 въвежда опората само за типове релации $R \in T_R$, които са в сила или между екземплярите на две различни понятия, или между два различни екземпляра на едно понятие. Следователно разглежданите тук ПКГ са насочени двуделни графи с най-много едно ребро между всеки два върха, които са свързани изобщо. Те могат да съдържат цикли (напр. G_2 на Фиг. 2.2).



Фигура 2.3а. Две ребра с етикети 1 и 2 между *c*-връх и *r*-връх на *n*-мерна релация



Фигура 2.3б. Примка за двумерна концептуална релация

Логическа интерпретация на простите концептуални графи

Дефиниция 2.3. [Sow84] Дефинираме оператора φ за превод на ПКГ с двумерни концептуални релации до формули на предикатното смятане от първи ред. Нека G е даден ПКГ; тогава φG е формула, конструирана както следва:

- Ако G съдържа k обобщени *c*-върха, присвояваме различни променливи x_1, x_2, …, x_k на всеки от тях;
- За всеки *c*-връх c от G, нека *identifier(c)* е променливата, присвоена на c ако c е обобщен връх или *marker(c)*, ако c е индивидуален връх;
- Представяме всеки *c*-връх c на G като едноместен предикат с име *type(c)* и аргумент *identifier(c)*;

- Представяме всеки r-връх r на G като двуместен предикат с име *type(r)*. Първият аргумент на предиката е *identifier*(c_1), където c_1 е c-върхът на G, свързан с входната дъга в r. Вторият аргумент на предиката е *identifier*(c_2), където c_2 е c-върхът на G, свързан с изходната дъга от r;
- Построяваме φG като съчленяваме кванторния префикс $\exists x_1 \exists x_2 \ldots \exists x_k$ към тялото на формулата, състоящо се от конюкция на всички едномерни и двумерни предикати, съпоставени на c-върховете и r-върховете на G. □

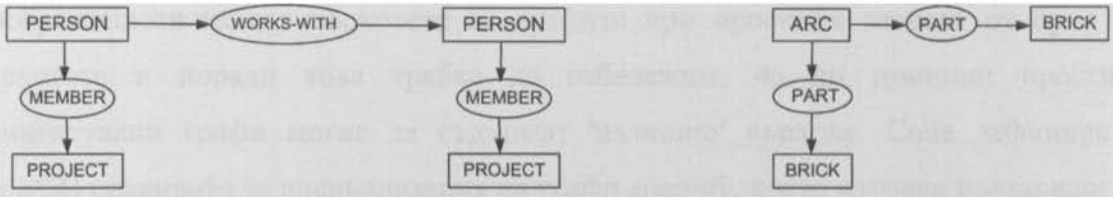
Логическата формула съдържа по един двумерен предикат за всяка концептуална релация r в графа. Ще наричаме тези предикати *елементарни конюнкти* и ще ги записваме като тройки $c_1\ r\ c_2$, където c_1 и c_2 са съответно 1-ви и 2-ри аргумент на r.

Пример 2.3. Тук е дадена логическата формула, съответстваща на G_2 от пример 2.2 *'Съществува човек, който обича себе си и Сю, която му яде пая'*:

```
∃x∃y∃z∃u∃v PERSON(x) & LOVE(y) & LOVE(z) & PIE(u) & EAT(v) &
& PERSON(Sue) & EXPR(x,y) & OBJ(y,x) & EXPR(x,z) &
& OBJ(z, PERSON(Sue)) & POSS(x,u) & PTNT(v,u) & AGNT(v, PERSON(Sue))
```

Тази формула съдържа седем елементарни конюнкта, съответстващи на седемте двумерни концептуални релации в G_2. □

Допустимо е множествата V_C и V_R да съдържат върхове с дублиращи се имена, тъй като изображението λ в дефиниция 2.2 може да присвои повтарящи се етикети на елементите на $V_C \cup V_R$. Фиг. 2.4 показва примерите: *'Съществува арка с две тухли като съставни части'* и *'Съществува човек, член на проект, който работи с друг човек, член на друг проект'*. Очевидно екземплярите, съответстващи на обобщените c-върхове BRICK, PERSON и PROJECT са различни, тъй като в логическите формули им се съпоставят различни екзистенциално-квантувани променливи. Но по дефиниция релациите свързват различни екземпляри и затова r-върховете за релациите PART и MEMBER също трябва да се дублират. Така формулите за двата ПКГ на Фиг. 2.4 съдържат конюнкти с дублирани имена на предикатите, които имат аргументи от различно-индексирани променливи. □



Фигура 2.4. Дублирани имена на върхове в ПКГ (примери от [Sow84] и [ChMu92]).

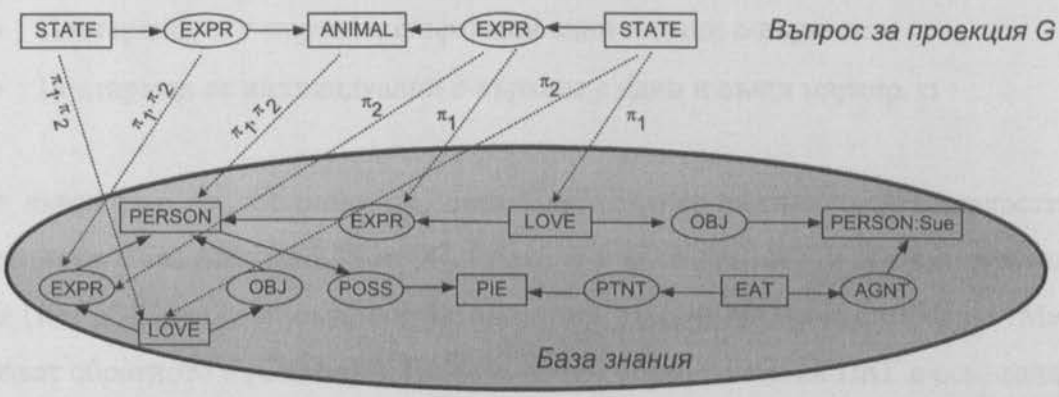
Проекция

Дефиниция 2.4. [Sow84] Нека G и H са два ПКГ с двумерни концептуални релации. Тогава **проекцията** π: $G \rightarrow H$ е ПКГ πG, където $\pi G \subseteq H$ и:

- За всяко понятие c в G, πc е понятие в πG и $type(\pi c) \leq type(c)$. Ако c е екземпляр с индивидуален маркер, тогава $c = \pi c$.
- За всяка концептуална релация $r(c_1,c_2)$ в G, πr е концептуална релация в πG като $type(\pi r) \leq type(r)$ и πr е в сила между πc_1 и πc_2, т.е. $\pi r(\pi c_1,\pi c_2)$ е в πG. □

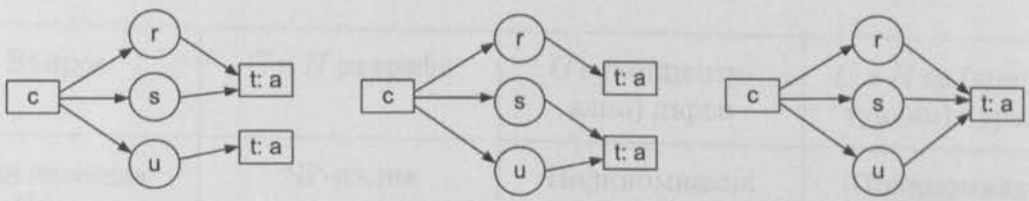
Дефиниция 2.5. [MuCh92] Нека G и H са два ПКГ. **Инективна проекция** π: $G \rightarrow H$ е такава проекция πG, където графът πG е изоморфен на G. □

Пример 2.4. Проекцията на ПКГ G върху ПКГ H намира в H концептуални шаблони, които са специализации на G. Инективната проекция осигурява извличане на изоморфни шаблони. Възможно е да има много проекции. Въпросът G на Фиг. 2.5 има празна проекция върху G_1 и няколко непразни проекции върху G_2. □



Фигура 2.5. Инективна проекция π_1 и проекция π_2 на въпрос G върху примерния G_2. При π_1, двата елементарни конюнкта на G са проектирани върху два различни конюнкта на G_2. При π_2, двата елементарни конюнкта на G са изобразени в един конюнкт на G_2

Изображенията между върховете на графите при проекция зависят от броя на върховете и поради това трябва да отбележим, че по принцип простите концептуални графи могат да съдържат 'излишни' върхове. Сова дефинира в [Sow84] правилото за специализация на графи *simplify*, което изтрива повтарящи се *r*-върхове между едни и същи екземпляри на понятия (тези *r*-върхове пораждат идентични предикати при логическата интерпретация и могат да бъдат изтрити, тъй като X&X=X). Освен 'излишни' *r*-върхове, ПКГ могат да съдържат излишно повторение на индивидуални *c*-върхове, както например графите *G* и *H* на Фиг. 2.6. Тези графи имат еднакви логически формули, и двата се проектират върху показаната на Фиг. 2.6 '*нормална форма*', но *G* не се проектира върху *H* и *H* не се проектира върху *G*. Повтарящите се индивидуални *c*-върхове могат лесно да бъдат обединени в линейно време [BaMu02].



ПКГ *G* ПКГ *H* '*нормална форма*' на *G* и *H*

$\varphi(G) = \varphi(H) = \exists x\, C(x)\ \&\ T(a)\ \&\ T(a)\ \&\ r(x,T(a))\ \&\ s(x,T(a))\ \&\ u(x,T(a))$

Фиг. 2.6. Графи *G* и *H*, които не могат да се сравняват чрез инективна проекция [BaMu02]

Дефиниция 2.6. [BaMu02] Даден ПКГ *G* е в **нормална форма** когато не съдържа:

- Повтарящи се *r*-върхове, свързващи едни и същи *c*-върхове и
- Повтарящи се индивидуални *c*-върхове с един и същи маркер. □

След въвеждане на дефинициите, нека коментираме важността и сложността на проекцията. Сова показва в [Sow84], че ако *H* и *G* са два ПКГ и *H* е специализация на *G* (т.е. $H \leq G$), тогава съществува проекция на *G* в *H*. По-късно Чен и Мюние доказват обратното в [ChMu92]. По този начин обработката на ПКГ е основана или върху проекцията, или върху правила за специализация и обобщение на графи.

От изчислителна гледна точка, пресмятането на проекцията е предизвикателство вече повече от 20 години. При зададен въпрос, неговите изображения върху графите в базата от знания се пресмятат граф по граф. Друга интересна задача е да се преброят всички проекции. Ако са дадени два ПКГ G и H, задачата да се определи дали $H \leq G$ е NP-пълна. Обаче има класове от ПКГ, за които съществуват полиномиални алгоритми за проекция, когато съответстващите им обикновени графи са дървета [MuCh92, Mug95]. Така най-ефективните подходи за извършване на проекция са всъщност приложение на алгоритми от теорията на графите.

Таблица 2.1 обобщава резултатите за сложност на задачата за изчисление на проекция. Показаните оценки важат за случая на пресмятания, които се извършват след постъпване на въпроса G в системата.

Въпрос	G и H са графи	G е (концептуално) дърво	G и H са (концептуални) дървета
Има ли проекция π: $G \rightarrow H$?	NP-пълна	Полиномиална	Полиномиална
Броят на проекциите π: $G \rightarrow H$ по-малък ли е от дадено число k?	NP-пълна	Полиномиална	Полиномиална
Има ли инективна проекция π: $G \rightarrow H$?	NP-пълна	NP-пълна	Полиномиална
Броят на инективните проекции π: $G \rightarrow H$ по-малък ли е от дадено число k?	NP-пълна	NP-пълна	NP-пълна

Таблица 2.1. Алгоритмична сложност на задачата за пресмятане на проекция с алгоритми, които извършват изчисления по време на изпълнение на операцията [Mug95]

2.3. Предварително кодиране на база от ПКГ като краен автомат

Ще работим с множеството от всички нормализирани ПКГ с двумерни концептуални релации в някое моментно състояние на база от знания, дефинирана спрямо опора *S*. Целта ни е да представим всеки ПКГ като дума в регулярен език. Изложението следва [AnMi08a] с примери от [AnMi08b и Ang09].

2.3.1 Линейно кодиране на ПКГ като низове от етикети на опората

Съгласно дефиниция 2.3, логическата формула на всеки ПКГ съдържа предикати $rel(c_1, c_2)$, където rel е етикет на *r*-връх и c_1, c_2 са или екзистенциално-квантувани променливи за обобщените *c*-върхове, или индивидуални маркери за индивидуалните *c*-върхове. Вече отбелязахме, че въвеждането на специфични за всеки граф индекси ще бъде пречка за разпознаване на еднаквите етикети в графите от базата и бъдещ въпрос за проекция. За да избегнем индексацията или специфичните имена на променливи, бихме могли да заместим всички променливи с имената на типовете понятия, към които те принадлежат. Така няма да имаме нужда от едномерни предикати. Освен това, типовете с индивидуални маркери – вложени във формулите като едноместни предикати – се представят с низове от вида *type*:*marker*, където *type* е етикетът на съответния *c*-връх в T_C (например на Фиг. 2.2 `PERSON:Sue` обозначава индивидуален *c*-връх от тип `PERSON`). Така едноместните предикати ще бъдат кодирани в двуместните. Тогава тялото на една логическа формула, състоящо се от конюнкции на двуместни предикати

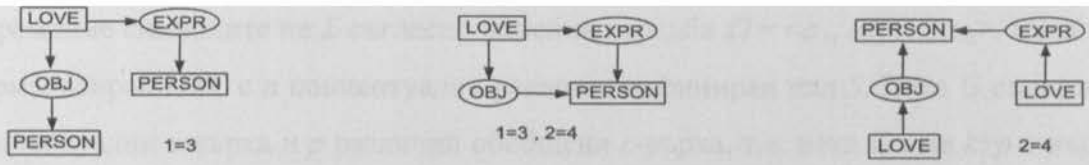
$$rel_1(concept_{11}, concept_{12}) \ \& \ \dots \ \& \ rel_n(concept_{n1}, concept_{n2})$$

може да бъде линеализирано като низ от символи – етикети на опората *S*:

$$concept_{11}\ rel_1\ concept_{12}\ \ concept_{21}\ rel_2\ concept_{22}\ \dots\ concept_{n1}\ rel_n\ concept_{n2}$$

където $concept_{ij}$, $1 \le i \le n$, j=1,2 са или етикети на типове, или низове *type*:*marker*. Обаче тази редица от етикети на *S* не показва кои са идентичните имена на върхове.

Пример 2.5. За да добием представа за проблемите, свързани с кодирането на топологичната структура на ПКГ, нека разгледаме Фиг. 2.7. Тя съдържа различни конфигурации на свързани елементарни конюнкти с дублиращи се *c*-върхове:



(а) (б) (в)

Фиг. 2.7. Два свързани конюнкта *expr(LOVE, PERSON) & obj(LOVE, PERSON)*: едни и същи етикети изразяват различни семантични твърдения за екземплярите на понятията

- Фиг. 2.7а: *Съществува любов, изпитвана от един човек и насочена към друг*,
- Фиг. 2.7б: *Съществува човек, който обича себе си*,
- Фиг. 2.7в: *Съществува човек, който изпитва любов и е обект на друга любов*.

Очевидно редицата от етикети на тройки, съответстващи на двата конюнкта

```
LOVE  EXPR  PERSON     LOVE  OBJ  PERSON
```

не указва кои са идентичните аргументи на двете релации. Забелязваме обаче, че можем да номерираме позициите на аргументите и след това да кодираме класовете от еквивалентни аргументи. Тогава индексът, описващ даден клас на еквивалентност, може да се присъедини към линейния запис на етикетите като *анотация*. По-долу показваме линеен низ и анотация за структурните конфигурации на Фиг. 2.7. Има 4 позиции на аргументи в линейния запис на етикетите на два свързани елементарни конюнкта; номерираме ги с 1, 2, 3 и 4. Тогава трите графа могат да бъдат кодирани както следва:

	Position 1		*Position 2*	*Position 3*		*Position 4*	*Annotation*
Фиг.2.7а:	LOVE	EXPR	PERSON	LOVE	OBJ	PERSON	1=3
Фиг.2.7б:	LOVE	EXPR	PERSON	LOVE	OBJ	PERSON	1=3,2=4
Фиг.2.7в:	LOVE	EXPR	PERSON	LOVE	OBJ	PERSON	2=4

Така анотацията, добавена към линейния запис, описва явно множествата от идентични аргументи и позволява да се различат графите на Фиг. 2.7 (а, б, в). Сега всеки ПКГ лесно може да се сведе до низ от етикети (имена на *понятия*, *релации* и имена на индивиди, записани във вида *понятие* : *индивид*) и съответна анотация. □

Дефиниция 2.7. Нека $S = (T_C, T_R, I, \tau)$ е опора на базата от знания. Дефинираме крайна азбука Σ с m символа, съставена от етикетите на опората (т.е.):

$$\Sigma = \{x \mid x \in T_C \text{ или } x \in T_R\} \cup \{x: i \mid x \in T_C,\ i \in I \text{ и } \tau(i)=x\}.$$

Нареждаме символите на Σ съгласно линейна наредба $\Omega = <a_1, a_2, \ldots, a_m>$. Нека G е нормализиран ПКГ с n концептуални релации, дефиниран над S. Нека G съдържа k индивидуални c-върха и p различни обобщени c-върха, т.е. нека G има $k+p$ c-върха, които заемат позициите на $2 \times n$ аргумента на релациите. Конструираме **линеен запис** на G както следва:

- Нека $f(G)$ е логическа формула, съпоставена на G съгласно дефиниция 2.3:

$\exists x_1 \exists x_2 \ldots \exists x_p \; type_1(x_1) \& \ldots \& type_p(x_p) \& type_{p+1}(marker_1) \& \ldots \& type_{p+k}(marker_k) \&$
$rel_1(concept_{11}, concept_{12}) \; \& \; \ldots \; \& \; rel_n(concept_{n1}, concept_{n2})$

  където $type_i \in T_C$ за $1 \le i \le p+k$, $rel_i \in T_R$ за $1 \le i \le n$, $marker_i \in I$ за $1 \le i \le k$ и $concept_{ij}$ за $1 \le i \le n$, $j=1,2$ е или една от променливите $x_1, x_2, \ldots, x_p$, или един от индивидуалните c-върхове $type_{p+1}(marker_1), \ldots, type_{p+k}(marker_k)$;

- За всички аргументи $concept_{ij}$, $1 \le i \le n$, $j=1,2$, които са равни на променлива x_u където $1 \le u \le p$, заместваме $concept_{ij}$ с низа $type_u : x_u$, където $type_u(x_u)$ е едноместен предикат в $f(G)$;

- За всички аргументи $concept_{ij}$, $1 \le i \le n$, $j=1,2$, които са равни на $type_{p+u}(marker_u)$, където $type_{p+u}(marker_u)$ е едноместен предикат в $f(G)$ за $1 \le u \le k$, заместваме $concept_{ij}$ с низа $type_{p+u} : marker_u$;

- Вземаме двуместните предикати в $f(G)$:

  $rel_1(concept_{11}, concept_{12}) \; \& \; \ldots \; \& \; rel_n(concept_{n1}, concept_{n2})$

  където $concept_{ij}$ за $1 \le i \le n$, $j=1,2$ е или един от обобщените c-върхове $type_1 : x_1$, $type_2 : x_2$, ..., $type_p : x_p$, или един от индивидуалните c-върхове $type_{p+1} : marker_1$, ..., $type_{p+k} : marker_k$

  Представяме ги като низ $seq(G)$ от n тройки-етикети с $3 \times n$ позиции:

  $concept_{11} \; rel_1 \; concept_{12} \; \ldots \; \ldots \; \ldots \; concept_{n1} \; rel_n \; concept_{n2}$

  Абстрахираме се от поднизовете $': x_1', \ldots, ': x_p'$ в $'type_1 : x_1', \ldots, 'type_p : x_p'$ съответно, като разглеждаме за момента $seq(G)$ като низ от символи на Σ, и сортираме $seq(G)$ тройка по тройка относно Ω. Когато пренареждаме етикетите $'type_1', 'type_2', \ldots, 'type_p'$ в процеса на сортирането, местим заедно с тях асоциираните поднизове - имена на променливи $': x_1', \ldots, ': x_p'$. Ако две съседни тройки се състоят от идентични символи на Σ, нареждаме

ги съгласно нарастващия ред на индексите на променливите $x_1, \ldots, x_p$, които присъстват като поднизове в аргументите $'type_1:x_1', \ldots, 'type_p:x_p'$;

- Записваме сортирания низ от етикетите на n тройки в $sortedSeq(G)$. Без загуба на общност, низът $sortedSeq(G)$ може да се означи като

$$concept_{11}\ rel_1\ concept_{12}\ \ldots\ concept_{n1}\ rel_n\ concept_{n2}.$$

Присвояваме индекс за позиция $v=2(i-1)+j$ на всеки аргумент $concept_{ij}$ за $1 \le i \le n, j=1,2$. Тогава $sortedSeq(G)$ може да се разглежда като

$$argument_1\ rel_1\ argument_2\ \ldots\ argument_{2n-1}\ rel_n\ argument_{2n}$$

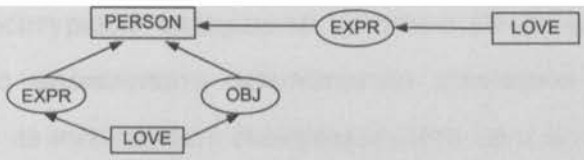
Построяваме класове от индекси както следва: за всяко множество от q идентични аргумента, където $argument_{v1}=argument_{v2}=\ldots\ldots=argument_{vq}$ и $2 \le q \le 2 \times n$, конструираме низ $V='v_1=v_2=\ldots=v_q'$, където $1 \le v_i \le 2 \times n$ за $i=1,2,\ldots,q$ и $v_1<v_2<\ldots<v_q$. Низът V е **клас на еквивалентни аргументи** за $sortedSeq(G)$.

- Нека $sortedSeq(G)$ има z различни множества от еквивалентни аргументи $\Re_1, \Re_2, \ldots, \Re_z$. За всяко множество $\Re_i$, нека съответният клас на еквивалентни аргументи бъде V_i, което е низ от цифри и символа '=' за $1 \le i \le z$. Сортираме $[V_1, \ldots, V_z]$ и от пренаредения списък $[V_1, V_2, \ldots, V_z]$ конструираме низа

$$annotation(G) = 'V_1, V_2, \ldots, V_z'$$

- В $sortedSeq(G)$ изтриваме поднизовете $':x_1', ':x_2', \ldots, ':x_p'$ от аргументите $concept_{ij}$, $1 \le i \le n, j=1,2$ които имат вида $'type_1:x_1', \ldots, 'type_p:x_p'$. След изтриването, $sortedSeq(G)$ съдържа само символи от Σ - етикети от $T_C \cup T_R \cup I$.
- Наредената двойка $<sortedSeq(G), annotation(G)>$ ще наричаме **линеен запис** на G относно $f(G)$ и Ω. □

Пример 2.6. В пример 2.5 са показани линейните записи на трите ПКГ от Фиг. 2.7. Те са единствени относно наредбата EXPR<LOVE<OBJ<PERSON, тъй като елементарните конюнкти на трите ПКГ съдържат две различни релации EXPR<OBJ и сортирането нарежда тройките по еднозначен начин – следователно, редът на аргументите е еднозначно-определен и като следствие анотациите са единствени. Но е лесно да намерим примери на ПКГ с множество възможни анотации, които се дължат на повтарящи се идентични тройки от символи на опората. Да разгледаме



Фигура 2.8. ПКГ с множество еквивалентни логически формули (поради изоморфизъм)

един подграф на G_2, даден на Фиг. 2.8, с две тройки 'LOVE EXPR PERSON', в които има два различни обобщени екземпляра от типа LOVE. Очевидно за този граф могат да се построят различни логически формули, в зависимост от присвояването на променливи на обобщените *c*-върхове. Ако x_1 се съпостави на PERSON и x_2-x_3 се съпоставят на двата обобщени екземпляра на LOVE, графът от Фиг. 2.8 ще има две логически формули поради изоморфизма между променливите x_2 и x_3:

$\exists x_1 \exists x_2 \exists x_3$ `PERSON(`x_1`)&LOVE(`x_2`)&LOVE(`x_3`)&expr(`x_2`,`x_1`)&expr(`x_3`,`x_1`)&obj(`x_2`,`x_1`)` и

$\exists x_1 \exists x_2 \exists x_3$ `PERSON(`x_1`)&LOVE(`x_2`)&LOVE(`x_3`)&expr(`x_2`,`x_1`)&expr(`x_3`,`x_1`)&obj(`x_3`,`x_1`)`.

Две анотации ще бъдат построени с използването на тези формули: съответно `'1=5,2=4=6'` и `'3=5,2=4=6'`. Но в момента ние не дискутираме изоморфизма между променливите. Вместо това предполагаме, че даден ПКГ G е вече кодиран като логическа формула и от нея започва конструирането на линейния запис на G. След построяването на анотацията, променливите се изтриват от записа и така изчезват и специфичните за графа индекси. Нека отбележим, че по принцип може да има много линейни представяния на етикетите на даден ПКГ G, в случай че не се изисква *sortedSeq* от дефиниция 2.7 да бъде сортиран низ. Съответните низове *annotation* осигуряват реконструкцията на изходния ПКГ. □

Лема 2.1. По дадени логическа формула $f(G)$ на ПКГ в нормална форма G и линейна наредба Ω на етикетите на опората, може да се построи единствен линеен запис на G. По даден линеен запис на ПКГ H, могат да се конструират логическа формула $f(H)$ и графично представяне на H.

Доказателство. Следва от конструкцията в дефиниция 2.7 и факта, че съществуват множество еквивалентни логически формули за даден ПКГ (поради изоморфизма между променливите), но фиксирането на една конкретна формула $f(G)$ прави записа еднозначен. □

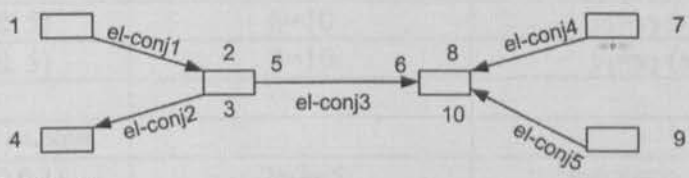
Анотациите, които осигуряват кодирането на топологичната структура на даден ПКГ, изброяват явно множествата еквивалентни аргументи на релациите. Така задачата за описание на възможните еквивалентности на n аргумента се свежда до задачата за разделяне на множество от n елемента в непресичащи се подмножества, като всяко подмножество описва клас еквивалентни аргументи. Броят на подмножествата се дава от числата на Бел B_1, B_2,... [MathWorld]. За нас са важни числата на Бел с четен индекс, тъй като броят на аргументите на двумерните релации е винаги четно число. В [AnMi08b] са разгледани 15-те начина за разделяне на множество от 4 елемента $\{x_1, y_1, x_2, y_2\}$ на непресичащи се подмножества. В Таблица 2.2 изброяваме тези подмножества и коментираме тяхната релевантност към решаваната от нас задача за описание на еквивалентни аргументи на две релации $r_1(x_1, y_1)$ & $r_2(x_2, y_2)$ при прости концептуални графи. Вижда се, че само

Разбиване на подмножества	**Аналогия с описанието на еквивалентни аргументи в ПКГ – примери и коментари**
1. $\{\{x_1\},\{y_1\},\{x_2\},\{y_2\}\}$	Не е релевантно – съответства на несвързан ПКГ
2. $\{\{x_1, y_1\}, \{x_2\}, \{y_2\}\}$	x_1=y_1 е невъзможно – не се допускат примки
3. $\{\{x_1, x_2\}, \{y_1\}, \{y_2\}\}$	Графът на Фиг. 2.7А е от този вид
4. $\{\{x_1\}, \{y_1, x_2\}, \{y_2\}\}$	Подграф на G_2: LOVE EXPR PERSON PERSON POSS PIE
5. $\{\{x_1, y_1, x_2\}, \{y_2\}\}$	x_1=y_1 е невъзможно – не се допускат примки
6. $\{\{x_1, y_2\}, \{y_1\}, \{x_2\}\}$	PERSON POSS PIE PHYS-OBJECT POSS PERSON Граф над примерния базис от Фиг. 2.1, означаващ (грубо) '*Съществува човек, който притежава пай и е притежаван от някакъв физически обект*'
7. $\{\{x_1\}, \{y_1, y_2\}, \{x_2\}\}$	Графът на Фиг. 2.7В е от този вид
8. $\{\{x_1\}, \{y_1\}, \{x_2, y_2\}\}$	x_2=y_2 е невъзможно – не се допускат примки
9. $\{\{x_1, y_1, y_2\}, \{x_2\}\}$	x_1=y_1 е невъзможно – не се допускат примки
10. $\{\{x_1, y_1\}, \{x_2, y_2\}\}$	x_1=y_1, x_2=y_2 са невъзможни – не се допускат примки
11. $\{\{x_1, x_2, y_2\}, \{y_1\}\}$	x_2=y_2 е невъзможно – не се допускат примки
12. $\{\{x_1, x_2\}, \{y_1, y_2\}\}$	Графът на Фиг. 2.7Б е от този вид
13. $\{\{x_1, y_2\}, \{y_1, x_2\}\}$	Няма такъв пример в базиса от Фиг. 2.1 – това са конюнкти $r_1(x_1,y_1)$ & $r_2(y_1,x_1)$ и r_1<r_2 според нередбата
14. $\{\{x_1\}, \{y_1, x_2, y_2\}\}$	x_2=y_2 е невъзможно – не се допускат примки
15. $\{\{x_1, y_1, x_2, y_2\}\}$	x_1=y_1=x_2=y_2 са невъзможни – не се допускат примки

Таблица 2.2. Класове на еквивалентност при разбиване на множество от 4 елемента на непресичащи се подмножества и съответни класове от еквивалентни аргументи на примерни ПКГ при наредба EXPR<LOVE<OBJ<PERSON<PHYS-OBJECT<PIE<POSS

шест подмножества от възможните 15 могат да се интерпретират като описание на класове еквивалентни аргументи на ПКГ с два елементарни конюнкта $r_1(x_1, y_1)$ & $r_2(x_2, y_2)$. Пример 2.7 показва, че в един конкретен ПКГ с 5 елементарни конюнкта има само 10 вида структурни конфигурации (а $B_{10} = 115975$). При проведените експерименти с ПКГ с до 10-12 елементарни конюнкта се оказва, че структурната вариативност на идентичните аргументи при ПКГ е много по-ограничена, отколкото са съответните числа на Бел (B_{20} и B_{24}, вж. част 2.6).

Пример 2.7. Ще изброим структурните конфигурации на подграфите в един ПКГ с 5 елементарни конюнкта, даден на Фиг. 2.9. Фокусираме се върху елементарните конюнкти, наименовани *el-conj1*, *el-conj2*, *el-conj3*, *el-conj4* и *el-conj5*. За удобство са показани само *c*-върховете. Те са номерирани от 1 до 10 според посоката на концептуалните релации, конюнкт по конюнкт, независимо от факта, че някои *c*-върхове са идентични като аргументи в повече от една релация. Например *el-conj1* има *c*-върхове с номера 1 и 2, където *c*-върхът 1 съответства на екземпляра в началото на входната дъга в съответната релация r_1 (която не е показана на фигурата) и аргумент 2 съответства на екземпляра в края на изходната дъга от r_1.



Фигура 2.9. ПКГ с 5 елементарни конюнкта и *c*-върхове, номерирани от 1 до 10.

Таблица 2.3 изброява 22-та подграфа-ПКГ на показания на Фиг. 2.9 ПКГ. Дадени са и структурните конфигурации в подграфите, като се предполага, че елементарните конюнкти във всеки подграф-ПКГ са записани чрез линеен низ на етикетите в наредба съответстваща на индекса на конюнктите *el-conj1 el-conj2 ... el-conj5*. Има 6 подграфа-ПКГ с по два елементарни конюнкта, които са изброени в таблицата с конкретните индекси на идентичните *c*-върхове. Но ако представим тези ПКГ с по два елементарни конюнкта като $r_1(x_1,y_1)\&r_2(x_2,y_2)$, виждаме 'типологията' при избраната наредба на тройките в линейния запис – структурите са 3 вида:

- *mun2-1* '$y_1=x_2$',

- *тип 2-2* '$x_1=x_2$' и
- *тип2-3* '$y_1=y_2$'.

За 6-те подграфа с по 3 конюнкта, представени като $r_1(x_1,y_1)\&r_2(x_2,y_2)\&r_3(x_3,y_3)$, има 4 типа структурни връзки. За 4-те подграфа с 4 конюнкта има 3 типа конфигурации. Експерименти с милиони случайно-генерирани ПКГ са представени в част 2.6. □

Подграфи-ПКГ, изброени чрез номерата на елементарни конюнкти	*Идентични аргументи според номерата на с-върховете от Фиг. 2.9*	*Топологична структура на подграфите при представяне във вид* $r_1(x_1,y_1)$ & ... & $r_k(x_k,y_k)$ *за* k=2,3,4,5
Подграфи от 1 ел. кон.		
*el-conj*1	-	-
*el-conj*2	-	-
*el-conj*3	-	-
*el-conj*4	-	-
*el-conj*5	-	-
Подграфи от 2 ел. кон.		
el-conj(1 & 2)	2=3	$y_1=x_2$ (*тип 2-1*)
el-conj(1 & 3)	2=5	$y_1=x_2$ (*тип 2-1*)
el-conj(2 & 3)	3=5	$x_1=x_2$ (*тип 2-2*)
el-conj(3 & 4)	6=8	$y_1=y_2$ (*тип 2-3*)
el-conj(3 & 5)	6=10	$y_1=y_2$ (*тип 2-3*)
el-conj(4 & 5)	8=10	$y_1=y_2$ (*тип 2-3*)
Подграфи от 3 ел. кон.		
el-conj(1&2&3)	2=3=5	$y_1=x_2=x_3$ (*тип 3-1*)
el-conj(1&3&4)	2=5,6=8	$y_1=x_2$, $y_2=x_3$ (*тип 3-2*)
el-conj(1&3&5)	2=5,6=10	$y_1=x_2$, $y_2=x_3$ (*тип 3-2*)
el-conj(2&3&4)	3=5,6=8	$x_1=x_2$, $y_2=x_3$ (*тип 3-3*)
el-conj(2&3&5)	3=5,6=10	$x_1=x_2$, $y_2=x_3$ (*тип 3-3*)
el-conj(3&4&5)	6=8=10	$y_1=y_2=y_3$ (*тип 3-4*)
Подграфи от 4 ел. кон.		
el-conj(1&2&3&4)	2=3=5,6=8	$y_1=x_2=x_3$, $y_3=y_4$ (*тип 4-1*)
el-conj(1&2&3&5)	2=3=5,6=10	$y_1=x_2=x_3$, $y_3=y_4$ (*тип 4-1*)
el-conj(1&3&4&5)	2=5,6=8=10	$y_1=x_2$, $y_2=y_3=y_4$ (*тип 4-2*)
el-conj(2&3&4&5)	3=5,6=8=10	$x_1=x_2$, $y_2=y_3=y_4$ (*тип 4-3*)
Подграфи от 5 ел. кон.		
el-conj(1&2&3&4&5)	2=3=5,6=8=10	

Таблица 2.3. Подграфи-ПКГ и типове връзки между идентични *с*-върхове.

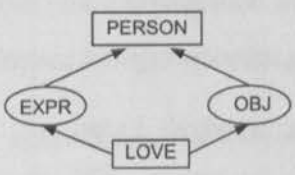
2.3.2. Предварително конструиране на минимален ацикличен краен автомат с маркери на заключителните състояния

Тук се предлага да се изброят явно всички възможни инективни обобщения на подграфи от базата, а това са фактически всички възможни въпроси за инективна проекция, които имат непразен отговор в дадената база от знания. Ще покажем как този концептуален ресурс може да бъде компресиран като минимален ацикличен краен автомат с маркери на заключителните състояния. Долната дефиниция ни помага да разпознаем и обработим на предварителната фаза само тези подграфи, които имат концептуална интерпретация спрямо опората.

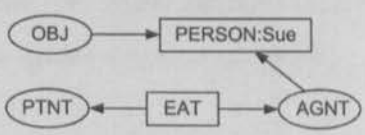
Дефиниция 2.8. Нека G е ПКГ с двумерни концептуални релации в база от знания, дефинирана над опора S. **Концептуален подграф** на G ще наричаме свързан граф G_{cs} със следните свойства:

- Като обикновен граф, $G_{cs} \subseteq G$ и
- G_{cs} е ПКГ, дефиниран над опората S. □

Пример 2.8. Фиг. 2.10а показва концептуален подграф на G_2, а графът на Фиг. 2.10б 'няма концептуален смисъл' над коя да е опора. По-долу под 'подграфи' ще разбираме концептуалните такива. □



Фигура 2.10а. Концептуален подграф на G_2



Фигура 2.10б. Свързани върхове на G_2, които не съставят концептуален подграф

Ще представим алгоритъм за конструиране на минимален ацикличен краен автомат с маркери на заключителните състояния, който кодира всички подграфи на базата от знания и техните инективни обобщения над опората. Съгласно твърдение 1.2, задаването на списък от думи (т.е. един краен регулярен език L), е достатъчно за дефиниране на ацикличен краен автомат, който разпознава L. Базата от знания в дадено състояние съдържа краен брой ПКГ с краен брой (концептуални) подграфи,

които имат краен брой инективни обобщения над опората. Следователно, можем да концентрираме усилията си върху построяване на краен регулярен език - списък от думи, които кодират всички подграфи и техните инективни обобщения. Ще използваме азбуката на всички етикети на опората, сортирана спрямо линейна наредба. Списъкът от думи също ще бъде лексикографски сортиран спрямо тази наредба. След конструиране на крайния регулярен език, ще приложим резултати от теорията на автоматите, за да конструираме минималния ацикличен краен автомат с маркери на заключителните състояния, който разпознава този език.

В алгоритъма се използват следните типове данни:

CHAR-типове: *sortedSeq*, *annotation*, *new_lin_labels*;

Списъци: *list_alternative_annot* – списък от низове, съставени от символите

{'1', '2',…,'9', '0', '=', ','};

Масив от списъци: *list_subgraphs*, *list_gen_graphs* ;

Масиви: *words_markers*(CHAR, <CHAR,CHAR,CHAR,CHAR>) и

sorted_words_markers(CHAR, {<CHAR,CHAR,CHAR,CHAR>,…, <CHAR, CHAR,CHAR,CHAR>}).

Алгоритъм 2.1. *Конструиране на минимален ацикличен краен автомат с маркери на заключителните състояния* $A_{KB} = (\Sigma, Q, q_0, F, \Delta, E, \mu_D)$, *който кодира всички инективни обобщения на подграфи в база от знания от нормализирани ПКГ с двумерни концептуални релации* $\{G_1, G_2, ..., G_n\}$, *дефинирани спрямо опора S.*

/* *Стъпка 1, дефиниране на крайна азбука* Σ: */

Нека $S = (T_C, T_R, I, \tau)$ е опора на $\{G_1, ..., G_n\}$. Нека $\Sigma = \{ x \mid x \in T_C$ или $x \in T_R \} \cup \{ x:i \mid x \in T_C, i \in I$ и $\tau(i)=x \}$. Нареждаме символите на Σ спрямо линейна наредба $\Omega = <a_1, a_2, \ldots, a_m>$. Тогава Σ е наредена азбука.

/* *Стъпка 2, индексиране на всички c-върхове в БЗ:* */

Съпоставяме различни индекси – цели числа – на всички *c*-върхове в базата от знания $\{G_1, G_2, ..., G_n\}$, с цел да осигурим тяхното третиране като различни екземпляри на типове понятия (тогава линейните записи на подграфи от базата не съдържат дублирани тройки от етикети на типове). Дефинираме азбука от символи на опората с индекси съгласно екземплярите на понятията в ПКГ:

$\Sigma_{KB} = \{a_{ij} | a_i \in \Sigma$, $1 \le i \le m$ и j е индекс на някой c-връх в базата с етикет $a_i, j \in \{u_1,\ldots,u_v\}$

където $u_1, u_2,\ldots,u_v$ са всички индекси, присвоени на a_i в базата или j='*none*' когато в базата няма индекс, съпоставен на $a_i\}$.

Нареждаме символите на Σ_{KB} съгласно линейната наредба

$\Omega_{KB} = <a_{1s1},\ldots, a_{1su}, a_{2p1},\ldots, a_{2pv}, \ldots.., a_{mq1},\ldots, a_{mqx}>$ където $s_1,s_2,\ldots s_u$ са индексите, присвоени на a_1; $p_1,\ldots,p_v$ са индексите, присвоени на a_2; ...., $q_1,\ldots q_x$ са индексите, присвоени на a_m и $s_1<s_2<\ldots<s_u$, $p_1<p_2<\ldots<p_v$, ..., и $q_1<q_2<\ldots<q_x$.

Така Σ_{KB} е наредена азбука. Дефинираме изображение λ: $\Sigma_{KB} \to \Sigma$ където $\lambda(a_{ij})=a_i$ за всяко $a_{ij} \in \Sigma_{KB}$, $1 \le i \le m$ и j е индекс, присвоен в Σ_{KB} на символа $a_i \in \Sigma$.

/* *Стъпка 3, намиране на всички (концептуални) подграфи в базата от знания*: */

for $i := 1$ **to** n **do begin**

list_subgraphs(i) := { $G^{sub\text{-}j}{}_i$ | $G^{sub\text{-}j}{}_i$ е концептуален подграф на G_i }; **end;**

/* *Стъпка 4, пресмятане и кодиране на всички инективни обобщения в масива sorted_words_markers:* */

var *main_index* := 1;

for each i **and** $G^{sub\text{-}j}{}_i$ в *list_subgraphs(i)* **do begin**

$<sortedSeq(G^{sub\text{-}j}{}_i), annotation(G^{sub\text{-}j}{}_i)>$:= `COMPUTE_LINEAR_RECORD`($G^{sub\text{-}j}{}_i$, Σ_{KB});

/* в $sortedSeq(G^{sub\text{-}j}{}_i)$ няма дублиращи се тройки от символи на Σ_{KB} */

list_gen_graphs(*i*,*j*) :=
`COMPUTE_INJ_GEN`($sortedSeq(G^{sub\text{-}j}{}_i)$, $annotation(G^{sub\text{-}j}{}_i)$, Σ_{KB}, Σ, λ);

/* всички инективни обобщения $G^{gen}{}_1$, $G^{gen}{}_2$, …, $G^{gen}{}_q$ на $G^{sub\text{-}j}{}_i$ се записват като тройки от символи на Σ в *list_gen_graphs*(*i*,*j*). При това k-тата тройка на $G^{gen}{}_1$, $G^{gen}{}_2$, …, $G^{gen}{}_q$ е изчислена като обобщение на k-тата тройка от $sortedSeq(G^{sub\text{-}j}{}_i)$. Топологичната структура на $G^{gen}{}_p$ е зададена в $annotation(G^{sub\text{-}j}{}_i)$ за $1 \le p \le q$ */

for each *i*,*j* и $G^{gen}{}_p$ в *list_gen_graphs*(*i*,*j*) **do begin**

$<sortedSeq(G^{gen}{}_p), annotation(G^{gen}{}_p), new\_lin\_labels(G^{sub\text{-}j}{}_i)>$:=
`ENSURE_PROJ_MAPPING`($sortedSeq(G^{sub\text{-}j}{}_i)$, $annotation(G^{sub\text{-}j}{}_i)$, Σ_{KB}, $G^{gen}{}_p$, Σ, λ);

list_alternative_annot :=
`COMPUTE_ISOMORPHISMS`($<sortedSeq(G^{gen}{}_p), annotation(G^{gen}{}_p)>$);

words_markers(*main_index, 1)* := *sortedSeq*($G^{gen}{}_p$);

words_markers(*main_index, 2)* := < *annotation*($G^{gen}{}_p$), *list_alternative_annot*, *new_lin_labels*($G^{sub\text{-}j}{}_i$), G_i>;

main_index := *main_index*+1; **end;**

end;

sorted_words_markers := **SORT-BY-FIRST-COLUMN**(*words_markers*) ;

/* обединение на дублирани редове в стълб 1 на масива *sorted_words_markers* */

while *sorted_words_markers*(*,1) съдържа *k>1 повтарящи се низа в стълб 1, започващи от ред p* **do begin**

sorted_words_markers(*p, 2*) := {*sorted_words_markers*(*p*,2), *sorted_words_markers*(*p*+1,2),…, *sorted_words_markers*(*p*+*k*-1,2)};

for 1≤*s*≤*k*-1 **do begin** **DELETE-ROW**(*sorted_words_markers*(*p*+*s*,*) **end**; **end**;

/* дефиниция на краен списък от думи над символите на Σ */

$L = \{w_1, \ldots, w_z \mid w_i \in$ *sorted_words_markers*(*,1), $1 \le i,j \le z$ и $w_i \le w_j$ относно Ω за $i \le j\}$.

/* <u>Стъпка 5, конструиране на минимален краен автомат</u>: */

Разглеждаме L като краен език над Σ, зададен от z думи, които са лексикографски сортирани спрямо Ω. На всяка дума $w_i \in L$ е съпоставен маркер, записан в *sorted_words_markers*(*i*,2) за $1 \le i \le z$.

Прилагаме алгоритъм от [Мих00, DMWW00] и построяваме директно минималния краен автомат с маркери на заключителните състояния $A_{KB}=\langle \Sigma, Q, q_0, F, \Delta, E, \mu \rangle$, който разпознава $L=\{w_1, w_2, \ldots, w_z\}$. По този начин

$F=\{q_{wi} | q_{wi}$ е край на пътя, започващ в q_0 с етикет w_i, за $w_i \in L$, $1 \le i \le z\}$,

$E=\{ M_i \mid M_i =$ *sorted_words_markers*$(i,2)$, $1 \le i \le z\}$ и

μ: $q_{wi} \rightarrow M_i$ където $q_{wi} \in F$, *sorted_words_markers*$(i,1) = w_i$ и

sorted_words_markers$(i,2) = M_i$ за $1 \le i \le z$. □

В алгоритъм 2.1 се използват следните функции:

function <*sortedSeq*(*G*), *annotation*(*G*)> = **COMPUTE_LINEAR_RECORD**(*G*, Σ).

Входните параметри задават един ПКГ в нормална форма *G*, представен като логическа формула, и наредена крайна азбука от символите на опората Σ. Тази функция построява линейния запис на *G*, следвайки конструкцията в дефиниция 2.7, и връща двойката низове <*sortedSeq*(*G*), *annotation*(*G*)>.

function *list_gen_graphs*(*i*, *j*) =
COMPUTE_INJ_GEN(*sortedSeq*($G^{sub\text{-}j}{}_i$), *annotation*($G^{sub\text{-}j}{}_i$), Σ_1, Σ_2, λ).

По даден линеен запис на ПКГ $G^{sub\text{-}j}{}_i$ в наредената азбука Σ_1, тази функция връща списък от етикетите на тройките на всички *q* инективни обобщения $G^{gen}{}_1$, $G^{gen}{}_2$,..., $G^{gen}{}_q$ на $G^{sub\text{-}j}{}_i$, пресметнати в наредената азбука Σ_2. Обобщенията са изчислени чрез изображението λ: $\Sigma_1 \rightarrow \Sigma_2$, което показва как символите на Σ_1 се обобщават от символите на Σ_2. Обобщенията в Σ_2 се изчисляват като низ от етикети на тройки, където *k*-тата тройка на $G^{gen}{}_1$, $G^{gen}{}_2$, ..., $G^{gen}{}_q$ обобщава *k*-тата тройка на *sortedSeq*($G^{sub\text{-}j}{}_i$). Низът от етикети на тройки на обобщенията може да не е сортиран спрямо Σ_2. Топологичната структура на $G^{gen}{}_1$, $G^{gen}{}_2$, ..., $G^{gen}{}_q$ е кодирана чрез *annotation*($G^{sub\text{-}j}{}_i$) тъй като редът на тройките в $G^{sub\text{-}j}{}_i$ и в неговите инективни обобщения е един и същ. Тъй като Σ_1 е крайна азбука, съдържаща различни индекси за всички *c*-върхове на $G^{sub\text{-}j}{}_i$, низът *sortedSeq*($G^{sub\text{-}j}{}_i$) не съдържа дублирани тройки от етикети на елементарни конюнкти. Инективните обобщения в азбуката Σ_2 могат да съдържат дублирани тройки, но редът на последните съответства на наредбата на тройките в $G^{sub\text{-}j}{}_i$. Така *annotation*($G^{sub\text{-}j}{}_i$) указва и идентичните екземпляри в обобщенията $G^{gen}{}_1$, $G^{gen}{}_2$,..., $G^{gen}{}_q$ на $G^{sub\text{-}j}{}_i$.

function <*sortedSeq*(G^{gen}), *annotation*(G^{gen}), *new_lin_labels*(*G*)> =
ENSURE_PROJ_MAPPING(*sortedSeq*(*G*), *annotation*(*G*), Σ_1, G^{gen}, Σ_2, λ)

Даден е линейния запис на ПКГ *G* - <*sortedSeq*(*G*), *annotation*(*G*)> - в наредена азбука Σ_1. Низът от етикети на инективното обобщение G^{gen} е даден в наредената азбука Σ_2, при което *k*-тата тройка на G^{gen} обобщава *k*-тата тройка на *sortedSeq*(*G*). Обобщението G^{gen} е изчислено с използване на изображението λ: $\Sigma_1 \rightarrow \Sigma_2$, което показва как символите на Σ_1 се обобщават от символите на Σ_2. Функцията **ENSURE_PROJ_MAPPING** извършва следното:

(1) Сортира G^{gen} спрямо Σ_2 и пресмята линейния запис на G^{gen}: <*sortedSeq*(G^{gen}), *annotation*(G^{gen})>,

(2) Проверява дали наредбата на *c*-върховете в *sortedSeq*(G^{gen}) съответства на наредбата на съответните специализирани *c*-върхове в *sortedSeq*(G). Ако това е така, се извършва присвояването *new_lin_labels*(G)=*sortedSeq*(G). Ако това не е така, функцията **ENSURE_PROJ_MAPPING** пренарежда символите в *sortedSeq*(G) по такъв начин, че *i*-тият символ да бъде специализация на *i*-тия символ в *sortedSeq*(G^{gen}) и записва пренаредените символи в *new_lin_labels*(G). И в двата случая, *new_lin_labels*(G) съдържа линейно-подредени тройки от етикети на елементарните конюнкти на G. Идентичните екземпляри на *new_lin_labels*(G) са кодирани чрез *annotation*(G^{gen}). По този начин е запомнена една инективна проекция π: $G^{gen} \rightarrow G$ от обобщение G^{gen} в Σ_2 към ПКГ G, зададен в Σ_1.

function *list_alternative_annot* =
COMPUTE_ISOMORPHISMS(<*sortedSeq*(G), *annotation*(G)>).

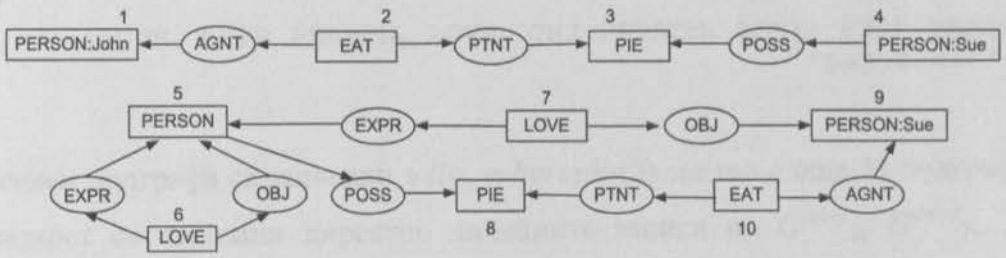
По даден линеен запис на ПКГ G, където *sortedSeq*(G) съдържа дублирани тройки от етикети на елементарни конюнкти, тази функция конструира сортиран списък от всички алтернативни анотации. Тези анотации съответстват на алтернативни конфигурации на идентичните екземпляри на обобщените *c*-върхове и отразяват възможните изоморфизми между променливите в логическата формула, от която е изчислен линейният запис на G. В случай че *list_alterantive_annot* е празен списък, G има само една анотация, която е вече изчислена при пресмятане на линейния запис и е записана в *annotation*(G).

Пример 2.9. Ще илюстрираме работата на Алгоритъм 2.1 над базата от знания в пример 2.2. На стъпка 1 се дефинира крайната азбука Σ, която се състои от етикетите на опората. Макар че от техническа гледна точка етикетите са низове от латински букви, ние ще ги разглеждаме като един символ. Можем да ги подредим в нарастващ ред съгласно линейната наредба в латинската азбука:

Σ = {ACT, AGNT, ANIMAL, ANIMATE, EAT, ENTITY, EVENT, EXPR, LOVE, OBJ, PERSON, PERSON:John, PERSON:Sue, PHYS-OBJECT, PIE, POSS, PTNT, STATE}

Ω: ACT<AGNT<ANIMAL<ANIMATE<EAT<ENTITY<EVENT<EXPR<LOVE<
OBJ<PERSON<PERSON:John<PERSON:Sue< PHYS-OBJECT<PIE<POSS<
<PTNT<STATE

На стъпка 2 се присвояват уникални индекси от 1 до 10 на 10-те различни *c*-върха в базата (така всички подграфи ще имат уникални линейни записи):



Дефинираме нова азбука Σ_{KB} с индекси и съответна нова линейна наредба Ω_{KB}:

Σ_{KB}={ ACT, AGNT, ANIMAL, ANIMATE, EAT_2, EAT_{10}, ENTITY, EVENT, EXPR, $LOVE_6$, $LOVE_7$, OBJ, $PERSON_5$, $PERSON:John_1$, $PERSON:Sue_4$, $PERSON:Sue_9$, PHYS-OBJECT, PIE_3, PIE_8, POSS, PTNT, STATE }

с наредба

Ω_{KB}: ACT<AGNT<ANIMAL<ANIMATE<EAT_2<EAT_{10}<ENTITY<EVENT<EXPR<
<$LOVE_6$<$LOVE_7$<OBJ<$PERSON_5$<$PERSON:John_1$<$PERSON:Sue_4$<
<$PERSON:Sue_9$<PHYS-OBJECT<PIE_3<PIE_8<POSS<PTNT<STATE

Ще поддържаме две азбуки: Σ_{KB} с индекси, в която различаваме отделните екземпляри на типовете понятия в базата, и Σ без индекси, където ще се задават бъдещите въпроси за проекция. При изчисление на обобщенията, изображението λ ще осигури третирането на всеки етикет на индексиран *c*-връх като неиндексиран. Така например λ: $LOVE_6$, $LOVE_7 \rightarrow$ LOVE.

На стъпка 3 се изчисляват всички подграфи на ПКГ от базата. Те се записват в масива *list_subgraphs*, като *i*-тият елемент на масива съдържа списък на всички подграфи на *i*-тия граф G_i от базата. За краткост тук ще разгледаме само 7-те подграфа на ПКГ, показан на Фиг. 2.8. Всички те са подграфи на G_2 и могат да се изброят както следва с използване на индексацията на екземплярите в базата от знания, въведена на стъпка 2:

$G^{sub\text{-}1}{}_2$: <'LOVE$_6$ EXPR PERSON$_5$', 'none'>,

$G^{sub\text{-}2}{}_2$: <'LOVE$_7$ EXPR PERSON$_5$', 'none'>,

$G^{sub\text{-}3}{}_2$: <'LOVE$_6$ OBJ PERSON$_5$', 'none'>,

$G^{sub\text{-}4}{}_2$: <'LOVE$_6$ EXPR PERSON$_5$ LOVE$_6$ OBJ PERSON$_5$', '1=3,2=4'>,

$G^{sub\text{-}5}{}_2$: <'LOVE$_6$ EXPR PERSON$_5$ LOVE$_7$ EXPR PERSON$_5$', '2=4'>,

$G^{sub\text{-}6}{}_2$: <'LOVE$_6$ OBJ PERSON$_5$ LOVE$_7$ EXPR PERSON$_5$', '2=4'>,

$G^{sub\text{-}7}{}_2$: <'LOVE$_6$ EXPR PERSON$_5$ LOVE$_6$ OBJ PERSON$_5$ LOVE$_7$ EXPR PERSON$_5$', '1=3,2=4=6'>

Тези седем подграфа са записани в *list_subgraphs*(2) заедно с още 55 подграфа. Тук за краткост са показани директно линейните записи на $G^{sub\text{-}1}{}_2$, $G^{sub\text{-}2}{}_2$,…,$G^{sub\text{-}7}{}_2$ спрямо наредбата Ω_{KB}, макар че те се конструират в началото на стъпка 4 чрез функцията **COMPUTE_LINEAR_RECORD**. Подграфите са записани в Σ_{KB} с уникални анотации. Поради наличието на индекси в Σ_{KB}, линейните записи в *list_subgraphs* не съдържат дублирани тройки от етикети на елементарни конюнкти в ПКГ.

На стъпка 4 се изчисляват всички инективни обобщения на подграфи в базата. Обобщенията се пазят като сортиран списък от думи над азбуката Σ. Анотациите от линейните записи на тези обобщения се превръщат в маркери, които се асоциират към линейните низове от етикети. Така в явен вид се съхраняват всички възможни въпроси за инективна проекция, които имат непразен отговор в базата. Данните се съхраняват в двата стълба на масива *sorted_words_markers*.

Нека разгледаме примери за изчисленията на стъпка 4. Таблица 2.4 показва избрани редове на *sorted_words_markers*, демонстриращи характерни случаи при този процес. Стълб 1 съдържа сортирани по тройки низове от символи в Σ. Стълб 2 изброява в Σ_{KB} подграфите, за които са изчислени обобщенията в съответните редове. Всички подграфи за съхранени заедно в съответния ред на стълб 2 и така са идентифицирани семантично-подобните подграфи на G_1 и G_2. Анотациите от линейните записи на обобщенията са част от маркерите в стълб 2 заедно с идентификатора на оригиналния ПКГ G_1 или G_2.

sorted_words_markers(i,1): *sortedSeq*(G^{gen})	sorted_words_markers(i,2): < *annotation*(G^{gen}), *list_alternative_annot*, *new_lin_labels*($G^{sub\text{-}j}{}_2$), G_2 >
ACT AGNT PERSON	<'none', [], 'EAT$_2$ AGNT PERSON:John$_1$', G_1 > <'none', [], 'EAT$_{10}$ AGNT PERSON:Sue$_9$', G_2 >
….	….
EAT OBJ PHYS_OBJECT	<'none', [], 'EAT$_2$ PTNT PIE$_3$', G_1 > <'none', [], 'EAT$_{10}$ PTNT PIE$_8$', G_2 >
….	….
LOVE EXPR ANIMAL LOVE OBJ ANIMAL	<'1=3,2=4', [], 'LOVE$_6$ EXPR PERSON$_5$ LOVE$_6$ OBJ PERSON$_5$', G_2 > <'2=4', [], 'LOVE$_7$ EXPR PERSON$_5$ LOVE$_6$ OBJ PERSON$_5$', G_2 >
….	….
LOVE EXPR ANIMATE LOVE EXPR ANIMATE LOVE OBJ ANIMATE	<'1=5,2=4=6', ['3=5,2=4=6'], 'LOVE$_6$ EXPR PERSON$_5$ LOVE$_7$ EXPR PERSON$_5$ LOVE$_6$ OBJ PERSON$_5$', G_2 >
LOVE EXPR ANIMATE LOVE OBJ ANIMATE STATE OBJ ANIMATE	<'1=3,2=4=6', [], 'LOVE$_6$ EXPR PERSON$_5$ LOVE$_6$ OBJ PERSON$_5$ LOVE$_7$ EXPR PERSON$_5$', G_2 >
LOVE EXPR PERSON	<'none', [], 'LOVE$_6$ EXPR PERSON$_5$', G_2 > <'none', [], 'LOVE$_7$ EXPR PERSON$_5$', G_2 >
….	….
PERSON POSS ENTITY	<'none', [], 'PERSON$_5$ POSS PIE$_8$', G_2 > <'none', [], 'PERSON:Sue$_4$ POSS PIE$_3$', G_1 >
….	….

Таблица 2.4. Примерни редове на масива *sorted_words_markers* в края на стъпка 4

Нека разгледаме едно обобщение в стълб 1 на Таблица 2.4, пресметнато за $G^{sub\text{-}7}{}_2$:

LOVE EXPR ANIMATE LOVE EXPR ANIMATE LOVE OBJ ANIMATE (*g*)

Линейният запис на $G^{sub\text{-}7}{}_2$ е конструиран от функцията **COMPUTE_LINEAR_RECORD** в Σ_{KB} и не съдържа тройки за елементарни конюнкти с идентични етикети. Сортирането на *sortedSeq*($G^{sub\text{-}7}{}_2$) е извършено в Ω_{KB} и затова двете тройки, започващи с LOVE$_6$, се появяват като 1ва и 2ра тройка в *sortedSeq*($G^{sub\text{-}7}{}_2$). Тогава *annotation*($G^{sub\text{-}7}{}_2$)='1=3,2=4=6'. Обаче (*g*) се пресмята в Σ; функцията **COMPUTE_INJ_GEN** връща за (*g*) следния низ от символи, който не е сортиран в Σ:

LOVE EXPR PERSON LOVE OBJ PERSON LOVE EXPR PERSON с анотация '1=3,2=4=6'

Функцията **ENSURE_PROJ_MAPPING** пренарежда (*g*) като сортиран низ *sortedSeq*(*g*), в който тройките са изброени в различен ред с *annotation*(*g*)='1=5,2=4=6' (понеже последните две тройки се разместват). Тази функция съответно пренарежда и

тройките на подграфа $G^{sub\text{-}7}{}_2$, за да осигури съответствие между екземплярите на *sortedSeq*(g) и техните специализации:

new_lin_labels($G^{sub\text{-}7}{}_2$) = 'LOVE$_6$ EXPR PERSON$_5$ LOVE$_7$ EXPR PERSON$_5$ LOVE$_6$ OBJ PERSON$_5$'

Именно пренареденият низ с етикети на $G^{sub\text{-}7}{}_2$ – *new_lin_labels*($G^{sub\text{-}7}{}_2$) – се съхранява в стълб 2. Така за всяко обобщение G от стълб 1, наредбата на екземплярите му съответства на наредбата на специализираните екземпляри в подграфите, записани в стълб 2. Анотацията *annotation*(G) за еквивалентни c-върхове в обобщението G е валидна също така и за подграфа *new_lin_labels* в стълб 2. По този начин е изчислена и запомнена една (потенциална) инективна проекция от G (в стълб 1) към подграф от базата (в стълб 2). При обощението (g) има още една особеност, тъй като то съдържа дублирани тройки с идентични етикети. Функцията `COMPUTE_ISOMORPHISMS` се извиква на стъпка 4, за да изчисли възможните изоморфизми между логическите формули на (g). Тази функция връща низа *list_alternative_annot,* който се записва в стълб 2 на Таблица 2.4. Анотацията на еквивалентните линейни записи на (g), които биха се конструирали от изоморфни логически формули, се изчислява предварително в режим off-line, за да се избегне конструирането й в реално време при изпълнение на заявката. Тази алтернативна анотация е много важна, тъй като въпросът за проекция може да бъде поставен на системата в графичен вид или като логическа формула, която е еквивалентна на използваната при предварителната фаза (но се различава от нея).

Някои обобщения в стълб 1 на Таблица 2.4 имат непразни проекции върху няколко подграфа, изброени в съответния ред на стълб 2. Поради това на края на стъпка 4 се конструират маркери-множества в стълб 2, като обединение на единични маркери, и тогава се изтриват редовете с повтарящи се стойности в стълб 1 на масива *sorted_words_markers*. Подграфите на Фиг. 2.7б и 2.7в също са изброени в стълб 2 с техните анотации като маркер-множество. Така концептуалният ахрив се подготвя на предварителната off-line фаза за отговор на въпроси в run-time.

Таблица 2.5 съдържа 43 различни инективни обобщения за $G^{sub\text{-}1}{}_2$, …, $G^{sub\text{-}7}{}_2$, които са изчислени над примерната опора. Анотациите им са групирани в 13 маркера:

```
ACT OBJ ANIMAL          M3
ACT OBJ ANIMAL    LOVE EXPR ANIMAL    M6
ACT OBJ ANIMAL    STATE EXPR ANIMAL   M6
ACT OBJ ANIMATE         M3
ACT OBJ ANIMATE    LOVE EXPR ANIMATE  M6
ACT OBJ ANIMATE    STATE EXPR ANIMATE      M6
ACT OBJ ENTITY          M3
ACT OBJ PERSON          M3
ACT OBJ PERSON    LOVE EXPR PERSON    M6
ACT OBJ PERSON    STATE EXPR PERSON   M6
ACT OBJ PHYS-OBJ        M3
LOVE EXPR ANIMAL        M8
LOVE EXPR ANIMAL    LOVE EXPR ANIMAL  M5
LOVE EXPR ANIMAL    LOVE EXPR ANIMAL    LOVE OBJ ANIMAL     M7
LOVE EXPR ANIMAL    LOVE OBJ ANIMAL   M9
LOVE EXPR ANIMAL    LOVE OBJ ANIMAL    STATE EXPR ANIMAL  M7a
LOVE EXPR ANIMAL    STATE EXPR ANIMAL       M10
LOVE EXPR ANIMATE       M8
LOVE EXPR ANIMATE    LOVE EXPR ANIMATE      M5
LOVE EXPR ANIMATE    LOVE EXPR ANIMATE    LOVE OBJ ANIMATE  M7
LOVE EXPR ANIMATE    LOVE OBJ ANIMATE       M9
LOVE EXPR ANIMATE   LOVE OBJ ANIMATE  STATE EXPR ANIMATE  M7a
LOVE EXPR ANIMATE    STATE EXPR ANIMATE     M10
LOVE EXPR PERSON        M8
LOVE EXPR PERSON    LOVE EXPR PERSON  M5
LOVE EXPR PERSON   LOVE EXPR PERSON  LOVE OBJ PERSON        M7
LOVE EXPR PERSON    LOVE OBJ PERSON   M9
LOVE EXPR PERSON   LOVE OBJ PERSON  STATE EXPR PERSON   M7a
LOVE EXPR PERSON   STATE EXPR PERSON        M10
LOVE OBJ ANIMAL  M3
LOVE OBJ ANIMAL    STATE EXPR ANIMAL  M6
LOVE OBJ ANIMATE        M3
LOVE OBJ ANIMATE    STATE EXPR ANIMATE      M6
LOVE OBJ ENTITY         M3
LOVE OBJ PERSON         M3
LOVE OBJ PERSON    STATE EXPR PERSON  M6
LOVE OBJ PHYS-OBJ       M3
STATE EXPR ANIMAL       M8
STATE EXPR ANIMAL   STATE EXPR ANIMAL       M10
STATE EXPR ANIMATE      M8
STATE EXPR ANIMATE    STATE EXPR ANIMATE    M10
STATE EXPR PERSON       M8
STATE EXPR PERSON    STATE EXPR PERSON      M10
```

Таблица 2.5. Сортиран списък на инективните обобщения на всички подграфи на ПКГ *'Съществува човек, който обича себе си, и изпитва и друга любов'* от Фигура 2.8.

M1: <'none', [], 'LOVE$_6$ EXPR PERSON$_5$', G_2>

M2: <'none', [], 'LOVE$_7$ EXPR PERSON$_5$', G_2>

M3: <'none', [], 'LOVE$_6$ OBJ PERSON$_5$', G_2>

M4: <'1=3,2=4', [], 'LOVE$_6$ EXPR PERSON$_5$ LOVE$_6$ OBJ PERSON$_5$', G_2>

M5: <'2=4', [], 'LOVE$_6$ EXPR PERSON$_5$ LOVE$_7$ EXPR PERSON$_5$', G_2>

M5a: <'2=4', [], 'LOVE$_7$ EXPR PERSON$_5$ LOVE$_6$ EXPR PERSON$_5$', G_2>

M6: <'2=4'>, [], <'LOVE$_6$ OBJ PERSON$_5$ LOVE$_7$ EXPR PERSON$_5$'>, G_2>

M6a: <'2=4', [], 'LOVE$_7$ EXPR PERSON$_5$ LOVE$_6$ OBJ PERSON$_5$', G_2>

M7: <'1=5,2=4=6', ['3=5,2=4=6'], 'LOVE$_6$ EXPR PERSON$_5$ LOVE$_7$ EXPR PERSON$_5$ LOVE$_6$ OBJ PERSON$_5$', G_2>

M7a: <'1=3,2=4=6', [], 'LOVE$_6$ EXPR PERSON$_5$ LOVE$_6$ OBJ PERSON$_5$ LOVE$_7$ EXPR PERSON$_5$', G_2>

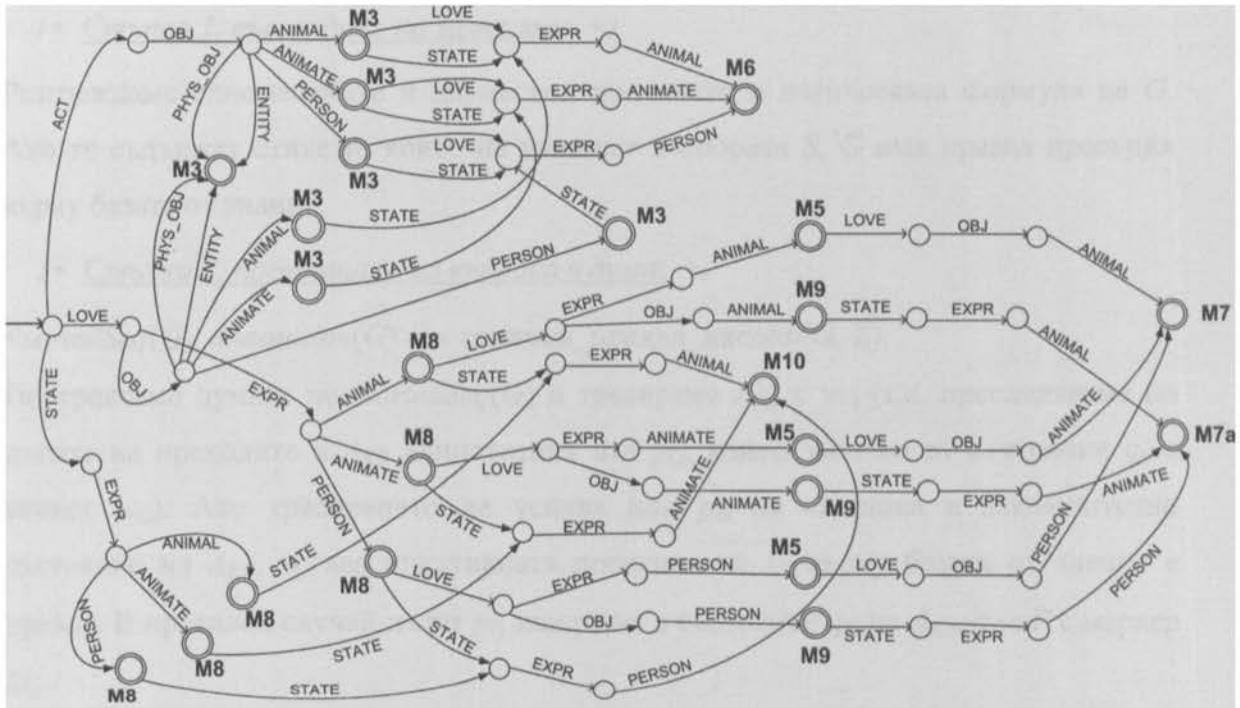
M8: M1 ∪ M2

M9: M4 ∪ M6a

M10: M5 ∪ M5a

Маркерите-'варианти' M5a, M6a и M7a се създават, когато функцията **ENSURE_PROJ_MAPPING** пренарежда низовете от етикети на подграфите по нов начин. Маркерите-множества M8-M10 се формират в стълб 2 на *sorted_words_markers* когато се групират дублиращите се стойности в стълб 1; тогава съдържанието на стълб 2 за тези редове се запомня като маркер-обединение, а редовете с дублирани стойности в стълб 1 се изтриват. Лексикографски-сортираният краен списък от думи със съответни маркери в Таблица 2.5 е вход за конструиране на минимален ацикличен краен автомат с маркери на заключителните състояния. Самата конструкция се извършва чрез алгоритъм, предложен в [Мих00, DMWW00].

На стъпка 5 A_{KB} се конструира директно. Фиг. 2.11 показва минималния ацикличен краен автомат с маркери на заключителните състояния, който кодира 43-те инективни обобщения, изброени в Таблица 2.5. Автоматът има 64 състояния и 83 дъги на преходите. На заключителните състояния са съпоставени 8 маркера. Този автомат разпознава регулярния език от 43 думи-обобщения, дадени в Таблица 2.5. □



Фигура 2.11. Минимален ацикличен краен автомат, кодиращ инективните обобщения на 7-те подграфа на ПКГ от Фигура 2.8.

2.4. Инективна проекция като трасиране на минимален ацикличен краен автомат по време на изпълнение на заявката

При получаване на заявка G, инективната ѝ проекция върху базата от знания се изчислява чрез трасиране на път в минималния ацикличен краен автомат, който е построен чрез алгоритъм 2.1 и кодира всички инективни обобщения на подграфи от базата. Предполагаме, че G се задава във вид на логическа формула над същата опора S, за която е построена базата от знания. Нека отбележим, че по принцип в S могат да участват типове, които не се срещат в базата в конкретния момент. Ще използваме наредената азбука Σ, която съдържа етикетите на опората:

Алгоритъм 2.2. *Намиране на всички инективни проекции на даден ПКГ G с двумерни концептуални релации върху база от знания от ПКГ с двумерни концептуални релации, която е кодирана по алгоритъм 2.1 като минимален ацикличен краен автомат с маркери на заключителните състояния* $A_{KB}=\langle \Sigma, Q, q_0, F, \Delta, E, \mu \rangle$:

/* *Стъпка 1, въвеждане на заявката:* */

Разглеждаме едноместните и двуместни предикати в логическата формула на G. Ако те съдържат етикети, които не участват в опората S, G има празна проекция върху базата от знания.

/* *Стъпка 2, превръщане на въпроса в дума:* */

$\langle sortedSeq(G), annotation(G)\rangle$:= **COMPUTE_LINEAR_RECORD**(G, Σ).

Построяваме думата w_G=$sortedSeq(G)$ и трасираме A_{KB} с w_G (т.е. проследяваме по дъгите на преходите в A_{KB} единствения път p_G, който започва от състояние q_0 с етикет w_G). Ако трасирането не успява или p_G не свършва в заключително състояние на A_{KB}, тогава инективната проекция на G върху базата от знания е празна. В противен случай пътят p_G завършва в състояние q_G на A_{KB}, $q_G \in F$ с маркер M_q.

/* *Стъпка 3, търсене на анотацията на въпроса G в маркетите*: */

Разглеждаме маркера M_q в q_G, който е множество от k единични маркера-четворки за $k \geq 1$:

$\{ \langle annotation_1, list\_alternative\_annot_1, new\_lin\_labels_1, G_{i1}\rangle, \ldots\ldots\ldots,$
$\langle annotation_k, list\_alternative\_annot_k, new\_lin\_labels_k, G_{ik}\rangle \}$

for $1 \leq j \leq k$ **do begin**

if $annotation(G) = annotation_j$ **then return** $\langle new\_lin\_labels_j, annotation(G)\rangle$ като инективна проекция на G върху G_{ij};

if $annotation(G) \in list\_alternative\_annot_j$

then $annotation(G) := annotation_j$;

return $\langle new\_lin\_labels_j, annotation(G)\rangle$ като инективна проекция на G върху G_{ij};

end

G има празна инективна проекция върху базата от знания. □

Теорема 1. Нека е дадена база от знания от прости концептуални графи с двумерни концептуални релации $\{G_1, G_2, \ldots, G_n\}$ над опората $S = (T_C, T_R, I, \tau)$ и нека $A_{KB} = \langle \Sigma, Q, q_0, F, \Delta, E, \mu \rangle$ е минимален краен автомат с маркери на заключителните

състояния, построен за тази база чрез алгоритъм 2.1. Нека G е прост концептуален граф с двумерни концептуални релации. Тогава всички инективни проекции на G върху $\{G_1, G_2, ..., G_n\}$ могат да се изчислят по алгоритъм 2.2.

Доказателство.

Ако някои от върховете на G (типовете понятия, типовете релации или върхове с индивидуални маркери) не са включени в опората S, то G няма инективна проекция върху $\{G_1, G_2, ..., G_n\}$ и алгоритъм 2.2 в стъпка 1 ще върне празното множество.

Нека G има инективна проекция G' върху базата от знания, където G' е концептуален подграф на някой граф G_i. Тогава всички върхове на G присъстват като етикети в опората S. Следователно, може да бъде построен линеен запис на G $<sortedSeq(G), annotation(G)>$ в стъпка 2 на алгоритъм 2.2, както и дума $w_G=sortedSeq(G)$ от етикети на опората, които съставят азбуката на A_{KB}.

Нека допуснем, G' не се съдържа в резултата, върнат при изпълнението на алгоритъм 2.2. Съществуват две възможни причини за това: *(1)* при трасирането на автомата A_{KB} с думата w_G не се достига до заключително състояние или *(2)* в маркера M_q на достигнато заключително състояние q не присъства графът G'.

Да разгледаме случай *(1)*, при който се допуска, че трасирането на A_{KB} с думата w_G води до състояние $q \notin F$. Тъй като G' е подграф на G_i, G' ще бъде включен в списъка *list_subgraphs*(i) на стъпка 3 при построяването на A_{KB} чрез алгоритъм 2.1. От друга страна, графът G е обобщение на G' и следователно G ще бъде включен в списъка *list_gen_graphs*(i,*) на стъпка 4 от алгоритъм 2.1. Тогава $w=sortedSeq(G)$ от линейния запис на G ще бъде включена в масива *sorted_words_markers*(*,1). Тъй като A_{KB} разпознава думите в *sorted_words_markers*(*,1), то трасирането на автомата с дума w ще звърши в състояние $q \in F$. Но линейният запис на G в алгоритъм 2.1 и в алгоритъм 2.2 се построява над една и съща опора, при една и съща наредба на символите на азбуката и тогава $w_G = w$. И така стигнахме до противоречие с допускането, че w_G не се разпознава от автомата A_{KB}.

Да разгледаме случай *(2)*, при който се допуска, че трасирането на A_{KB} с думата w_G води до състояние $q \in F$, но G' не участва в маркера M_q на достигнато заключително състояние q. Следвайки аналогични разсъждения като при случай *(1)*, G е p-тото обобщение на графа G', който е j-тия подграф на G_i. Следователно на стъпка 4 от алгоритъм 2.1 се изчислява тройката

$$<sortedSeq(G), annotation(G), new\_lin\_labels(G')>$$

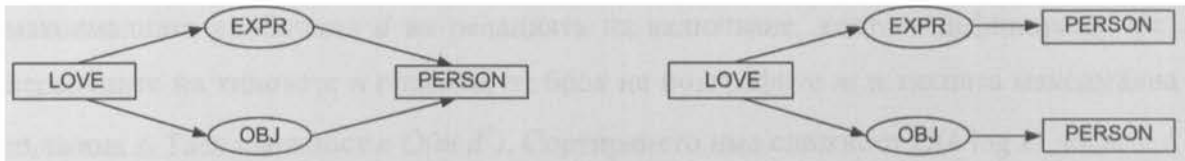
и списъкът от алтернативните анотации *list_alternative_annot* на G ако съществуват такива, където *new_lin_label*(G') съдържа графа G' евентуално преподреден спрямо наредбата в G. От тук и съгласно конструкцията на A_{KB} на стъпка 5 от алгоритъм 2.1 следва, че G' ще присъства в маркера на думата $w = sortedSeq(G)$. И поради детерминираността на A_{KB} маркерът на състоянието, до което ще се достигне при трасирането на A_{KB} с думата w ще съдържа графа G', с което достигнахме до противоречие с допускането.

Следователно G' се съдържа в резултата, върнат при изпълнението на алгоритъм 2.2. Тогава всички инективни проекции на G върху $\{G_1, G_2, ..., G_n\}$ могат да се изчислят по алгоритъм 2.2. □

Пример 2.10. Фиг. 2.12 показва два ПКГ и техните линейни записи. Графът на Фиг. 2.12А има инективна проекция върху графа от Фиг. 2.8, за който е постоен автоматът на Фиг. 2.11. След трасиране на A_{KB} от Фиг. 2.11 с думата 'LOVE EXPR PERSON LOVE OBJ PERSON' стигаме до маркера M9 (т.е. M4∪M6a), в който участва анотацията '1=3,2=4' на графа от Фиг. 2.12А. Но графът на Фиг. 2.12Б няма инективна проекция върху графа от Фиг. 2.8, тъй като неговата анотация '1=3' не участва в маркера M9. Така думата

'LOVE EXPR PERSON LOVE OBJ PERSON'

принадлежи на регулярния език, разпознаван от автомата на Фиг. 2.11, но това не е достатъчно условие за наличие на инективна проекция. Сложността на изчисленията на тази фаза касае само въпроса и не зависи от големината на базата от знания. □



LOVE EXPR PERSON LOVE OBJ PERSON
1=3, 2=4
(А)

LOVE EXPR PERSON LOVE OBJ PERSON
1=3
(Б)

Фигура 2.12. Въпроси за инективна проекция като думи в регулярен език, построен над азбука от етикетите на опората.

2.5. Алгоритмична сложност

Двете фази на изчисление на проекцията – предварителна и при обработка с цел отговор на потребителската заявка – се изпълняват поотделно. Поради това оценяваме сложността на компонентите им поотделно.

Предварителната (off-line) фаза включва всички изчисления, които не зависят от конкретен въпрос за инективна проекция. Тя трансформира базата от знания в специфичен концептуален ресурс, който е унифицирано, оптимално и компресирано представяне, подпомагащо операцията проекция. Пресмятанията имат експоненциална сложност. Нека отделим 5 главни компоненти в тях:

- ***Намиране на всички концептуални подграфи на ПКГ в базата от знания и представянето им в линеен запис <sortedSeq,annotation>***. При даден ПКГ с n c-върха, намирането на всички негови подграфи има сложност $O(2^n)$. Сортирането на списък от символи има сложност $O(k \log k)$, където k е броят на елементите в списъка. Описанието на идентичните аргументи *annotation* може да се изчисли за линейно време относно броя на c-върховете в графа. Общо този компонент има сложност $O(n\ 2^n)$.

- ***Пресмятане на всички инективни обобщения на подграфите в базата от знания, представяне на обобщенията като линейни записи, и пренадеждане на линейния запис на някои подграфи с цел осигуряване на взаимно-еднозначно съответствие между техните върхове и върховете в съответните сортирани обобщения.*** Сложността на изчисление на всички инективни обощения зависи от

максималната дълбочина d на релацията на включване, която е дефинирана чрез йерархиите на типовете в опората, от броя на подграфите m и тяхната максимална дължина s. Тази сложност е $O(m\, d^s)$. Сортирането има сложност $O(k \log k)$, където k е дължината на низа символи за сортиране. Изчисляването на идентичните аргументи в *annotation* за даден ПКГ може да стане в линейно време. Общата сложност на този компонент е $O(m\, d^s (\log m + s \log d))$.

- ***Построяване на всички алтернативни анотации за всички обобщения (чрез пресмятане на изоморфизмите между променливите в логическите формули за тези ПКГ, които имат дублиращи се тройки от етикети в линейния запис).*** Нека низа *sortedSeq*(G) от линейния запис на ПКГ G съдържа m групи от дублирани тройки с дължини съответно по k_1, k_2, ..., k_m елементарни конюнкти. Изоморфизмите съответстват на пермутациите на дублираните тройки, които заемат съседни позиции след сортирането на *sortedSeq*(G). Изчислението им за всеки подграф има сложност $O(k_1!\, k_2! \ldots k_m!)$.

- ***Построяване на минимален ацикличен краен автомат с маркери на заключителните състояния $A_{КВ}$.*** Както е показано в [Мих00, DMWW00], сложността на конструкцията е $O(n \log m)$, където n е общият брой на символите във входния списък лексикографски-сортирани думи и m е броят на състоянията в $A_{КВ}$.

Крайният автомат $A_{КВ}$ може да бъде поддържан като алтернативно представяне на базата от знания. Сравнително лесно е да се добави нов подграф към базата (т.е. към нейния краен автомат), но промените и обновяването изискват ново конструиране на автомата. Тук ще разглеждаме обновяването като още един компонент, който се изпълнява off-line, отделно от пресмятанията на потребителската заявка в реално време. Сложността му е както следва:

- ***Поддръжка на базата от знания чрез добавяне и изтриване на думи от крайния автомат $A_{КВ}$.*** Дадена дума може да бъде добавена или изтрита от автомата за линейно време, зависещо от дължината на думата n. Обаче добавяне или изтриване на цял ПКГ изисква добавяне или изтриване на думи, съответстващи

на инективните обобщения на всичките му подграфи, което изисква изчисления със сложност $O(m_1 d^{s_1})$, където m_1 е броят обобщенията, s_1 е максималната дължина на подграфа и d е дълбочината на релацията за включване.

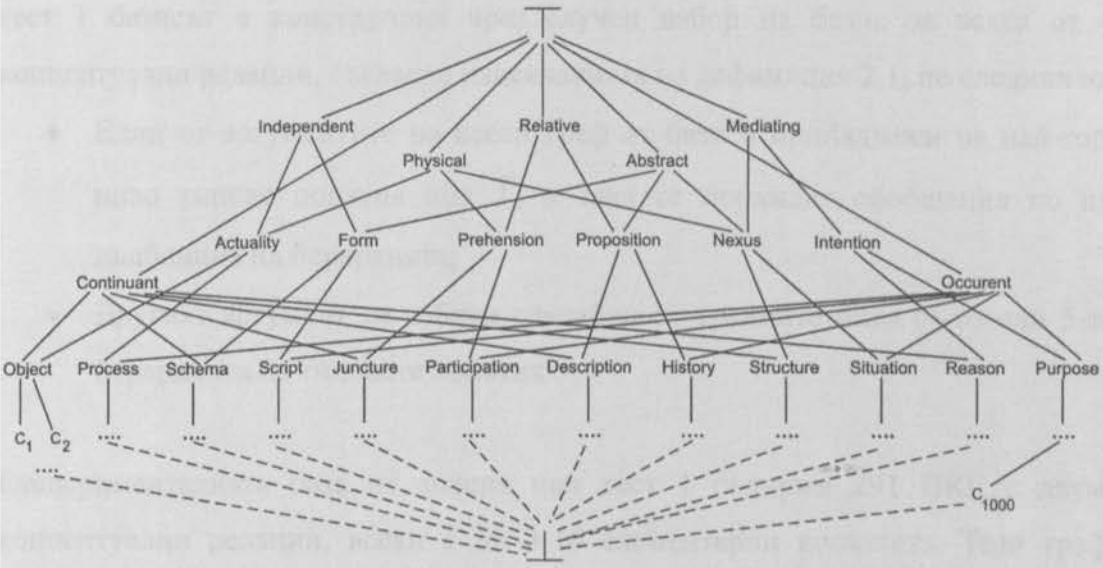
Изчисленията в реално време (в run-time) се извършват след постъпване на конкретна заявка от потребителя. Те използват наготово ресурса, подготвен на предварителната фаза, и имат два главни компонента след задаване на въпрос G в логическа форма:

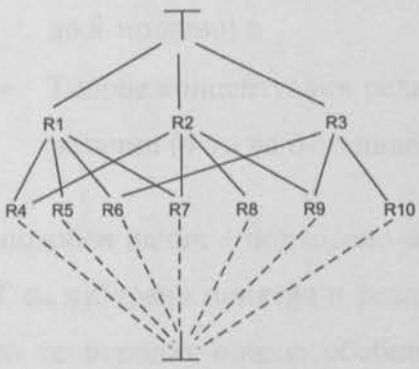
- ***Представяне на G като сортиран низ w_G от символи на опората*** със сложност $O(n \log n)$, където n е броят на символите в G, ***и пресмятане на идентичните агрументи в низа annotation*** за линейно време $O(n)$;
- ***Трасиране на крайния автомат A_{KB} с дума w_G.*** Сложността на тази задача е $O(n)$, където n е броят на символите в G. Големината на базата от знания няма значение, тъй като сложността зависи само от входната дума. Всички инективни проекции на G върху базата се намират наведнъж.

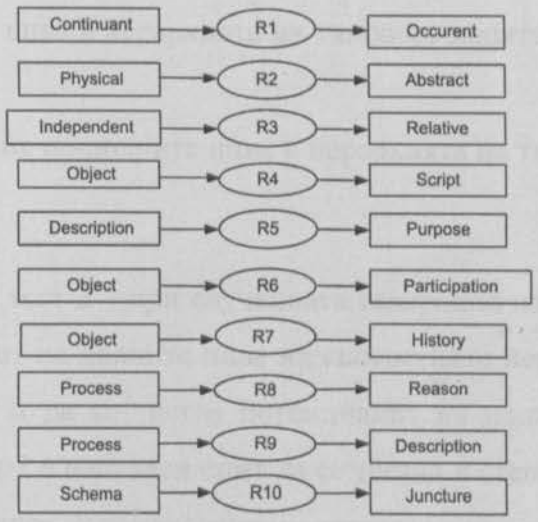
Предимствата на предлагания двуфазов подход се виждат ясно при сравнение между горните оценки за сложност на run-time изчисленията и показаните в Таблица 2.1. Освен това чрез получения на предварителната фаза 'концептуален автомат' лесно се решават и други важни задачи, например става елементарно да се провери дали даден ПКГ е еквивалентен на съществуващ граф в базата от знания.

2.6. Експериментални тестове

За да се добием представа за степента на компресия на инективните обобщения при кодирането им в минимален краен автомат, досега сме генерирали случайно две независими бази знания от ПКГ с двумерни концептуални релации в нормална форма. При генерацията на тестовите данни се задават някои основни параметри: дълбочина и широчина на йерархиите на типовете и брой типове, брой на графите в базата и дължина на графите като брой на елементарните им конюнкти. В първия тестов набор от данни опората е генерирана случайно, а при втория тест за връх на йерархията на типовете понятия - вж. Фиг. 2.13 - се използва най-горният слой на







Фигура 2.13. Опора използвана в тест 2, с горно ниво на типове понятия от [Sow00]

йерархия, предложена от Сова в [Sow00]. Опората при тест 2 е получена чрез добавяне на 1000 типа понятия към най-горния слой по случаен начин и с използване на 10 абстрактни концептуални релации (нека отбележим, че е рядкост някой автор да предложи набор от релации; в литературата се срещат предимно йерархии от типове понятия). Таблица 2.6 резюмира количествата данни при двата теста.

Фиг. 2.13 показва също как при тест 2 ръчно е конструиран базисът на опората за всяка от 10-те концептуални релации R_1, R_2, ..., R_{10}; като аргументи са избрани типове понятия от най-горното ниво, с цел да се осигурят повече обобщения. При тест 1 базисът е конструиран чрез случен избор на базис за всяка от 40-те концептуални релации, съгласно изискванията на дефиниция 2.1, по следния начин:

- Един от аргументите на всеки граф от базиса принадлежи на най-горното ниво типове понятия под $\top$, и така се пораждат обобщения по цялата дълбочина на йерархията;
- Другият аргумент се избира случайно от средните нива (4-то или 5-то) на йерархията на типовете понятия.

Експерименталната база от знания при тест 1 съдържа 291 ПКГ с двумерни концептуални релации, всеки с по 3-10 елементарни конюнкта. Тези графи са генерирани случайно с избор на:

- Типове понятия от по-долните нива в йерархията на типовете понятия (6-то до 8-мо ниво) и
- Типове концептуални релации от по-долните нива в йерархията на типовете релации (4-то до 6-то ниво).

По подобен начин е подходено и при тест 2 – при случайната генерация на 329-те ПКГ са избирани понятия и релации от по-долните нива на съответните йерархии, за да се породят повече обобщения и да се тества потенциалът на подхода за компресиране на обобщения. При тест 2 е направен опит да се тества и степента на разнообразие на анотациите, в зависимост от топологичната структура на графите.

Брой на:	Тест 1	Тест 2
1. Типове понятия в йерархията на опората	600	1025
2. Типове концептуални релации в йерархията на опората	40	10
3. Максимална дълбочина на йерархиите: на типовете понятия на типовете концептуални релации	 8 6	 20 2
4. Максимална широчина на йерархиите: на типовете понятия на типовете концептуални релации	 10 17	 98 7
5. Брой надтипове в йерархиите (за тип)	2,32 средно	2,00029 средно (максимално 8)
6. ПКГ в базата от знания	291	329
7. Елементарни конюнкти (дължина на ПКГ)	3-10 средно 4,5	3-12 средно 5,65
8. (Концептуални) подграфи в базата от знания	6 753	11 146
9. Инективни обобщения на всички подграфи	10 436 190	140 031 027
10. Видове анотация (видове топологични структури на ПКГ)	13 885	3 618
11. Състояния в минималния КА, построен от алгоритъм 2.1	2 751 977	23 956 007
12. Преходи в минималния КА, построен от алгоритъм 2.1	3 972 096	43 347 641
13. Байтове във входния текстов файл за алгоритъм 2.1 в UNICODE (т.е., обем на сортирания списък от линейните записи на всички инективни обобщения)	891,4 MBytes	~ 13 GBytes
14. Байтове във файла на минималния КА, построен от алгоритъм 2.1, но без маркерите (запазени са само указатели към тях)	52,44 MBytes	612,73 MBytes
15. Степен на компресия входен файл / минимален КА	~ 17 пъти	~ 21,2 пъти
16. Байтове във входния файл, архивиран с bzip2	21,8 MBytes	
17. Отношение на големината минимален КА / zip	2,4	

Таблица 2.6. Два набора тестови данни за Алгоритъм 2.1, подготвени чрез случайна генерация на опори и бази ПКГ с двумерни концептулани релации в нормална форма

Базата от знания при тест 2 се състои от прости концептуални графи, които са формирани чрез 4 структурни шаблона за различните размерности графи. Един шаблон съответства на линейна свързаност на елементарните конюнкти. Друг шаблон съответства на 'звездовидна свързаност': графите се състоят от елементарни конюнкти, които имат точно по един общ аргумент (а съответните планарни графи изглеждат като звезда). Тогава всички подграфи имат еднаква топологична структура, на което се дължи и относителната липса на 'вариации' в низовете, описващи идентичност на аргументите при изброяване на анотациите.

Нека разгледаме редовете на Таблица 2.6. Тест 2 е проведен над по-големи йерархии и с повече и по-дълги ПКГ (редове 1-7 в таблицата). Поради това при тест 2 има повече подграфи (ред 8) и броят на всички инективни обобщения е по-голям (ред 9). При тест 2 много обобщения имат сходна топологична структура и затова броят анотации е по-малък (ред 10). По-високата степен на компресия при тест 2 (ред 15) може да се обясни с факта, че в езика има повече думи с еднакви начала и еднакви краища. Но и при двата експеримента компресията на входния списък от думи е значителна, когато този списък се кодира като минимален краен автомат, и напомня резултатите, получени при представяне на морфологични речници като минимални ациклични крайни автомати (вж. [Мих00]). Самият минимален автомат е постоен по алгоритъма, предложен в [Мих00], и за конструкцията се използва съществуващ софтуер. При тест 1 големината на получения минимален краен автомат беше сравнена с големината на архивирания текстов файл, съдържащ сортиран списък на всички обобщения. Автоматът е само 2,4 пъти по-голям от zip-архива (ред 17), но осигурява търсене (трасиране) по зададена входна дума за линейно време. Както е казано и в Таблица 2.6 ред 14, подграфите на базата не са запазени в маркерите на автомата и не са пресметнати като част от неговия обем. И при двата експеримента маркерите на заключителните състояния на автомата съдържат само указатели-индекси на подграфите. По принцип самите подграфи би следвало да се запомнят на външна памет, докато автоматът може да стои в оперативната памет поради сравнително неголемия му обем и при двата теста (ред 14).

Текстовият файл, подаден на входа на процедурите за конструиране на минималния автомат, съдържа лексикографски-сортирани редове съответстващи на всички инективни обобщения за експеримента. Всеки ред има следния формат:

<*sortedSeq*(G_i) от линейния запис на графа G_i, *Marker*>

където $1 \le i \le 10436190$ за тест 1 и $1 \le i \le 140031027$ за тест 2. И за двата експеримента, минималният автомат е построен в Уникод над азбука от символи на опората, която се разглежда като подазбука на Уникод. Тъй като опорите съдържат сравнително малко символи – под 1100, на всеки етикет от опората се съпоставя един символ в Уникод. За описание на анотациите, азбуката на автоматите съдържа цифрите 0,1,…,9. Необходими са освен това три символа-разделители, които показват

- къде свършва думата от езика на автомата във входния низ – т.е. къде започва анотацията в поредицата входни символи и
- как са разделени вътрешно класовете на еквивалентност в анотацията.

Нека подробно разгледаме азбуката на езика при тест 1:

- 600 етикета на типове понятия C_1, C_2, …, C_{600}, които се превръщат в 600 символа в Уникод,
- 40 етикета на типове концептуални релации R_1, R_2, …, R_{40}, които се превръщат в 40 символа в Уникод,
- цифри 0,1,2,3,4,5,6,7,8,9, от които се конструират позициите на 20-те аргумента на релациите при ПКГ с дължина до 10 елементарни конюнкта, и
- разделители ',', '=' и '&'.

При изброяването на символите във файла на всички обобщения, подготвен за конструиране на автомата, се оказа, че някои типове понятия не се срещат в конкретните графи поради случайния избор. Така че азбуката на тест 1 се състои от по-малко от 653 символа. Азбуката на тест 2 се състои от по-малко от 1047 символа. При това положение е лесно за всеки етикет на тип от опората да се отдели по един символ в Уникод, който е достатъчно голям (65536 символа), за да 'обслужи' много по-обширно множество от етикети.

Тъй като при построяване на линейния запис на обобщенията подграфите се пренареждат, за да съответстват техните екземпляри на съответните в сортираните обобщения, имаше опасения от произвеждане на твърде много подграфи и в резултат - неуправляемо количество данни. Затова беше важно да се сравни броят на различните видове анотации за подграфи с различен размер и да се види, че броят на концептуалните подграфи е сравнително малък – което оправдава изброяването на техните обобщения по метода на грубата сила, понеже вариативността на анотациите е 'под контрол'. Таблица 2.8 показва броя на подграфите за случайно-генерираните бази от ПКГ в двата експеримента. Дадено е и съответното число на Бел за множество с брой елементи, равен на броя на аргументите на елементарните конюнкти. От стойностите в таблицата можем да направим две заключения, макар че са проведени само два независими теста:

Брой елементарни конюнкти в инективните обобщения	Брой аргументи за разделяне в класове на еквивалентност	Брой различни анотации в експерименталните бази знания		Съответно число на Бел (максимален брой възможни класове на еквивалентност)
		Тест1	**Тест2**	
2	4	5	5	B_4=15
3	6	42	35	B_6=203
4	8	238	139	B_8=4 140
5	10	695	342	B_{10}=115 975
6	12	1654	461	B_{12}=4 213 597
7	14	3369	761	B_{14}=190 899 322
8	16	2863	769	B_{16}=10 480 142 147
9	18	3831	271	B_{18}=682 076 806 159
10	20	1188	513	B_{20}=51 724 158 235 372
11	22		281	B_{22}=4 506 715 738 447 320
12	24		41	B_{24}=445 958 869 294 805 000
	TOTAL:	**13885**	**3618**	**450 518 001 943 669 000**

Таблица 2.8. Брой на различни анотации в експерименталните бази от знания при тест 1 и тест 2, сравнен с числото на Бел за съответния брой аргументи на елементарните конюнкти

- Броят на концептуалните подграфи е сравнително ограничен, т.е. смислените формули в даден език за представяне на знанията са по-малко от всички конюнктивни формули от разглеждания тук вид;

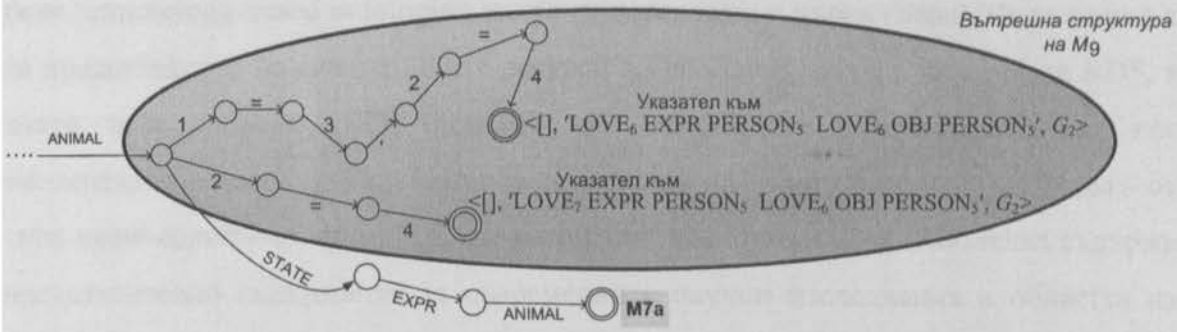
- Базисът на опората налага много силно ограничение върху топологичната структура на инективните обобщения. Това е ясно и интуитивно, но тук виждаме нагледни доказателства в каква степен формулите си 'приличат'.

И при двата теста количеството обобщения е управляемо при съвременните изчислителни характеристики на стандартни персонални компютри.

Независимо от факта, че броят на топологичните структури на подграфите и инективните им обобщения остава сравнително ограничен, вътрешното представяне на анотациите също е подчинено на идеята за осигуряване на максимална ефективност при търсене по време на изпълнение на заявката. В проведените тестове анотацията е реализирана като заключителна част от думата, вж. Фиг. 2.14. На фигурата е "разгъната" картината на маркера в заключителното състояние **M9** от Фиг. 2.11, което се намира след 6-тия символ на думата

```
LOVE  EXPR  ANIMAL  LOVE  OBJ  ANIMAL  STATE  EXPR  ANIMAL
```

Така се осигурява претърсване на анотациите чрез трасиране в линейно време.



Фигура 2.14. Разширяване на думите на автомата с анотация, кодираща еднакви *c*-върхове.

Експериментите са направени над случайно-генерирани данни поради липсата на голям обем реални данни. Ние сме се интересували главно от количествените показатели. И двата теста показват важността на понятието 'концептуален подграф'; виждаме, че алгоритмите за извършване на проекция по време на изпълнение на подадена заявка неизбежно обработват много графи (или формули), които нямат концептуален смисъл в разглеждания свят. Това е неизбежно при

работа със 'сурови данни' и напълно излишно при предложения от нас двуфазов подход, който на предварителната off-line фаза отделя и оптимизира само значещите подграфи и техните обобщения.

Когато се планираха експериментите за ефективна обработка на концептуална информация, трябваше да оценим дали обемът на тестовите данни съответства на съвременните стандарти. За целта разгледахме средната големина на наличните концептуални ресурси. Днес в света има няколко много големи онтологии като CyC[12] и LinKBase®[13], които се развиват десетки години, съдържат милиони понятия и релации и в тях са вложени стотици човеко-години труд. В нашия случай се направи сравнение с обема на наличните ресурси в академичните среди. При проучване на онтологичната библиотека OntoSelect [OnSelect] установихме, че тя съдържа информация за 1420 онтологии като представителен списък на ресурси от формализирани описания на знания за света. Тези онтологии са разработени в различни проекти и с различни цели. Само около 50 от тях съдържат повече от 600 понятия и това са предимно онтологии, използващи терминологични номенклатури (или terminology-based ontologies, както ги нарекохме в първа глава). Формализмът за представянето на семантичните ресурси в OntoSelect често е диалект на RDF, а както вече казахме, RDF-тройките са подобни на разглежданите от нас елементарни конюнкти. Релациите са двумерни и най-често изразяват свойства – от типа *agent-capable-of*, *citizen-of*, *has-part* и т.н. Ако приемем, че OntoSelect съдържа представително съдържание за съвременните научни изследвания в областта на Семантичния интернет, то тогава обемът на разглежданите от нас тестови данни може да се определи като 'представителен' и 'със средна големина'. Всъщност OntoSelect показва философията на днешните научни прототипи на интелигентни системи, при които се извличат множество по-малки онтологии с цел да се свържат под унифициран горен слой (upper model). Следователно, обемът на данните в нашите експерименти съответства на среден по големина изследователски тест.

---

[12] Собственост на Cycorp (www.cyc.com), проектът е започнал през 1984 год.
[13] Собственост на L&C (Language and computing, http://www.landcglobal.com/)

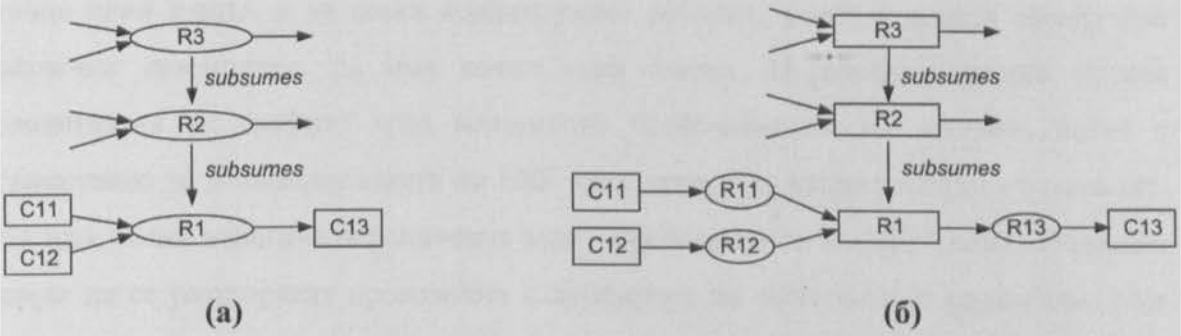
2.7 Ограничения на предложения подход и възможни обобщения

По-долу ще коментираме ограниченията на класа ПКГ, разглеждан в предложения тук изчислителен модел, и възможни обобщения на подхода.

Разглеждат се само ПКГ с двумерни концептуални релации

Както беше показано в част 1.1, философският и концептуален спор дали всяка *n*-мерна релация може да се представи чрез набор от *n* двумерни релации е до известна степен методологически. Представените становища отразяват различията между практиката и теорията на представянето на знанията.

Да предположим обаче, че някой ден открием многомерни концептуални релации, които инженерите на знанията не успеят да декларират като набор от двумерни релации поради фатална загуба на съществена семантична информация. Тогава, ако в базата от знания има например тримерна релация *R1* (вж. Фиг. 2.15а), възможно ли е чисто технически да обработим такава база чрез предлагания двуфазов модел на пресмятания?



Фигура 2.15 Тримерна концептуална релация, моделирана чрез двумерни релации.

Отговорът е, че е възможно, ако поддържаме вътрешна информация за 'кодирането' на *n*-мерните релации чрез 'служебни' етикети на изкуствено-добавени понятия и релации (вж. Фиг. 2.15б и 'изкуствените' елементи – понятието *R1* и релациите *R11*-*R12* и *R13*). Тъй като *n*-мерните релации се специализират до релации със същата размерност, може цялата база от знания и всички инективни обобщения на

подграфите да се прехвърлят в запис с двумерни релации. Едни и същи процедури за 'превод до двумерни релации' ще третират както графите от базата, така и потребителските заявки. Освен това минималният краен автомат, конструиран чрез алгоритъм 2.1, съдържа всички 'правилни' инективни обобщения, така че всеки въпрос с непразна проекция върху базата е кодиран в автомата във вид *<дума, анотация>*. Така че е възможно да се поддържа системен интерфейс между оригиналните *n*-мерни релации и тяхно вътрешно представяне чрез двумерни релации, без дори потребителят да е информиран за това. Сложността на превода до двумерни релации не е особено обременителна както за фазата на предварителната обработка, така и при изпълнение на конкретна потребителска заявка в реално време. (По принцип такъв е и подходът на практиците от W3C [W3C], които без теоретични дискусии минават направо към двумерните релации).

Разглеждат се само нормализирани ПКГ

Всички ПКГ, обработвани чрез алгоритъм 2.1 и алгоритъм 2.2, трябва да бъдат представени в нормалната си форма: тоест, за всеки екземпляр на понятие да има точно един *c*-връх и за всяка концептуална релация, която е в сила между два различни екземпляра, да има точно един *r*-връх. Нормалната форма описва семантиката на графите чрез минимален брой символи на опората, което е съществено за интерпретацията на ПКГ като думи над крайна азбука от символи. По този начин налагаме ограничение върху представянето на ПКГ (които очевидно могат да се разширяват произволно с дублиране на понятията и релациите, така както конюнктивните формули могат да се разширяват чрез дублиране на конюнкти). Но това ограничение не засяга семантиката на ПКГ и изразните им средства да представят знания за света. Така че в нашия подход става дума за едно незначително опростяване, дължащо се на прагматичния стремеж да се постигне изброяване на крайното множество от обобщения на подграфите в дадена база от знания в определен момент.

Общ случай на проекция

Всички ПКГ в нормална форма с непразни инективни проекции върху дадена база от знания съдържат най-много k елементарни конюнкта, където k е броят на концептуалните релации в най-дългия ПКГ от базата. Затова в определен момент можем да изброим off-line всички въпроси-ПКГ с непразни инективни проекции и да ги компресираме в минимален краен автомат. Обаче такова изброяване е невъзможно в общия случай на проекция, когато няма изискване πG да бъде изоморфен на G и затова въпросите с непразна проекция могат да имат произволен брой елементарни конюнкти. Ако дължината на въпроса се ограничи до някаква разумна константа за дадения момент, предложените по-горе разглеждания могат да бъдат повторени аналогично с цел да се построи краен автомат, кодиращ всички ПКГ-въпроси с непразна проекция за определена дължина на въпросите. Маркерите и анотациите трябва да се усложнят, за да се изброят явно всички изображения на екземпляри на понятия от потенциалните въпроси към екземпляри на понятия в базата от знания. Така ще се запомнят произволни проекции, а не само инективните.

Трябва да отбележим обаче, че построеният по алгоритъм 2.1 минимален автомат съхранява в себе си информация и за някои неинективни проекции, в случай че въпросът е по-къс от максималния граф в базата. Да разгледаме например графа на Фиг. 2.7а с анотация '1=3' и да си го представим като въпрос към базата знания, компресирана в автомата на Фиг. 2.11. Графът на Фиг. 2.7а се проектира в графа на Фиг. 2.7б, който има анотация '1=3,2=4'. Трасирането автомата от Фиг. 2.11 с дума LOVE EXPR PERSON LOVE OBJ PERSON води до маркера M9, който съдържа подграф с анотация '1=3,2=4'. Едно тривиално разширение на алгоритъм 2.2 ще позволи да се намират чрез трасиране и проекции от този вид, в случаите когато анотацията на въпроса се съдържа като низ в анотацията на отговора. Според автора, най-ценна за практическите приложения е операцията инективна проекция, която позволява търсене на изоморфни концептуални шаблони. Поради това нашето внимание е фокусирано върху нея.

Предложеният двуфазов подход реализира 'един такт' от задачата за търсене на концептуална информация и това е именно намирането на специализация, като всеки екземпляр се проектира в друг екземпляр и всяка релация – в друга. Практическата стойност на подхода се заключава в ускореното пресмятане на специализации.

По принцип търсенето на информация в човешки смисъл се извършва на по-едри стъпки; нека си припомним примера в част 2.1, където понятието ПОЛИТИЧЕСКО СЪБИТИЕ се специализира до по-частните понятия «*среща на министрите на ЕС*», «*изявление на президента*» и т.н. В обзорната първа глава споменахме, че при концептуалните графи се отделя голямо внимание на дефинициите на типовете и на операциите за разширяване и свиване на типове. На пръв поглед споменатият в част 2.1 пример показва комбинация от неколкократно прилагане на специализации и операции за свиване/разширение на тип:

- *Политическо събитие* е специализирано до *среща на политически фигури*,
- *Политическа фигура* е специализирано до *финансов министър на страна от ЕС* и т.н.

По-внимателно вглеждане в човешкия подход за обработка на концептуална информация ни показва, че комбинирането на няколко операции наведнъж е често срещано, с което отговорът на даден въпрос става по-ефективен.

Фиг. 2.16 ни подсказва как е реализирана специализацията в примера от част 2.1. Вижда се, че извличането на единичен подграф-специализация няма да е особено информативно в случай на отговори на естествен език – освен ако не се покаже и контекстът на цялостния факт, от който е извлечен отговора чрез инективна проекция на въпроса в отделни екземпляри и релации между тях. Тук предложеният подход също ни носи предимство, защото в маркера на заключителното състояние ние съхраняваме указател към оригиналния ПКГ, от

който е извлечена инективната проекция – т.е. имаме бърз начин да достигнем до контекста.

Фигура 2.16. Отговор чрез инективна проекция с показване на контекста: въпросът се проектира върху графа '*На среща на финансовите министри на страните от ЕС, (проведена) миналия вторник в Брюксел, (имаше) обсъждане на цената на суровия петрол*'

На Фиг. 2.16 е използван специфичният референт за група индивиди с краен брой членове [ФИНАНСОВ-МИНИСТЪР-на-ДЪРЖАВА-ЧЛЕН-на-ЕС: plural], който е представен в първа глава. Тук не коментираме как 'МИНАЛА' седмица се проектира в 'МИНАЛ' вторник (на английски този затрудняващ момент се губи, понеже и в двата случая се казва 'LAST'). Забелязваме възможността за изказване на факта по различни начини:

- *... министрите обсъдиха цената на суровия петрол ...*
- *... имаше обсъждане на цената на суровия петрол ...*
- *... проведе се обсъждане на ....*

Обикновено при кодиране на знанието в декларативни концептуални структури се загубват по-тънките стилистични детайли на естествения език.

Като заключение на тази част бихме казали следното: ако някой е в състояние да 'превежда' текстове на естествен език до вътрешни семантични представяния като показаното на Фиг. 2.16, задачата за автоматично отговаряне на въпроси би се решавала много по-успешно, например чрез предложения тук двуфазов подход. За

жалост самият превод е невъзможен в голяма предметна област, поне засега, и се извършва със задоволителна точност в ограничени светове (и то след многогодишен труд за разработка на съответните алгоритми и програми). Освен това, ако искаме да сме последователни в търсене на аналогия с подхода на Гугъл за предварителна off-line обработка, би следвало да извършим и запомним предварително и умозаключенията над фактите в базата: например, да разгънем

[ФИНАНСОВ-МИНИСТЪР-на-ДЪРЖАВА-ЧЛЕН-на-ЕС: plural]

до изброяване на 27 екземляра, по един за всяка страна, които са свързани с релация AGNT към СРЕЩА и с релация AGNT към ОБСЪЖДАНЕ. Така бихме били готови да отговорим чрез специализация на въпрос от рода на «*В каква среща участва финансовият министър на България миналия вторник*». Но тази забележка е изцяло в сферата на предположенията как би могла да се развие областта. Доколкото днес се извършват автоматични изводи в процеса на отговори на въпроси, те се правят само в реално време след получаване на заявката на потребителя.

ГЛАВА 3:

Концептуални структури и обработка на естествен език

В тази глава са представени резултати, свързани с приложения на концептуалните структури в изследователски прототипи за автоматична обработка на езика. Ще опишем алгоритъма за подбор на знанията за генератора EGEN, който произвежда обяснения на български и немски език в техническа област, и няколко експериментални компонента за анализ на естествения език – за разрешаване на анафорични връзки, за обработка на отрицание в изречения от специален тип и за извличане на знания от текст. Тъй като авторът е бил отговорен за обработката на концептуалните структури при създаването на споменатите по-горе прототипи на интелигентни системи, в част 3.2 се резюмират оригиналните особености на използваните концептуални ресурси.

3.1. Концептуални структури при генерация на обяснения на естествен език в техническа област

Тук са представени накратко приносите на автора при реализацията на компонента за извличане на знания в проектите DB-MAT (1992-1995) и DBR-MAT (1996-1998), финансирани от Фондация 'Фолксваген' (Германия)[14]. Личните приноси са свързани с:

- Дизайн на формàта и съдържанието на системните концептуални ресурси и
- Проектирането и реализацията на алгоритмите, които извличат релевантни факти от базата от знания при зададена заявка за генерация, и ги подават на генератора за вербализация на обяснението (т. нар. Query Mapper).

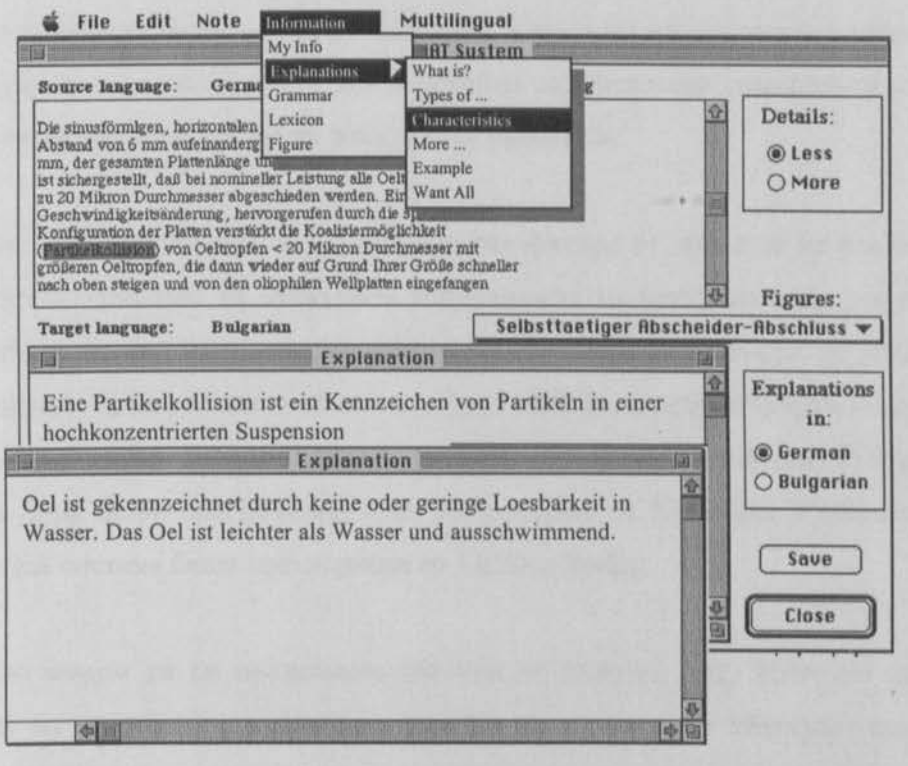
Идеята на проектите DB(R)-MAT се заражда през 90-те години на 20-ти век, когато в компютърната лингвистика активно се изследват начините за изграждане на системи, базирани върху знание. По това време се разработва и един от най-големите прототипи на система, базирана върху знание, в рамките на проекта *KBMT* (Knowledge-Based Machine Translation) [GoNi91]. В DB-MAT целта е да се реализира система, генерираща при поискване обяснения на факти от предметната област за нуждите на преводач, който превежда технически документ (и търси разяснения, за да го преведе по-добре). Известно е, че преводачите влагат не повече от 2/3 от времето си в оформяне на самия текст на превода, а над 1/3 от усилията им са свързани с търсене на информация

[14] http://nats-www.informatik.uni-hamburg.de/~dbrmat/db-mat.html, последно посещение 13 април 2009 год. Вж. също http://www.lml.bas.bg/~galja/db-mat

относно областта на превода и подбор на необходимите термини на другия език [FHA90, KiWi90]. Задачата на DB-MAT може да се опише и по следния начин: *Имаме една база от декларативно-представени знания, създадена за решаване на дадена задача на изкуствения интелект. Как да добавим лексикон към тази база и да я употребим като източник на знания за генерация на обяснения на естествен език?* Тази задача е съзвучна и с популярната преди 15-20 години идея за създаването на терминологични банки знания, която се поставя от 'терминолози' - предимно лингвисти, изучаващи т.нар. LSP (Languages for Special Purposes) и се развива в конференциите 'Terminology and Knowledge Engineering'. По същество организацията на терминологията е организация на знанието [GaBu96], поради неразривната връзка между концептуалните представяния и естествения език като носител на семантична информация. Текстови описания на терминологично знание обикновено се дават в тезаурусите и в специални речници на терминологията, които са от енциклопедичен тип и са най-често едноезични. През 90-те години обаче се оформят идеи за натрупване на електронни банки с формализирани обяснения на значението на термините; има проекти за създаване на концептуални йерархии, които са като 'вътрешни скелети' на терминологичните банки и към тях са 'закачени' самите термини на различни езици [MSBE92, Mey94, EMe95]. Водят се дискусии дали трябва да има една унифицирана концептуална йерархия в многоезична терминологична банка или пък много йерархии, по една за всеки естествен език, и преводните съответствия на термините да се задават чрез идентифициране на съответни една на друга концептуални единици в двата езика [AhDa90]. Идеята за концептуална съставка или 'концептуална координата' във всяка терминологична колекция е толкова популярна, че се налага в стандартите за терминологични банки по ISO [ISO96]. И до днес се счита, че терминологичният ресурс е 'съвременен', ако е организиран около вътрешна концептуална йерархия, и тази препоръка се отправя към проектите за съставяне на терминологични колекции [ETB05]. Но и досега липсват убедителни публично-достъпни примери за успеха на такова терминологично начинание в голям мащаб. Контекстът на проекта DB-MAT е именно концептуалното моделиране на терминологични колекции в двуезичен план. В началото на проекта беше моделирана юридическа терминология; впоследствие за целите на генерацията беше избрана техническата област на пречистване на отпадни води. Значимите концептуални единици представляват понятия, чиито наименования са термини – т.е. сравнително установени лексикални единици с фиксирано значение и добре-определена грануларност [vHAn94].

3.1.1. Интерфейс на DB-MAT – работно място на преводача

Тъй като проектът DB-MAT създава прототип на система за компютърно-подпомогнат човешки превод (MAT - Machine-Aided Translation), интерфейсът е проектиран като работно място на преводач, който превежда между български и немски език (Фиг. 3.1). Преводачът чете текста на входния език – на фигурата това е немски, в прозореца `'Source language'` и пише веднага текст на изходния език – в случая на български, в прозореца `'Target language'`. Като работно място за преводача, DB-MAT предлага граматични сведения (главно по морфология, за склонение на думите) и достъп до вградени едноезични речници. Това става от менюто `'Information'`, което е отворено на Фиг. 3.1, чрез което се получават сведения за езика в `'Source language'`. Чрез менюто `'Multilingual'` се предлагат сведения от двуезични немско-български или българо-немски лексикони (в зависимост коя е двойката входен-изходен език за превод). Тези функции са типични за работно място на преводача, например ESPRIT-проектът *Translator's Workbench* предлага освен това вграждане на коректори за граматика, пунктуация и стил; преводаческа памет; многоезична терминологична банка и достъп до система за напълно автоматичен машинен превод [TWB91].



Фигура 3.1. Работно място на потребителя-преводач в системата DB-MAT (прозорецът `'Explanations'` с генерираното обяснение на немски език е разположен над екрана с българския текст; допълнително е монтиран и втори пример обяснение) [vHAn96]

Като изследователски проект, DB-MAT се насочва към една нестандартна задача – да предложи на преводача обяснения на знанията в предметната област. Това са кратки текстове, генерирани динамично при поискване. Заявката на потребителя се формулира чрез осветяване на термин в полето на изходния език и избиране на `'Explanations'` от менюто. На Фиг. 3.1 е избран терминът Partikelkollision и за него са поискани обяснения относно характеристиките или свойствата на понятието. Самите обяснения могат да бъдат генерирани на немски или български език в зависимост от желанието на потребителя; това подпомага процеса на превода, тъй като преводачът вижда термините на другия език – не само конкретния избран термин, но и термините за непосредствено-свързаните с него понятия. Обясненията могат да бъдат по-детайлни или по-общи; този параметър също се контролира чрез бутон на главния интерфейс. На Фиг. 3.1 е показано и обяснението, генерирано в отговор на потребителска заявка: *Сблъскването на частици е свойство на частиците във високо-концентрирана суспензия* (на немски език). Това изречение вербализира намерените в базата от знания факти; при добавяне на повече факти, то ще изглежда по друг начин. На монтирания втори екран за обяснения е даден друг текст, генериран динамично също на немски език след запитване за свойствата на *Öl* (*масло*): *Маслото се характеризира с никаква или ниска разтворимост във вода. Маслото е по-леко от водата и изплуващо.* При генерацията на това обяснение в две изречения са обединени намерените в базата от знания четири характеристики на понятието с етикет *Öl*.

Освен това чрез съответното меню са достъпни фигури от областта на пречистване на отпадни води. Показват са сканирани изображения на най-типичните илюстрации в учебниците и техническите енциклопедии, които бяха използвани за извличане на концептуалните графи в базата от знания. През 1995 беше демонстриран генераторът за немски език с автор Калина Бончева на LPA Prolog за Макинтош. Генераторът за български език беше програмиран през 1998 година от Кристина Тутанова. По това време цялата система беше прехвърлена на SICStus Prolog.

Днес също можем да си представим система от подобен вид, която по желание на преводача му предоставя динамично достъп до извадки от Интернет-документи и списък препратки към релевантни помощни (едно- и многоезични) източници. Най-вероятно днес биха се извличали направо параграфи от текстовете, с предлагане на стотици алтернативи. В проекта DB-MAT обаче съществото е извличане на

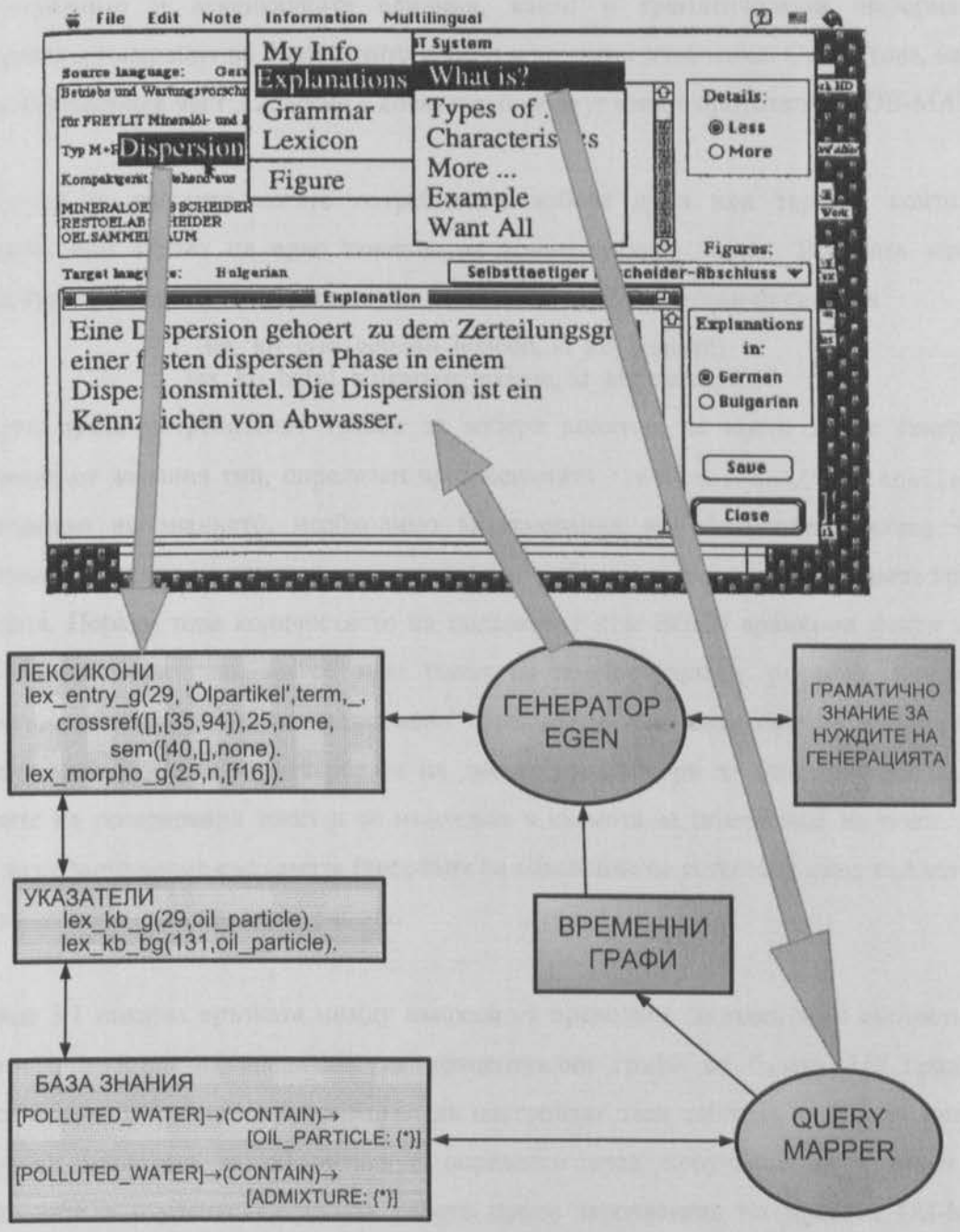
фрагменти от концептуални графи, обединението им в едно цяло и разказването им на естествен език с цел произвеждане на компактно обяснение.

3.1.2. Архитектура на прототипа DB-MAT

Системата има няколко главни компонента, които са показани на Фиг. 3.2. След избиране на термин от входния текст и на потребителска заявка за него (чрез съответния елемент на някое меню), посоченият термин се свежда до основната му форма, по нея се търси в лексикона на системата и оттам – което е важно за нашите разглеждания – в списък указатели, задаващи съответствията между етикетите на концептуалните единици в базата от знания и думите/термините като лексикални елементи. Веднага отбелязваме, че има отделни указатели за връзка от немския и българския лексикони към концептуалните етикети, като последните са на английски език. Това решение беше избрано поради необходимостта да се управлява концептуалната и лексикална грануларност на понятията в базата и на термините в двата езика – немски и български.

След успешното намиране на етикет в базата от знания, той - заедно с код за вида на потребителската заявка - се подава на модул за извличане на релевантно знание. Модулът е наречен Query Mapper и формира (под-)факти на твърдения в базата от концептуални структури, които трябва да се обяснят на̀ преводача в отговор на неговата заявка. Извлеченото знание се подава на генератора EGEN като файл от временни графи, който се пази до края на текущата сесия с цел да се избягват повторенията (т.е. да не се разказва повторно факт, който е бил обяснен преди няколко стъпки в диалога). Генераторът ползва лексиконите при 'обличане на фактите с думи', при което му е необходима съдържащата се там морфологична информация, но разполага и със собствено граматично знание за синтактични рамки на глаголите, граматика за синтезиране на клаузи с вложени изречения, правила за поставяне на членуване и пунктуация и т.н. Освен това генераторът избира дискурсна схема за запълване с подадените временни графи. След формиране на отговор, генераторът отваря прозореца `'Explanation'` и показва в него текста на съответния език. Така приключва една стъпка в диалога между преводача и системата. Потребителят може отново да освети термин от активните прозорци `'Source language'` и `'Explanation'`

и да зададе отново въпрос към системата или да продължи с превода в прозореца 'Target language' (който е разположен под 'Explanation' на Фиг. 3.1 и Фиг. 3.2). Следващ въпрос ще предизвика същата поредица от действия, като извлечените временни графи ще бъдат сравнени с вече показваните. Ще се генерира текст само по новото знание, освен ако не се поиска обяснение на другия език.



Фигура 3.2. Основни компоненти и езикови ресурси в системата DB-MAT и тяхна роля при изпълнение на потребителската заявка

3.1.3. Извличане на знание в отговор на въпрос за обяснения

Ще представим компонент Query Mapper, описан в [AnBo96a, AnBo96b и AnBo96c]. Той осигурява едно от постиженията на системата DB-MAT – а именно, че базата от знания е независима от процеса на генерация и по този начин може да бъде заменена с друга, стига да бъдат заменени и системните лексикони, указателите за връзка между концептуалните и лексикалните единици, както и граматическата информация осигуряваща генерация на съответните прости и вложени изречения. Освен това, както ще бъде показано в част 3.2, лесно е добавянето на друг език в прототипа на DB-MAT.

Обяснение се генерира когато потребителят избере дума или термин, които са указатели към етикет на едно понятие от концептуалния модел. Връзката между лексикона и базата се задава чрез показаните на Фиг. 3.2 указатели-предикати

lex_kb_g(id_german_lexicon, id_kb_concept).
lex_kb_bg(id_bulgarian_lexicon, id_kb_concept).

С други думи, потребителят трябва да избере понятие, за което да се генерира обяснение от желания тип, определен чрез менютата `'Information'/'Explanation'`. Извличането на знанието, необходимо за генерация на обяснението, става чрез инективна проекция на предварително зададени шаблони върху концептуалните графи от базата. Поради това количеството на подаваните към EGEN временни факти не е известно предварително, но се знае типът на концептуалните релации, които са включени в подаваните за 'разказване' фрагменти. По своеобразен начин тези концептуални релации играят ролята на дискурсни маркери за отношенията между клаузите на генерирания текст и се използват в схемата за генериране на текст. Тук няма да се занимаваме със самата генерация на обяснения на естествен език, тъй като тя е извън обхвата на настоящата работа.

Таблица 3.1 показва връзката между въпроса на преводача, зададен чрез съответните менюта и процеса на извличане на концептуални графи от базата. По принцип напредналите потребители биха могли да настройват тази таблица за своите нужди. Видът на менютата за обяснения е определен след проучване за нуждите на преводачите в студентска курсова работа преди започването на проекта DB-MAT [KiWi90]. Оказва се, че преводачите имат нужда от обяснения, но не желаят да четат дълги специализирани текстове с цел да получат допълнителна информация. В рамките

на проекта бяха избрани шаблони, които фрагментират обясненията на по-кратки порции, и това беше извършено след внимателно проучване на структурата на текста в енциклопедични речници. Всеки ред от Таблица 3.1 фиксира съдържанието на бъдещото обяснение чрез списъка концептуални релации за извличане. Например, когато преводачът избере дума или термин от текста за превод (т.е. понятие X) и попита `'What is'`, той получава обяснение, съдържащо:

#	Подменю	Елемент на подменюто	Концептуални релации, които се извличат за формиране на отговора	Наследяване
q1:	*What is?*		*Types of ... /All* и ATTR, CHAR, PART-OF	да
	Types of ...			
q2:		*All*	над- и под-понятия и понятия-сестри	
q3:		*General*	всички над-понятия от йерархията	
q4:		*Concrete*	всички под-понятия от йерархията	
q5:		*Similar*	всички понятия-сестри от йерархията	
	Characteristics			
q6:		*All*	всички по-долни подменюта заедно	да
q7:		*Attributes*	ATTR, CHAR	да
q8:		*Who*	AGNT	
q9:		*Object*	OBJ, PTNT	
q10:		*How*	INSTR	
q11:		*Where*	LOC, DEST, FROM, IN, TO	
q12:	*More ...*		всички други релации, които не са включени в горните редове	
q13:	*Examples*		индивидуални екземпляри	
q14:	*Want all*		всичко заедно без повторения	да

Таблица 3.1. Видове въпроси от менюто `'Explanation'` и релации за извличане на отговор

- концептуалната околност на X – т.е. отговорите на въпроса *Types of ... /All*,
- атрибути на X – т.е. понятия, свързани към X с релацията ATTR,
- характеристики на X – т.е. понятия, свързани към X с релацията CHAR,
- части на X и сведения за други понятия, в които X е част – т.е. понятия, свързани към X с релацията PART-OF.

Зад Таблица 3.1 стои списък шаблони на прости концептуални графи, които се използват като въпроси за инективна проекция към базата от знания. Например, за извличането на концептуалната релация OBJ се използват шаблони от следните видове (където [T:?] означава «кое да е понятие»):

[избрано-понятие] → (OBJ) → [T:?]
[избрано-понятие] ← (OBJ) ← [T:?]
[избрано-понятие] → (OBJ) → [T:?] → (ATTR) → [T:?]
[избрано-понятие] ← (OBJ) ← [T:?] → (ATTR) → [T:?]
[избрано-понятие] → (OBJ) → [T:?] → (CHAR) → [T:?]
[избрано-понятие] ← (OBJ) ← [T:?] → (CHAR) → [T:?]

Шаблоните за проекция за 'размножени' до графи с максимален за базата брой релации ATTR и CHAR, като търсенето започва с опити за проекция на по-дългите шаблони. Целта е да се извлече факт, при който избраното понятие X е във връзка OBJ с друго понятие Y с характеристики Z_1, Z_2, ... Тази особеност на DB-MAT – че извлеченото знание формира смисъла на едно бъдещо изречение – определено влияе на решенията как да се декларират графите в базата (вж. част 3.2). Ситуациите се извличат като едно цяло понятие при намиране на релевантни знания с компонента Query Mapper. Дефинициите на тип се вербализират винаги като цяло, т.е. не се правят проекции с цел намиране на подграфи от дефинициите на типове.

Освен извличането на подграфи и организирането им в списък временни факти, които се подават за генерация, компонентът Query Mapper извършва следното:

- изтрива повтарящите се факти в списъка временни структури, като запазва информация откъде са били извлечени и с кой шаблон (за да се избегнат повторения при последващ въпрос);
- изтрива факти с повтарящо се съдържание съгласно исканата детайлност, например ако са извлечени

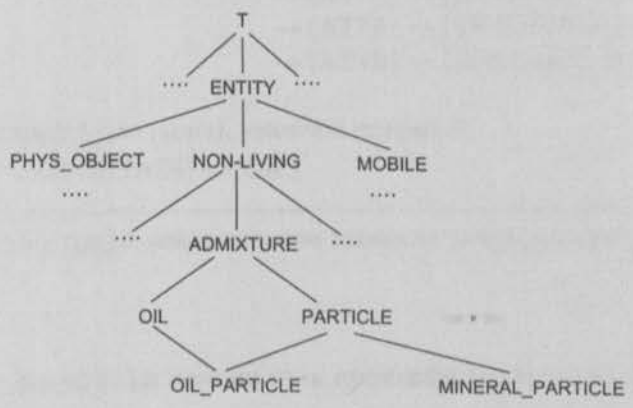
  [POLLUTED_WATER] → (CONTAIN) → [OIL] и
  [POLLUTED_WATER] → (CONTAIN) → [OIL_FRAGMENT: {*}]

  долният факт може да бъде изтрит при по-общ отговор, тъй като OIL е едно от надпонятията на OIL_FRAGMENT в йерархията на типовете понятия;
- пази временните графи до следващия въпрос, понеже потребителят може да освети някой термин от прозореца с показаното обяснение и да продължи да пита, например за обяснение на другия език, така че по принцип е възможно временните графи да бъдат използвани повторно.

Тъй като ефектът от работата на компонента Query Mapper се вижда най-добре чрез генерираните обяснения (а и така навлизаме в специфичните проблеми на управление на лексикалната и концептуална грануларност, при които дадено концептуално съдържание трябва да бъде облечено с думи в определена граматична форма на някой

естествен език), по-долу ще разгледаме примери за извлечени факти и съответни генерирани обяснения както на немски, така и на български език.

Пример 3.1. [AnBo96b] Да разгледаме фрагмент от йерархията на типовете понятия и пет концептуални графа в базата от знания на DB-MAT, представени на Фиг. 3.3. В граф 2 и граф 3 виждаме използването на ситуации (***situation***, специфичен контекст указващ възможността да се извърши операцията свиване на тип, който се третира като едно цяло понятие при извличане на подграфи от базата). Ще дадем примери за поведението на системата при различни въпроси, с избор на различни стойности за детайлност чрез радио-бутоните `Details(Less,More)` на основния екран на системата. Тези бутони позволяват на потребителя да контролира количеството характеристики обяснени за по-долните понятия, с цел да научава повече свойства наведнъж. В Таблица 3.1 е показано как се наследяват характеристиките от по-горните понятия при извличане на подграфи, но само за определени видове заявки (вж. стойностите *'да'* в последния стълб на Таблица 3.1).



концептуален граф 1:
[POLLUTED_WATER] → (CHAR) → [DISPERSION] → (OF) → [PARTICLE: {*}]
→ (CHAR) → [CONCENTRATION] → (OF) → [PARTICLE: {*}]
→ (CONTAIN) → [ADMIXTURE: {*}] .

концептуален граф 2:
[PRECIPITATION] → (OBJ) → [SITUATION:
[POLLUTED_WATER] → (CONTAIN) → [OIL_PARTICLE: {*}]
→ (ATTR) → [LIGHTER_THAN_WATER]
→ (ATTR) → [SWIMMING_UP]
→ (ATTR) → [ROUGHLY_DISPERSED]] .

концептуален граф 3:
[INDUSTRY] → (ATTR) → [OIL_PROCESSING]
→ (RESULT) → [SITUATION: [POLLUTED_WATER] → (CONTAIN) → [OIL]] .

концептуален граф 4:
[OIL_PARTICLE] → (CHAR) → [DENSITY] .

концептуален граф 5:
[PARTICLE] → (CHAR) → [DIMENSION] .

Фигура 3.3. Твърдения в базата от знания на системата DB-MAT (версия описана в [AnBo96b])

Нека предположим, че извличането на знания за отговора става само от ресурса, показан на Фиг. 3.3. Ако потребителят избере в текста термина 'отпадъчна вода' на някой език и запита чрез менюто за обяснения `'Characteristics/Attributes'`, при избора на релевантно знание ще бъдат претърсени дефиницията на типа POLLUTED_WATER и конкретните факти (концептуални графи) за отделни екземпляри на понятието POLLUTED_WATER, които са свързани с други понятия чрез концептуалните релации ATTR и CHAR. От граф 1 чрез инективна проекция ще бъде извлечен временният концептуален граф 1 от Таблица 3.2.

Временен концептуален граф 1, извлечен от граф 1:
[POLLUTED_WATER] → (CHAR) → [DISPERSION]
→ (CHAR) → [CONCENTRATION]

Временен концептуален граф 2, извлечен от граф 1:
[POLLUTED_WATER] → (CONTAIN) → [ADMIXTURE: {*}]

Временен концептуален граф 3 (ситуация), извлечен от граф 2:
[POLLUTED_WATER] → (CONTAIN) → [OIL_PARTICLE: {*}]
→ (ATTR) → [LIGHTER_THAN_WATER]
→ (ATTR) → [SWIMMING_UP]
→ (ATTR) → [ROUGHLY_DISPERSED]

Временен концептуален граф 4 (ситуация), извлечен от граф 3:
[POLLUTED_WATER] → (CONTAIN) → [OIL]

Таблица 3.2. Временни графи, извличани при заявки за генериране на обяснения

От временния граф 1 генераторът на немски език произвежда следното изречение:

Abwasser is gekennzeichnet durch Dispersion und Konzentration.
(Отпадъчната вода се характеризира с дисперсия и концептрация.)

Ако граф 1 в базата от знания изглеждаше по следния начин:

[POLLUTED_WATER] -
→ (CHAR) → [SITUATION: [DISPERSION] → (OF) → [PARTICLE: {*}]]
→ (CHAR) → [SITUATION: [CONCENTRATION] → (OF) → [PARTICLE: {*}]]
→ (CONTAIN) → [ADMIXTURE: {*}]

то тогава временният граф 1 би съдържал целите ситуации като контекст:

[POLLUTED_WATER] -
→ (CHAR) → [SITUATION: [DISPERSION] → (OF) → [PARTICLE: {*}]]
→ (CHAR) → [SITUATION: [CONCENTRATION] → (OF) → [PARTICLE: {*}]]

и генерираното обяснение би изглеждало както следва:

Abwasser is gekennzeichnet durch Dispersion der Partikeln und Konzentration der Partikeln.

(Отпадъчната вода се характеризира с дисперсия на частиците и концептрация на частиците.)

Ще дадем и примери за смяната на нивото на детайлност на обясненията. Това се извършва чрез евристична стратегия, като стремежът е при `'Details=Less'` да се вербализират факти за по-горните понятия в йерархията на типовете понятия, а при `'Details=More'` да се разказват факти за понятията от по-долните нива с наследяване на характеристиките на над-понятията. Да разгледаме временните графи 2, 3 и 4 в Таблица 3.2, извлечени за понятието POLLUTED_WATER при въпрос `'More … '`, когато се вербализира концептуалната релация CONTAIN. За самото понятие POLLUTED_WATER, което е фокус на обяснението, няма характеристики за наследяване от неговите надпонятия. Затова при вземане на решение за детайлността се разглеждат извлечените факти. Забелязваме, че и трите временни графа описват някакви примеси – ADMIXTURE, OIL, OIL_PARTICLE - които се съдържат в отпадъчната вода. И трите понятия са дефинирани в йерархията на типовете понятия от Фиг. 3.3, но OIL и OIL_PARTICLE не са единствени наследници на надтиповете си. Затова се взема решение винаги да се показва най-горното понятие ADMIXTURE и едно от OIL или OIL_PARTICLE според желаната детайлност. При `'Details=Less'` генераторът ще съедини временни графи 2 и 4 и ще произведе следното обяснение:
Abwasser enthält Öl und Beimischungen. (*Отпадъчната вода съдържа масло и примеси*).
При `'Details=More'` EGEN ще съедини временни графи 2 и 3 и ще генерира:
Abwasser enthält Beimischungen und grobdisperse und ausschwimmende Ölpartikeln welche leichter als wasser sind.
(*Отпадъчната вода съдържа примеси и грубодиспергирани и изплуващи маслени частици, които са по-леки от водата.*)

Всеки факт в базата от знания може да бъде разказан в обяснение, стига да се постави необходимия въпрос. Очевидно едни и същи факти (подграфи) ще бъдат извличани при отговор на различни въпроси, тъй като концептуалните релации свързат винаги две понятия – а потребителският въпрос касае едното от тях. По този начин временните графи от Таблица 3.2 могат да бъдат извличани и по други поводи (не само при поискване на обчснения за `'отпадъчна вода'`). Например, ако потребителят избере в текста понятието `'Beimischung'` (примес) и поиска обяснение за него чрез менюто `'More … '`, отново ще бъде извлечен временния граф 2

[POLLUTED_WATER] → (CONTAIN) → [ADMIXTURE: {*}]

и ще бъде генерирано обяснението:

Beimischungen sind enthalten in Abwasser.

(Примеси се съдържат в отпадъчната вода.)

Избраното от потребителя понятие 'Beimischung' става глобален фокус на бъдещото обяснение. Естествената му роля в генерираните изречения е да заеме мястото на подлога. В този случай отговорът на системата е в страдателен залог, тъй като временният граф 2 се трасира срещу стрелките на релациите при конструиране на изречението. По-горе показахме вербализация на временния граф 2 като

Abwasser enthält Beimischungen (*Отпадъчната вода съдържа примеси)*

при избран фокус 'отпадъчна вода'.

Ще дадем и един пример за генериран дискурс по понятията от базата на Фиг. 3.3. За понятието OIL_PARTICLE на въпрос 'What is?' се генерира следното обяснение при 'Details= Less':

Ölphasen (Ölpartikel[1]) gehören zu Partikeln[2]. Die[3] Ölphasen sind gekennzeichnet durch Dichte[4]. Die ausschwimmenden[5] und grobdispersen[6] Ölphasen, welche leichter als Wasser sind[7], sind enthalten in Abwasser[8].

Маслените[9] частици са частици. Маслените частици се характеризират с плътност. Маслени частици[10], които се съдържат в отпадъчна вода[10], са изплуващи, грубодиспергирани и по-леки от водата.

В този пример 1 е синоним от лексикона; 2 е надтип от концептуалната йерархия; 3 е определителен член, поставен поради предишното споменаване на обекта Ölphasen; 4 е характеристиката *плътност*; 5 и 6 са характеристиките *изплуващ* и *грубодиспергиран* в съответното съгласуване; 7 е реализирано като подчинено изречение, тъй като в лексикона не е намерено единично прилагателно за изказването му; 8 е повърхнинна реализация в страдателен залог, тъй като графът *'отпадъчната вода съдържа ....''* се вербализира в изречение, където Ölphasen са заели ролята на фокус (и подлог) и поради това граматиката предлага само възможност за изказване на конкретния факт в страдателен залог; 9 е членуване на български, което не се среща в немския текст; 10 е пример за неудачно членуване на български от тактическия компонент.

При 'Details=More' в горния дискурс би настъпила една промяна:

Die Ölphasen sind gekennzeichnet durch Dichte und Dimension

(Маслените частици се характеризират с плътност и размер),

където свойството *размерност* е наследено от надпонятието PARTICLE. Тук следва да се отбележи, че системата DB-MAT се развиваше в продължение на три години (след първата демонстрация на генератора на немски език през 1995 год.). През това време бяха фиксирани няколко сценария за усъвършенстване на процеса на извличане на знанията с цел подобряване на качеството на генерираните обяснения. Също така се предложиха начини за прецизиране на декларативно-представеното знание. В приложените към този труд авторски статии се описват различни състояния на прототипа, които са последователни стъпки в развитието на проекта. Таблица 3.3 съдържа генерирани обяснения на български език, които илюстрират съдържанието на базата от знания в заключителната фаза на проекта.

3.1.4. Моделиране на потребителя

В [NeAn99] е представен един евристичен подход за наблюдение на поредицата от заявки на потребителя и модифициране или на шаблона, по който се извлича знанието за разказване в следващия отговор, или на временните графи запазени от предишния въпрос. След запознаване с принципите на генерация в DB-MAT става ясно, че има необходимост системата да се погрижи за кохерентността поне на две съседни обяснения. По-горе показахме генерираните изречения в страдателен и действителен залог, които идват от един и същи временен граф:

Маслените частици се съдържат в отпадъчната вода и

Отпадъчната вода съдържа маслени частици.

Съгласно дадените в Таблица 3.1 менюта и релации за управление на извличането на знания от базата, първото обяснение ще се изведе при задаване на въпроси q1, q6 или q7 за 'маслени частици', а второто – при въпроси q1, q6 или q7 за 'отпадъчна вода'. Ако потребителят задава въпросите за двете понятия непосредствено един след друг, няма смисъл при втория въпрос да се вербализира току-що разказан факт от отговора на първия. По принцип, в DB-MAT не се поддържа профил на потребителя – например, той не отговаря на въпросник дали е специалист в предметната област или начинаещ, но системата натрупва някои наблюдения относно поведението му. Реализиран е агент, който следи поредицата от въпроси и понятия и в зависимост от ситуацията взема решение за изтриване или включване на факти във файла с временните графи, подавани към генератора EGEN.

Подменю и негов елемент	Избрано понятие и генериран отговор на съответния въпрос
What is?	**Избрано понятие**: *аерозол* **Обяснение**: *Аерозолът е колоид на течни частици или твърди частици в газ. Аерозолът е колоид съдържащ газ. Димът и мъглата са аерозоли.* **Избрано понятие**: *колоид* **Обяснение**: *Колоидът е смес от микроскопични фрагменти на дисперсна фаза, която се състои от извънредно малки и равномерно и устойчиво разпределени частици, , в непрекъсната среда .*
Types of ...	
All	**Избрано понятие**: *отделяне с центрофуга* **Обяснение**: *Отделяне с центрофуга е отделяне на примеси както утаяване, флотация и филтрация.* **Избрано понятие**: *величина* **Обяснение**: *Величините са обекти както силите, веществата, институциите и физическите обекти. Размерност, плътност, съпротивление, относително тегло, концентрацията и степен на раздробеност са величини.*
General	**Избрано понятие**: *степен на раздробеност* **Обяснение**: *Степен на раздробеност е величина.*
Concrete	**Избрано понятие**: *степен на раздробеност* **Обяснение**: *Дисперсията е степен на раздробеност.* **Избрано понятие**: *синтетична емулсия* **Обяснение**: *Боята е синтетична емулсия.*
Similar	**Избрано понятие**: *колоид* **Обяснение**: *Колоидите са колоид съдържащ газ или колоид съдържащ течност.*
Characteristics	
Attributes	**Избрано понятие**: *газ* **Обяснение**: *Всеки газ се характеризира с неброимост.* **Избрано понятие**: *колизия на частиците* **Обяснение**: *Колизия на частиците е характеристика на маслени частици в високо-концентрирана суспензия.*
Who	**Избрано понятие**: *маслени частици* **Обяснение**: *Маслени частици които изплуват.*
Object	**Избрано понятие**: *гравитация* **Обяснение**: *Маслени частици в спокойна вода са обект на действие на гравитацията.*
More ...	**Избрано понятие**: *отпадъчна вода* **Обяснение**: *Отпадъчната вода съдържа масло, частици и грубо-дисперигирани, изплаващи и по-леки от водата маслени частици.* **Избрано понятие**: *маслени частици* **Обяснение**: *Грубо-дисперигирани, изплаващи и по-леки от водата маслени частици се съдържат в отпадъчна вода.* **Избрано понятие**: *колизия на частиците* **Обяснение**: *Колизия на частиците води до намаляване на скорост на изплуване и утаяване.*
Examples	**Избрано понятие**: *вертикален маслоуловител* **Генерирано**: *Сепараторът на Шел е вертикален маслоуловител.*

Таблица 3.3. Оригинални примери за генерирани отговори на български език.

Динамична промяна на количеството факти в генерираното обяснение може да става и чрез временна модификация на съдържанието на Таблица 3.1. Това се прави в следните случаи:

- Ако е отговорено на въпрос q6 или q7 и веднага след това се постави въпрос q1, то при отговора на q1 не се вербализират характеристиките, разказани в q6 или q7;
- Избягва се повторението между отговорите на въпросите q1 и q2-q5, ако тези въпроси са поставени непосредствено един след друг; т.е., избягва се повторна вербализацията на концептуалната йерархия или части от нея;
- Избягва се повторението между отговорите на въпросите q6 и q7-q11, ако тези въпроси са поставени непосредствено един след друг; т.е., избягва се повторна вербализацията на характеристики, местоположение и т.н.;
- Избягва се повторението между съдържанието на отговорите на въпросите q14 и q1-q13, ако такива въпроси са поставени непосредствено един след друг.

Както вече казахме, изтриването на току-що разказани временни графи е друг начин да се избегне дублиране на съдържанието на две последователно-генерирани обяснения. Това се случва често при различни понятия и въпроси, в зависимост от съдържанието на базата от знания. Освен горното обяснение за *маслените частици, които се съдържат в отпадъчната вода*, нека покажем и по-сложни случаи. Например, въпрос q8 за 'утаяване' ще предизвика генерацията на обяснението

Маслените частици, които са по-тежки от водата, се утаяват.

Непосредствено-следващ въпрос q8 за 'маслени частици' би следвало отново да включи това обяснение, но дублирането ще бъде избегнато по решения на моделиращия агент. Така отговорът ще съдържа само две изречения:

Маслените частици, които са по-леки от водата, изплуват.

Изплуващите маслени частици се залепят.

Показаните тук модификации на основния алгоритъм позволяват генерирането на по-естествени обяснения в диалогов режим, които са адаптирани към поредицата заявки в текущата сесия и съдържанието на вербализираните от системата знания.

3.1.5. Оценка на подхода

Подобно на всеки експериментален прототип, DB-MAT има свои силни и слаби страни. Един основен проблем е, че не е възможна генерацията на дълги сложни изречения, вербализиращи много големи графи и особено дефиниции на типове в базата от знания (например, с 20 понятия и 18 релации). Също така може да се извадят множество отделни фрагменти-подграфи с разнообразна структура, които да се подадат на генератора, и да се окаже, че той не е в състояние да ги разкаже. На практика алгоритъмът за подбор на релевантното знание работи независимо от лингвистичното осигуряване на системата. Запознатите с проблемите на повърхнинната реализация знаят, че е изключено да се генерират много дълги изречения; също така не могат да се очакват чудеса от генератора, като например да изкаже произволно сложен факт в няколко съседни изречения. Поради това в DB-MAT се налага да се използват концептуални шаблони, които извличат сравнително малки по обем графи и позволяват всяка клауза на подадено знание да се изкаже в едно изречение. Остава открит въпросът дали потребителят може да обобщи получените стъпка по стъпка обяснения и да построи по-цялостна представа за предметната област, но всъщност това е спорно и при текстове, генерирани от човека. В DB-MAT се разказва за 'непосредствената околност' на избраното понятие с надеждата, че получените обяснения подпомагат разбирането на текста за превод.

Друг проблем е наличието на известна изкуствена фрагментарност в генерирания текст и еднообразен стил на обясненията. Казано в термините на компютърната лингвистика, на EGEN му липсва т. нар. *domain communication knowledge* [KKR91] – знание как да се говори за отделните понятия; поради това генераторът произвежда един и същи тип обяснения за всеки обект, без да взема под внимание неговата значимост в предметната област, мнението на потребителя дали понятието не е тривиално и така нататък. Това личи и в приведените по-горе примери: например не е необходимо да се вербализира очевидният факт, че *маслените частици* са *частици* и др. под. Качеството на генерираните обяснения силно зависи от кодираното знание: неговите повторения, импликации и парафрази водят до появата на повторения и смислови дефекти в генерираните обяснения. Подобрения могат да настъпят чрез известно усложняване на концептуалния ресурс на системата, но от съществено значение е усложняването на езиковите ресурси. Въвеждането на допълнително знание обаче би увеличило

значително ръчния труд по конструиране на лексикалните, граматични и концептуални ресурси на системата.

Нека изброим и основните предимства на така описания подход за извличане на знания за целите на генерацията. Най-интересното положително свойство, осигурено от компонента Query Mapper е, че EGEN започва да 'говори' веднага щом му се зададе база от знания, лексикон и връзки между тях в съответния вътрешен формат. Обясненията се генерират динамично и се обработват всички налични концептуални графи от базата. Системата работи над нови, неизвестни графи, като прилага операциите за разширение или свиване на тип с цел да уголеми или намали обема на подаденото към генератора знание. Вложените графи (ситуациите) са създадени с цел описание на концептуални единици с по-голяма грануларност, което позволява установяване на връзки от елементите на базата към сложните немски съществителни, записани като една дума в немския лексикон. Включването на нов език изисква добавяне на съответни езикови данни и граматическа информация, но не и промяна на концептуалния ресурс. Българският език е добавен към немския генератор именно по този начин.

Модулността е едно друго голямо предимство на DB-MAT. Както се вижда от Фиг. 3.2, предметната област може да се смени с друга при промяна на четири ресурса:

- базата от знания,
- лексикона с термини и важни думи от новата област,
- списъка указатели, показащи връзката между термините и понятията, и
- граматическото знание, необходимо за генерацията (особено шаблони за изказване на по-важните глаголи от новата област).

Добавянето на друг естествен език става чрез включване на следните три ресурса за новия език, стига в базата да има понятия и контексти с нужната грануларност:

- лексикон с термини, важни думи, морфологичен модел и списък преводни съответствия,
- списък указатели, показащи връзката между термините на новия език и понятията в базата от знания, и
- граматическо знание, необходимо за генерацията на съответния език.

Тази модулност превръща системата в интересно решение за езиково инженерство. Поради компактността си и сравнителната си простота – която го прави стабилен и

предсказуем – прототипът на DB-MAT е подходящ за приложения в разнообразни области със специфичен стил на дискурса.

Както беше показано в обзорната първа глава, засега автоматичната генерация като цяло има своите ограничения (например не може да се генерира дълъг разказ на произволна тема), но в редица предметни области са необходими именно кратки, еднотипни съобщения. В Европа има голям интерес към прилагане на автоматичната генерация при осигуряване на достъп на гражданите до публична информация, а се очаква и Европейският парламент до 2015 год. да приеме директиви в тази насока и да задължи страните-членки да улеснят максимално разпространението на публична информация до всички граждани. Пример за това е един наскоро завършен проект за многоезикова генерацията на кратки текстове за наличие на замърсители във въздуха е MARQUIS (*Multimodal air quality information service for general public*)[15], финансиран от Европейската комисия по програмата eContent през 2005-2007 като информационна среда за разпространение на публична информация на 6 езика до европейските граждани. Текстовете за наличното замърсяване в конкретен район се показват в интернет, съобщават се заедно с метеорологичните прогнози по телевизията и радиото и се изпращат като съобщения на мобилни оператори на болнични заведения и абонирани за услугата потребители с респираторни заболявания. По този начин всеки час могат да се произвеждат нови текстове с дължина 1-5 изречения, като процесът е напълно автоматичен и използва данни, събирани от метеорологичните служби (които работят в цяла Европа по достатъчно стандартизиран начин). В тясната област могат да се изучат предварително всички класове потребители и да се създадат схеми за генерация на отделни типове обяснения, съобразени с моделите на потребителите. Технологията, представена в части 3.1 и 3.2, по своята гъвкавост и изчистена постановка за обработка на концептуалната информация е не по-лоша от алгоритмите за генерация на MARQUIS и също включва възможности за моделиране на потребителя. Тя е налице за българския език и лесно би се разширила към езиците на околните страни. Така конструираните алгоритми и решения за концептуално моделиране биха намерили и практическа реализация.

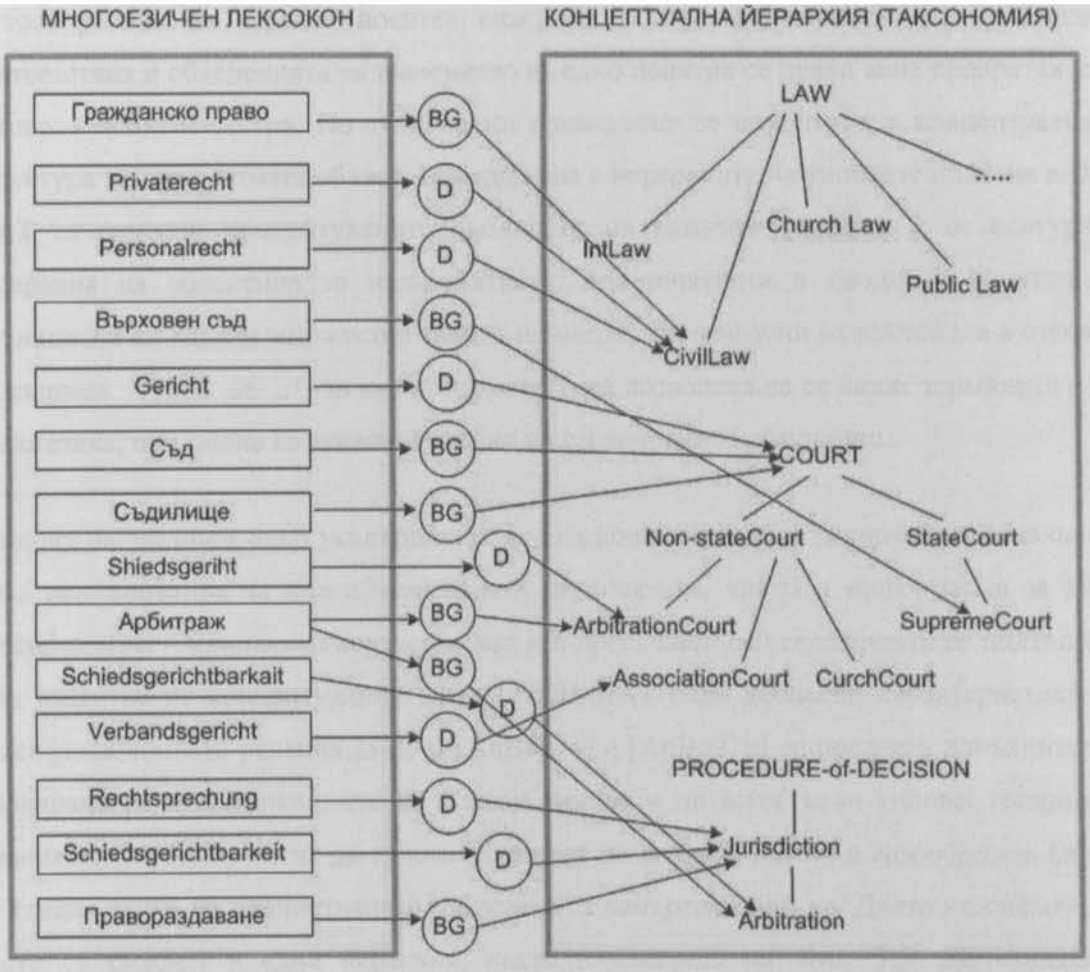
[15] http://www.epsiplus.org/products/weather_and_environment/marquis, последно посещение 26 април 2009.

3.2. Извличане на декларативни представяния на знания с цел подготовка на концептуален ресурс за обработка на естествен език

Ще представим най-съществените проектантски решения при дефиниране на знания за дадена предметната област. Тези решения са тясно свързани с необходимостта да се осигури унифицирано представяне на лексикални единици с различна грануларност на поне два езика – български и немски. Взето е предвид и типичното за немския език използване на сложни съществителни, които на практика функционират като една дума, но съответстват на българска фраза от няколко думи. В разглежданата техническа област най-важна роля играят термините: езикови единици с относително добре дефинирана и неизменна семантика, които реферират към концептуални елементи с относително стандартизирана, езиково-независима грануларност.

3.2.1. Концептуална йерархия на типове понятия като модел на термините

Всички важни термини са етикети на понятия от йерархията на понятията в опората на базата от знания. Макар системата да е развивана в продължителен период от време с прогресивно усъвършенстване на концептуалния модел, някои от постановките са основополагащи и бяха фиксирани в началния етап на проекта. Такава е постановката за наличие на единна концептуална йерархия с различни по грануларност понятия, към която са свързани термините от двуезичните лексикони. Йерархията е единна, понеже предметната област е една, а естествените езици са мйого. Фиг. 3.4 илюстрира принципите на свързване на лексикални и концептуални елементи в областта на правото, което беше първата експериментална предметна област в DB-MAT [Ang95]. Виждат се постановките за концептуално моделиране в терминологичните колекции, споменати в [AhDa90] и [ISO96], които бяха дискутирани в началото на тази глава. Термините '*Privaterecht*' и '*Personalrecht*' са синоними на немски, понеже реферират към едно и също понятие; терминът *'гражданско право'* е техен превод, тъй като реферира към същото понятие. На немски има термини като например *'арбитражен съд'*, които не са релевантни към сегашното юридическо устройство на държавните съдилища в България. Други понятия като българския *'върховен съд'* липсват в немското правораздаване (това са т.нар. terminological gaps) и т.н. Така концептуалната йерархия е нещо като идеален и максимално-покриващ модел на възможните понятия от света, в който всички многоезични термини намират своя адекватен референт. Такава йерархия



Фигура 3.4. Концептуална йерархия като посредник (или скелет) в терминологичен модел

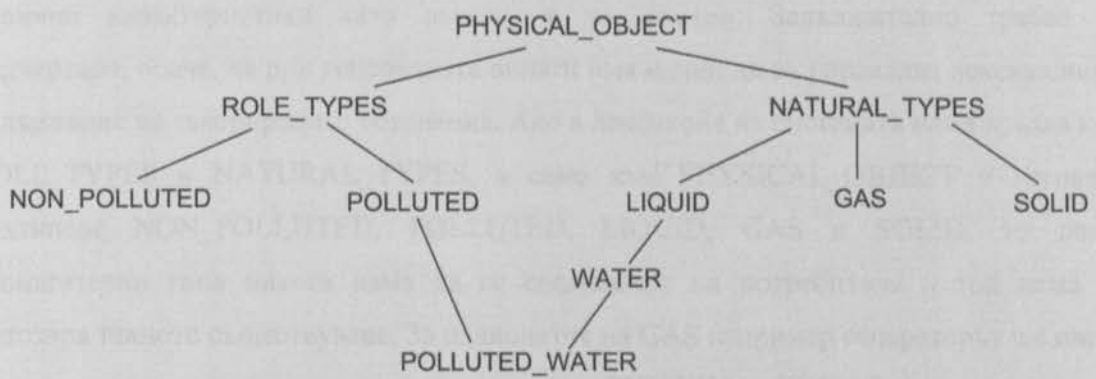
обаче е трудна за ръчно създаване и поддържане в една абстрактна и променяща се област като правото (или поне правото в България през 90-те години). По тази причина на един по-късен етап в DB-MAT беше избрана техническа предметна област, в която обектите са материални тела или физически величини.

Проектирането на йерархията е повлияно от предположението, че тя трябва да осигури генерация на типични обяснения за преводача. Много често в речниците се цитират понятията от непосредствената околност на разглежданото, например:

> 1546 **fixed-length record**, **F-mode record** || запись *ж* фиксированой длины || Aufzeichnung *f* mit fester Länge || enregistrement *m* à longueur fixe
> *Record, whose length is equal to the length of the other records, which are logically or physically related to it* (see also **variable-length record**) [ExDi78]

В този речник на хартиен носител виждаме, че при изброяването на преводните съответствия и обясненията за значението на едно понятие се прави явна препратка към неговото понятие-сестра. По този начин преводачът се ориентира в концептуалната структура на предметната област. Поради това в йерархията на типовете понятия в DB-MAT са описани концептуалните околности на важните термини и се осигурява генерация на обяснения за над-понятията, под-понятията и сродните понятия. В Таблица 3.3 са дадени множество такива примери, произведени от системата в отговор на въпроса `'Types of ...'` за някой термин. Това позволява да се видят термините и на двата езика, при смяна на заявката за език на генерираното обяснение.

По-долу ще опишем едно усъвършенстване на концептуалната йерархия с добавяне на явно наименование за класификационата перспектива, както и информация за вида класификация – *изчерпваща/неизчерпваща* и в *пресичащи се/непресичащи се* подтипове. При развитие на концептуалния модел в DB-MAT бяха добавени характеристики на класификационната релация *IS-A*. В [AnBo97a] и [AnBo97b] се предлага допълнително маркиране на класификациите на ролеви типове и на естествени типове. Например, физическите обекти могат да се класифицират на *твърди*, *течни* и *газообразни*. Обаче от гледна точка на пречистването, обектите са *замърсени* или *не*. Двете класификации често се смесват в една йерархия, както е показано на Фиг. 3.5. Допълнително въведените служебни понятия ROLE_TYPES и NATURAL_TYPES позволяват да се въведе характеристика на класификацията чрез релацията ISA_KIND. Клаузите за ISA_KIND означават:



isa_kind(physical_object, natural_types, natural).
isa_kind(physical_object, role_types, ecology).
char(natural_types, exclusive).

Фигура 3.5. Класификация от различни перспективи в пресичащи се подтипове

"*физическите обекти са класифицирани в естествени типове NATURAL_TYPES*" и "*физическите обекти от гледна точка на екологията са класифицирани в ролеви типове ROLE_TYPES*". Така се въвежда наименование на перспективата, за която е валидна класификацията – в този случай *екология*. Би могло да има и други класификационни перспективи, които се отнасят към PHYSICAL_OBJECT. Освен това за важни класификации се отбелязва дали подтиповете са непресичащи се или не. Например физическите обекти като NATURAL_TYPES са разделени на изключващите се подтипове LIQUID, GAS и SOLID и това е декларирано в клаузата char(natural_types, exclusive). Самите подтипове на NATURAL_TYPES и ROLE_TYPES се пресичат, но това е естествено при смесване на ролевите класификации в единна таксономия. От йерархията на типовете могат да се генерират обяснения от следния вид:

*'Физическите тела са твърди, течни **или** газообразни, а от гледна точка на екологията те могат да бъдат замърсени **и** незамърсени.'*

Тук съюзът *'или'* вербализира класификацията на взаимно-изключващи се подтипове, докато *'и'* вербализира класификацията с незададени характеристики за взаимно-изключване. Изразът *'от гледна точка на'* показва наличието на перспектива при класификация в роли. Тази информация е подсилена и от модалността *'могат да бъдат ... '*, която показва възможността обектите да имат различни характеристики.

При извличането на спомагателни типове като ROLE_TYPES и NATURAL_TYPES сме се опирали на опита в други големи онтологии като Komet/Penman Upper Model [BHR95], които съдържат във вид на сервизни (dummy) класификационни типове типични характеристики като `String` и `Non-string`. Задължително трябва да подчертаем, обаче, че при генерацията винаги има начин да се управлява лексикалното съдържание на генерираните обяснения. Ако в лексикона на системата няма връзка към ROLE_TYPES и NATURAL_TYPES, а само към PHYSICAL_OBJECT и неговите подтипове NON_POLLUTED, POLLUTED, LIQUID, GAS и SOLID, то двата спомагателни типа никога няма да се споменават на потребителя и той няма да подозира тяхното съществуване. За надпонятие на GAS например генераторът ще взема първият по-висок лексикализиран тип, в случая PHYSICAL_OBJECT, и ще произвежда обяснения като:

... газът е физически обект ...

Така типовете ROLE_TYPES и NATURAL_TYPES служат само за вътрешна организация на йерархията и представляват спомагателни понятия, към които могат да

се асоциират подходящи характеристики относно перспективата за класификация и вида на подтиповете (пресичащи се или взаимно-изключващи се).

Тук е редно да коментираме и обясненията в случай на повече от един надтип, които са нещо различно от коментираните по-горе класификационни перспективи. При зададен въпрос за `'Types of …'`, в ред случаи е добре да се вербализира наличието на повече от един надтип. За базата от знания на Фиг. 3.3 например би следвало да се произведе обяснението

... Като масло, маслените частици имат плътност; като частици, те имат размер ...

За съжаление някои от тези видове обяснения не бяха включени в генераторите на DB-MAT поради липса на ресурси за работа по лингвистичното осигуряване на системата.

3.2.2. Типове понятия с различна грануларност

Дискретната структура на концептуалните графи подпомага моделирането на терминологични единици с различна лексикална и концептуална грануларност. Базата от знания в DB-MAT се състои от:

- *опора* със зададени таксономии на типовете понятия и типовете концептуални релации,
- *дефиниции на типове* - специален вид графи, които задават необходимите и достатъчни условия за принадлежност на екземпляр към даден тип, или представят условия, верни за всички индивиди от този тип, и
- *отделни концептуални графи* (твърдения за предметната област), които са анализи на различни формули, кодиращи свойствата на обектите.

Вече дискутирахме една друга особеност на концептуалните графи – съществуване на контексти, нещо като «концептуални скоби», чрез които няколко понятия се групират заедно. Тази възможност да се обособяват концептуални единици с различна грануларност се оказва много полезна при конструиране на връзките между многоезичния терминологичен лексикон на системата и концептуалните единици в базата. Както вече споменахме, процесът на извличане на знанието е независим

Фигура 3.6. Поддържане на концептуални и лексикални единици с различна грануларност

от комуникативните цели (и от всякакви лингвистични съображения за строежа на текста) и по този начин може да се извлече частичен фрагмент от даден граф. Това налага задаване на връзки от всяка концептуална единица към думите в лексикона, понеже предварително не е ясно какъв фрагмент ще бъде подаден за изказване в схемата и кога. Тъй като дефиницията на типа се вербализира винаги като едно цяло, на Фиг. 3.6а е показано как връзките към лексиконите се реализират на ниво тип (в случая понятието Ölpartikel е наименовано с една съставна дума на немски и с номинална фраза от две думи на български). При извличането на факти чрез проекция от даден концептуален граф в базата, обаче, не е ясно какъв фрагмент ще бъде извлечен като релевантен на бъдещ въпрос и на какъв език ще се поиска обяснение. Към генератора може да бъде подаден целият контекст (наименован с една съставна немска дума) или неговата вътрешна част (в която и двете понятия си имат наименование от по една дума на български език). В случая, показан на Фиг. 3.6б, генераторът EGEN ще генерира обяснение както следва: при заявка за обяснение на немски, ще намери лексикалния указател към контекста и ще използва съставната немска дума Abwasser; при заявка за обяснение на български, няма да намери такъв указател, ще влезе вътре в контекста и ще изгенерира номиналната фраза 'отпадъчна вода' по правилата, налични в граматиката за конструиране на номинални фрази от прилагателно и съществително нарицателно. Така, благодарение на различните по грануларност концептуални единици в базата, има различни възможности за 'закачване' на указатели към лексикона на необходимите концептуални позиции и за създаване на сравнително унифициран подход при подбора на думите. Това решение е много удачно при многоезикова генерация и улеснява свободното извличане на фрагменти от знание, тъй като за всяка

единица има лексикално название на всички езици (което или е фиксирано, или се генерира в момента).

Тъй като указателите между лексиконите и базата от знания са изградени чрез посочване на идентификатори на два елемента, системата има нужда от уникални идентификатори както за единиците в лексикона, така и за конструкциите в базата от знания. През 1996 год. беше създадено работното място за дефиниране на концептуални графи CGLex [ADTB97], в което се поддържат системни идентификатори и вътрешни имена на всички концептуални релации, понятия, ситуации, контексти и графи-тела на дефиниции на типове. Средата е базирана на менюта и позволява добавяне на анотации-коментари на естествен език към контекстите, графите и дефинициите на типове, а анотациите осигуряват начин за бързо търсене в графите чрез ключови думи на естествен език. Работното място CGLex беше хармонизирано с по-рано създадени прологово представяне и среда за поддържане на лексиконите [PEK95, BoEu95, vHa95] и това осигури сравнително лесно ръчно управление на указателите между базата от знания и лексиконите на български и немски език. Във финалната си версия системата DB-MAT работи над около 100 концептуални графа и 120 термина, макар че таксономията съдържа над 200 понятия (част от тях са спомагателни и не се показват пред потребителя).

3.2.3. Конвенции при извличане на знанията от текстови източници

Тъй като извличаното в DB-MAT знание се подава към генератор с ограничени възможности за конструиране на изречния и свързан дискурс, още при проектирането на формàта на фактите в базата се наложи създаване на стандарти как да се декларират фактите от предметната област. В тази секция е дадено резюме на решенията за оформяне на концептуалните графи в декларативни представяния.

Структуриране на елементите в базата от знания

Грануларността на понятията съответства на думи винаги, когато е възможно. Всяка концептуална единица, която ще бъде разказана чрез съществително, прилагателно или наречие, се представя в базата от знания като понятие. Глаголите се появяват в текста като вербализация на понятия или концептуални релации. Предлозите се генерират от

концептуални релации; от релацията OF се появява и падежът *Genitive* на немски език, който вербализира притежание. Множествено число се поставя на броимите съществителни, когато в съответните понятия в базата от знания присъства референтът множество {*}. Съюзите И/ИЛИ се появяват при изброяване. Членуването става чрез правилата на съответната граматика, която определя и пунктуацията. В генерираните обяснения не се срещат междуметия и частици. Почти не се употребяват местоимения поради ограничените възможности за реализация на референтни изрази. Тъй като термините се кодират като понятия или ситуации, тяхната грануларност е фиксирана. Веднъж определена, тя влияе върху грануларността на останалите градивни елементи в базата от знания.

Конвенции при извличане на релации и реда на аргументите им

Посоката на стрелките на концептуалните релации е много важна, тъй като генераторът се опитва да постави първия аргумент на релацията в позицията на подлог в действителен залог и строи "utterance path" за разказване на графа по посока на стрелките на релацията. Ако за някои релации като THME, PTNT, CONTAIN се наложи позицията на подлога да се заеме от втория аргумент – например защото той е фокус – генераторът търси шаблон за вербализиране в страдателен залог, тъй като графът ще бъде обходен в посока обратна на стрелките. Грануларността на релациите е също много важна, понеже често релациите задават тематичните роли на глаголите и са опорни точки в граматичния шаблон за разказване на графа. Поради това в DM-MAT с течение на времето бяха изработени стандарти за кодиране на знания за целите на генерацията:

- Кодиране на събития като понятия:
  a) [CHASE] → (AGNT) → [CAT] (действие → агент-подлог),
  b) [CHASE] → (THeME) → [MOUSE]; [EAT] → (PTNT) → [PIE] (действие-преходен глагол → пряко допълнение),
  c) [GO] → (INST) → [BUS] (действие → непряко допълнение),
  d) [EAT] → (MANNER) → [FAST] (действие → поясняващо наречие),
  e) [GO] → (ALONG) → [PATH] → (TO) → [PLACE] (посока на движение).

  При a), b) и d), концептуалната релация не се вербализира. За c) и d) се задават граматични шаблони със съответни предлози;

- Кодиране на състояния:

a) [PLATE] → (PART-OF) → [SEPARATOR] (част → цяло),

b) [ENTITY] → (LOC) → [PLACE] (местоположение на обект),

c) [PHYSICAL-OBJECT] → (MADE-FROM) → [MATERIAL] (материал).

- Кодиране на атрибути и характеристики:

a) [BALL] → (ATTR) → [RED] (обект → атрибут),

b) [BALL] → (CHAR) → [COLOR:red] (обект → характеристика, чиято стойност е измежду елементите на предвартелно зададено множество).

- Кодиране на предложни фрази:

a) [CAT] → (ON) → [MAT],

b) [DISPERSION] → (OF) → [PARTICLE: {*}]

Фиксирането на стандарти за изразяване на концептуалните отношения между елементите в базата от знания прави възможно и формулирането на краен брой въпроси за инективна проекция, чрез които да се извличат подграфи както е показано в част 3.1.

3.2.4. Броимост-неброимост

Тъй като алгоритмите за обработка на декларативно-представено знание в изкуствения интелект оперират върху типове понятия и техни екземпляри, важно е да разграничим в таксономията броимите и неброими типове. Няма единно мнение дали 'броимостта' е езиково-зависима или е свързана с понятието на концептуално ниво. Забелязваме, обаче, че съответният естествен език играе роля в представите ни за броимост: на английски *news* (*новина-новини*) е неброимо, в единствено число, докато на български новината е абстрактно, но броимо съществително [Mol92] и в този смисъл може да се говори за една новина, т.е. за един екземпляр на понятието НОВИНА. В редица авторитетни източници, свързани с фундаменталните аспекти на представяне на знанията, свойството 'броимост' се разглежда като основно и задължително при концептуалното моделиране, например:

- в [Hay85] 'броимостта' е съществен и вътрешно-присъщ атрибут на физическите обекти;
- за Никола Гуарино 'броимостта' е категория в горното ниво на неговата онтология още от първите версии на нейното публикуване [Gua97];

- в известната part-whole таксономия [WCH97] са отделени два случая, свързани с разграничаване на броими и неброими типове: *portion-mass* (например *slice-pie*) и *stuff-object* (например *steel-car*);
- Сова предлага в [Sow84] и [Sow92] използването на измеряеми екземпляри, които съответстват на конкретни количества материал, за да се представят различни индивидуални количества от типове като ВОДА, СОЛ, ВРЕМЕ и т.н.

Ние обаче се интересуваме от концептуално моделиране в контекста на многоезиковата генерация и се обръщаме и към източниците, които третират понятията във връзка с естествения език. Роберт Дейл казва в [Dal88], че 'на обектите не е присъщо да са броими или неборими; те се разглеждат като такива от определена перспектива'. Например, в готварските реценти 'ориз' често е неброимо и тогава казваме 'две чаши ориз', докато специалистът по растителни болести би могъл да разглежда ориза като 'изброимо множество от индивидуални зрънца'. В [Hob95] се подчертава, че контекстът на употреба на думите в естествения език променя перспективата за класификация на понятията, например *пътят* е *линия* – когато планираме пътуване, *повърхнина* – когато шофираме по него, и *обемен* обект – когато пропадаме в дупка на пътя. Виждаме, че включването на естествения език в разглежданията веднага води до неопределеност, динамизъм на интерпретацията и изискване за гъвкава смяна на концептуалния модел в съответствие с контекста (т.е. принципните затруднения, обсъдени в първа глава).

При концептуалното моделиране в DB-MAT се отчита необходимостта от генериране на текстове с използване както на броими, така и на неброими съществителни. В [Ang98] е изказано предположението, че затворените светове се характеризират не само с ограничен брой понятия и релации, а и с ограничен и предварително-фиксиран набор от перспективи към всеки обект. Освен това в затворените светове има естествено-определена грануларност на понятията. Предлага се да се смесят броими и неброими типове в единна таксономия. Подходът за вербализация е: когато се говори за самия материал, тогава се използва неброимо съществително в единствено число; когато обаче фокусът е върху конкретни свойства като форма или очертание, се реферира към определени индивиди чрез броими съществителни и тогава е възможна употребата както на единствено, така и на множествено число. Стремежът е извличане на знанията по подходящ начин, с отчитане на указателите между понятията в базата от знания и елементите на системните лексикони.

На Фиг. 3.7 е показано как броими и неброими обекти се смесват в единна класификация. Типовете OIL_DROP и OIL_PARTICLE са подтипове съответно на броимите DROP и PARTICLE, а също и на OIL, което е ВЕЩЕСТВО. Имаме нужда от маркер, указващ как и къде се осъществява преходът между броими и неброими. Това е PACKAGER – ключова дума, която означава, че OIL_DROP и OIL_PARTICLE са „количества от веществото". Въведената на Фиг. 3.5 релация ISA_KIND показва, че PACKAGER е класификационна перспектива от вид 'роля' (подобно на PROFESSION) и не е неизменна характеристика на екземплярите на подтиповете, които удовлетворяват тази роля в определен момент. Ролята PACKAGER съчетава отношенията *portion-mass* и *stuff-object*, които са типични релации за част от цялото [WCH97].

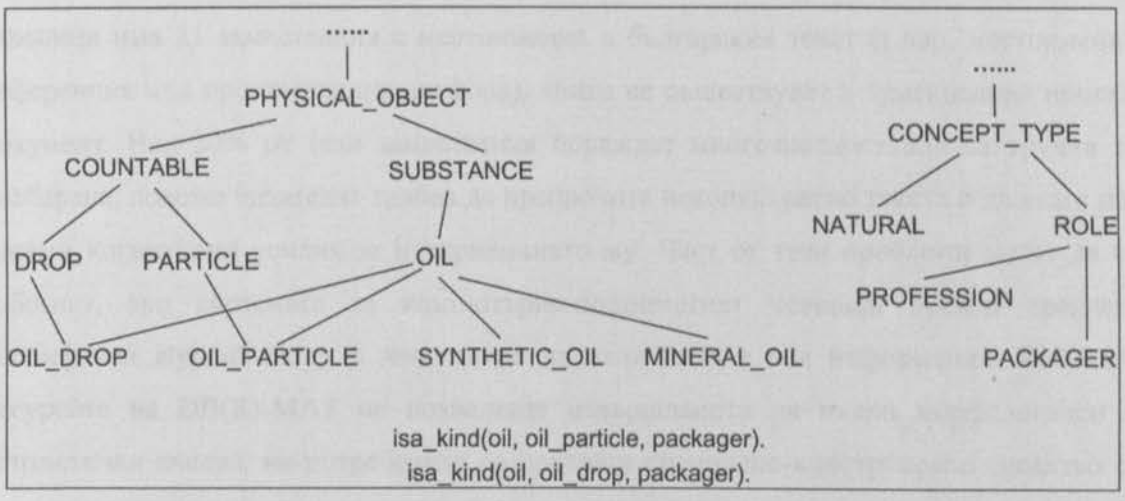
Тъй като OIL_DROP и OIL_PARTICLE са броими, генераторът може да произведе обяснението в единствено число

Всяка маслена капка има диаметър по-малък от 0,05 mm,

а също така – от гледна точка на OIL да генерира текста

Маслото е синтетично масло или минерално масло. Маслото се появява във вид на маслени капки и маслени частици. Като масло, маслените частици имат плътност и относително тегло.

При разработката на изложените тук идеи за усъвършенстване на таксономията авторът се е ръководел от убеждението, че йерархията трябва да съдържа всички надтипове, които потребителите могат да намерят в различни текстови източници. Но при съвместяване на много класификации в една база от знания, системата се нуждае от специални средства за съответно управление и обработка на съдържанието на базата.



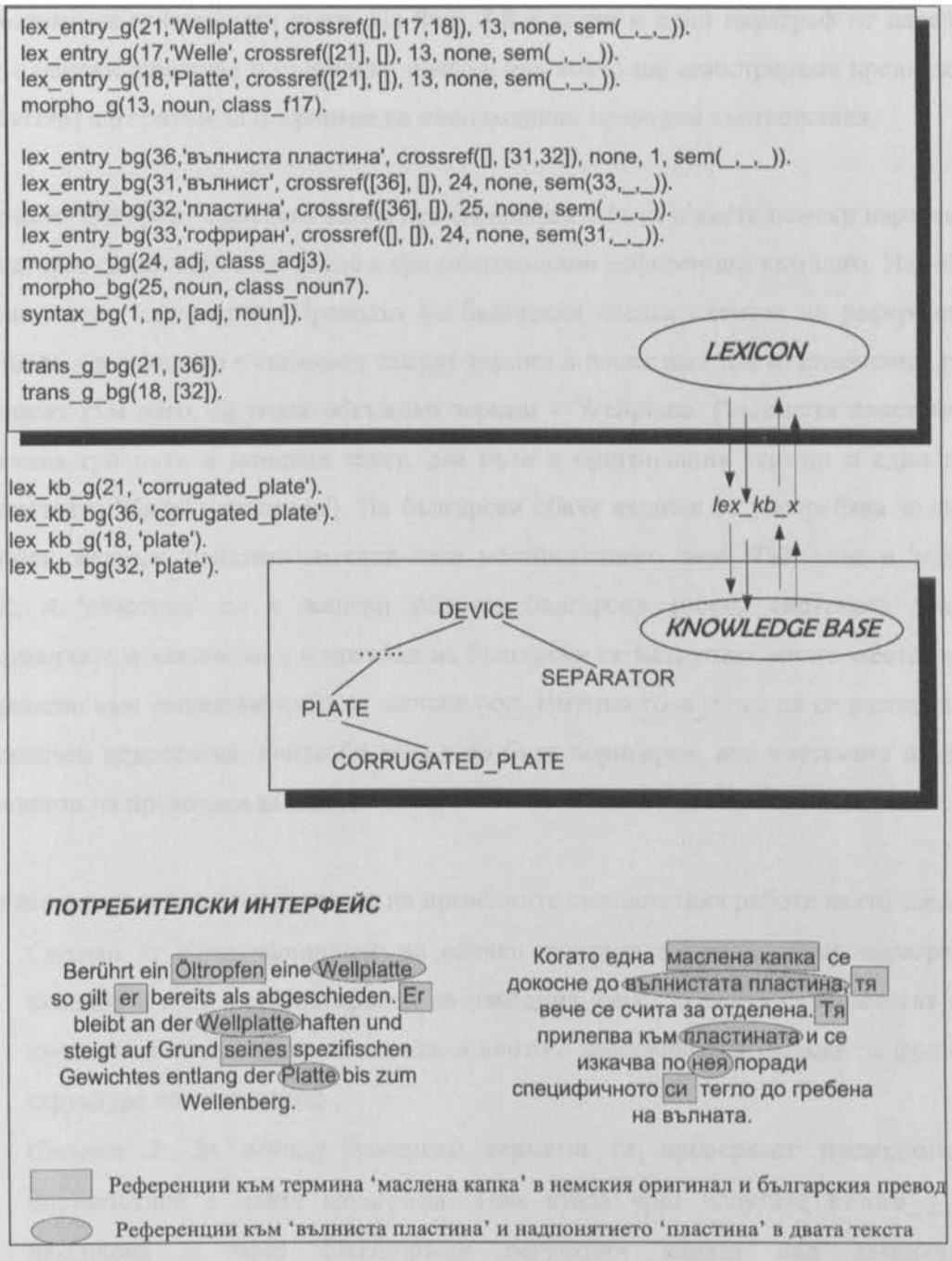
Фигура 3.7. Съчетаване на броими и неброими типове в единна таксономия

3.3. Елементи на разбиране на естествен език с използване на знания

В тази част са представени резултати в областта на семантичния анализ на естествения език, при което се откроява ролята на знанието за предметната област, декларирано явно в концептуалния ресурс на системата.

3.3.1. Подобряване на човешкия превод чрез следене на референцията

В рамките на проектите DB-MAT и DBR-MAT е разработено средство за следене на преводите на терминология и изпращане на предупреждение (*warning*) към потребителя за евентуално многозначен или нередактиран превод на основните понятия. Идеята за такава автоматична проверка възникна след изследване на стила на 80 страници немско-български превод в областта на пречистване на отпадни води. По принцип при превода професионалните преводачи предпочитат да следват лексикалната и синтактичната структура на входния текст винаги, когато е възможно – т.е. превеждат дума по дума, изречение по изречение и референция по референция. Отчасти те правят това, понеже не разбират добре значението на текста, особено в техническа област, а и така се съхранява максимално стила на оригинала и – не на последно място – се пести време за превод. По този начин преводачите често съхраняват евентуално съществуващите многозначности във входния текст, за да бъдат разрешени от човека-читател на другия език. В процеса на превод, обаче, поради смяна на родовете на съществителните при заместване с местоимения, могат да възникнат нежелани многозначности в изходния текст. Например в споменатия по-горе превод, в 80-те страници има 21 замествания с местоимения в българския текст (т.нар. местоименна референция или прономинална анафора), които не съществуват в оригиналния немски документ. Над 20% от тези замествания пораждат многозначност или са трудни за разбиране, понеже читателят трябва да препрочита неколкократно текста и да влага по-големи когнитивни усилия за възприемането му. Част от тези проблеми могат да се избегнат, ако системата за компютърно-подпомогнат човешки превод предлага специфичен style-checking с използване на наличната в нея информация. Тъй като ресурсите на DB(R)-MAT не позволяват извършването на пълен морфологичен и синтактичен анализ, на потребителя се предлага специално-конструирано средство за контролиране на превода на терминологията [AKvH98].



Фигура 3.8. Системни ресурси и проследяване на превода на термините в паралелни текстове

На Фиг. 3.8 са представени системните ресурси – лексикони, база от знания и предикати-указатели `lex_kb_g` и `lex_kb_bg` за съответствията между елементите на лексикона и базата от знания. По принцип в лексиконите не се съдържат всички думи, употребени в представителен корпус от текстове в предметната област, и по тази причина системата не разполага със запас от обща лексика извън термините и другите

необходими за генерацията думи. На Фиг. 3.8 е даден и един параграф от паралелен текст – немски оригинал и български превод, над който ще илюстрираме предложения в [AKvH98] алгоритъм за откриване на неподходящи преводни съответствия.

Понятието 'Öltropfen' ('маслена капка') се споменава 4 пъти в двете немски изречения – веднъж чрез самия термин и после с три местоименни референции към него. На немски терминът е от мъжки род. Преводът на български следва схемата на референция в оригинала – в началото е споменат самият термин и после има три местоимения, които се отнасят към него. Другият обсъждан термин - 'Wellplatte' ('вълниста пластина') се споменава три пъти в немския текст, два пъти с оригиналния термин и един път с надпонятието 'Platte' ('пластина'). На български обаче веднъж се употребява 'вълниста пластина', веднъж 'пластина' и след това местоимението 'нея'. Тъй като и 'маслена капка', и 'пластина' са в женски род на български (което системата знае от информацията в лексикона), в превода на български се натрупват много местоименни референции към съществителни от женски род. Именно това може да се разгледа като стилистичен недостатък, който би могъл да бъде коригиран, ако системата привлече вниманието на преводача към него.

Създаденият модул за проследяване на преводните съответствия работи както следва:

- *Стъпка 1*: Идентифициране на всички термини в оригиналния параграф на входния език и неговия превод на изходния език. За целта се използват едно-езичните лексикони на системата, в които е записана информация за фразовата структура на термините;
- *Стъпка 2*: За всички намерени термини се проверяват преводните им съответствия в двата параграфа. Това става чрез клаузите `trans_g_bg` в лексикона и чрез филтриращи регулярни изрази над лексикалните характеристики, които подпомагат откриването на фразовите термини;
- *Стъпка 3*: Ако има термин с неоткрит превод на стъпка 2, в изходния текст се търси парафраза на термина от оригиналния език, която може да бъде от два вида:
    - Термин за надпонятието (или надпонятие от таксономията, или съществително-опора на фразата от синтактичната информация `syntax_g` и `syntax_bg` за термин-фраза);

- Списък с изброяване на подпонятията - директни наследници от йерархията на термина с неоткрит превод в изходния език.

- *Стъпка 4*: Ако не се намери преводно съответствие чрез парафраза на стъпка 3, в изходния текст се търси местоимение, което реферира към търсения термин. Ако няма такова местоимение, модулът предполага наличие на *грешка.*

  Ако има такова местоимение, модулът търси в разглеждания параграф на изходния език термин, към който местоимението реферира. Ако няма такъв термин, се предполага наличие на грешка. Ако има повече от един обект, към който може да реферира местоимението, се извежда предупреждение за евентуална многозначност и проверка на стила *(warning).*

В скицирания алгоритъм са интегрирани най-прости техники за откриване на обекти, към които реферира местоимение – като намиране на съществително в съответния род и число в предишното изречение. В примера на Фиг. 3.8 ще бъде предположено, че местоимението 'нея' от второто изречение на български език има за потенциален референт както 'пластина' и 'вълниста пластина' (които се възприемат като едно и също понятие), така и 'маслена капка' от първото изречение. Поради това ще се изведе предупреждение.

Алгоритъмът използва максимално информацията, налична в лексикона и базата от знания, за да предостави допълнителна помощ на човека-преводач. Преводът от немски на български в техническа област предполага натрупване на повече фрази в българското изречение, чрез които се реализира превода на сложните немски съществителни; тези фрази могат да се преплитат в текста по нетривиален начин. Поради тези съображения предложеното средство за контролиране на дискурсните референти е една полезна услуга.

3.3.2. Извличане на концептуални структури от анализиран текст

В контекста на съществуваща база от знания в областта на финансите, авторът е изследвал подходи за обогатяване на базата с нови факти чрез анализ на специализиран текст и проверка на консистентността на извлечените концептуални структури [BDA01]. Както беше казано в обзорната първа глава, понастоящем е невъзможно да се

извлича формално представяне на смисъла на изказванията от произволен текст в голям мащаб. Дори при успешен синтактичен анализ на едно отделно изречение е трудно да се установи как точно да бъде конструирана логическата му форма. Една от причините е видът четене, който при човешката интерпретация се определя според контекста. Нека разгледаме следните изречения в областта на финансовите пазари, дадени на английски език:

```
(1.1) Newly issued securities are traded on primary market.
(1.2) All newly issued securities are traded on primary markets.
(1.3) Every newly issued security is traded on primary markets.
(1.4) A newly issued security is (to be) traded on primary market.
(1.5) Primary market operates with newly issued securities.
```

Тези изречения се състоят от едни и същи значещи думи и очевидно представят твърдения за всички екземпляри на понятието SECURITY. Такова е дори (1.4), което е изказано в единствено число, но поради т.нар. 'обобщено четене' се възприема като твърдение за всички екземпляри от типа. Примерът ни дава основание да заключим, че за успешното извличане на концептуални структури от текст се нуждаем от някакво 'нормиране' на езика, с цел драстично редуциране на многозначностите. Например, в единствено число ще се говори само за един обект, а в множествено – за много. Ако броят на обектите е определен, числото винаги се споменава явно. Ако се изказва твърдение за всички екземпляри на понятието, задължително това се прави с употреба на думата 'всички' и т.н. Така стигаме до понятието 'контролиран език' (controlled language), което възниква около 1930 год. и е доста широко разпространено. Обикновено контролираният език се свързва с идеята за задаване на технически спецификации по естествен за човека начин, който същевременно е еднозначно-преводим във формално вътрешно представяне с оглед компютърна обработка [ABMHS95]. Големи фирми като Ксерокс и Британ Аероспейс от десетилетия подготвят голямо количество от документацията си на специфични контролирани езици и така осигуряват машинния й превод. Съвременен пример за контролиран език е *Attempto Controlled English*, който е създаден с цел да се подпомогне натрупването на декларативно-представени знания и извършване на умозаключения над тях [FuSch02]. *Attempto* се използва и в европейската мрежа за върхови постижения REWERSE [REW08] и се разглежда като едно от перспективните направления за напредъка на семантичния интернет. Така че, една важна задача при проектирането и разработката на среди за извличане на знания от текст, е да се ограничи структурата на входния

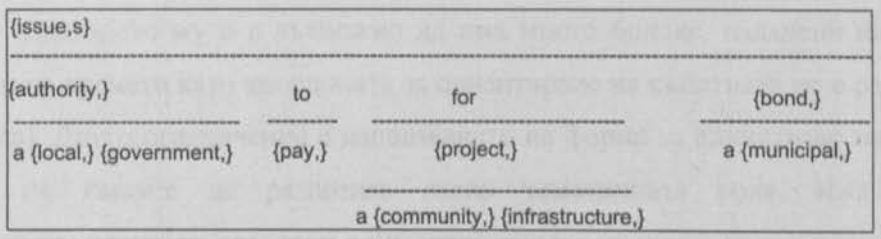
параграф по естествен за потребителя начин, като се предвиди възможност човекът-оператор да подобрява итеративно въвеждания текст, докато системата 'разбере' входа.

Предложеният в [BDA01] прототип CGExtract използва един важен компонент за формален анализ на входния английски текст – а именно, средата Parasite [Ram05] във версията ѝ от 1999 година. Системата Parasite извършва морфологичен и синтактичен анализ на входното изречение на английски език, произвежда логическо представяне и поставя в него явни имена на тематични роли (т.е. семантични отношения между глагола и другите главни фрази в изречението). Формирането на факти в базата от знания започва от резултата на Parasite, като CGExtract хармонизира понятията и релациите на създавания концептуален граф с вече съществуващите в базата графи.

Ще илюстрираме подхода чрез пример, в който са дадени резултатите на Parasite и след това е показана построената от CGExtract конструкция. Нека на входа на системата Parasite постъпва едно изречение на английски език (в което грижливо са нанесени всички неопределителни членове и се говори в единствено число, сегашно време съгласно обичайните правила за налагане на ограничения при контролираните езици):

```
A local government authority issues a municipal bond to pay for a
community infrastructure project.
```

Морфологичният анализ на думите в изречението се извършва с помощта на вградените в Parasite лексикони за английския език. След това се прави синтактичен разбор на входа; системата съдържа правила за английския синтактис, които покриват над 80% от конструкциите в езика. В разултат на синтактичния анализ се произвежда дърво на зависимостите във входното изречение, което е показано на Фиг. 3.9. За всяка дума или фраза в изречението е намерена една доминираща я дума и отношенията на подчинение са изобразени като прави линии над съответната дума/фраза. Например, `municipal` се подчинява на `bond` и го пояснява, а всички думи и фрази в изречението са подчинени на глагола `issue` в главното изречение.



Фигура 3.9. Синтактичен разбор на изречение във формализма на депендентната граматика

За всяко успешно анализирано изречение Parasite строи логически модел. Той е композиран от терми обозначаващи думите в изречението, съдържанието на правилата за синтактичен анализ и постулати за значенията на думите, в които са указани техните тематични роли. Служебните думи *lambda* и *theta*, които се използват в правилата и постулатите се появяват във финалната логическа форма, както и специфичните вложени части на предикатите на значенията, които са композирани чрез унификация. Таблица 3.4 съдържа модела на примерното изречение. В него се срещат идентификатори за събитието ISSUE и участващите обекти; променливи свързващи отделните терми – характеристики на обектите; тематични роли AGENT, FOR, OBJECT за семантичните отношения между глагола и допълненията му и релацията PURPOSE, за да укаже връзката между действието ISSUE и вложеното изречение *to pay for a community infrastructure project*.

```
issue(#354).
    theta(#354, agent, #356).
    theta(#354, for, #357).
    theta(#354, object, #358).
government(#356, lambda(A,local(A,lambda(B,authority(B))))).
infrastructure(#357,lambda(A,community(A,lambda(B,project(B))))).
municipal(#358,
        lambda(A,
                bond(A)
                    & purpose(A,
                            lambda(B,
                                    theta(#359,agent,B)
                                        & C.#359
                                            & pay(lambda(D,theta(#359,identity,D)))
                                                & lambda(D,theta(#359,identity,D))
                                                    is event)))).*
```

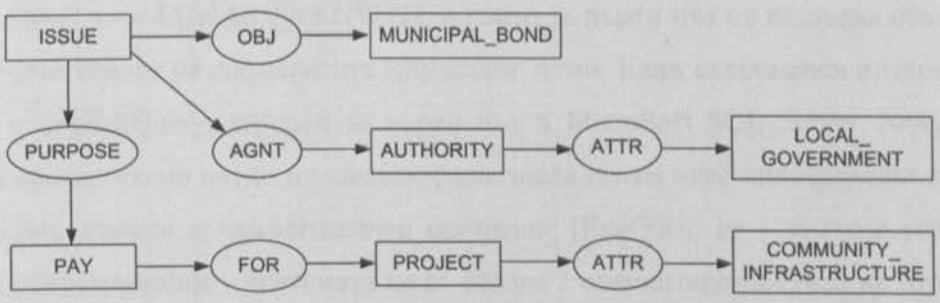
Таблица 3.4. Модел на изречение, произведен от системата Parasite

В модела са отразени някои типични техники на подхода за дефиниране на формална семантика на естествения език – например, действието ISSUE има свой екземпляр, който се случва в определен момент от време; така за всеки BOND има конкретен момент на издаването му и е възможно да има много бонове, издадени на различни дати (макар че времето като координата за ориентиране на събитията не е разгледано в този пример). Друго ограничение е използването на форма за единствено число, която позволява на Parasite да разпознае лесно тематичната роля. Има и някои конвенционални решения, свързани с интерпретацията на група от три последователни съществителни на английски език: прието е, че първите две са модификатори на

третото, и така първото се третира като модификатор на второто. Модулът CGExtract работи над дървото на синтактичните зависимости от Фиг. 3.9 и модела от Таблица 3.4. Преди всичко трябва да се реши кои думи от изречението съответстват на типовете понятия и релации в базата от знания. След преглеждане на имената на типовете в базата CGExtract намира понятия, чиито етикети съвпадат със споменатите във входното изречение обекти MUNICIPAL_BOND, AUTHORITY, PAY, PROJECT, LOCAL_GOVERNMENT и COMMUNITY_INFRASTRUCTURE. Връзките на подчинение от дървото на Фиг. 3.9

LOCAL_GOVERNMENT → AUTHORITY и
COMMUNITY_INFRASTRUCTURE → PROJECT

се интерпретират като концептуалната релация ATTRibute. С помощта на тематичните роли AGENT, OBJECT, FOR и PURPOSE от модела, CGExtract генерира показания на Фиг. 3.10 концептуален граф. При генерацията се пропускат неинстанцииранитe терми в модела, които са дошли от композицията на логическите форми на заложените в системата постулати на значенията. Генерираният граф се начертава пред инженера на знанията за проверка с помощта на графичната среда CGWorld [DST01]. Този пример показва, че дори когато автоматичното извличане на знанията се гради върху една от най-солидните прототипни среди за формален семантичен анализ от гледна точка на компютърната лингвистика – каквато е Parasite, има много нетривиални въпроси за решаване при прехода към декларативно концептуално представяне. Проблемите са свързани главно с композицията на концептуалните единици в базата от знания (а те трябва да се породят от думите в текста), тяхното наименоване във възприетото ниво на концептуална грануларност и открояването на концептуалните релации (които често са неявно-представени в езика или се подразбират в качеството им на връзки между главните фрази в изречението).



Фигура 3.10. Знание за предметната област, извлечено от изречение на естествен език след дълбок синтактичен и семантичен анализ

Освен извличане на твърдения от входен параграф на контролиран английски език, CGExtract извършва следното:

- Обработва заявки за добавяне на нови типове към йерархията на понятията, от подходящо формулирани изречения като например *'A subsidy is a financial institution'*;
- На основата на разпознатите от Parasite референции в две съседни изречения, строи по-сложни графи от кратки параграфи. Например, при подаване на две входни изречения

  ```
  A local government authority issues a municipal bond to pay for
  a community infrastructure project. The interest of a municipal
                        bond is income-tax free.
  ```

  споменаването на *'a municipal bond'* във второто изречение ще се разпознае като референция към екземпляра на това понятие в първото изречение, поради което към графа на Фиг. 3.10 ще бъдат добавени още факти:

  [MUNICIPAL_BOND] ← (OF) ← [INTEREST] → (ATTR) → [INCOME_TAX_FREE]
- Поддържа интерфейс за преглеждане на наличните концептуални структури от инженера на знанията.

3.3.3. Обработка на отрицание в контекста на таксономия на типовете

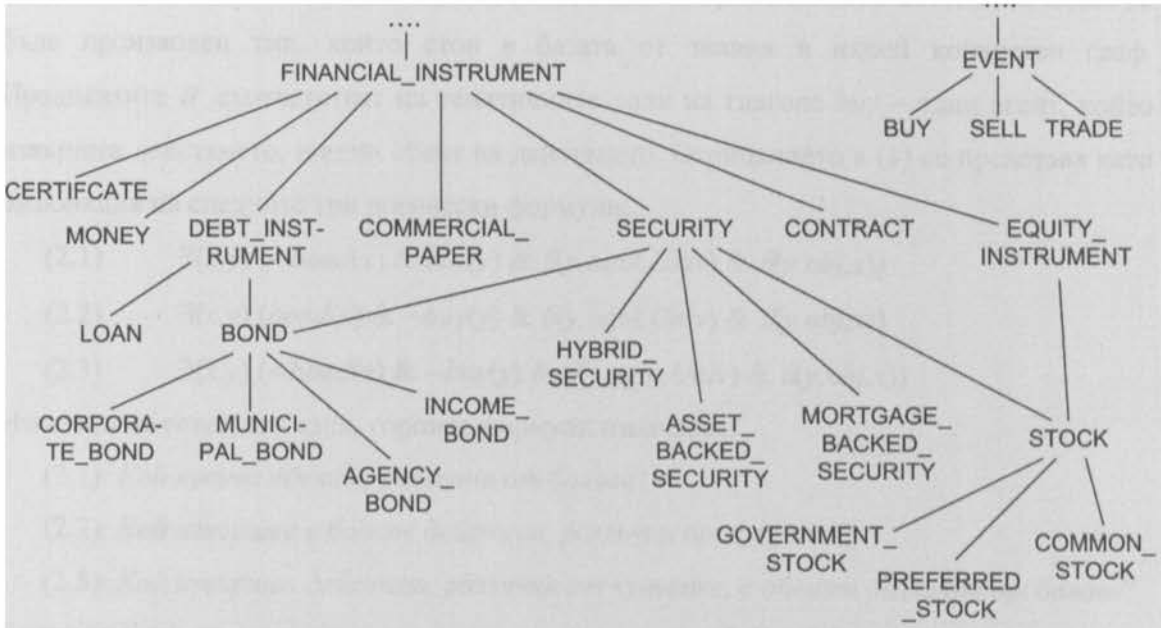
В изкуствения интелект най-често се използват декларативно-представени твърдения с положително знание. Компютърната лингвистика също не е напреднала особено в обработката на отрицанието и думите-маркери за квантификация – *всички*, *много*, *няколко* и т.н. Интересът към интерпретацията на английските *each* и *every* и на въпросителните местоимения възниква с развитието на естествено-езикови интерфейси към бази данни като LUNAR [WKNW72], в който за първи път се изследва обхватът и относителната тежест на споменатите 'служебни' думи. Един съвременен интерфейс от този вид е EnglishQuery, вграден за първи път в MicroSoft SQL Server 2000, който обработва множествено число и успешно разпознава голям клас положителни въпроси с относително сложен и немногозначен синтаксис [EngQ00]; но – както е установил авторът експериментално - системата не се справя с елементарни случаи на отрицание и не е особено интелигентна в обработката на квантори. По разбираеми причини, малко системи за анализ и разбиране на естествен език се насочват към третиране на отрицанието и кванторите, и това са предимно научно-изследователски прототипи.

Автоматичната обработка на отрицанието на естествен език е трудна поради поне две причини:

- Системата трябва да определи коя е отречената фраза на изречението, т.е. какъв е обхватът на отрицанието, да реши къде се поставят скобите в логическото представяне на изречението. Този обхват обаче често се преплита с обхвата на кванторите (напр. *'ние не видяхме всяко дете'*) и др.под.;
- Отрицанието в едно изречение означава отрицание на събитие или състояние, обект, характеристика, информация за местоположение, време и начин, контекст и т.н. Семантичният анализатор трябва да интерпретира отрицанието, за което са му необходими детайлни знания за света на дискурса и нетривиални техники за извод. Създаването на последните също представлява немаловажен проблем.

В [BSA02] е предложен експериментален модул, който обработва отрицанието чрез заместване на отречения тип с неговите сродни типове (братя или сестри) в затворения свят на концептуалната йерархия. По този начин може да се отговаря на въпроси, поставени пред системата на ограничен английски език. Отговорът се оформя също на английски в прости изречения с помощта на граматически правила за наредба и съгласуване на главните фрази в изречението. Демонстрационният прототип работи в предметната област на финансовите пазари и е част от среда за разглеждане на съществуваща база от знания чрез въпроси и отговори. Така на практика инеженерът на знанията и експертът в предметната област получават интерфейс за проверка на съдържанието на фактите в концептуалния ресурс. За съжаление, при наблюдения на създадените прототипи за обработка на отрицание се вижда, че броят на семантичните интерпретации на изречение с отрицание нараства факториално спрямо броя на участващите в него понятия [Poe00]. Това важи и за разглеждания от нас подход.

Ще опишем накратко предложения в [BSA02] модул с фокус върху дизайна на базата от знания и обработката на концептуалните структури. Фиг. 3.11 показва част от йерархията на типовете понятия, която служи за контекст при интерпретацията на отрицанието в затворения свят на ограничената предметна област. Дадени са и шест концептуални графа, от които ще се извлича отговор на въпрос чрез инективна проекция.



концептуален граф 1:
[PENSION_FUND] ← (AGNT) ← [BUY] → (OBJ) → [GOVERNMENT_STOCK: {*}]
→ (LOC) → [PRIMARY_MARKET] .

концептуален граф 2:
[DEMANDER] ← (AGNT) ← [SELL] → (OBJ) → [BOND: {*}] → (LOC) → [PRIMARY_MARKET] .

концептуален граф 3:
[COMPANY] ← (AGNT) ← [TRADE] →(OBJ)→ [CORPORATE_BOND: {*}] →(CHAR) → [NEWLY_ISSUED] .

концептуален граф 4:
[BROCKER]←(AGNT)←[SELL]→(OBJ)→[STOCK: {*}]→(CHAR)→[MATURITY]→(ATTR)→[SHORT_TERM] .

концептуален граф 5:
[BROCKER] ← (AGNT) ← [SELL] → (OBJ) → [HYBRID_SECURITY: {*}]
→ (LOC) → [STOCK_EXCHANGE: NYSE] .

концептуален граф 6:
[STOCKHOLDER] ← (AGNT) ← [TRADE] → (OBJ) → [HYBRID_SECURITY: {*}]
→ (LOC) → [STOCK_EXCHANGE] .

Фигура 3.11. Примерна база от знания в предметната област на финансовите пазари

Въпросът постъпва в системата като изречение:

Who does not buy bonds?

След морфологичен и синтактичен анализ, при които се разпознава наличие на отрицание, въпросът се превежда до логическа форма с квантор за всеобщност, построена по аналогия с въпросите в базите данни: `'Select all ... '`

(1) $\neg (\forall (x,y) (bond(x) \& buy(y) \& \theta(y,agnt,Univ) \& \theta(y,obj,x)))$

Обектът, за който се отнася въпроса, е маркиран с променливата *Univ*; това може да бъде произволен тип, който стои в базата от знания в някой конкретен граф. Предикатите θ съответстват на тематичните роли на глагола *buy* – един агент, който извършва действието, и един обект на действието. Отрицанието в (1) се представя като дизюнкция на следните три логически формули:

(2.1) $\exists(x,y)\ (\neg bond(x)\ \&\ buy(y)\ \&\ \theta(y,agnt,Univ)\ \&\ \theta(y,obj,x))$

(2.2) $\exists(x,y)\ (bond(x)\ \&\ \neg buy(y)\ \&\ \theta(y,agnt,Univ)\ \&\ \theta(y,obj,x))$

(2.3) $\exists(x,y)\ (\neg bond(x)\ \&\ \neg buy(y)\ \&\ \theta(y,agnt,Univ)\ \&\ \theta(y,obj,x))$

Изказани на естествен език, горните формули означават:

(2.1): *Кой купува обекти, различни от бонове*?

(2.2): *Кой извършва с бонове действия, различни от купуване*?

(2.3): *Кой извършва действия, различни от купуване, с обекти различни от бонове*?

Системата ще намери отговори и на трите въпроса (2.1), (2.2) и (2.3) в рамките на наличното знание, чрез извършване на инективна проекция върху графите, и ще ги разкаже на потребителя.

Най-просто е да покажем семантичната интерпретация на отрицанието в (2.2). Сродните понятия на BUY от йерархията са SELL и TRADE и те се третират като отрицания на BUY:

(2.2.1) [UNIV] ← (AGNT) ← [SELL] → (OBJ) → [BOND]

(2.2.2) [UNIV] ← (AGNT) ← [TRADE] → (OBJ) → [BOND]

Тези два графа ще бъдат използвани като въпроси за инективна проекция в базата от знания и намерените факти ще бъдат включени в отговора на (1). Тук не се занимаваме с въпроса, че значението на TRADE покрива отчасти значението на BUY; след като двата типа са декларирани като различни, те ще бъдат обработвани по този начин в процедурите за изводи.

С цел опростяване на записа при описание на отрицанията в (2.1) и (2.3), ще въведем множеството *compliment*(c), където c е понятие в базата от знания, а *compliment*(c) са всички понятия, различни от c, които са сродни на c поради наличие на общ надтип, но не са наследници на c. Ще използваме множествата:

- *parent*(c) = { $x \mid x \neq c$, $c<x$ и не съществува тип y, $y \neq c$ и $y \neq x$, така че $c<y<x$ };
- *sibling*(c) = { $x \mid (\exists y)\ (y = parent(c) \cap parent(x))$ };

- $son(c) = \{ x \mid (\exists k)\ (\exists y_1, y_2, \ldots, y_k)\ (y_i \neq c$ и $y_i \neq x$ за $1 \leq i \leq k$, $y_i \neq y_j$ за $1 \leq i,j \leq k$ и $i \neq j$, и такива, че $y_1 \in parent(x), y_2 \in parent(y_1), y_3 \in parent(y_2), \ldots, c \in parent(y_k))\}$.

Тогава дефинираме:

$$compliment(c) = \{ sibling(c) \cup \{ x \mid (\exists y)\ (x \in son(y),$$ където $y \in sibling(c))\ \} \setminus son(c),$$

където '\' е знакът за теоретико-множествено изваждане.

Например, за BOND от Фиг. 3.11 имаме:

- *parent*(BOND) = { DEBT_INSTRUMENT, SECURITY };
- *sibling* (BOND) = { LOAN, HYBRID_SECURITY, ASSET_BACKED_SECURITY, MORTGAGE_BACKED_SECURITY, STOCK };
- *son*(BOND) = {CORPORATE_BOND, MINUCIPAL_BOND, AGENCY_BOND, INCOME_BOND};
- *compliment*(BOND) = { LOAN, HYBRID_SECURITY, ASSET_BACKED_SECURITY, MORTGAGE_BACKED_SECURITY, STOCK, GOVERNMENT_STOCK, PREFERRED_STOCK, COMMON_STOCK }

След въвеждането на множеството *compliment* за всяко понятие, можем да запишем (2.1) като

(2.1.1) [UNIV] ← (AGNT) ← [BUY] → (OBJ) → [UNIV: *compliment*(BOND)],

а (2.3) като

(2.3.1) [UNIV] ← (AGNT) ← [SELL] → (OBJ) → [UNIV: *compliment*(BOND)] и

(2.3.2) [UNIV] ← (AGNT) ← [TRADE] → (OBJ) → [UNIV: *compliment*(BOND)].

Тук [UNIV: *compliment*(BOND)] означава кое да е понятие от множеството *compliment*(BOND), което има 8 елемента. Зад всеки от горните графи стоят по 8 въпроса за инективна проекция.

Въпросите (2.1.1) се проектират върху базата и непразна проекция има само върху граф 1. В нея UNIV се специализира до PENSION_FUND, а [UNIV: *compliment*(BOND)] – до GOVERNMENT_STOCK. От специализацията ще се генерира отговор на въпроса (2.1): *Кой купува обекти, различни от бонове*?

Въпросите (2.2.1) и (2.2.2) имат непразна проекция съответно върху граф 2 и граф 3. При тези проекции се извличат съответно специализациите:

[DEMANDER] ← (AGNT) ← [SELL] → (OBJ) → [BOND]

[COMPANY] ← (AGNT) ← [TRADE] → (OBJ) → [CORPORATE_BOND]

Тези графи са отговор на въпроса (2.2): *Кой извършва с бонове действия, различни от купуване*?

Въпросите (2.3.1) и (2.3.2) водят до намиране на отговор на (2.3): *Кой извършва действия, различни от купуване, с обекти различни от бонове*? Те имат непразна проекция върху графи 4, 5 и 6 и извличат специализациите:

[BROCKER] ← (AGNT) ← [SELL] → (OBJ) → [STOCK]

[BROCKER] ← (AGNT) ← [SELL] → (OBJ) → [HYBRID_SECURITY]

[STOCKHOLDER] ← (AGNT) ← [TRADE] → (OBJ) → [HYBRID_SECURITY]

След оформяне на отговор на контролиран английски език, системата извежда следния отговор:

```
Pension funds buy government stocks.
Demanders sell bonds.
Companies trade corporate bonds.
Brokers sell stocks and hybrid securities.
Stockholders trade hybrid securities.
```

Естествено описаният тук прототип може да се използва и за вербализация на прости факти при отговор на положителни въпроси. Този софтуерен модул е използван при извличане на базата от знания в проекта Ларфласт, за който ще стане дума в следващата глава.

ГЛАВА 4:

Използване на концептуални структури в интелигентни среди за обучение

Тук са представени резултатите на автора при разработка на концептуалните ресурси на средата за обучение STyLE (Scientific Terminology Learning Environment), която беше създадена в рамките на проекта ЛАРФЛАСТ (1999-2001). Приносите на автора са:

- Уточняване на принципите за концептуално моделиране на терминологичната колекция и по-специално формалното дефиниране на типове-роли,
- Проектиране на архитектурата на системата STyLE, базирана върху знания и
- Създаване на база от знания във вид на концептуални графи в областта на финансите.

Тъй като авторът беше научен координатор на проекта[16], а колективът от България осъществи интеграцията на средата и оценяването й с реални потребители, вижданията на автора за ролята на знанието в една интелигентна система за обучение оказаха съществено влияние при проектирането на компонентите и тяхната интеграция. STyLE е разпределена среда в интернет и подпомага автономни занимания на студенти при изучаване на финансова терминология. Студентите изучават едновременно както предметната област - икономика, така и английски (чужд или втори) език. Термините се разглеждат и като езикови единици, и като етикети на понятия в базата от знания.

4.1. Архитектура на STyLE

Ларфласт е научно-технологичен проект и изследва начините за интеграция на:

- ***техники за обучение, базирани върху знания*** – които предполагат наличие на декларативно описание на предметната област и използване на понятията:
  - в метаданните на учебните материали,
  - в модела на студента,
  - като основа за адаптивността на средата, и

[16] Проектът *LARFLAST* (*Learning Foreign Language Scientific Terminology*) бе финансиран от Европейската комисия по програмата Copernicus'98 със седем партньора: Университет в Лийдс, Великобритания; Университет на Манчестър, Великобритания; LIRM – Монпелие, Франция; Център по изкуствен интелект на Румънската академия на науките, Букурещ; Държавен университет на Симферопол, Украйна; ФМИ, Софийски Университет "Св. Кл. Охридски" и Виртех ООД България. Административен координатор на проекта беше проф. Джон Селф от Университета в Лийдс.

- ***съвременни езикови технологии*** – които осигуряват възможности за:
  - въвеждане на потребителски отговори на свободен английски език,
  - динамично обновяване на архива с подходящи четива, за да се показват при нужда на обучаемия с цел преподаване на допълнителни знания, и
  - генериране на обяснения на контролиран английски език с цел обяснение на съдържанието на езиковите и концептуални ресурси.

Интегрираният прототип STyLE е разпределена среда върху четири сървъра в интернет, които обменят помежду си следните данни:

- персонален код на текущия потребител-студент и списък от клаузи с резултатите му при изпълнение на тестове (т. нар. модел на обучаемия - *learner model*), и
- идентификатор на понятието/термина, за който студентът допуска грешки и евентуално има нужда от допълнителна информация.

Така концептуалният модел на предметната област и термините в лексикона се превръщат в информационен скелет, осигуряващ възможност за хармонизация на работата на отделните компоненти на средата. Подобно на проекта DB-MAT, връзките между понятията като концептуални елементи и термините като лексикални единици осигуряват кохерентността на системните ресурси и съвместната им обработка.

Ще скицираме архитектурата и компонентите на STyLE, които са представени в [BKNA00] и [ABKTNS02] в две различни фази на разработката. Фиг. 4.1 показва компонентите и ресурсите във финалната версия на средата [ABKTNS02]. Главните компоненти на системата са следните:

1. `Комуникационен модул`;

2. `Модул за диагностика` – разбира и оценява отговорите на обучаемия при изпълнение на два вида тестове: *(i)* с фиксиран отговор, който се избира от меню, и *(ii)* с отговор, който се въвежда от клавиатурата на свободен английски език. Проверката на тестовете от вид *(ii)* се реализира чрез средата `Parasite` (която осигурява лексикален, морфологичен, синтактичен и семантичен анализ на отговора) и модула за логически доказателства `STyLE-Parasite` (той следи дали отговорът е *между* най-специфичния и най-общия очакван отговор, които са дефинирани предварително като логически формули от експерт по формална лингвистична семантика). Правилният отговор трябва да съдържа необходимите думи в нужната степен на общност;

3. Педагогически (планиращ) агент – решава какво да се направи на следващата стъпка и адаптира системата към нуждите и предпочитанията на студента;

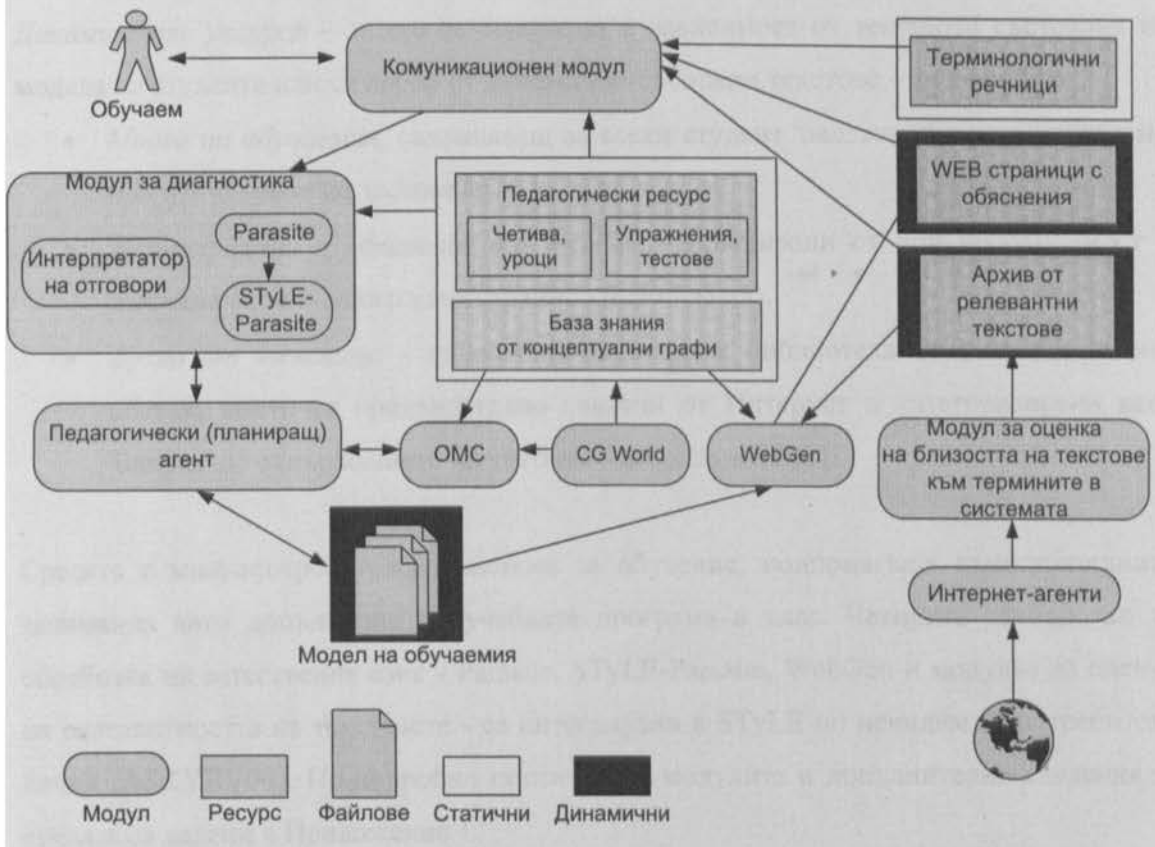
4. Отворен модел на студента (ОМС) – модул за диалог между STyLE и студента при откриване на грешки в усвоеното знание за предметната област;

5. Генератор WebGen - модул за генерация на кратко обяснение (до един екран) с дефиниции и употреби на термините, които студентът не е заучил добре;

6. Интернет-агенти – самостоятелни компоненти, които в режим off-line търсят по мрежата текстове с подходящо съдържание;

7. Модул за оценка на близостта на текстове към термините на STyLE – който изчислява коефициент на релевантност за намерените в интернет текстове и създава динамичен архив от четива за показване пред студента;

8. Среда за графично представяне на концептуални графи CGWorld –която позволява създаване и поддържане на системната база и подпомага изобразяването на графи пред студента.



Фигура 4.1. Архитектура на интелигентната среда за обучение STyLE

Средата съдържа *статични* и *динамични* ресурси. Статичните остават неизменни при работата на системата, докато динамичните се влияят от поведението на студента или се обновяват с течение на времето. *Статичните ресурси*, показани на Фиг.4.1, са:

- *Педагогически ресурс от учебни обекти* - четива-уроци и упражнения-тестове;
- *База знания от концептуални графи* - декларативно-представени факти за предметната област, които формират концептуалния ресурс на системата. Елементите на базата от знания са свързани с речниците от термини чрез специална анотация. Етикетите на понятията се използват и като метаданни в анотацията на учебните обекти от педагогическия ресурс. Например предварително е зададено кое понятие от базата и кое негово свойство се тестват от съответното упражнение (и по този начин – при грешен отговор - системата може да включи в модела на студента информация, че *'студентът X не знае понятието Y'*). Концептуалният модел ще бъде разгледан по-долу;
- *Терминологични речници* с граматична информация и текстови обяснения за значението на термините.

Динамичните ресурси – които се генерират в зависимост от текущото състояние на модела на студента или са набор от динамично-събирани текстове – са следните:

- *Модел на обучаемия*, съхраняващ за всеки студент 'бележник' с резултатите му при изпълнение на тестовете;
- *Web-страници с обяснения от WebGen* - генерирани от при необходимост с дължина от най-много една страница (един екран);
- *Архив от текстове* - динамично-обновявана библиотека от текстове-учебни обекти, които са предварително свалени от Интернет и категоризирани като 'близки' до съдържанието на учебния материал в STyLE.

Средата е многопотребителска система за обучение, подпомагаща самостоятелните занимания като допълнение на учебната програма в клас. Четирите технологии за обработка на естествения език - Parasite, STyLE-Parasite, WebGen и модулът за оценка на релевантността на текстовете - са интегрирани в STyLE по невидим за потребителя начин [ASKYBV04]. По-подробно описание на модулите и допълнителни сведения за средата са дадени в Приложение 1.

4.2. Концептуално моделиране на естествени типове и роли в STyLE

При проектирането на онтологията в областта на финансите бяха използвани и усъвършенствани някои предишни решения на автора, залегнали в проекта DB-MAT. Прецизирана е дефиницията на типове-роли, с цел отчитане на съвременните тенденции при деклариране на знанията в определена предметна област, включително при проектиране на класове в обектно-ориентираното програмиране.

Известно е, че класическите онтологични примитиви са понятия/класове и отношения/релации между тях [Pol08]. През 80-те години на миналия век, при възникването на проблематиката за представяне на знанията в изкуствения интелект, вниманието се насочва към моделиране на статично знание за света. Първоначално интересът е фокусиран върху т. нар. естествени типове. Според [Sow84], класификациите в естествени подтипове са основани върху неизменни твърди свойства, така че естественият тип на даден обект не може да се променя, докато обектът съществува (например, PERSON е естествен подтип на ANIMATE). Гуарино посочва, че неизменността на основните характеристики е само едно от важните свойства на естествените подтипове, които напомнят субстанциите на Аристотел поради тяхната цялостност и независимост [Gua92]. Например, един обект принадлежи на класа PERSON независимо от другите обекти, тъй като носи в себе си всички свойства, необходими и достатъчни да бъде разпознат като екземпляр на този клас. Освен естествените типове, обаче, има и друг вид типове-роли, които отразяват динамичното поведение на екземплярите в предметната област - например, *човек* може да бъде *дете*, *студент*, *служител*, *лекар*, *родител* и много други неща в различни периоди от живота си [Sow84]. Индивидите, които играят дадена роля, са свързани с други индивиди – например човек е *родител* само ако има дете, или *служител* само ако е назначен на работа. Така че ролите са свързани с контексти и характеристики, които указват какви са свойствата на екземпляра, изпълняващ дадената роля [Gua92]. И двата класически източника за концептуално моделиране - [Sow84] и [Gua92] – не дават отговор на въпроса дали и как да се смесват естествени и ролеви типове в концептуалната йерархия.

В последните години интерес към ролите възниква и в областта на обектно-ориентираното програмиране и многоагентните системи. Статията [Ste00] посочва, че

роли са разглеждани още в мрежовия модел на данните през 60-те години на 20-ти век, преди релационният модел да се наложи като по-прост и универсален. Фактът, че един обект може да играе много различни роли, означава, че ролите също класифицират екземплярите – например, хората могат да се класифицират като *студенти-в-настоящия-момент* и такива, които не изпълняват тази роля. Но класификациите чрез роли са многопрофилни, динамични и се променят с времето. В друга статия Фридрих Стайман обсъжда дуалността в поведението на ролите [Ste05]. От една страна, те изглеждат като подтипове (например, *родител* е подтип на *човек*) и тогава многото роли могат да се моделират като пресичащи се подтипове. Така екземплярите, които изпълняват ролята, могат да мигрират свободно и динамично между типовете. От друга страна, ролите обединяват екземпляри със стабилни характеристики на стереотипно поведение – например *клиентът* е *лице* или *организация*, а *родителят* е *човек* или *животно* и т.н., поради което ролите могат да се разглеждат и като надтипове. Следователно ролите пораждат специфични йерархии, които допълват класификационната таксономия в естествени типове.

В тази глава е описано едно приложно решение на автора за концептуално моделиране на роли-термини в STyLE, което подпомага изучаването на чуждоезикова терминология. По принцип е трудно да се мисли за универсално решение, тъй като опитът на WordNet показва, че значенията на думите формират мрежа, а не йерархия [WNet]. Напоследък във философията също се разглеждат подходи за динамично описание на обектите и процесите [Pet08]. Успешното моделиране на роли е извънредно важно, тъй като би позволило софтуерните системи да се специфицират в термините на агенти, които изпълняват главните услуги за потребителя, например агенти в система за обучение [SGPO'D08]. Но засега в литературата срещаме отделни модели на роли в конкретни предметни области и то без поддръжка на динамично поведение на екземплярите.

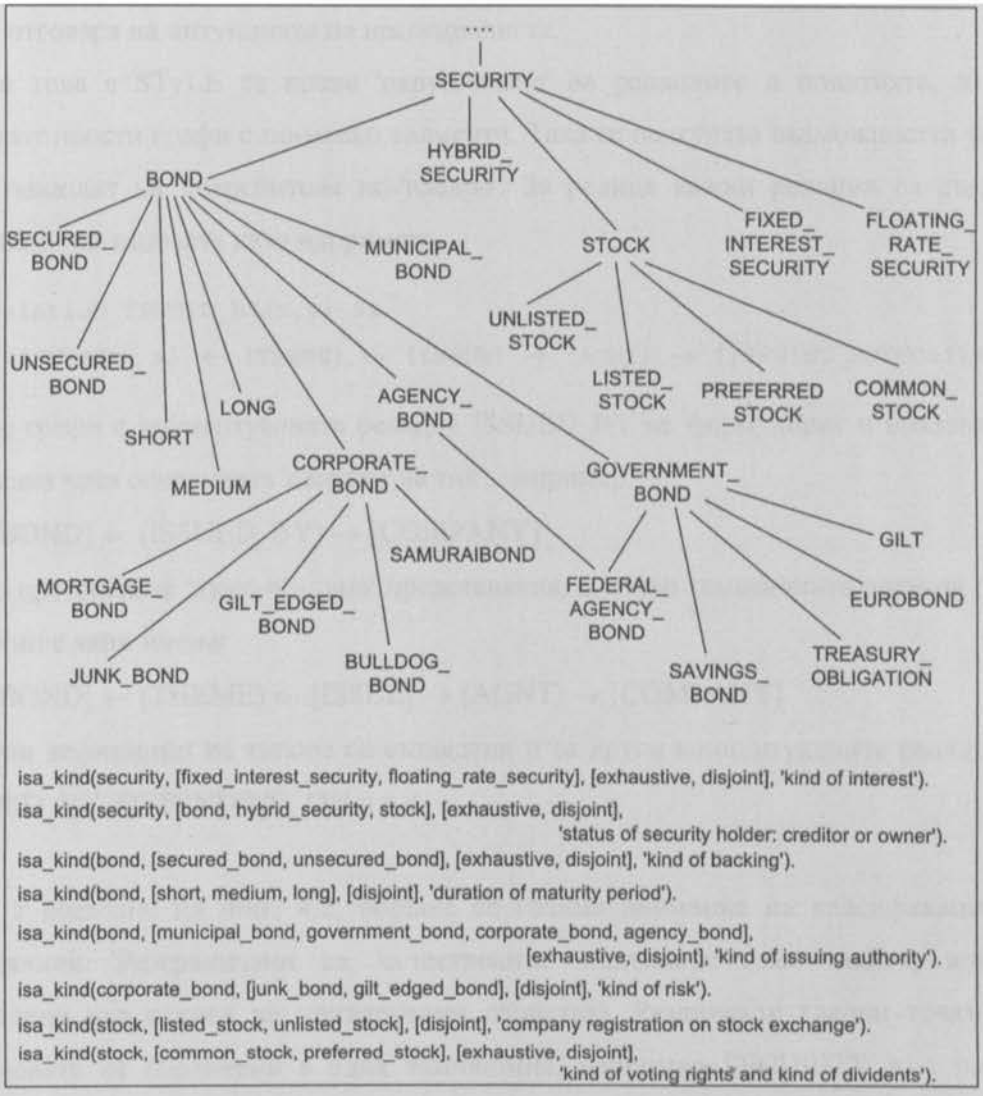
4.2.1. Избор на етикетите и класификация на естествените типове

Финансовата област е доста обширна, затова базата от знания в проекта Ларфласт покрива само отделни нейни подобласти. При построяването на онтологията в STyLE са отчитани следните въпроси:

- Кои понятия, релации и факти са от особено значение за студента при изучаването на терминология на финансовия английски език?
- Защо се извличат и декларират в базата понятието *X*, релацията *Y*, факта *Z* – т.е. какво ще се прави с тях?
- Как точно да се кодират специфичните фрагменти от знанието, с цел да се отговори на нуждите на проекта и да се осигури по-лесна обработка?

Една от особеностите на STyLE е, че при модула OMC (Отворен модел на студента, вж. Фиг. 4.1) студентът вижда в графичен вид фактите в базата от знания или техни проекции. Най-просто е графите да се показват пред обучаемия с оригиналните етикети на понятията (за разлика от описаната в предишната глава генерация на естествен език, при която вътрешните етикети във всички случаи остават скрити за потребителя). При извличането на йерархията на типовете са възприети следните принципи [ANBN00]:

- Съдържанието на таксономията е съобразено с мнението на експертите-преподаватели, кои са най-важните понятия за заучаване в езиков курс по финансов английски език. Например две приоритетни понятия са BOND и STOCK. Части от йерархията на тези типове са показани на Фиг. 4.2. Класификационните перспективи са описани детайлно за важните понятия, тъй като знанието за тях се тества в многобройни упражнения и съответни фрагменти от йерархията се показват пред студента при нужда;
- Етикетите на понятията са съставени чрез пълно изписване на думите в съответните термини, без промени и съкращения. Неизбежно е въвеждането на надтипове, които не съответстват на термини, например надтипът на SECURITY е наречен PRODUCT_OF_FINANCIAL_MARKET. За тези 'фиктивни' термини се избират етикети-обяснения, които описателно декларират значението на типа. Унифицираният подход към наименоване на концептуалните единици позволява превода им на друг език без особени затруднения. По този начин е възможно интерфейсът за обучаемия да се локализара при желание на преподавателя;
- При описанието на класификационната перспектива, за която вече говорихме в предишната глава (вж. Фиг. 3.7), предикатите isa_kind/4 за естествените типове са разширени с нови аргументи. Изброяват се явно както подтиповете, така и характеристиките на вида класификация. Включено е подробно обяснение на перспективата, с цел то да се покаже на потребителя като текстов низ.



Фигура 4.2. Таксономия от финансови понятия и класификационни перспективи

Във връзка с проектирането на модула ОМС и идеята да се показват концептуални графи на потребителя-неспециалист, в проекта беше проведено изследване доколко графичният формат е разбираем за обикновените потребители - в случая, студенти по икономика без специална подготовка по информатика [Dim01]. Беше установено, че хората интуитивно разбират прости концептуални графи с ограничен брой върхове. Основна роля при разбирането играят понятията, наименовани с термини. Има два по-затруднителни елемента в изображението:

- явните имена на концептуалните релации могат да бъдат объркващи, когато са неочаквани;

- посоката на концептуалните релации също може да бъде объркваща, ако не отговаря на интуицията на неспециалиста.

Поради това в STyLE се прави 'окрупняване' на релациите и понятията, за да се формират прости графи с по-малко елементи. Така се осигурява възможността фактите да се показват на потребителя по-'наедро'. За редица важни релации са създадени дефиниции на типовете като например:

```
relation ISSUED_BY(x,y) is
[SECURITY: x] ← (THEME) ← [ISSUE] → (AGNT) → [ISSUING_AUTHORITY: y]
```

Прости графи с концептуалната релация ISSUED_BY се формулират и показват пред обучаемия чрез операцията 'свиване на тип', например

[BOND] ← (ISSUED_BY) → [COMPANY]

вместо съответните 'дума-по-дума' представяния, в които тематичните роли на глагола са дадени с явни имена:

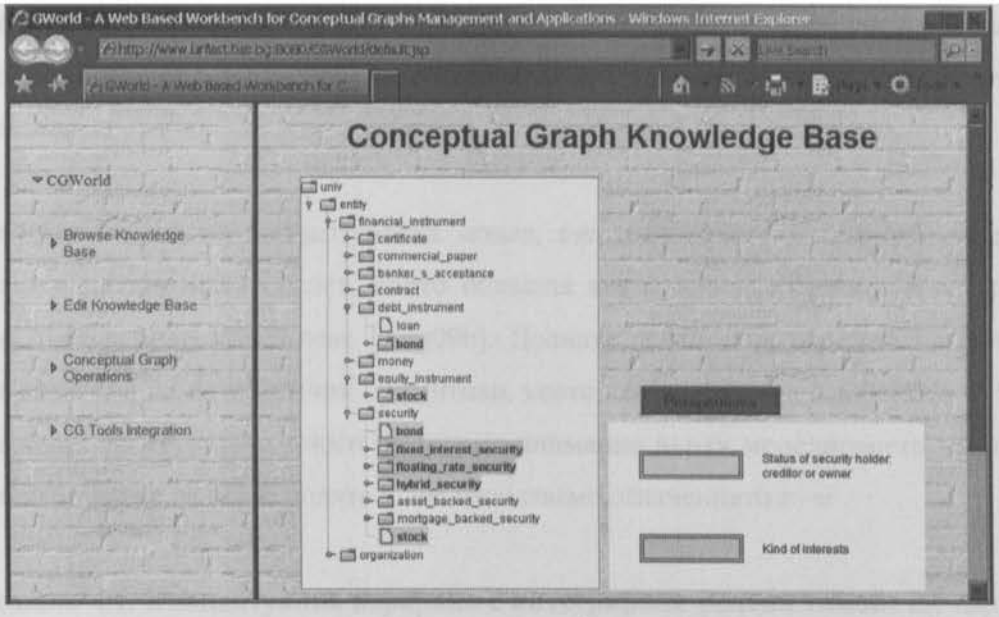
[BOND] ← (THEME) ← [ISSUE] → (AGNT) → [COMPANY]

Подобни дефиниции на типове са създадени и за други концептуалните релации като HOLDED_BY, DEPENDING_ON и т.н.

Както е показано на Фиг. 4.2, обръща се голямо внимание на класификационните перспективи. Разграничени са 'естествените' подтипове (т.е. такива, които са определени въз основа на неизменяеми свойства). Различните гледни точки върху подтиповете са обединени в една таксономия, например SECURITY има различни подтипове относно две 'естествени' класификации:

- Относно гледната точка *'kind of interest'* типът SECURITY се разделя на два подтипа: FIXED_INTEREST_SECIRITY и FLOATING_RATE_SECIRITY;
- Относно гледната точка *'status of security holder: creditor or owner'* типът SECURITY се разделя на три подтипа: BOND, HYBRID_SECURITY и STOCK.

И двете класификации са изчерпващи и непресичащи се, което е отбелязано явно, тъй като по премълчаване класификацията на Фиг. 4.2 е от пресичащи се типове и е направена по неизчерпващ начин. Фиг. 4.3 показва как таксономията се изобразява пред обучаемия с помощта на средата CGWorld, с различни цветове за различните видове класификации, а името на перспективата се дава като 'легенда' в долния десен ъгъл на екрана. Потребителят има възможност на разглежда цялата йерархия чрез кликване върху тип, който се отваря по обичаен за Windows начин.



Фигура 4.3. Визуализация на таксономия и класификационните й перспективи [Доб07]

4.2.2. Онтологичен модел на ролеви типове

Явното разграничение на типове-роли е необходимо за създаване на адекватен концептуален модел на терминологичните колекции (макар че терминолозите обикновено не използват роли). Но реални примери ни убеждават, че много термини означават роли на важни понятия от предметната област.

Пример 4.1. Нека разгледаме следните оригинални дефиниции на финансови термини, взети от сайта [InvWorld]:

- **broker/dealer** - any individual or firm in the business of buying and selling securities for itself and others. … When acting as a broker, a broker/dealer executes orders on behalf of his/her client. When acting as a dealer, a broker/dealer executes trades for his/her firm's own account. ....
- **investor** - an individual or firm who commits money to investment products with the expectation of financial return. …
- **bull** - an investor who believes that a particular security, a sector, or the overall market is about to rise. Opposite of bear.
- **bear** - an investor who believes that a security, a sector, or the overall market is about to fall. Opposite of bull.

Горните термини указват роли на лицата и фирмите в областта на финансовите пазари. Роли са различни от естествената класификация на *individual* като *animate* и на *firm* като *organisation*. □

С цел прецизиране на концептуалния модел, ще дефинираме интегриран модел на знанията в предметната област, който обхваща както класификацията в естествени типове, така и ролевите типове [Ang09b]. Долната дефиниция отрежда на ролевите типове значение на онтологични примитиви, което съответства на изказаното в [Ste05] предложение да се отдели много по-голямо внимание върху моделирането на ролите. Когато тип z може да играе ролята u, ще използваме означението $z{:\sim}u$.

Дефиниция 4.1. Концептуална йерархия с интегрирани ролеви типове ще наричаме наредената 5-орка $H = (N, R, T, I, \lambda)$, където:

- N е крайно, частично-наредено множество от различни естествени типове понятия. Частичната наредба дефинира йерархията на типовете понятия: за $x,y \in N$, $x \leq y$ означава, че x е естествен подтип на y. Тогава x е специализация на y и y е обобщение на x. Универсалният тип T (top) обобщава всички типове в N: за $x \in N$ съществуват $k \geq 0$ различни типа $x_1, x_2, \ldots, x_k \in N$ такива, че $x \leq x_1$, $x_i \leq x_{i+1}$ за $1 \leq i \leq k\text{-}1$ и $x_k \leq \mathsf{T}$. Всички типове в N обобщават абсурдния тип $\perp$ (bottom);
- R е крайно, частично-наредено множество от различни ролеви типове. $N \cap R = \varnothing$. За $x,y \in R$, $x \leq y$ означава, че x е ролев подтип на y. Тогава x е специализация на y и y е обобщение на x. За $x \in R$, съществуват $z \in N$, $u \in R$ и $k \geq 0$ различни типа $x_1, x_2, \ldots, x_k \in R$ такива, че $x \leq x_1$, $x_i \leq x_{i+1}$ за $1 \leq i \leq k\text{-}1$, $x_k \leq u$ и $z{:\sim}u$. За $z \in N$ и $u \in R$, $z{:\sim}u$ означава,че z може да играе ролята u;
- T е крайно множество от 5-торки, които характеризират класификациите на ролевите типове в подтипове:

$$T = \{ < c, \{ c' \mid c' \leq c \}, [\,'exhaustive' \mid 'non\text{-}exhaustive'\,], [\,'disjoint', 'joint'\,], \\ 'definition\text{-}of\text{-}c' > \text{където } c \in R \};$$

- I е множество от различни индивиди, които обозначават специфицирани екземпляри на понятия. $N \cap I = \varnothing$ и $R \cap I = \varnothing$.
- λ е изображение от I към $N \cup R$ и съпоставя индивиди на естествените и ролеви типове. Така λ дефинира съответствието и принадлежността (*conformity*) на екземплярите към естествените и ролеви типове понятия. □

Тази дефиниция определя само структурата на таксономията, построена върху естествените типове, и връзките на йерархии от ролеви типове към някои естествени типове. Изображението λ съпоставя екземпляри на понятия и по принцип може да поддържа и динамични съответствия. Както видяхме в глава 2 при дефиниция 2.1, в подобни определения се фиксира и ролята на концептуалните релации, но тук ние прецизираме само разграничението на типовете-роли от естествените типове с цел приложение на получената онтология в обучението. От тази гледна точка са определени подробно характеристиките на ролевите класификации: дали те са изчерпващи или не, дали са пресичащи се или не. Към всяко понятие-роля може да се включи и текстова дефиниция, която да се показва пред обучаемия.

Пример 4.2. Ще разгледаме концептуална йерархия с интегрирани ролеви типове, построена за термините от пример 4.1 съгласно дефиниция 4.1:

N = {INDIVIDUAL, ANIMATE, ENTITY, FIRM, ORGANISATION}, където
INDIVIDUAL≤ANIMATE, ANIMATE≤ENTITY, FIRM≤ORGANISATION и
ORGANISATION≤ENTITY;

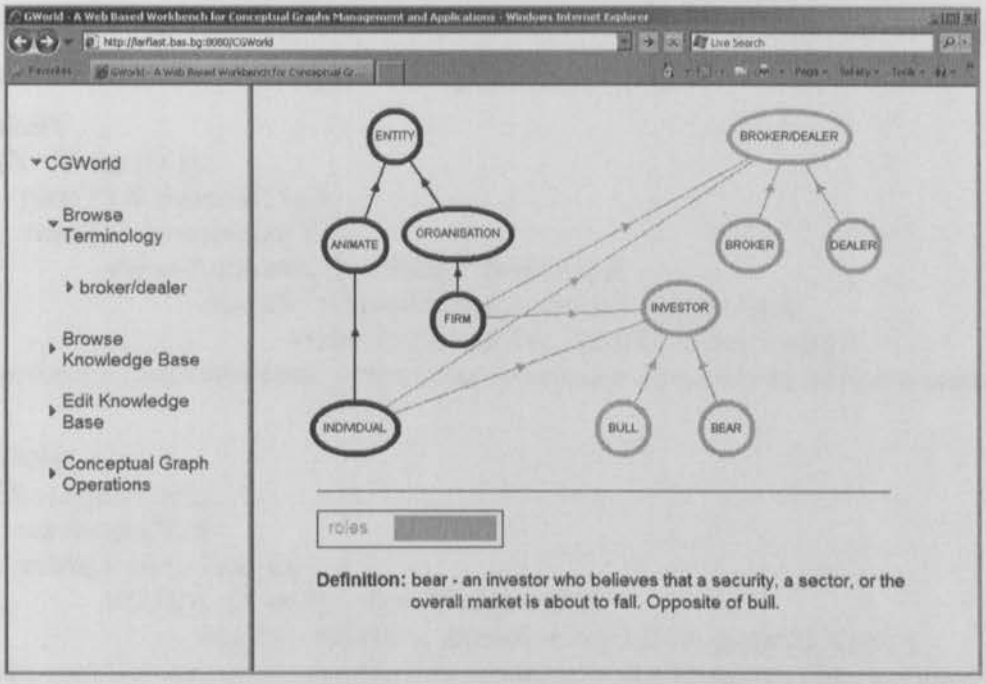
R = {BROKER/DEALER, BROKER, DEALER, INVESTOR, BULL, BEAR}, където
BROKER≤BROKER/DEALER, DEALER≤BROKER/DEALER, BULL≤INVESTOR,
BEAR≤INVESTOR, INDIVIDUAL:~ INVESTOR, FIRM:~ INVESTOR,
INDIVIDUAL:~ BROKER/DEALER, FIRM:~ BROKER/DEALER;

T = { < BROKER/DEALER, {BROKER, DEALER}, '*exhaustive*', '*disjoint*', '*def-brdealer*' >,
< INVESTOR, {BULL, BEAR}, '*exhaustive*', '*disjoint*', '*def-investor*' > };

I = {'Google Investor'},

λ: 'Google Investor' → ORGANISATION, λ: 'Google Investor' → INVESTOR. □

Фиг. 4.4 показва как се визуализират естествените и ролеви типове от Пример 4.2 в средата CGWorld. Низовете '*def-brdealer*' and '*def-investor*' кореспондират на определенията, дадени в Пример 4.1. В текстовите полета, предвидени от дефиниция 4.1, могат да се включат произволни изречения за показване пред студентите. Графът на Фиг. 4.4. е подготвен чрез средата CGWorld и запомнен при извличането на концептуалния модел от около 300 важни английски термина в областта на финансите. Предложеното решение се използва при изучаване на втори език – една много важна област, тъй като повечето хора учат чужди езици в продължение на целия си живот. Концептуалният ресурс и подхода могат да бъдат интегрирани и в друга система, в която онтологията служи за опорна конструкция на дигитализирано съдържание.



Фигура 4.4. Концептуален скелет на английски финансови термини

4.2.3. Интегриране на постулати на значенията на думите

В STyLE като отделен компонент работи системата Parasite, която превежда отговорите на студента, зададени на свободен английски език, до вътрешни предикатно-аргументни структури. Поради това възникна нетривиалният проблем за хармонизиране на постулатите на значенията на думите, използвани в Parasite, и дефинициите на типовете-термини в базата от знания. Това е необходимо поради факта, че студентът вижда както диагностиките на Parasite при грешен отговор, така и съдържанието на базата от знания. Беше необходимо да се съгласува таксономията от понятия в предметната област със скритата йерархия на думите така, както те са описани чрез постулатите на лексикалните значения. От гледна точка на формалната лингвистична семантика в компютърната лингвистика, постулатите са: '*... твърдения за връзките между значенията на думите. Това не са дефиниции на значенията, нито необходими и достатъчни условия за валидност на знанието в света, а описания на взаимни ограничения върху значенията, които влизат в сила когато съответните думи се композират в изречение*' [Ram95]. Тази особеност при дефиниране на аксиомите за значенията на думите в дисциплината 'разбиране на естествен език' е подчертана и при обсъжданията на Фиг. 1.5б в първа глава. Ще дадем примери за

постулати (MP – meaning postulates), използвани при тестване на отговорите на обучаемите във финалната версия на STyLE [ASKBV04]:

```
lexicalMP(
forall(X::{budget(X)},
        plan(X) & financial(X) &
        exists(Y::{summarize(Y)},
                exists(I::{income(I)}, theta(Y,$object,I) &
                        exists(E::{expenditure(E)}, theta(Y,$object,E) &
                                exists(T::{period(T)}, theta(Y, $over, T)))))) ).
```

(*Бюджетът е финансов план, резюмиращ приходите и разходите за даден период*)

```
lexicalMP(
forall(X::{capacity(X)},
        maximum(X) &
        exists(Y::{produce(Y)},
                exists(Z::{firm(Z)}, theta(Y,$agent,Z) &
                        forall(U::{unit(U)}, theta(Y,$object,U) & count(U,X))))) ).
```

(*Капацитетът е максималният брой на всички изделия, произвеждани от фирма Z*)

```
lexicalMP(
forall(X :: {export(X)},
        (good(X) or service(X))
        & exists(Y :: {sell(Y)}, theta(Y,$object,X)
                & exists(Z :: {theta(Y,$agent,Z)},
                        exists(ZC :: {country(ZC)}, location(Z, ZC)
                                & exists(T::{buyer(T)}, theta(Y,$to,T)
                                        & exists(TC::{country(TC)},
                                          location(T, TC)
                                          & not(ZC = TC)
                                          & forall(D :: {to(X, D) & country(D)},
                                            TC = D)))))) ).
```

(*Експорт X на стоки или услуги е продажба на обекта X от агент Z, намиращ се в страна ZC на купувач T, намиращ се в различна страна TC. Всеки екземпляр на понятието 'експорт' се изнася в точно една страна TC*)

На абстрактно ниво, постулатите на значенията кодират факти подобни на онези, които са представени в базата от знания. Но постулатите имат различен синтаксис и са ориентирани към контролиране правилността на думите, очаквани в отговорите на определени упражнения от педагогическия ресурс. По тази причина постулатите на значенията са фокусирани върху определени аспекти на семантиката на думите, в специфичното лигвистично проявление на тематичните роли на глаголите и т.н.. Горните примери показват, че в постулатите се включват и факти, които бихме могли да наречем 'знания за света', но те са кодирани около променливи и терми, вложени в декларациите на тематичните роли, поради което не е лесно да бъдат разпознати от

автоматични процедури. С цел хармонизиране на двата вида семантични представяния в STyLE –твърдения *(i)* за понятията и релациите в базата от знания и *(ii)* за значенията на използваните в предметната област думи, бяха извършени следните стъпки за унифициране на съдържанието:

- Всички прости концептуални графи, дефинирани ръчно в базата от знания, са преведени автоматично до логически формули по алгоритмите на [Sow84];
- Въз основа на тези логически формули, ръчно са произведени постулати на значенията в синтаксиса на Parasite.

Във финалната си версия STyLE работи над около 300 термина от областта на финансовите пазари. Проверката на упражнения на свободен английски език се извършва над ограничен брой думи, за които има около 150 постулата на значенията.

4.3. Термините като връзка между езиковите и концептуални компоненти и ресурси на системата

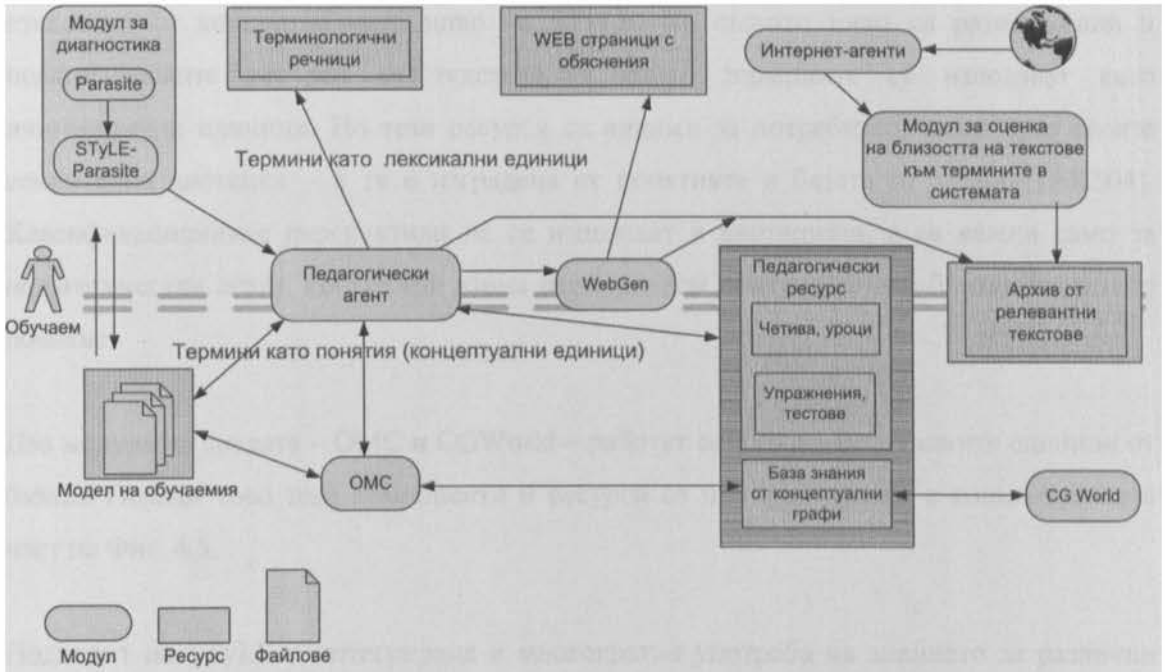
В трета глава бяха изложени вижданията на автора, че термините-понятия са връзката между лексикалните и концептуални ресурси на системата (вж. указателите на Фиг. 3.2). В средата STyLE е приложено същото решение, но в едноезиков план; усложнението идва от хетерогенността на компонентите и ресурсите в системата.

Макар че указателите

`термин от лексикона` ↔ `идентификатор на понятие в базата от знания`

не са показани като отделен ресурс на Фиг. 4.1, те съществуват и осигуряват връзките между модулите и ресурсите на системата. Тези указатели позволяват обучаемият да вижда термините в текста като думи, а в базата от знания – като етикети на понятия. На Фиг. 4.5 е илюстриран начинът на използване на лексикалните и концептуални единици:

- Parasite и STyLE-Parasite разглеждат термините само като единици от техните лексикони и/или постулати на значенията. След завършване на дадено упражнение, те подават на Планиращия агент информация за идентификатора на упражнението и резултата на студента;



Фигура 4.5. Термини-думи и термини-понятия в STyLE

- В многоезиковите речници термините са описани като лингвистични единици. Тези речници са достъпни или от главния интерфейс на системата, или по препоръка на Планиращия агент в текущата учебна ситуация;
- Интернет-агентите търсят по финансови сайтове текстове, в които се срещат разглежданите термини;
- Модулът за изчисление на близостта разглежда само текстовете и термините като съвкупност от думи. Той изчислява коефициент на релевантност на всеки текст към всеки термин в системата.

Всички изброени до тук модули и ресурси се намират в горната половина на Фиг. 4.5, над чертата, разделяща езиковото от концептуалното ниво на STyLE.

Планиращият агент работи с концептуалните единици, но ползва и указателите, за да превърне намерените по мрежата текстове в динамичен архив от допълнителни четива. Когато Модулът за оценка на близостта присвои мярка за релевантност на някой текст, Планиращият агент превръща текста в учебен обект със съответна семантична мета-анотация. Тъй като Планиращият агент, WebGen и Архива от текстове боравят и с термините, и с понятията, те са разположени на Фиг. 4.5 върху чертата, разделяща

езиковото от концептуалното ниво на STyLE. На същото ниво са разположени и педагогическите ресурси от текстове, в които термините се използват като лингвистични единици. Но тези ресурси са видими за потребителя само чрез своята семантична анотация – а тя е изградена от понятията в базата от знания [AKS04]. Класификационните перспективи не се използват в анотацията, а са важни само за педагогическия агент, когато той взима решение кои понятия са най-близки за дадено понятие.

Два модула на средата – OMC и CGWorld – работят само с концептуалните единици от базата. Поради това тези компоненти и ресурси се намират изцяло в концептуалната част на Фиг. 4.5.

Подходът на STyLE е интегрирана и многократна употреба на знанието за различни цели:

- за показване пред студентите в модула OMC,
- за анотиране на учебните обекти,
- при избор на стратегия за адаптиране на действията на системата към текущата учебна ситуация, включително навигация в педагогическия ресурс и т.н.

Идентификаторите на понятията са въведени ръчно в анотацията на упражненията и оттам те се разпространяват автоматично по всички останали модули и ресурси на системата. Така анотационната схема е независима от предметната област. Лесно могат да се добавят нови статични ресурси – педагогически, концептуални и лексикални - с подходящи метаданни. STyLE може да работи и над друга предметна област, ако статичните ресурси са декларирани във възприетия формат.

Ще покажем повече детайли за използването на концептуални етикети в анотацията на учебните обекти, тъй като това е основата за адаптивната стратегия на средата. Към всички учебни обекти се прикрепят следните унифицирани метаданни [AKS04]:

- `CourseID` – идентификатор на курса на провежданото обучение;
- `TopicID` – идентификатор на темата (урока) в курса за обучение;
- `Unit_No` – номер на секцията в урока – най-малката единица, на която се базира адаптацията при предоставяне на материали на студента;

- `Unit_No_Drill_No` – идентификатор на упражнение за учебен обект тип "*exercise*" или адрес в Интернет, откъдето е свален обектът тип "*reading*";
- `Вид на учебния обект` – може да бъде два типа: *exercise* или *reading*,
- `List_of_TestedAspects` – списък на тестваните аспекти, празен за четивата и списък от тройки `[Aspect, PropositionId, Weight]` за упражненията;
- `Label`:
  - `ConceptLabel` за упражнения, показва за кое понятие се отнася упражнението или
  - списък от двойки `[Term, RelevanceScore]` за четива (защото един текст може да бъде подходящо четиво за повече от един термин и повече от един урок),
- `Autorship` – Кой е създал обекта за обучение, и
- `CreationDate` – Дата на създаване на обекта.

Тройките `[Aspect, PropositionId, Weight]` в `List_of_TestedAspects` имат следното значение:

- `Aspect` – представя основни типове аспекти, които се тестват, а именно
  - *bdef*: основна дефиниция,
  - *afact*: допълнителен факт,
  - *rel*: възможни свойства кодирани с концептуални релации – агент, атрибут, характеристика, инструмент и др.
- `PropositionId` – уникален идентификатор на факта (прост концептуален граф) в базата от знание;
- `Weight` – предварително дефинирано от експерта-преподавател число в интервала [1, 100], отразяващо важността на тестваната информация за разбирането на изучаваната област при обучение на чуждоезикова терминология.

Измежду така изброените по-горе анотационни полета, три се пренасят в Модела на студента при изпълнение на текущото упражнение. Моделът съдържа четири вида клаузи:

- ***know*** – студентът *X* знае факта (термина) *Y* в изучаваната област. Клаузата се записва, когато студентът дава правилен отговор на текущото упражнение;

- ***not_know*** – студентът *X* не знае факта (термина) *Y*. Клаузата се записва, когато студентът дава грешен отговор на текущото упражнение;
- ***self_not_know*** – при отговор на текущото упражнение за тестване на термина *Y*, студентът *X* избира от меню "*не знам*";
- ***know_wrongly*** –студентът *X* има неправилна престава за понятието *Y*, и знанията му трябва да се подобрят. Клаузата се записва при частично правилен отговор на текущото упражнение.

Всички клаузи са предикати с един и същи брой аргументи:

```
predicate_name(UserID, Concept_Label, [List_of_domain_facts],
    Unit_No_Drill_No, NumberOfTest, Annotation, LM_ID).
```

Измежду тези седем аргумента, три се пренасят от анотацията на изпълняваното от студента упражнение. Тъй като отразява понятията в предметната област, Моделът на студента се намира в концептуалната част на Фиг. 4.5. По-детайлно описание на модела и адаптивните свойства на STyLE е предложено в [Кал07].

4.4. Лингвистична семантика на текста vs. концептуално представяне на знанията за предметната област

В края на изложението на авторските резултати, можем да обобщим направените наблюдения относно природата на 'семантичните представяния' в разгледаните приложения. В този труд построихме представяния от две гледни точки:

- *в компютърната лингвистика за езикови единици* – фрази, изречения или кохерентен дискурс, с цел разбиране на естествен език или автоматично извличане на факти/знания от текст. Прототипи от този вид бяха разгледани в част 3.3;
- *при конструиране на онтологии и бази от знания* – т.е. при построяване на явни, декларативни ресурси от логически твърдения относно свойствата на понятията и релациите в предметните области на приложните системи. Концептуалното моделиране в DB-MAT и STyLE беше разгледано съответно в части 3.2 и 4.2.

Общото между двата вида представяния е изискването за експлицитно деклариране на отношенията между единиците, съответно на единиците на текста (започвайки от

думите) или единиците на концептуалния модел (започвайки от понятията). Затова често говорим за отношения или релации и при двете представяния:

- тематични роли на глаголите като семантични отношения между думите в изреченията, дискурсни релации като отношения между клаузите в текста и т.н., както и
- концептуални релации между понятията в базата от знания, свойства и атрибути на понятията и т. н.

От гледна точка на човека, много от тези релации се подразбират по премълчаване. Затова много от тях не се означават със специални думи в изказванията на естествен език. При компютърния модел, обаче, въвеждането на явно име е задължително.

Подходът на компютърната лингвистика е ориентиран към обработка на думите в конкретни изречения и текстове. Анализът се осъществява в рамките на лингвистични теории и вътрешните семантични представяния са построени съгласно конвенциите на тези теории. Целта е да се създаде възможно най-пълен модел на лингвистичните явления във входния текст. При наличие на конкретен входен текст, обаче, вече имаме конкретна употреба и тя влия на значението на думите. Достатъчно е да припомним цитираните изказвания, че 'на обектите не е присъщо да са броими или неборими; те се разглеждат като такива от определена перспектива' [Dal88] и че 'пътят е линия – когато планираме пътуване, повръхнина – когато шофираме по него, и обем – когато пропадаме в дупка' [Hob95]. От гледна точка на компютърния лингвист, семантичното представяне на текста се построява едва след контекстуалната интерпретация на изказването.

При концептуалните представяния стремежът е точно обратният - да се постигне езиково- и контекстно-независимо описание на предметната област. Например таксономията на Фиг. 4.2 и класификационните перспективи в нея не са извлечени от никой конкретен текст (още по-малко от конкретно изречение), а резюмират съдържанието на най-различни източници. В този смисъл инженерите на знанията и философите се стремят към универсална цел, чиято практическа осъществимост е под въпрос извън тесни предметни области.

ГЛАВА 5:
Заключение и насоки за бъдеща работа

Извличането, представянето и обработката на знанията е ключова област за символните пресмятания в изкуствения интелект. Понастоящем тя е обект на бурно развитие с оглед нуждите на възникващите глобални информационни приложения, при които голяма част от данните за заобикалящия ни свят ще бъде съхранена във вид на дигитални обекти. Навигацията в огромните информационни ресурси е възможна само чрез използване на онтологии или фолксономии, изградени над етикети - думи от естествения език. Компютърната лингвистика също поднася все по-добри прототипи на системи за добиване на знания от текста и така подпомага процеса на полу-автоматично конструиране на онтологии и фолксономии. В този смисъл научната дейност на автора е насочена към едни от най-актуалните теми в информатиката и очертава бъдещи насоки за усъвършенстване на семантично-базираните системи.

Можем да изброим резултатите, описани в представения труд според съдържанието на втора, трета и четвърта глава както следва:

Във втора глава е предложена технология за съхранение и обработка на концептуална информация като регулярен език. Под 'концептуална информация' тук разбираме прости концептуални графи – екзистенциално-квантувани, положителни конюнктивни формули над двуместни предикати, които могат да се интерпретират като твърдения за фактите в една добре дефинирана предметна област. Търсенето на концептуални шаблони чрез инективна проекция се реализира посредством двуфазов алгоритъм. На предварителната фаза се извършват всички подготвителни пресмятания с експоненциална сложност, които не зависят от потребителски въпроси. Предимствата на подхода проличават след компресирането на обобщенията в минимален краен автомат с маркери на заключителните състояния. В реално време, конкретната заявка се обработва почти с линейна сложност. Доколкото има по-сложни изчисления, те третират само

превод на потребителския въпрос до дума от регулярния език и по тази причина са с ограничен обем. Предложеният подход позволява бързо намиране на инективна проекция и проверка на идентичност на графи. Минималният краен автомат е полезен и в още един аспект: той позволява да се обособят 'подобни' прости концептуални графи, които се състоят от еднакви предикати и аргументи, свързани по различен начин (например графите на Фиг. 2.7а, 2.7б и 2.7в). Обикновено такива твърдения не се разглеждат като 'близки' в изкуствения интелект, но – под влияние на методите за измерване на подобие между документи на естествен език – можем да кажем, че те са сродни, понеже се отнасят за едни и същи понятия и релации.

Оценките за сложност на алгоритмите и проведените експерименти показват, че при реализацията са необходими значителни изчислителни ресурси като време и памет. Но при двуфазовия подход ресурсоемката обработка се извършва предварително, когато няма ограничения за времето. Фактите от базата се съхраняват на външна памет, но минималният краен автомат е компактен и при средна по обем база може да стои в оперативната памет. Експериментите потвърждават очакванията за добра скалируемост на подхода, който радикално подобрява времето за изчисление на инективна проекция. Конструкцията включва иновативни аспекти:

(i) въведена е унифицирана азбука на представянето на простите концептуални графи, която осигурява сравняване на етикетите между всички съществуващи и бъдещи графи в разглеждания свят. Лексикографската наредба на символите от тази азбука позволява еднозначно кодиране на графичните структури. Така става възможно да се изброят в сортиран списък от думи етикетите на всички подграфи, а маркерът за топологичните им връзки да се третира като анотация към всяка дума. Всички подграфи от базата, които са отговори на потребителски въпроси за търсене на концептуални шаблони, се съхраняват като маркери. Така подграфите с еднакви обобщения се оказват групирани в един и същи маркер;

(ii) въведено е кодиране на цялата база от знания като единен минимален ацикличен краен автомат с маркери на заключителните състояния. Това прави скоростта на трасиране на думи в него независима от големината на базата.

Както казахме в увода на дисертацията, задачата за обработка на извънредно големи концептуални ресурси е много актуална. Алгоритмите за ефективно търсене и оптимизация са задължителен елемент от бъдещите семантичните системи. С появата на почти неограничена по обем външна памет, главното предизвикателство ще бъде намирането на техники за бързо търсене в реално време. Предложеният подход е възможно решение в тази насока. Той дава насоки за решаване на актуалните проблеми за ефективно търсене в големи семантични ресурси и поставя нови изследователски проблеми за преосмисляне на алгоритмите за пресмятания в двуфазов план, с изнасяне на колкото е възможно повече изчисления на предварителен етап.

В трета глава е представена архитектура на DB-MAT – прототип за генериране на обяснения в техническа предметна област, по зададена база от знания във вид на концептуални графи. Предложен е алгоритъм за извличане на фрагменти от знания, които да се разказват на потребителя на естествен език като системен отговор на заявка за обяснение на факти относно определено понятие. Извличането се реализира чрез инективна проекция на шаблони върху базата от знания. Тъй като става дума за енциклопедично знание в техническа област, предполага се, че обобщеният неспецифициран екземпляр е типичен представител на понятието и затова всички извлечени факти за този екземпляр се разглеждат като реферирящи към един обект – поради което всички прости клаузи се обединяват в един параграф. Предложени са и решения за моделиране на контекста на диалога с цел избягване на повторенията в съседни обяснения. Получената архитектура е изградена на модулен принцип; възможно е цялата система да се прехвърли върху друга предметна област и лесно да се добави друг език за генериране на обяснения – например елементарно е да се добави английски език, както това е направено в прототипа DB-MAT за български.

Концептуалното моделиране при извличането на знанията за предметната област е извършено с решаващото участие на автора. В системата има единна таксономия и една база от знания, като на концептуално ниво се диференцират единици с различна грануларност (към които реферират съответно термини на немски и

български с различна лексикална грануларност). Проблемът с грануларността на лексикалните и концептуални елементи в многоезиков план дотолкова засяга 'кухнята и вътрешното know-how' на отделните системи, макар че рядко се дискутира детайлно в научни публикации. Авторът познава само още едно решение от подобен вид – споменатата в част 1.2.2 идея на онтологията МикроКосмос, състояща се от примитиви (роли и свойства), до които се декомпозират (и диференцират) значенията на думите в различните езици. В DB-MAT това решение е невъзможно, тъй като екземплярите на понятията се вадят от цялата база и се подават на генератора за оформяне на текстови обяснения; поради това се наложи моделиране на различни по грануларност единици чрез контекст в рамките на всеки конкретен концептуален граф.

Въведени са перспективи при класификацията в йерархията, които позволяват прецизиране на съдържанието на генерираните текстове. Предложен е и начин за съчетаване на броими и неброими типове в една концептуална йерархия, чрез явно деклариране на класификационната перспектива.

Използвайки създадените дълбоки онтологии, авторът е изследвал аспекти на разбирането на естествен език от компютър. В контекста на DB-MAT е предложен алгоритъм за тестване на коректността на човешкия превод, в който се следи превода на референциите към термините и се сигнализира за евентуална многозначност. В контекста на база от знания във финансовата област са предложени алгоритми и програмни реализации на модули, подпомагащи извличането на факти от текст и тестването на съдържанието на базата от знания с отрицателни въпроси.

В четвърта глава е предложена архитектура на интелигентна система за обучение, базирана върху знание. Основната цел е да се съвместят зад единен потребителски интерфейс модули за обработка на естествен език, които ползват термините като думи и лексикални единици, и модули за обработка на концептуална информация, които третират термините като понятия. Авторът е разработил принципите за

построяване на онтологията. Решенията за концептуално моделиране разширяват използваните в DB-MAT перпективи с оглед детайлно деклариране на свойствата на класификацията тип-подтипове. В концептуалния модел са включени дефиниции на типове, чрез които се формират типове с по-голяма грануларност за показване пред обучаемия в компактен вид. Следвайки съвременните схващания за моделиране на ролите като онтологични примитиви, в настоящия труд се предлага решение за обособяване на ролите в отделно множество типове. Дефинирана е концептуална йерархия, в която са смесени класификация от естествени подтипове и йерархии на ролеви типове. Полученият семантичен модел е тестван в среда за изучаване на чуждоезикова терминология в областта на финансите. Онтологията е основа за адаптивността на системата, тъй като е главният ресурс, осигуряващ възможностите за навигиране и вземане на решение какво да се направи на следващата стъпка. Семантичната анотация на учебните обекти също се основава върху концептуалния модел на предметната област. Предложеното решение за анотация и навигация над метаданните може да се приложи и в други системи, в други предметни области.

Авторът ще продължи работата си по темата на настоящия труд в следните направления:

Главна научна задача е да се задълбочат изследванията на предложения двуфазов подход за търсене на специализации, както и да се очертаят възможностите за интегрирането му в алгоритми за извършване на умозаключения чрез поддържане на базата от знания в два алтернативни формата: традиционен и компресиран във вид на минимален краен автомат. Ще бъдат построени и прототипи за изучаване на подхода в реални условия. Приложенията ще бъдат създадени в рамките на проекта ЕВТИМА 'Ефективно търсене на концептуални шаблони с приложение в медицинската информатика', финансиран в конкурс ИДЕИ-2008 на Националния фонд 'Научни изследвания' чрез договор ДО2-292/18.12.2008 за периода 2009-2011.

Авторът се надява, че с развитието на информационните услуги в България ще стане възможно и практическото приложение на технологията за генерация на кратки обяснения. Вече скицирахме един възможен сценарий за генерация на съобщения относно степента на замърсяване на въздуха. По подобен начин, полезна услуга в здравеопазването би била автоматичната генерация на кратки персонализирани съобщения и изпращането им като SMS до кръг от абонати, както и много други задачи за автоматично разпространение на еднотипна информация.

Предложените в дисертацията решения за концептуално моделиране (перспективи, роли и управление на понятия с различна грануларност) имат пряко приложение в алгоритмите за анотация и навигиране при достъп до дигитални архиви. Макар че това все още не е видимо за широката публика у нас, основните проблеми в тази област са свързани не само с техническото разработване, но и с организацията на дигиталните обекти и анотирането им с метаданни на естествен език. Това се наблюдава при развитието на ултра-големи дигитални архиви като Europeana[17] и Learning Resource Exchange[18], където главните проблеми на вече създадените репозитории са навигацията в множествата от обекти и хармонизирането на анотациите им. Представените в четвърта глава постижения представляват полезни решения, които могат да се интегрират в практически системи.

ПРИНОСИТЕ на дисертацията са следните:

Научни приноси:

1. Дефиниран е линеен запис на прости концептуални графи с двумерни концептуални релации. Записът е единствен с точност до изоморфизъм между променливите, присвоени на върхове с еднакви етикети в графа. Използвана е азбука от етикетите на опората на базата от знания.

[17] http://www.europeana.eu/portal/, портал за търсене в културните репозитории на Европа, последно посещение 26 април 2009.
[18] http://lreforschools.eun.org/LRE-Portal/Index.iface, архив от около 30000 учебни обекта и други материали на повече от 20 езика, създаден от European schoolnet заедно с 16 европейски министерства на образованието, последно посещение 26 април 2009.

2. Конструиран е двуфазов алгоритъм за намиране на инективна проекция по зададена заявка, при който времето за отговор зависи само от дължината на заявката, а не от размера на базата от знания. На предварителната (off-line) фаза, цялата база от прости концептуални графи с двумерни концептуални релации се запазва като ацикличен минимален краен автомат с маркери на заключителните състояния. По време на изпълнение на заявката (в run-time), въпросът се свежда до дума на регулярен език и с нея в линейно време се трасира минималният краен автомат с маркери на заключителните състояния. Така пресмятанията в една NP-пълна задача са разделени на два компонента: *(i)* предварителен с експоненциална сложност и *(ii)* работещ в реално време с линейна сложност.
3. Дефинирана е концептуална йерархия, в която са смесени класификация от естествени подтипове и йерархии на ролеви типове. Показан е един критерий за определянето на даден тип като роля: екземплярите му да сменят динамично типа си. Дефиницията задава подход към конструирането на онтология: да се фиксират класификациите на естествените типове, към които да се добавят йерархиите на ролевите типове. Разграничаването на ролите като онтологични примитиви осигурява адекватно моделиране на семантичния компонент в терминологични колекции.

Научно-приложни приноси:

1. Конструиран е алгоритъм, извличащ по избрано понятие фрагменти от знания, които да бъдат разказани на естествен език при генериране на обяснения в техническа област. Извличането се извършва чрез инективна проекция на концептуални шаблони върху базата от знания. Алгоритъмът е апробиран в системата DB-MAT, която по въпрос на потребителя генерира обяснения на немски и български език в техническа област.
2. Конструиран е алгоритъм за подпомагане на човешкия превод на технически текстове от немски на български език чрез следене на референцията, който е реализиран в системата DB-MAT.

3. Конструиран е алгоритъм за извличане на концептуални структури от анализиран текст в областта на финансите, както и за обработка на въпроси с отрицание към база от знания в тази област.
4. Предложена е архитектура на интелигентна среда за изучаване на чуждоезикова терминология, базирана върху знание, в която са интегрирани различни езикови и семантични технологии. Предложена е анотационна схема с метаданни на архив с учебни обекти.
5. Предложен е модел за разграничение на езиковите и концептуалните ресурси в система за обработка на естествен език, базирана върху знания. Въведена е схема за описание на връзките между понятията в базата от знания и многоезиковия лексикон на системата DB-MAT, с цел третиране на различната грануларност както между концептуалните и лексикални структури, така и между немските и български термини.
6. Предложени са две прецизирани решения при концептуалното моделиране на знания за дадена предметна област:
   (i) въвеждане на явно-наименовани перспективи в концептуалните йерархии;
   (ii) моделиране на категорията броимост-неброимост;

Приложни приноси:

1. Предложен е подход за моделиране на диалога в системата DB-MAT, който редуцира повторенията в обяснения, генерирани в резултат на изпълнение на последователни заявки.
2. Извършено е концептуално моделиране в две предметни области:
   - на пречистване на отпадни води, като получената база знания е вградена в прототип за генерация на многоезикови обяснения;
   - на финансовите пазари, като получената онтология с интегрирани ролеви типове е вградена в среда за самообучение при изучаване на английска финансова терминология.

Проекти, свързани с резултатите на дисертацията:

- **1992-1995** - *DB-MAT: Немско-български компютърно-подпомогнат превод, базиран върху знания*, финансиран от Фондация "Фолксваген" – Германия. Координатор: Валтер фон Хан (Хамбургски университет). Ръководител на българската група: Галя Ангелова;
- **1995-1996** - *DOC-GEN: Генериране на хипертекстова документация на естествен език от вътрешните спецификации на обектно-ориентирания CASE-Tool OBLOG*, изпълнен за португалската фирма OBLOG Software S.A. Ръководител Галя Ангелова;
- **1995-1997** - *CGLex: Среда за извличане на концептуални графи, базирана върху естествен език*, проект I-420/94 с Националния фонд "Научни изследвания" с ръководител Галя Ангелова;
- **1996-1998** - *DBR-MAT: Немско-българско-румънски компютърно-подпомогнат превод, базиран върху знания*, финансиран от Фондация "Фолксваген" – Германия. Втора фаза на *DB-MAT*. Координатор: Валтер фон Хан (Хамбургски университет). Ръководител на българската група: Галя Ангелова;
- **1998-2001** - *LARFLAST Learning Foreigh Language Scientific Terminology*, Copernicus'98 JRP 977074, финансиран от Европейската комисия. Научен координатор: Галя Ангелова като външен сътрудник на ФМИ на СУ "Св. Кл. Охридски";
- **2001-2003** - *Balric-Ling: Балкански регионални информационни центрове за осведомяване на стандартизация на лингвистични ресурси и среди в модерни приложения на езиковите технологии*, IST-2000-26454, финансиран от Европейската комисия в 5-тата Рамкова програма. Координатор: Галя Ангелова (един от първите проекти с много партньори, координирани от България в Рамкова програма на ЕК);
- **2001-2004** – Работен пакет 4 "*Езикови технологии*" в проекта на ИПОИ-БАН BIS-21 Center of Excellence ICA1-2000-70016, финансиран от Европейската комисия в 5-тата Рамкова програма. Ръководител на пакета: Галя Ангелова;

- **2005-2007** – Работен пакет "*Системи, базирани върху знания*" в проекта на ИПОИ-БАН BIS-21++ Center of Competence FP6-2004-ACC-SSA-2, финансиран от Европейската комисия в 6-тата Рамкова програма. Ръководител на пакета: Галя Ангелова;
- **2009-2011** – *ЕВТИМА Ефективно търсене на концептуални шаблони с приложение в медицинската информатика*, проект по конкурс ИДЕИ ДО2 292/18.12.2008 с Националния фонд "Научни изследвания" с ръководител Галя Ангелова.

Апробация на получените резултати

Статиите по дисертационния труд са публикувани в период от 15 години (1994 – 2009). Резултатите, описани в тези статии, са представяни многократно от автора пред научна аудитория и рецензенти по проекти. Голяма част от докладите на конференции, свързани със съвместни публикации, са изнасяни от по-младите сътрудници на автора (винаги, когато е било възможно).

Лекции по покана:

- Едноседмичен курс лекции по *'Концептуални графи и обработка на естествен език'*, Факултет по комуникации, Университет на Аалборг – Дания, декември 1997 г.;
- Доклад по покана на Международния семинар ОнтоЛекс, Септември 2000, Созопол,
- Поканена лекция във Факултета по природо-математически науки на Университета в Скопие, Македония, март 2007,
- Доклад по покана на 37-мата Пролетна конференция на Съюза на Математиците в България, Боровец, април 2008.

Сесии за финално оценяване на международни проекти:

- Май 1998, София – заключителен семинар на проекта DBR-MAT (финансиран от Фондация Фолксваген, Германия), с външни експерти проф.

Крушид Ахмад (Университет в Съри, Великобротания) и проф. Сюзън Армстронг (Университет в Женева, Швейцария);

- Ноември 2001, София - финално ревю на проекта Ларфласт (финансиран от Европейската комисия), с рецензент проф. Джон Нербън (Университет в Гронинген, Холандия).

Доклади, изнесени от автора на международни научни събития:

- International Conference *Machine Translation - Ten Years on,* Cranfield, United Kingdom, November 1994,
- International Conference *Recent Advances in Natural Language Processing* RANLP-95, Batak, Bulgaria, September 1995,
- 16th International Conference on Computational Linguistics COLING-96, Copenhagen, Denmark, August 1996,
- 7th International Conference *Artificial Intelligence – Methodology, Systems, Applications* AIMSA-96, Sozopol, Bulgaria, September 1996,
- International Conference *Recent Advances in Natural Language Processing* RANLP-97, Batak, Bulgaria, September 1997.
- 5th International Conference on Conceptual Structures (ICCS-97), Seattle, USA, August 1997,
- 6th International Conference on Conceptual Structures (ICCS-98), Montpellier, France, August 1998,
- 8th International Conference AIMSA-98, Sozopol, Bulgaria, September 1998,
- International Workshop *Web Information Technologies: Research, Education and Commerce* (WITREC-2000), Univ. Montpellier II, France, 2000,
- 8th International Conference on Conceptual Structures ICCS-2000, Darmstadt, Germany, August 2000,
- 9th International Conference *Artificial Intelligence – Methodology, Systems, Applications* AIMSA-2000, Varna, Bulgaria, September 2000,
- International Symposium dedicated to the 60th anniversary of Walther von Hahn, University of Hamburg, April 2002,

- Workshop on *Semantics, Ontologies and eLearning* (WOSE-04), in conjunction with the Federated conference *On the Move to Meaningful Internet Systems 2004*: OTM 2004, Agia Napa, Cyprus, October 2004,
- International workshop "*Language Resources: Integration & Development in e-learning & in Teaching Computational Linguistics*", in conjunction with LREC 2004, Lisbon, Portugal, May 2004,
- International workshop "*eLearning for Computational Linguistics and Computational Linguistics for eLearning*", in conjunction with COLING 2004, Geneva, Switzerland, August 2004,
- 13th International Conference on Conceptual Structures ICCS-2005, Kassel, Germany, July 2005,
- 14th International Conference on Conceptual Structures ICCS-2006, Aalborg, Denmark, July 2006,
- Юбилейна международна конференция, посветена на 30тата годишнина на Секцията по математическа лингвистика в Института по математика и информатика на БАН, София, юли 2007,
- International workshop *Íomhá: Images, ontology and multi-modal heritage access,* Тринити-Колидж, Дъблин, Ирландия, 22 май 2008,
- 16th International Conference on Conceptual Structures ICCS-2008, Toulouse, France, July 2008.

Доклади на семинари:

- Семинари на Секцията по Лингвистично моделиране на ИПОИ-БАН – многократно,
- Семинар по *Динамични онтологии* с ръководители М. Хаджийски и В. Петров – двукратно през ноември 2007 и юни 2008 г.,
- Работна среща *Българските галерии през 21 в. - дигитализация и онлайн достъп*, Национална Художествена Галерия - София, 12-13 март 2008,
- Семинар *Европейски проект MINERVA eC*, организиран от Централната библиотека на Българската академия на науките, 26 февруари 2008 г.,

- Семинар по Компютърна лингвистика със студенти и преподаватели в Югозападния университет „Н. Рилски" – Благоевград, май 2007,
- Фирмен семинар *Обработка на Естествен език* в АПИС, юни 2007,
- Семинар в Университета на Роскилде – Дания, ноември 2000,
- Семинари на Катедрата по обработка на естествен език, Факултет по информатика на Хамбургския университет – многократно.

ПУБЛИКАЦИИ по ДИСЕРТАЦИЯТА (31 с 95 цитирания)

В списания и поредици, издадени в чужбина:

[AnBo96a] Angelova, G. and K. Bontcheva. *DB-MAT: Knowledge Acquisition, Processing and NL Generation using Conceptual Graphs*. In: Eklund, E., G. Ellis and G. Mann (Eds.) *Conceptual Structures: Knowledge Representation as Interlingua*, Proc. of the 4th Intern. Conference on Conceptual Structures (ICCS-96), Sydney, Australia, August 1996, Lecture Notes in Artificial Intelligence Vol. 1115, Springer, pp. 115-129.

[AnBo96b] Angelova, G. and K. Bontcheva. *DB-MAT: a NL Based Interface to Domain Knowledge.* In Ramsay, A. (Ed.), *Artificial Intelligence: Methodology, Systems, Applications*, Proc. 7th Int. Conference AIMSA-96), Sozopol, Bulgaria, September 1996, IOS Press, Vol. 35 in the series "Frontiers in AI and Applications", pp. 218-227.

[ADTB97] Angelova, G., S. Damyanova, K. Toutanova and K. Bontcheva. *Menu-Based Interface to Conceptual Graphs: the CGLex Approach.* In: Lukose, D., H. Delugach, M. Keeler, L. Searle and J. Sowa (Eds.), *Conceptual Structures: Fulfilling Peirce's Dream*, Proceedings of the *CGTools Workshop*, Seattle, USA, August 1997, Lecture Notes in Artificial Intelligence Vol. 1257, Springer, pp. 603-606.

[AnBo97a] Angelova, G. and K. Bontcheva. *Task-Dependent Aspects of Knowledge Acquisition: a Case Study in a Technical Domain.* In: Lukose, D., H. Delugach, M. Keeler, L. Searle and J. Sowa (Eds.), *Conceptual Structures: Fulfilling Peirce's Dream*, Proceedings of the 5th Int. Conf. on Conceptual Structures (ICCS-97), Seattle, USA, August 1997, Lecture Notes in Artificial Intelligence Vol. 1257, Springer, pp. 183-197.

[Ang98] Angelova, G. *Manual Acquisition of Uncountable Types in Closed Worlds.* In: Mugnier, M.-L. and M. Chein (Eds.), *Conceptual Structures: Theory, Tools and Applications,* Proceedings of the 6th Int. Conference on Conceptual Structures (ICCS-98), 1998, Springer, Lecture Notes in Artificial Intelligence Vol. 1453, pp. 351-358.

[AKvH98] Angelova, G., O. Kalaydjiev, and W. v. Hahn. *The gain of failures: Using side effects of anaphora resolution for term consistency checks.* In: Giunchiglia, F. (Ed.), *Artificial Intelligence: Methodology, Systems, Applications*, Proceedings of the 8th Int.

Conference AIMSA-98, Bulgaria, September 1998, Lecture Notes in Artificial Intelligence Vol. 1480, Springer, pp. 1-13.

[NeAn99] Nenkova, A. and G. Angelova. *User Modelling as an Application of Actors.* In: Tepfenhart, W. and W. Cyre (Eds.), *Conceptual Structures: Standards and Practices,* Proceedings of the 7th Intern. Conference on Conceptual Structures (ICCS-99), Virginia Tech, USA, July 1999, Springer, Lecture Notes in Artificial Intelligence 1640, Springer, pp. 83-89. JRC/ISI journal impact factor 0,530.

[BKNA00] Boytcheva, S., O. Kalaydjiev, A. Nenkova and G. Angelova. *Integration of Resources and Components in a Knowledge-Based Web-environment for Terminology Learning.* In: Cerri, S. A. and D. Dochev (Eds.) *Artificial Intelligence: Methodology, Systems, Applications*, Proc. of the 9th Int. Conf. AIMSA-2000, Springer, Lecture Notes in Artificial Intelligence 1904, pp. 210-220. JRC/ISI journal impact factor 0,253.

[BSA02] Boytcheva, S., A. Strupchanska and G. Angelova. *Processing Negation in NL Interfaces to Knowledge Bases.* In: Priss, U., D. Corbett, and G. Angelova (Eds.) *Conceptual Structures: Integration and Interfaces.* Proc. of the 10th Int. Conference on Conceptual Structures (ICCS-2002), Bulgaria, Borovets, July 2002, Springer, Lecture Notes in Artificial Intelligence 2393, pp. 137-150.

[AKS04] Angelova, G., O. Kalaydjiev and A. Strupchanska. *Domain Ontology as a Resource Providing Adaptivity in eLearning.* In: Meersman, R., Z. Tari and A. Corsaro (Eds.) Proc. *On the Move to Meaningful Internet Systems 2004*: OTM 2004 Confederated Conference and Workshops, *Workshop on Semantics, Ontologies and eLearning* (WOSE-04), Springer, Lecture Notes in Computer Science 3292, pp. 700-712. JRC/ISI journal impact factor 0,513.

[Ang05] Angelova, G. *Language Technologies Meet Ontology Acquisition.* In: Dau, F., M.-L. Mugnier, and G. Stumme (Eds.): *Conceptual Structures: Common Semantics for Sharing Knowledge,* Proceedings of the 13th Int. Conference on Conceptual Structures (ICCS-2005), Springer, Lecture Notes in Artificial Intelligence 3596, pp. 367-380. JRC/ISI journal impact factor 0,302.

[Ang09a] Angelova, G. *Efficient Computation with Conceptual Graphs.* In: Hitzler, P. and H. Schärfe (Eds.) *Conceptual Structures in Practice.* Chapman & Hall/CRC Studies in Informatics Series, Boca Raton FL, May 2009, ISBN 978-1-4200-6062-1, pp. 73-98.

В списания и поредици, издадени в България:

[AnMi08a] Angelova, G. and S. Mihov. *Conceptual Information Compression and Efficient Pattern Search.* Serdica Journal of Computing, ISSN 1312-6555, Vol. 2 (2008), pp. 369-402.

[Анг08б] Ангелова, Г. *Автоматична генерация на обяснения в техническа област.* Сп. Автоматика и информатика, издание на Съюза по автоматика и информатика "Джон Атанасов", София, 2008, ISSN 0861–7562, кн. 4/2008, стр. 13-16.

[Ang09b] Angelova, G. *Ontological Approach to Terminology Learning*. Доклади на БАН, 2009 (под печат). JRC/ISI 2007 journal impact factor 0,106 (2007).

Научни публикации в пълен текст в рецензирани сборници трудове на международни конгреси и конференции

Сборници трудове на конгреси и конференции, издадени в чужбина:

[vHAn94] v. Hahn, W. and G. Angelova. *Providing Factual Information in MAT.* Proc. of the International Conference *Machine Translation - Ten Years on,* Cranfield, United Kingdom, 1994, pp. 11/1-11/16.

[AnBo96c] Angelova, G. and K. Bontcheva. *NL Domain Explanations in Knowledge Based MAT.* In: Proc. 16th Int. Conference on Computational Linguistics COLING-96, Copenhagen, Denmark, August 1996, Vol. 2, pp.1016-1019.

[vHAn96] v. Hahn, W. and G. Angelova. *Combining Terminology, Lexical Semantics and Knowledge Representation in Machine Aided Translation.* In Galinski, C. and K.-D. Schmitz (Eds.), *Terminology and Knowledge Engineering*, Proc. of the 4th Int. Congress on TKE-96, Vienna, August 1996, INDEKS Verlag, Frankfurt/M., pp. 304-314.

[ANBN00] Angelova, G., A. Nenkova, S. Boycheva and T. Nikolov. *Conceptual graphs as a knowledge representation core in a complex language learning environment.* In: Stumme, G. (Ed.) *Working with Conceptual Structures: Contributions to ICCS 2000*, Proc. of the 8th International Conference on Conceptual Structures ICCS-2000, Darmstadt, Germany, August 2000, Shaker Verlag, pp. 45-58.

[BDA01] Boytcheva S., P. Dobrev and G. Angelova. *CGExtract: towards Extraction of Conceptual Graphs from Controlled English.* In: G. Mineau (Ed.), *Conceptual Structures: Extracting and Representing Semantics*, Contributions to ICCS-2001, the 9th Int. Conf. on Conceptual Structures, Stanford, California, August 2001, pp. 89-116. CEUR Workshop Proceedings, ISSN 1613-0073, vol. 41 (електронно издание http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-41/).

[ABKTNS02] Angelova G., S. Boytcheva, O. Kalaydjiev, S. Trausan-Matu, P. Nakov and A. Strupchanska. *Adaptivity in Web-based CALL.* In: Frank van Harmelen (Ed.), Proceedings of ECAI-2002, the 15th European Conference on AI, IOS Press, 2002, pp. 445-449.

[BVSYA04] Boytcheva S., I. Vitanova, A. Strupchanska, M. Yankova, and G. Angelova. *Towards the assessment of free learner's utterances in CALL.* In: Delmonte, R., P. Delcloque, and S. Tonelli (Eds.) *NLP and Speech Technologies in Advanced Language Learning Systems*, Proceedings of the InSTIL/ICALL 2004 Symposium on Computer Assisted Language Learning, Venice, 2004, pp. 187-190.

[ASKBV04] Angelova G., A. Strupchanska, O. Kalaydjiev, S. Boytcheva and I. Vitanova. *Terminological Grid and Free Text Repositories in Computer-Aided Teaching*

of Foreign Language Terminology. In Monachesi, Vertan, v. Hahn and Jekat (Eds.), Proc. "*Language Resources: Integration & Development in e-learning & in Teaching Computational Linguistics*", Int. Workshop at LREC-04, Lisbon, 2004, pp. 35-40.

[ASKYBV04] Angelova G., A. Strupchanska, O. Kalaydjiev, M. Yankova, S. Boytcheva and I. Vitanova. *Towards deeper understanding and personalisation in eCALL.* In Lemnitzer, L., Meurers, and Hinrichs (Eds.), Proceedings "*eLearning for Computational Linguistics and Computational Linguistics for eLearning*", Int. Workshop at COLING 2004, Geneva, 2004, pp. 45-52.

[AnMi08b] Angelova, G. and S. Mihov. *Finite State Automata and Simple Conceptual Graphs with Binary Conceptual Relations*. In: Eklund, P. and O. Haemmerlé (Eds.), Supplementary Proceedings of ICCS-2008, the 16th Int. Conference on Conceptual Structures, CEUR Workshop Proceedings 2008, Vol. 354, pp. 139-148 (електронно издание http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-354/).

Сборници трудове на конгреси и конференции, издадени в България:

[Ang95] Angelova, G. *Naive Lexicon or Cryptic Formalisms? User Support in Machine Aided Translation.* In: Mitkov, R., N. Nicolov and N. Nikolov (Eds.), *Recent Advances of Natural Language Processing* 1995, Proc. International Conference RANLP-95, Batak, Bulgaria, September 1995, pp. 286-292.

[AnBo97b] Angelova, G. and K. Bontcheva. *Refining Domain Ontologies for Flexible Explanation Generation.* In: Mitkov, R., N. Nicolov and N. Nikolov (Eds.), *Recent Advances of Natural Language Processing* 1997, Proc. International Conference RANLP-97, Batak, Bulgaria, September 1997, pp. 258-263.

[Ang00] Angelova, G. *Ontologies for Natural Language Processing Applications.* Доклад по покана на OntoLext 2000. In: K. Simov and A. Kiryakov (Eds.), *Ontologies and Lexical Knowledge Bases*, Proceedings of OntoLex 2000, Sozopol, Bulgaria, 8-10 September 2000, pp. 1-15.

[BKSA01] Angelova G., S. Boytcheva, O. Kalaydjiev and A. Strupchanska. *Between language correctness and domain knowledge in CALL.* In: Mitkov, R., G. Angelova, K. Bontcheva, N. Nicolov and N. Nikolov (Eds.), *Recent Advances in Natural Language Processing* 2001, Proceedings of the EuroConference RANLP-2001, Tzigov Chark, Bulgaria, pp. 40-46.

[KaAn02] Kalaydjiev, O. and G. Angelova. *Adaptive Hypermedia in eLearning.* In Turlakov, H. and L. Boyanov (eds.) *Next Generation Network Computing*, Proc. Next Generation Network Workshop, Rousse, Bulgaria, October 2002, pp. 87-98.

[Анг08а] Ангелова, Г. *Езиковите технологии днес и утре*. Доклад по покана на Пролетната Конференция на СМБ, Боровец, април 2008. В: Сборник трудове на 37-мата Пролетна конференция на Съюза на Математиците в България, ISSN 1313-3330, София, 2008, стр. 68-85.

ЛИТЕРАТУРА

[Allen95] Allen, J. *Natural Language Understanding*. The Benjamin/Cummings Publ. Co., 1994.

[Ang95] Angelova, G. *Naive Lexicon or Cryptic Formalisms? User Support in Machine Aided Translation.* In: Mitkov, R., N. Nicolov and N. Nikolov (Eds.), Proc. International Conference Recent Advances of Natural Language Processing (RANLP'95), Batak, Bulgaria, September 1995, pp. 286 - 292.

[Ang98] Angelova, G. *Manual Acquisition of Uncountable Types in Closed Worlds.* In: Mugnier, M.-L. and M. Chein (Eds.), *Conceptual Structures: Theory, Tools and Applications,* Proceedings of the 6th Int. Conference on Conceptual Structures (ICCS-98), Montpellier, France, August 1998, Lecture Notes in Artificial Intelligence Vol. 1453, pp. 351-358.

[Ang00] Angelova, G. *Ontologies for Natural Language Processing Applications.* Доклад по покана на OntoLext 2000. In: K. Simov and A. Kiryakov (Eds.), *Ontologies and Lexical Knowledge Bases*, Proceedings of OntoLex 2000, Sozopol, Bulgaria, 8-10 September 2000, pp. 1-15.

[Ang05] Angelova, G. *Language Technologies Meet Ontology Acquisition*. In: Dau, F., M.-L. Mugnier, and G. Stumme (Eds.): *Conceptual Structures: Common Semantics for Sharing Knowledge,* Proc. of the 13th Int. Conference on Conceptual Structures (ICCS-2005), Springer, Lecture Notes in Artificial Intelligence 3596, pp. 367-380.

[Ang09a] Angelova, G. *Efficient Computation with Conceptual Graphs*. In: Hitzler, P. and H. Schärfe (Eds.) *Conceptual Structures in Practice*. Chapman & Hall/CRC Studies in Informatics Series, Boca Raton FL, 2009, ISBN 978-1-4200-6062-1, pp. 73-98.

[Ang09b] Angelova, G. *Ontological Approach to Terminology Learning*. Доклади на БАН, 2009 (под печат).

[AnBo96a] Angelova, G. and K. Bontcheva. *DB-MAT: Knowledge Acquisition, Processing and NL Generation using Conceptual Graphs*. In: Eklund, E., G. Ellis and G. Mann (Eds.) *Conceptual Structures: Knowledge Representation as Interlingua*, Proc. of the 4th Intern. Conference on Conceptual Structures (ICCS-96), Sydney, Australia, August 1996, Lecture Notes in Artificial Intelligence Vol. 1115, pp. 115-129.

[AnBo96b] Angelova, G. and K. Bontcheva. *DB-MAT: a NL Based Interface to Domain Knowledge.* In Ramsay, A. (Ed.), *Artificial Intelligence: Methodology, Systems, Applications*, Proc. 7th Int. Conference AIMSA-96), Sozopol, Bulgaria, September 1996, IOS Press, Vol. 35 in the series "Frontiers in AI and Applications", pp. 218-227.

[AnBo96c] Angelova, G. and K. Bontcheva. *NL Domain Explanations in Knowledge Based MAT.* In: Proc. 16th International Conference on Computational Linguistics COLING-96, Copenhagen, Denmark, August 1996, Vol. 2, pp.1016-1019.

[AnBo97a] Angelova, G. and K. Bontcheva. *Task-Dependent Aspects of Knowledge Acquisition: a Case Study in a Technical Domain.* In: Lukose, D., H. Delugach, M. Keeler, L. Searle and J. Sowa (Eds.), *Conceptual Structures: Fulfilling Peirce's Dream*, Proceedings of the 5th Int. Conference on Conceptual Structures (ICCS-97), Seattle, USA, August 1997, Lecture Notes in Artificial Intelligence Vol. 1257, pp. 183-197.

[AnBo97b] Angelova, G. and K. Bontcheva. *Refining Domain Ontologies for Flexible Explanation Generation.* In: Mitkov, R., N. Nicolov and N. Nikolov (Eds.), Proc. of the Int. Conference "Recent Advances of Natural Language Processing (RANLP'97)", Batak, Bulgaria, September 1997, pp. 258-263.

[AgDo08] Agre G. and D. Dochev. An *Approach to Technology Enhanced Learning by Application of Semantic Web Services*. "Cybernetics and Information Technologies", Vol. 8 (2008), № 3, pp. 60-72.

[AhDa90] Ahmad, K., A. Davies et al. *A Methodology for Building Multilingual Termbases and Special-Purpose Lexica*. Technical Report of the project TWB (ESPRIT project 2315), University of Surrey 1990.

[AnMi08a] Angelova, G. and S. Mihov. *Conceptual Information Compression and Efficient Pattern Search*. Serdica Journal of Computing Vo. 2, 2008, pp. 369-402.

[AnMi08b] Angelova G. and S. Mihov. *Finite State Automata and Simple Conceptual Graphs with Binary Conceptual Relations*. In: P. Eklund, O. Haemmerle (Eds), Supplementary Proceedings of the 16th Int. Conf. on Conceptual Structures (ICCS'08), CEUR Workshop Proceedings 2008, ISSN 1613-0073, 139-148..

[ABKTNS02] Angelova, G., Sv. Boytcheva, O. Kalaydjiev, S. Trausan-Matu, P. Nakov, and A. Strupchanska. *Adaptivity in Web-Based CALL*. In: Frank van Harmelen (Ed.), Proc. of ECAI-2002, the European Conference on AI, IOS Press, 2002, pp. 445-449.

[ABMHS95] Arnold, D., L. Balkan, S. Meijer, R. Humphreys, and L. Sadler. *Input: Controlled Languages*. In: *Machine Translation - an Introductory Guide*, 1995, достъпна на http://www.essex.ac.uk/linguistics/clmt/MTbook/HTML/book.html, последно посещение 3 април 2009.

[ADTB97] Angelova, G., S. Damyanova, K. Toutanova, and K. Bontcheva. *Menu-Based Interface to Conceptual Graphs: the CGLex Approach*. In: Lukose, D., H. Delugach, M. Keeler, L. Searle and J. Sowa (Eds.), *Conceptual Structures: Fulfilling Peirce's Dream*, Proceedings of the *CGTools Workshop*, Seattle, USA, August 1997, Lecture Notes in Artificial Intelligence Vol. 1257, pp. 603-606.

[AKS04] Angelova, G., O. Kalaydjiev and A. Strupchanska. *Domain Ontology as a Resource Providing Adaptivity in eLearning.* In: Meersman, R., Z. Tari and A. Corsaro (Eds.) Proc. *On the Move to Meaningful Internet Systems 2004*: OTM 2004 Confederated Conference and Workshops, *Workshop on Semantics, Ontologies and eLearning* (WOSE-04), Springer, Lecture Notes in Computer Science 3292, pp. 700-712.

[AKvH98] Angelova, G., Kalaydjiev, O., and v. Hahn, W. *The gain of failures: Using side effects of anaphora resolution for term consistency checks.* In: Giunchiglia, F. (Ed.), *Artificial Intelligence: Methodology, Systems, Applications*, Proceedings of the 8th Int. Conference AIMSA-98, Bulgaria, September 1998, Lecture Notes in Artificial Intelligence Vol. 1480, pp. 1 - 13.

[ANBN00] Angelova, G., A. Nenkova, S. Boycheva and T. Nikolov. *Conceptual graphs as a knowledge representation core in a complex language learning environment.* In: Stumme, G. (Ed.) *Working with Conceptual Structures: Contributions to ICCS 2000*, Proc. of the 8th International Conference on Conceptual Structures ICCS-2000, Darmstadt, Germany, August 2000, Shaker Verlag, pp. 45-58.

[ASKBV04] Angelova G., A. Strupchanska, O. Kalaydjiev, Sv. Boytcheva and I. Vitanova. *Terminological Grid and Free Text Repositories in Computer-Aided Teaching of Foreign Language Terminology.* In Monachesi, Vertan, v. Hahn and Jekat (Eds.), Proc. Workshop "*Language Resources: Integration & Development in e-learning & in Teaching Computational Linguistics*", organised at LREC 2004, Lisbon, pp. 35-40.

[ASKYBV04] Angelova G., A. Strupchanska, O. Kalaydjiev, M. Yankova, S. Boytcheva and I. Vitanova. *Towards deeper understanding and personalisation in eCALL.* In Lemnitzer, L., Meurers, and Hinrichs (Eds.), Proc. "*eLearning for Computational Linguistics and Computational Linguistics for eLearning*", Workshop COLING 2004, Geneva, pp.45-52.

[Bag05] Baget, J.-F. *RDF Entailment as a Graph Homomorphism.* In: Gil, Y., E. Motta, V. R. Benjamins, and M. A. Musen (Eds.). The Semantic Web - Proc. 4th International Semantic Web Conference, ISWC 2005, Galway, Ireland, 2005, Lecture Notes in Computer Science 3729, Springer 2005, pp. 82-96.

[Boy02] Boytcheva, Sv. *ILP techniques for free-text input processing.* In Proc. of AIMSA-2002, 10th Int. Conf. on Artificial intelligence: Methodology, Systems and Applications, Lecture Notes on Artificial Intelligence 2443, Springer 2002, pp. 101–110.

[Bru98] Brusilovsky, P. *Methods and techniques of adaptive hypermedia.* In: P. Brusilovsky, A. Kobsa and J. Vassileva (eds.): Adaptive Hypertext and Hypermedia, 1998, pp. 1–43.

[Bru04] Brusilovsky, P. *Adaptive Educational Hypermedia: From generation to generation.* Proc. of 4th Hellenic Conference on Information and Communication Technologies in Education, Athens, Greece, 2004, стр. 19–33.

[BDA01] Boytcheva Sv., P. Dobrev and G. Angelova. *CGExtract: towards Extraction of Conceptual Graphs from Controlled English.* In: G. Mineau (Ed.), *Conceptual Structures: Extracting and Representing Semantics*, Contributions to ICCS-2001, the 9th Int. Conference on Conceptual Structures, Stanford, California, August 2001, pp. 89-116. CEUR Workshop Proceedings, vol. 41, достъпен на http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-41/, ISSN 1613-0073, последно посещение 3 април 2009.

[BoEu95] Boynov, N. and L. Euler. *The Structure of the Lexicon and its Support in DB-MAT.* Project DB-MAT, Technical report 1-95, May 1995.

[BHR95] Bateman, J., R. Henschel and F. Rinaldi. *The generalized upper model 2.0.* Technical report, GMD/Institut für Integrierte Publikations- und Informationssysteme, Darmstadt, 1995.

[BKNA00] Boytcheva, Sv., O. Kalaydjiev, A. Nenkova and G. Angelova. *Integration of Resources and Components in a Knowledge-Based Web-environment for Terminology Learning.* In: Cerri, S. A. and D. Dochev (Eds.) *Artificial Intelligence: Methodology, Systems, Applications*, Proc. of the 9th Int. Conference AIMSA-2000, Bulgaria, Springer, Lecture Notes in Artificial Intelligence 1904, pp. 210-220.

[BKSA01] Boytcheva, S., O. Kalaydjiev, A. Strupchanska, and G. Angelova. *Between Language Correctness and Domain Knowedge in CALL.* In: Angelova, Bontcheva, Mitkov, Nikolov, and Nikolov (Eds.) *Recent Advances in Natural Language Processing* 2001, Proc. EuroConference RANLP-2001, Tzigov Chark, Bulgaria, 2001, pp. 40-46.

[BLHL01] Berners-Lee, T., J. Hendler and O. Lassila. *The Semantic Web. Scientific American Magazine.* May 2001

[BaMu02] Baget, J.-F. and M.-L. Mugnier. *Extensions of Simple Conceptual Graphs: the Complexity of Rules and Constraints.* Journal of AI Research, vol. 16, 2002, pp. 425-465.

[BrMi07] Brusilovsky, P. and E. Millán. *User models for adaptive hypermedia and adaptive educational systems.* In: P. Brusilovsky, A. Kobsa and W. Neidl (eds.): *The Adaptive Web: Methods and Strategies of Web Personalization.* Lecture Notes in Computer Science, Vol. 4321, Springer 2007, pp. 3-53.

[BSA02] Boytcheva, S., A. Strupchanska and G. Angelova. *Processing Negation in NL Interfaces to Knowledge Bases.* In: Priss, U., D. Corbett, and G. Angelova (Eds.) *Conceptual Structures: Integration and Interfaces.* Proc. of the 10th Int. Conference on Conceptual Structures (ICCS-2002), Bulgaria, Borovets, July 2002, Springer, Lecture Notes in Artificial Intelligence 2393, pp. 137-150.

[BVSYA04] Boytcheva S., I. Vitanova, A. Strupchanska, M. Yankova, and G. Angelova. *Towards the assessment of free learner's utterances in CALL.* In: Delmonte, R., P. Delcloque, and S. Tonelli (Eds.) *NLP and Speech Technologies in Advanced Language*

Learning Systems, Proceedings of the InSTIL/ICALL 2004 Symposium on Computer Assisted Language Learning, Venice, 2004, pp. 187-190.

[Cun99] Cunnigham, H. *Information extraction – an user guide*. Research Memo CS-99-07, Computer Science Department, University of Sheffield, 1999 (http://www.dcs.shef.ac.uk/~hamish/IE), последно посещение 5 май 2009.

[CrCo04] Croitoru, M. and E. Compatangelo. *A combinatorial approach to conceptual graph projection checking*, In: Proceedings of the 24th Int. Conference of the British Computer Society, Special Group on AI (SGAI'2004), pp. 130-143.

[CCH02] Callan, J., B. Croft and E. Hovy. *The Energy Data Collection (EDC) project: Deep Focus on Hydra-Headed Metadata*. Електронна версия на http://www.digitalgovernment.org/news/stories/2002/images/metadatafinal.pdf, последно посещение 6 май 2009.

[CDMTM01] Cerri, S.A., S. Dikareva, D. Maraschi, and S. Trausan-Matu. *Web Server Based Architectures for Language Learning: LARFLAST Agents generating CALL Dialogues*, In: Proc. Int. Conference on Computer Assisted Language Learning, Univ. of Exeter, UK, 2001.

[CL-ISO07] Common Logic, http://www.common-logic.org, последно посещение 10 април 2009.

[ChMu92] Chein, M. and M.-L. Mugnier. *Conceptual Graphs: fundamental notions*. Revue d'Intelligence Artificielle, Vol. 6, No.4, 1992, pp. 365-406.

[ChTo03] Chakarova, I. and G. Totkov. *On transferring of traditional learning materials into virtual learning environment*. In: Rachev and Smrikarov (Eds.) *Proc. CompSysTech 2003: E-Learning,* Rousse, Bulgaria, June 2003. ACM Press, New York, pp. 611-616.

[Dal88] Dale, R. *Generating Referring Expressions in a Domain of Objects and Processes*. Ph.D. Thesis, University of Edinburgh, 1988.

[Dau09] Dau, F. *Formal Logic with Conceptual Graphs*. Book chapter in P. Hitzler and H. Scharfe (Eds), *Conceptual Structures in Practice*, Chapman & Hall/CRC Studies in Informatics Series, ISBN 978-1-4200-6062-1, 2009.

[Dim01] Dimitrova, V. *Interactive Open Learner Modelling*, PhD in Artificial Intelligence in Education, Computer Based Learning Unit, Leeds University, 2001.

[Dim02] Dimitrova, V. *STyLE-OLM Interactive Open Learner Modelling*. International Journal of Artificial Intelligence in Education, 2002, Vol. 13, pp. 54-69.

[DMWW00] Daciuk, J., S. Mihov, B. Watson, and R. Watson, *Incremental Construction of Minimal Acyclic Finite State Automata,* Journal of Computational Linguistics, Vol. 26, Issue 1, 2000, pp. 3-16.

[DoSt01] Dochev D. and K. Staykova. *A Multilingual System for Automatic Generation of Technical Manual Texts*. Proc. of the Int. Conf. on Computer Systems and Technologies CompSysTech'2001, Sofia, 21-22 June 2001, pp. II.14.1-5, 2001.

[DoTo00] Dobrev, P. and K. Toutanova. *CGWorld – a Web-based workbench for conceptual graphs management and applications*. Proc. ICCS-2000, Shaker Verlag, Aachen 2000, ISSN 0945-0807, pp. 243–257.

[DRDK01] Dikarieva, S., Ronginsky, V., Dikariev, E. and H. Kulikova, *Strategies for Interaction between Learners and Systems and Multilingual Terminology Resources in STyLE*, Larflast project, report 4.3, 2001, delivered to the European Commission.

[DSB01] Dimitrova, V., J. Self, and P. Brna. *Applying Interactive Open Learner Models to Learning Technical Terminology*. In Bauer, Gmytrasiewicz, Vassileva (Eds.), Proc. 8th Int. Conference on User Modelling 2001, Sonthofen, Germany. Published by Springer, 2001, Lecture Notes in Artificial Intelligence Vol. 2109, pp. 148-157.

[DST01] Dobrev, P., A. Strupchanska, and K. Toutanova. *CGWorld-2001 - new features and new directions*. In: ICCS-2001 *CGTools Workshop*, July 2001, Stanford University, Електронен сборник трудове на http://www.cs.nmsu.edu/~hdp/CGTools/proceedings/, последно посещение 18 април 2009.

[EngQ00] *English Query*, http://msdn.microsoft.com/en-us/library/aa198281(SQL.80).aspx, последно посещение 18 април 2009.

[ExDi78] *Explanatory dictionary of computing machinery and data processing. Russian-English-German-French*. Москва, Издательство „Руский язык", 1978.

[EMe95] Eck, K. and I. Meyer. *Bringing Aristotle into the 20th Century: A Tool and Approach for Constructing Definitions within a Terminological Knowledge Base*. In: R. A. Strehlow and S. E. Wright (Eds.), *Standardizing and Harmonizing Terminology: Theory and Practice*. Philadelphia, American Society for Testing and Materials, 1995, pp. 83-101.

[ETB05] eContent EUROTERMBANK Project, *Deliverable D1.2 Final Methodology Report*, December 2005, достъпно на *http://project.eurotermbank.com/uploads/D1.2 Final Methodology Report.pdf*, последно посещение 9 април 2009.

[FHA90] Fulford, H., M. Hoege and K. Ahmad. *Translator's Workbench: User Requirements Study*. Technical Report of the project TWB (ESPRIT project 2315), University of Surrey 1990.

[FLDC86] Fagrues, J., M.C. Landau, A. Dugourd, and L. Catach. *Conceptual Graphs for Semantics and Knowledge Processing*. In: IBM J. Res. and Develop. Vol. 30 (1), January 1986, pp. 70-79.

[FaNé98] Faure, D. and C. Nédellec. *ASIUM: Learning subcategorization frames and restrictions of selection*. In Kodratoff, Y. (Ed.), Proceedings of the 10th Conference on Machine Learning (ECML 98), Workshop on Text Mining, Chemnitz, Germany, April 1998.

[FuSch02] Fuchs, N. E. and U. Schwertel. *Reasoning in Attempto Controlled English*. Technical Report, 2002. Достъпен на http://www.ifi.uzh.ch/rerg/publications/. Последно посещение 17 април 2009.

[GaBu96] Galinski, C. and G. Budin. *Terminology*, Section 12.5 in Cole, R., Mariani, J., Uszkoreit, H., Zaenen, A. and Zue, V. (Eds.) *Survey of the State of the Art in Human Language Technology*, 1996, pp. 371-374. Достъпно на http://www.lt-world.org/HLT_Survey/master.pdf, последно посещение на 2 април 2009.

[Gru93] Gruber, T. *A translation approach to portable ontology specifications*. Knowledge Acquisition 5, 1993, pp. 199-199.

[Gua92] Guarino, N. *Concepts, Attributes, and Arbitrary Relations*. Journal of Data and Knowledge Engineering Vol. 8, No. 3, 1992, pp. 249-261.

[Gua97] Guarino, N. *Some Organizing Principles for a Unified Top-Level Ontology*. In: Working Notes AAAI Spring Symposium on Ontological Engineering, Stanford 1997.

[GaKn02] Gamper, J. and J. Knapp. *Review of intelligent CALL systems*. Journal of Computer Assisted Language Learning Vol. 15, No. 4, 2002, pp. 329-342.

[GoNi91] Goodman, K. and S. Nirenburg (Eds.). *The KBMT project: A Case Study in Knowledge-Based Machine Translation*. Morgan Kaufmann Publishers, Inc., 1991.

[GPMM05] Gómez-Pérez, A. and D. Manzano-Macho. *An overview of methods and tools for ontology learning from texts*. The Knowledge Engineering Review, Vol. 19(3), pp. 187 – 212.

[GSP08] Govedarova N., S. Stoyanov and I. Popchev. *Hybrid Ontology and CBR-Based Search in BULCHINO Catalogue*. In Proc. of the International Conference "Informatics in the Scientific Knowledge 2008", 26-28 June 2008, Varna, pp. 205-216

[Hay85] Hayes, P. *The Second Naïve Physics Manifesto*. In: Brachman, R. and H. Levesque (Eds.) Readings in Knowledge representation, Morgan Kaufmann Publishers 1985, pp. 468-485.

[Hea92] Hearst, M. *Automatic Acquisition of Hyponyms from Large Text Corpora*, in Proc. of the 14th Int. Conf. on Computational Linguistics COLING-1992, pp. 539-545.

[Hea09] Hearst, M. *Search User Interfaces*. Cambridge University Press, 2009, ISBN 9780521113793, достъпна на http://searchuserinterfaces.com, последно посещение 26 юни 2009.

[Hob85] Hobbs, J. *Ontological Promisquity*. Proc. 23rd Annual Meeting of ACL, Chicago, IL, July 1985, pp. 61-69.

[Hob95] Hobbs, J. *Sketch of an Ontology Underlying the Way We Talk about the World*. International Journal of Human-Computer Studies Vol. 43, 1995, pp. 819-830.

[Hop71] Hopcroft, J. *An n log n algorithm for minimizing states in a finite automaton*. In: Kohavi, Z. (Ed.), The Theory of Machines and Computation, Academic Press, 1971, pp. 189-196.

[HMU83] Hopcroft, J., Motwani R. and J. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, Reading, MA, 1983.

[HaPe08] Hadjiski, M. and V. Petrov. (Eds.) *Ontologies - Philosophical and Technological Problems*. Sofia, "Prof. Marin Drinov Publishing House", 2008.

[HaSt03] Handschuh, S. and S. Staab. *CREAM: CREAting Metadata for the Semantic Web*. Computer Networks: The International Journal of Computer and Telecommunications Networking, Volume 42 , Issue 5, 2003, pp. 579 – 598.

[ISO96] ISO 860 *Terminology work – Harmonization of concepts and terms*, TC 37, 1996.

[InvWorld] *Financial Glossary at Investor Words*, http://www.investorwords.com, последно посещение 14 май 2009.

[Kal01] Kalaydjiev, O. *An Intelligent Pedagogical Agent in CALL*. In Bauer, Gmytrasiewicz, Vassileva (Eds.), Proceedings 8th International Conference on User Modelling 2001, Sonthofen, Germany, Doctoral Symposium, Lecture Notes in Artificial Intelligence Vol. 2109, Springer 2001, pp. 283–285.

[Kal06] Kalaydjiev, O. *Using Ontology for Proper Planning in Computer Assisted Language Learning*. In: Angelova, G. ., K. Boyanov, K. Fillyov and Vl. Getov (Eds.). *J. V. Atanasoff Information Days 2006*. Proc. of the Young Researchers Session, within the frame of IEEE 2006 J.V. Atanasoff Int. Symposium on Modern Computing. Sofia, October 2006, pp. 61-66.

[KaAn02] Kalaydjiev, O. and G. Angelova. *Adaptive Hypermedia in eLearning*. In Turlakov, H. and L. Boyanov (Eds.) *Next Generation Network Computing*, Proc. Next Generation Network Workshop, Rousse, Bulgaria, October 2002, pp. 87-98.

[KoDo94] Koychev, I. and M. Dobreva. *Application of Learning by Examples Method to Dating Medieval Bulgarian Manuscripts*. International Journal of Information Theory and Applications, vol. 2(5), 1994.

[KKR91] Kittredge, R., T. Korelsky, and O. Rambow. *On the need for domain communication knowledge*. Computational Intelligence, Vol. 7, Issue 4, November 1991, pp. 305-314.

[KiWi90] Kieselbach, C. and H. Wienschiers. *Studie zur Anforderungsspezifikation einer computergestützten Übersetzerumgebung*. Studienarbeit, Universität Hamburg. 1990.

[Leh96] Lehmann, F. *Big Posets of Participating and Thematic Roles*. Invited lecture at ICCS-96, In: Eklund, P., G. Ellis, and G. Mann (eds.), Conceptual Structures: Knowledge representation as Interlingua, Lecture Notes in Artificial Intelligence 1115, Springer, 1996, pp. 50-74.

[LSOMM08] Lemnitzer, L., K. Simov, P. Osenova, E. Mossel and P. Monachesi. *Using a domain-ontology and semantic search in an eLearning environment*. In: Innovative Techniques in Instruction Technology, E-learning, E-assessment, and Education. Springer Netherlands, (2008), 279-284.

[LVKSECM07] Lemnitzer, L., C. Vertan, A. Killing, K. Simov, D. Evans, D. Cristea, P. Monachesi, *Improving the search for learning objects with keywords and ontologies,* In: Creating New Learning Experiences on a Global Scale. Second European Conference on Technology Enhanced Learning, Lecture Notes in Computer Science Vol. 4753, 2007, 202-216.

[McK85] McKeown, K. *Using discourse strategies and focus constraints to generate natural language text*. Cambridge University Press, 1985.

[Mey94] Meyer, I. *Helping Terminologists Do Knowledge Engineering: Some Linguistic Strategies and Computer Aids. Actualité Terminologique*, December 1994, pp. 6-10.

[Mit90a] Mitkov R. *Generating explanations of geometrical objects*. Computers and Artificial Intelligence, Vol. 9, No.6, 1990.

[Mit90b] Mitkov R. *A text generation system for explaining concepts in geometry*. Proceedings of the COLING '90 Conference, Helsinki 1990, pp. 425-427.

[Mol92] Molhova, J. *The noun – a Contrastive English – Bulgarian Study*. Издателство на Софийския Университет „Св. Кл. Охридски', София, 1992.

[Mug95] Mugnier, M.-L. *On Generalization / Specialization for Conceptual Graphs*. Journal of Experimental and Theoretical Computer Science, Vol. 7, 1995, pp. 325-344. (публикувана също като Research Report LIRMM 93-003, January 1993).

[MaCe01] Maraschi, D. and St. Cerri. *Position paper of the Larflast team (Monpellier)*. Technical report 5.3, Larflast project, 2001, delivered to the European Commission.

[MuCh92] Mugnier, M.-L. and M. Chein. *Polynomial Algorithms for Projection and Matching,* In: Pfeiffer, H. and T. Nagle (Eds.) Conceptual Structures: Theory and

Implementation, Proceedings of the 7th Annual Workshop on Conceptual Graphs (AWCG'92). Springer, LNAI 754, 1992, pp. 239-251.

[MLS06] Monachesi, P., L. Lemnitzer, and K. Simov, *Language Technology for eLearning*, In the Proc. of the First European Conf. on Technology Enhanced Learning (EC-TEL-2006), Springer Lecture Notes in Computer Science, Volume 4227, 2007, 667-672.

[MiSi89] Mitkov R. and G. Simeonova. *Generating sentences and discourse: Models and computer programs*. In: Mathematics and mathematical education, Proceedings of the Spring Conference of the Union of Bulgarian Mathematicians, Sofia, 1989.

[MSBE92] Meyer, I., D. Skuce, L. Bowker, and K. Eck. *Towards a new generation of terminological resources: an experiment in building a terminological knowledge base*. In: Proc. of the 14th conf. on Computational linguistics COLING-92, Volume 3, pp. 956-960.

[MaTh88] Mann, W. and S. Thompson. *Rhetorical Structure Theory: Toward a functional theory of text organization*. Text, 1988, Vol. 8, No. 3, pp. 243-281.

[MathWorld] Weisstein, E. *Bell number*. From MathWorld, A Wolfram Web Resource. http://mathworld.wolfram.com/BellNumber.html, последно посещение 20 април 2009.

[Nak00a] Nakov, Pr. *Getting better results with latent semantic indexing*. Students Presentations at ESSLLI-2000, Birmingham, UK, 2000, pp. 156-166.

[Nak00b] Nakov, P. *Web Personalisation Using Extended Boolean Operations with Latent Semantic Indexing*. Proc. AIMSA-2000, Varna, Bulgaria, September 2000, Lecture Notes in AI 1904, Springer, pp. 189-198.

[Ner02] Nerbonne, J. *Computer-Assisted Language Learning and Natural Language Processing*. In: Mitkov, R. (Ed.) Handbook of Computational Linguistics, Oxford University Press, 2002, pp. 670–698.

[Nik08] Nikolova, I. *Language Technologies for Instructional Resources in Bulgarian*, In proceedings of 13th Student Session at ESSLLI, 4-15 August 2008, Hamburg, Germany. Electronic proceedings http://staff.science.uva.nl/~kbalogh/StuS13/StuS13_Proceedings.pdf последно посещение 20 април 2009.

[NeAn99] Nenkova, A. and G. Angelova. *User Modelling as an Application of Actors*. In: Tepfenhart, W. and W. Cyre (Eds.), *Conceptual Structures: Standards and Practices*, Proceedings of the 7th Intern. Conference on Conceptual Structures (ICCS-99), Virginia Tech, USA, July 1999, Springer, Lecture Notes in Artificial Intelligence 1640, pp. 83-89.

[NMB96] Nirenburg, S., K. Mahesh, and S. Beale. *Measuring Semantic Coverage*. In Proc. COLING-96, Copenhagen, Denmark 1996,

[NMR95] Nikolov, N., Ch. Mellish, G. Ritchie. *Sentence Generation from Conceptual Graphs*. In: G. Ellis, R. Levinson, W. Rich, J. Sowa (eds.), Conceptual Structures: Applications, Implementation and Theory. Proc. ICCS'95, LNAI 954, pp. 74-88.
[NKPPR96] Nerbonne, J., L.Karttunen, G.Proczeky, E. Paskaleva and T.Roosmaa. *Reading more in Foreign Languages*, Proc. Fifth Applied Natural Language Processing Conference, April 1997, Washington, ACL.

[NLPLC04] *Natural Language Processing in Medical Coding*. White paper of Language and Computing (April 2004), http://www.landcglobal.com, последно посещение 20 април 2009.

[NiNi96] Nikolov, R. and I. Nikolova. *A Virtual Environment for Distance Education and Training*, IFIP WG3.6 Conference, Vienna, September 1996.

[NPP06] Nisheva-Pavlova, M., P. Pavlov. *On the Applicability of Protege/OWL in Building Software Tools for Intelligent Search in Digitized Collections of Manuscripts*. Review of the National Center for Digitization, Belgrade, Vol. 9 (2006), pp. 13-17.

[OnSelect] *OntoSelect Ontology Library*, http://olp.dfki.de/ontoselect/, последно посещение 30 март 2009.

[Pan06] Paneva D. *Use of Ontology-based Student Model in Semantic-oriented Access to the Knowledge in Digital Libraries*, In the Proceedings of the Fourth HUBUSKA Open Workshop *"Semantic Web and Knowledge Technologies Applications"*, 12 September, 2006, Varna, Bulgaria, pp. 31-41. Достъпна на http://mdl.cc.bas.bg/dessi/Desislava %20Paneva_files/publications.html, последно посещение 20 април 2009.

[Pet08] Petrov, V. *Process Ontology as an Expression of the Idea of Dynamism in Philosophy*. In: Hadjiski, M. and V. Petrov (eds.), *Ontologies – Philosophical and Technological Problems*. Sofia, Prof. Marin Drinov Academic Publishing House, 2008, pp. 67-77.

[Poe00] Poesio, M. Semantic Analysis. In Dale, R., H. Moisl and H. Somers (Eds.), *A Handbook of Natural Language Processing*, Marcel Dekker, Inc., 2000, Chapter 6, pp. 93-122.

[Pol08] Poli, R. *SOfia Lectures on Ontology*. In: Hadjiski, M. and V. Petrov (eds.), *Ontologies – Philosophical and Technological Problems*. Sofia, Prof. Marin Drinov Academic Publishing House, 2008, pp. 7-66.

[PDPD07] Pavlova-Draganova, L., D. Paneva and L. Draganov. *Knowledge Technologies for Description of the Semantics of the Bulgarian Iconographical Artefacts*, In Proc. of 5th HUBUSKA Open Workshop *"Knowledge Technologies and Applications"*, 2007, Slovakia. На http://mdl.cc.bas.bg/dessi/Desislava%20Paneva_files/publications.html, последно посещение 20 април 2009.

[PEK95] Petermann, H., Euler, L. and K. Bontcheva. *CGPro - A PROLOG Implementation of Conceptual Graphs.* Technical Report FBI-HH-M-251/95, University of Hamburg, October 1995.

[PKOMK04] Popov, B., Kiryakov, A., Ognyanoff, D., Manov, D. and Kirilov, A. *KIM – a Semantic Platform for Information Extraction and Retrieval.* Journal of Natural Language Engineering 10 (3/4), 2004, pp. 375-392.

[PaMi98] Paskaleva, E. and S. Mihov. *Second Language Acquisition from Aligned Corpora*, Language Teaching and Language Technology, Sake Jager, John A. Nerbonne, A. J. van Essen (Eds.), Swets & Zeitlinger Lisse, The Netherlands, 1998, pp. 43 – 52.

[PaNP06] Pavlov, P. and M. Nisheva-Pavlova. *Knowledge-based Search in Collections of Digitized Manuscripts: First Results.* Proc. of the 10th ICCC Int. Conf. on Electronic Publishing (Bansko, 14-16 June 2006), FOI-COMMERCE, Sofia, 2006, pp. 27-35. Достъпна на http://mdl.cc.bas.bg/dessi/Desislava%20Paneva_files/publications.html, последно посещение 20 април 2009.

[PaPa06] Pavlov R. and D. Paneva. *Personalized and Adaptive Learning – Approaches and Solutions*, In the Proceedings of the CHIRON Open Workshop *"Visions of Ubiquitous Learning"*, 20 June, 2006, Stockholm, Sweden, pp. 2-13.

[PRL07] Paneva, D., K. Rangochev and D. Luchev. *Knowledge Technologies for Description of the Semantics of the Bulgarian Folklore Heritage*, In: Proc. of the 5th Int. Conference "Information Research and Applications" – i.Tech 2007 (ITA 2007 - Xth Joint Int. Scientific Events on Informatics), 2007, Varna, Bulgaria, vol. 1, pp. 19-25. Достъпна на http://mdl.cc.bas.bg/dessi/Desislava%20Paneva_files/publications.html, последно посещение 20 април 2009.

[Ram05] Ramsay, A. *PrAgmatics=ReAsoning about the Speaker's InTEnsions*, Parasite Manual, http://www.informatics.manchester.ac.uk/~allan/PARASITE/manual.pdf, 2005, последно посещение 17 април 2009.

[Ram95] Ramsay, A., *Theorem Proving for Intensional Logic*, Journal of Automated Reasoning 14, 1995, pp. 237-255.

[REW08] *REWERSE, Reasoning on the Web with Rules and Semantics*, Мрежа за върхови постижения в тематика IST на 6-тата Рамкова Програма на ЕК, 2004-2008, страница в интернет http://rewerse.net/, последно посещение 17 април 2009.

[RaSe00] Ramsay, A. and H. Seville. *What did he mean by that?* Proc. Int. Conf. AIMSA-2000, Springer, LNAI 1904, 2000, pp. 199–209.

[ReSp04] Reinberger, M.-L. and P. Spyns. *Discovering Knowledge in Texts for the Learning of DOGMA-inspired Ontologies.* In the Proceedings of the ECAI-2004 Workshop on Ontology Learning and Population: *Towards Evaluation of Text-based Methods in the Semantic Web and Knowledge Discovery Life Cycle*, 2004, pp. 19-24.

[RSPD04] Reinberger, M.-L., Spyns, P., Pretorius, A.J. and Daelemans, W. *Automatic Initiation of an Ontology*. In R. Meersman, Z. Tari (Editors), Proc. of Int. Conf. CoopIS/DOA/OBDASE 2004, Springer, LNCS 3290, 2004, pp. 600-617.

[RoSch97] Roche E. and Y. Schabes (Eds.) *Finite State Language Processing*, MIT Press, Cambridge, Massachusetts, 1997.

[Sil93] Silberztein, M. *Dictionnaires électroniques et analyse automatique de textes: le système INTEX*. Masson, Paris, 1993.

[Ste00] Steimann, F. *On the representation of roles in object-oriented and conceptual modelling*, Journal of Data and Knowledge Engineering Vol. 35, No. 1, 2000, pp. 83-106.

[Ste05] Steimann, F. *The Role Data Model Revisited*. In: Boella, G., J. Odell, L. van der Torre, and H. Verhagen (Eds.), *Roles, An Interdisciplinary Perspective*. Proceedings of the AAAI Fall Symposium. AAAI Press, 2005, pp. 128-135. Електронен сборник трудове на http://www.aaai.org/Library/Symposia/Fall/fs05-08.php, последно посещение 14 май 2009.

[Sow84] Sowa, J. *Conceptual Structures - Information Processing in Mind and Machine*. Reading, MA Addison Wesley 1984.

[Sow91] J. Sowa. *Towards the Expressive Power of Natural Language*. In: J. Sowa (Ed.), Principles of Semantic Networks, Morgan Kaufmann Publishers, 1991, pp. 157-190.

[Sow92] Sowa, J. *Conceptual Graphs Summary*. In: Nagle, T., J. Nagle, L. Gerholz, and P. Eklund (Eds.): *Conceptual Structures: Current Research and Practice*, Ellis Horwood 1992, pp. 3-52.

[Sow00] Sowa, J. *Knowledge Representation: Logical, Philosophical, and Computational Foundations*, Brooks Cole Publishing Co., Pacific Grove, CA, 2000. Виж също *The Challenge of Knowledge Soup*, http://www.jfsowa.com/talks/challenge.pdf (последно посещение на 13 април 2009).

[Sow08] Sowa, J. *Conceptual Graphs*. In: van Harmelen, F., V. Lifschitz, and B. Porter (Eds.), *Handbook of Knowledge Representation*, Elsevier, 2008, Chapter 5, pp. 213-237.

[SAP99] Simov, K., Angelova, G. and E. Paskaleva. *MORPHO-ASSISTANT: The Proper Treatment of Morphological Knowledge*. In: Proceedings of the 13th Int. Conference COLING'90, Helsinki, Finland, 1990, Vol. 3, pp. 453-457.

[StDo00] Staykova K. and D. Dochev. *Development of lexico-grammar resources for natural language generation (experience from AGILE project)*. In: Cerry S. and D. Dochev (Eds.) Artificial Intelligence: Methodology, Systems, Applications, Lecture Notes in Artificial Intelligence 1904, Springer-Verlag, pp. 242-251, 2000.

[SDK01] Strupchanska A., P. Dobrev and K. Toutanova. *CGWorld-2001 – new features and new directions,* ICCS 2001 *Workshop on CG Tools*, July 2001, Stanford Univ., USA. Достъпно на http://www.cs.nmsu.edu/~hdp/CGTools/proceedings/papers/CGWorld.pdf, последно посещение 24 април 2009.

[SDK02] Strupchanska A., P. Dobrev and K. Toutanova. *CGWorld – from Conceptual Graph Theory to the Implementation*. ICCS 2002 *Workshop on CG tools*, July 2002, Borovets, Bulgaria, http://www.lml.bas.bg/iccs2002/acs/CGWorld2002.pdf, последно посещение 19 април 2009.

[SDPPDS07] Staykova K., Dochev D., Paneva D., Pavlova-Draganova L., Saraydarova V. *Development of Domain Ontology, Targeted to Creation of Learning Materials from Digital Archives*. In: Christodoulakis S. (Ed.) Proc. of Workshop on "Cross-Media and Personalized Learning Applications on top of Digital Libraries" LADL 2007, 20 September 2007, Budapest, Hungary, pp. 91-100.

[SGP08] Stoyanov S., N. Govedarova and I. Popchev. *A Knowledge-Intensive CBR Application for Content-Based Retrieval in BULCHINO Catalogue*, In Proc. of the Conference "Concurrency, Specification and Programming 2008", Gross Vaeter, Germany, 29 September - 1 October 2008, Vol. 1, pp. 168-180.

[SGPO'D08] Stoyanov, S., I. Ganchev, I. Popchev and M. O'Droma. *An approach to the development of infostation-based elearning architectures*. Comptes rendues de l'Académie bulgare des Sciences, Vol. 61, No. 9, 2008, pp. 1189-1198.

[SJA04] Snow, R., D. Jurafsky, and Y. Andrew. *Learning syntactic patterns for automatic hypernym discovery*, Advances in Neural Information Processing Systems 17, 2004.

[SSD08] Saraydarova V., K. Staykova, D. Dochev. *Extensions of Logos Domain Ontology "Bulgarian Iconographical Objects"*. In: Stockinger P., D. Dochev (Eds.) Proceedings of the Second LOGOS Open Workshop "Cross-Media and Personalised Learning Applications with Intelligent Content" LAIC 2008, 2008, Varna, pp. 67-78.

[StTo09] Stefanov, K. and K. Todorova. *Computing Ontology Creation*. In: Proceedings of International Congress MASSEE2003, Borovets, Bulgaria, pages 40-49.

[SVPM08] Stoyanov, S., V. Valkanova, I. Popchev and I. Mihov. *A Scenario-Based Approach for the Creation of a Virtual Environment for Secondary School Instruction*. "Cybernetics and Information Technologies", Vol. 8 (2008), № 3, pp. 86-96.

[TExtr] Term Extractor, http://lcl2.di.uniroma1.it/termextractor, последно посещение 25 април 2009.

[TrMa00a] Trausan-Matu, St. *WebGen - A Prototype of the Generator of Immersive Contexts on the Web*. Larflast project, 2000, Report 6.2 after month 22, delivered to the European Commission.

[TrMa00b] Trausan-Matu, St. *Metaphor Processing for Learning Terminology on the Web*. Proc. AIMSA-2000, Varna, Bulgaria, September 2000, Lecture Notes in AI 1904, Springer, pp. 233-241.

[TWB91] Kugler, M., G. Heyer, R. Kese, B. von Kleist-Retzow, and G. Winkelmann. *The Translator's Workbench: An Environment for Multi-Lingual Text Processing and Translation.* In: Proc. Machine Translation Summit III, Washington, DC, USA, July 1991.

[Vit99] Vitanova, I. *Learning Foreign Language Terminology: the User Perspective*. Larflast report 8.1, August 1999, delivered to the European Commission.

[vHa95] v. Hahn, W. *Linguistic Resources of DB-MAT.* Project DB-MAT, Technical report 2-95, July 1995.

[vHAn94] v. Hahn, W. and G. Angelova. *Providing Factual Information in MAT.* Proc. of the International Conference *Machine Translation - Ten Years on,* Cranfield, United Kingdom, November 1994, pp. 11/1 - 11/16.

[vHAn96] v. Hahn, W. and G. Angelova. *Combining Terminology, Lexical Semantics and Knowledge Representation in Machine Aided Translation.* In Galinski, C. and K.-D. Schmitz (Eds.), *Terminology and Knowledge Engineering*, Proc. of the 4th International Congress on TKE-96, Vienna, Austria, August 1996, INDEKS Verlag, Frankfurt/M., pp. 304-314.

[VaBo06] Vassileva D. and B. Bontchev. *Self-adaptive hypermedia navigation based on learner model characteristics*. Proc. of Int. Conference on Education, organised by IADAT (the International Association for the Development of Advances in Technology), Barcelona, Spain, July 2006.

[VPG88] P. Velardi, M. Pazienza, M. De'Giovanetti. *Conceptual Graphs for the Analysis and Generation of Sentences*. In: IBM J. Res. and Develop. Vol. 32 (2), March 1988, pp. 251-267.

[VW07] Vander Wal, Thomas. *Folksonomy Coinage and Definition*. http://vanderwal.net/folksonomy.html, последно посещение 25 април 2009.

[W3C06] Noy, N. and A. Rector (Eds.) *Defining N-ary Relations on the Semantic Web. W3C Working Group Note 12 April 2006.* http://www.w3.org/TR/2006/NOTE-swbp-n-aryRelations-20060412, последно посещение 25 април 2009.

[W3C-Lang] *Resource Description Framework* (RDF) http://www.w3.org/RDF/ и *Web Ontology Language* (OWL) http://www.w3.org/2004/OWL/, последно посещение 25 април 2009.

[Wer95] Wermelinger, M. *Conceptual graphs and first order logic*. In Ellis, G., R. Levinson, W. Rich and J. Sowa (Eds.): *Conceptual Structures: Applications, Implementation and Theory*, Proc. ICCS-1995, the 3rd Int. Conference on Conceptual Structures, Springer, Lecture Notes in Artificial Intelligence 954, 1995, pp. 323-337.

[Wi09] Wilks, Y. *Ontotherapy, or How to Stop Worrying About What There Is*. Invited talk at RANLP-2007, to appear in Nicolov, N., G. Angelova and R. Mitkov (Eds.), *Recent Advances in Natural Language Processing V. Selected papers from RANLP-07*. Series "Current Issues in Linguistic Theory", John Benjamins, Amsterdam, 2009.

[Woo70] Woods, W. A. *Transition network grammars for natural language analysis*. Comm. ACM, 1970, 13(10): pp. 591-606.

[WCH97] Winston, M., R. Chaffin and D. Herrmann. *A Taxonomy of Part-Whole Relations*. Cognitive Science Vol. 11, 1987, pp. 417-444.

[WSG97] Wilks, Y., B. M. Slator, and L. M. Guthrie. *Electric words: dictionaries, computers, and meanings*. The MIT Press, 1996, 289 pages.

[WKNW72] Woods, W., R. Kaplan, and B. Nash-Webber. *The Lunar Sciences Natural Language Information System*. Final Report, Bolt, Beranek and Newman Report 2378, Cambridge, Massachusetts, 1972.

[WNet] *WordNet – a Lexical Database for the English Language*. http://wordnet.princeton.edu/, последно посещение 13 април 2009.

[Zar09] Zarri, G. P. *Representation and Management of Narrative Information, Theoretical Principles and Implementation*. Springer Series *Advanced Information and Knowledge Processing* 2009, 302 p.

[Анг08а] Ангелова, Г. *Езиковите технологии днес и утре*. Доклад по покана на Пролетната Конференция на СМБ, Боровец, април 2008. В: Сборник трудове на 37-мата Пролетна конференция на Съюза на Математиците в България, ISSN 1313-3330, София, 2008, стр. 68-85.

[Анг08б] Ангелова, Г. *Автоматична генерация на обяснения в техническа област*. Сп. *Автоматика и информатика*, издание на Съюза по автоматика и информатика "Джон Атанасов", София, 2008, ISSN 0861–7562, кн. 4/2008, стр. 17-24.

[АБВСБХ84] Ангелова, Г., А. Борковский, В. Вернер, Г. Стрейны-Бринзой и Вл. Хорошевский. *Языки программирования для исскуственного интеллекта*. В: Заключителен отчет на Работна група 18 на КНВВТ "*Представяне на знанията в човеко-машинни и робототехнически системи*", Москва, ИЦ на АН на СССР, ВИНИТИ, 1984, Том С, стр. 31-72.

[БаЛинк] Балрик-Линг: http://www.larflast.bas.bg/balric/index/index.htm, от панела вляво: *Морфологични ресурси, Анализатор, Демо за българския език*. Последно посещение 21 април 2009.

[БГ-19] Проект „*Езикови технологии в среди за електронно обучение: иновации, приложения, изследвания*", BG-19 към Нац. фонд "Научни изследвания", 2005-2007, http://www.fmi-plovdiv.org/index.jsp?id=520&ln=1, последно посещение 6 май 2009.

[БДБ04] Bultreebank, проект финансиран от Фондация Фолксваген 2000-2004 с български ръководител Кирил Симов, вж. www.bultreebank.org, последно посещение 21 април 2009.

[Бултра] Система Bultra http://www.bultra.com, Последно посещение 21 април 2009.

[БВУ] *Български виртуален университет*. http://www.bvu-bg.eu/ Последно посещение 8.04.2009.

[Дас07] *DaskaL, езиково-независим продукт за създаване и интерактивното използване на упражнения и тестове за езиково обучение*. Институт за Български Език на БАН, http://dcl.bas.bg/programs_bg.html.

[ДиД95] Дичева, Дарина. *Интелигентни системи за обучение*. София, Издателство Софтех, 1995, 144 стр.

[Доб07] Добрев, П. *Представяне, управление и използване на знания с приложения при обработка на естествен език*. Дисертация за присъждане на образователната и научна степен доктор по научната специалност „Компютърни системи, комплекси и мрежи", София, 2007.

[ДКНП08] Деврени-Куцуки, А. и М. Нишева-Павлова. *ONTO-PEDIA: Онтология, представяща модел на просветната система в България в периода 1940-1954 г.* В: Електронно списание "Дидактическо моделиране", ИМИ-БАН. www.math.bas.bg/~omi/DidMod/Articles/AnnaDevreni-Ontopedia.pdf, последно посещение 27 април 2009.

[Кал07] Калайджиев, О. *Интелигентна система за обучение, базирана на езикови технологии*. Дисертация за присъждане на образователната и научна степен доктор по научната специалност „Информатика", София, 2007.

[Кру07] Крушков, Хр. *Интегрирана компютърна среда за обучение по български език*. http://rdesc.uni-plovdiv.bg/hdk/ICTP2001.htm. Последно посещение 20 април 2009.

[Ман03] Манев, Кр. *Увод в дискретната математика*. Трето издание, КЛМН, София, 2003, 337 стр.

[Мих00] Михов, Ст. *Минимални ациклични автомати: конструкции, алгоритми, приложения.* Дисертация за присъждане на образователната и научна степен доктор по научната специалност „Информатика", София, 2000 (достъпна на http://www.lml.bas.bg/~stoyan, последно посещение 20 април 2009.)

[МКБ10] *Международна класификация на болестите версия 10*, http://www.rcz-varna.com/index.php?ch=22, последно посещение 25 април 2009.

[НКПИД03] *Национална класификация на продуктите по икономически дейности*, Версия 2003 (НКПИД-2003), http://www.nsi.bg/Classifics/Classifications.htm, последно посещение 25 април 2009.

[ОсСи07] Осенова, П. и К. Симов. *Формална граматика на българския език.* Институт по паралелна обработка на информацията, БАН, София, България, 2007, 128 страници. Вж. http://www.bultreebank.org/bgpapers/FormalGrammarBG.pdf

[ОТекст09] *Списък публикации на сътрудници на ОнтоТекст* http://www.ontotext.com/publications/index.html, последно посещение 28 април 2009.

[Пас07] Паскалева, Е. *Компютърна морфология – ресурси и инструменти.* ИПОИ-БАН, София 2007, ISBN 978-954-92148-1-9, 150 стр.

[ПоДа90] Попчев, И. и Л. Даковски (ред.) *Изкуствен интелект – проблеми и приложения.* Държавно издателство «Техника», София, 1990 г.

[Сто08] Стоянов, С. *Проект 'Електронен каталог на Българското културно-историческо наследство'*, Семинар Европейски проект MINERVA eC', организиран от Централна библиотека на БАН на 28 февруари 2008, София. Информационен бюлетин на ЦБ-БАН, брой Брой 4 (14). Презентация достъпна на http://cl.bas.bg/about-central-library/bulletin-of-central-library-of-bas/volume-14/presentations/bulchino_presentation.pps, последно посещение 25 април 2009.

[СемТех09] Проект *Семантични Технологии за Интернет услуги и технологично-поддържано обучение* на ИИТ-БАН с ръководител ст.н.с. д-р Генадий Агре, вж. http://sinus.iit.bas.bg/

[УК09] Проект *Умна книга* на ФМИ-СУ с ръководител доц. д-р Ив. Койчев, финансиран от НФНИ през 2009-2011 год., вж. http://dse.fmi.uni-sofia.bg/smartbook.html

Списък на фигурите в текста

Списък на таблиците в текста

Кратко описание на проекта ЛАРФЛАСТ и средата STyLE

Проектът Ларфласт изследва възможностите за интеграция на езикови и семантични технологии в системи за обучение. Като най-естествено приложение беше избрано подпомогнатото от компютър изучаване на чужд език, тъй като се предполага, че предимствата на езиковите технологии ще се открояят най-ясно при изучаването на естествен език. А възможностите на семантичните технологии биха проличали най-добре в област, която се поддава сравнително лесно на концептуално моделиране. Поради това в проекта беше създадена прототипна среда за изучаване на английска финансова терминология, наречена STyLE. Характерна особеност на STyLE е автоматичната проверка на коректността на отговор на студента, въведен в системата на свободен английски език.

Има няколко типа грешки при отговаряне на тестове, които се дължат както на незнание на езика, така и на непознаване на предметната област [Vit99]:

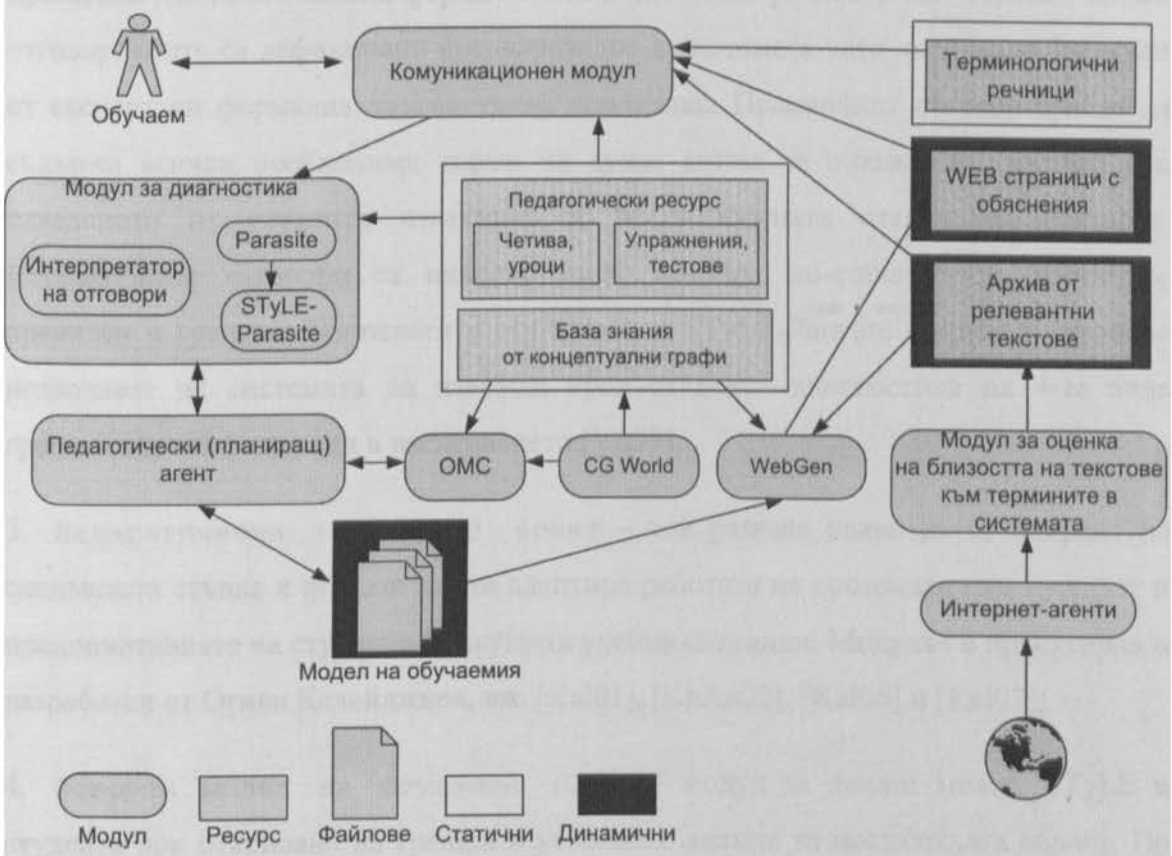
- *Езикови грешки* – в правописа на термина, морфологични грешки при употреба на словоформите му, синтактични грешки при конструиране на изречения с термина;
- *Неразбиране на въпроса, поставен в теста*;
- *Коректно разбиране на въпроса, съчетано с липсващо знание за правилния термин* – което води до използване на парафрази и термини за надпонятия;
- *Коректно разбиране на въпроса, съчетано с липсващо знание за предметната област* - което води до използване на по-специфични термини или даване на частично-верни, непълни или грешни отговори.

Средата STyLE моделира гореизброените ситуации за възникване на грешни отговори чрез интеграция на модули за обработка на естествен език и семантични технологии за обработка на знанието за предметната област.

При описание на архитектурата и компонентите на STyLE ще използваме Фиг. 4.1 от четвърта глава, което позволява по-лесно проследяване на съдържанието. Главните компоненти на системата са следните:

1. `Комуникационен модул` на главния сървър на системата – той *(i)* поддържа интерфейса с потребителя (показва му тестове или четива от предварително създадения педагогически ресурс и приема отговорите му) и *(ii)* осигурява връзката с отдалечените сървъри;

2. `Модул за диагностика` – разбира и оценява отговорите на обучаемия при изпълнение на тестовете. В STyLE има два вида тестове: *(i)* с фиксиран отговор, който се избира от меню, и *(ii)* с отговор, който се въвежда от клавиатурата на свободен английски език. Поради това проверката на верността на отговора се извършва от два различни компонента. Упражненията с фиксиран отговор – а те са



Фигура 4.1. Архитектура на интелигентната среда за обучение STyLE

от типа *'избор-по-меню'*, *'подреждане на изречения'* и *'попълване на празното място'* – се проверяват от `Интерпретатора на отговори`. Той сравнява низа на въведения отговор с предварително-зададен низ или множество от низове. Проверката на тестовете, чиито отговори са въведени на свободен английски език, се реализира чрез средата `Parasite` (създадена от Алън Рамзи и колектив от Университета в Манчестер, вж. [RaSe00] и [Ram05]) и модулът за логически доказателства `STyLE-Parasite` (създаден от Светла Бойчева, вж. [BKSA01], [Boy02] и [BVSYA04]). Средата Parasite осигурява лексикален, морфологичен, синтактичен и семантичен анализ на отговора на студента, като диагностицира евентуални лингвистични грешки. При анализиран отговор Parasite извежда логическа форма (модел) на въведения кратък английски текст. Но логическата форма се композира само за подаденото изречение, извън контекста на текущото упражнение. С цел проверка за съответствие с теста, компонентът STyLE-Parasite проверява дали логическата форма е *между* най-специфичния и най-общия очакван отговор, които са дефинирани предварително в системата като логически формули от експерт по формална лингвистична семантика. Правилният отговор трябва да съдържа всички необходими терми на думи, които се очаква да участват във въведеното от студента изказване, и то в нужната степен на общност. Неправилните отговори са няколко вида: по-общ, по-специфичен, частично-правилен и грешен. Разпознатите от Parasite и STyLE-Parasite грешки в отговора позволяват на системата да извежда пред студента диагностика на 4-те вида грешки, идентифицирани в изследването [Vit99];

3. `Педагогически (планиращ) агент` – той решава какво да се направи на следващата стъпка и по този начин адаптира работата на системата към нуждите и предпочитанията на студента в текущата учебна ситуация. Модулът е проектиран и разработен от Огнян Калайджиев, вж. [Kal01], [KaAn02], [Kal06] и [Kal07];

4. `Отворен модел на студента (ОМС)` – модул за диалог между STyLE и студента при откриване на грешки в усвоеното знание за предметната област. По желание на обучаемия, в определени учебни ситуации се стартира диалог с ОМС с цел изясняване на погрешно-заучените понятия, при който ОМС показва на

обучаемия концептуални структури в графичен формат. Компонентът OMC е създаден в Университета на Лийдс от Ваня Димитрова, вж. [DSB01], [Dim01] и [Dim02];

5. Генератор WebGen - модул за генерация на кратко обяснение (до един екран) с дефиниции и употреби на термините, които студентът не е заучил добре. Текстът се показва в html-страница и е оформен на контролиран английски език. Генераторът се извиква, когато студентът поиска допълнителни сведения в текущата учебна ситуация. Модулът е създаден от Стефан Траушан-Мату в Центъра по изкуствен интелект на Румънската академия на науките, вж. [TrMa00a] и [TrMa00b]);

6. Интернет-агенти – самостоятелни компоненти, които в режим off-line търсят по мрежата текстове с подходящо съдържание и ги подават на модула за проверка на близостта до термините на системата STyLE. Агентите са създадени от групата на Стефано Чери в LIRM – Монпелие, Франция, вж. [MaCe01] и [CDMTM01];

7. Модул за оценка на близостта на текстове към термините на STyLE – този модул изчислява коефициент на релевантност за намерените в интернет текстове, които се свалят главно от финансови сайтове, и създава динамичен архив от четива за показване пред студента при нужда в текущата учебна ситуация. Модулът прилага статистически методи за обработка на естествен език и е създаден от Преслав Наков, вж. [Nak00a] и [Nak00b];

8. Среда за графично представяне на концептуални графи CGWorld – тя позволява създаване и поддържане на системната база от концептуални графи и подпомага изобразяването на графи пред студента в модула OMC. Средата е разработена от Павлин Добрев и Албена Струпчанска с участието на Кристина Тутанова (вж. [DoTo00], [SDK01] и [SDK02]) и [Dob07]).

В STyLE има *статични* и *динамични* ресурси. Статичните остават неизменни при работата на системата, докато динамичните се влияят от поведението на студента

или се обновяват с течение на времето. Статичните ресурси, показани на Фиг.4.1, са:

- *Педагогически ресурс на системата*, създаден от преподавателя-експерт Ирена Витанова, с два вида учебни материали: *(i)* четива-уроци, които резюмират най-важните сведения от предмета, и *(ii)* упражнения-тестове, чрез които се проверяват знанията на студентите. Тези тестове покриват областта на финансовите пазари и позволяват цялостна проверка на знанията на студентите за разглежданите понятия;
- *Декларативно-представени факти за предметната област*, огранизирани в база от концептуални графи, които формират концептуалния ресурс на системата. Елементите на базата от знания са свързани с речниците от термини чрез специална анотация. Етикетите на понятията се използват и като метаданни в анотацията на учебните обекти от педагогическия ресурс. Например предварително е зададено кое понятие от базата и кое негово свойство се тестват от съответното упражнение (по този начин – при грешен отговор - системата може да включи в модела на студента информация, че *'студентът X не знае понятието Y'*). Концептуалният модел ще бъде разгледан по-долу;
- *Многоезични терминологични речници* с граматична информация и текстови обяснения за значението на термините (разработени под ръководството на Светлана Дикарева в Държавния Университет на Симферопол, Украйна, вж. [DRDK01].

Динамичните ресурси – които се генерират в зависимост от текущото състояние на модела на студента или са набор от динамично-събирани текстове – са следните:

- *Модел на студента*, съхраняващ за всеки студент 'бележник' с резултатите му при изпълнение на тестовете (вж. [Кал07]);
- *Текстови обяснения от най-много една страница (един екран)*, генерирани от WebGen при заявка, поставена чрез Комуникационния модул;
- *Динамично-обновявана библиотека от текстове-учебни обекти*, които са предварително свалени от Интернет и категоризирани като 'близки' до

съдържанието на учебния материал в STyLE. За всеки текст се пази мярка за подобие към изучаваните термини. Така текстът може да се показва като четиво при нужда, ако студентът поиска допълнителна информация за конкретен термин. Генераторът WebGen извлича от тези текстове примери на употреби на термините, което позволява обогатяването на STyLE с повече примери за контекста на употреба на термините. Динамичният ресурс от текстове беше високо оценен от преподавателите, които винаги имат нужда от нови примери при подготовка на учебния материал и домашните задания.

От гледна точка на обучаемия, средата STyLE е съвкупност от html-страници, които са достъпни от всяка точка в интернет. Студентът получава следните услуги:

- да чете основните уроци по темата на финансовите пазари,
- да проверява съдържанието на многоезиковите терминологични речници,
- да изпълнява упражненията от педагогическия ресурс, като препоръката е да се следва предложения от Планиращия агент път за навигация в педагогическия ресурс. В случай на грешка, обучаемият избира измежду следните възможности:
    - да последва някоя от избраните от Планиращия агент хипервръзки и да прочете допълнителните четива, предоставени от системата;
    - да обсъди грешките си с модула ОМС;
    - да избере от друга услуга (за четене или тестване), предлагана от средата;
    - да напусне интернет-страниците на STyLE.

Средата е многопотребителска система за обучение, подпомагаща самостоятелните занимания като допълнение на учебната програма в клас. За всеки обучаем се поддържа кохерентен диалог, независимо кой от сървърите на системата произвежда текущата html-страница. Студентите оцениха положително интеграцията на учебни текстове и упражнения в едно цяло и възможността веднага да видят къде правят грешки, с предложение за допълнителни четива по

темата. Обикновено те следват предложенията на Педагогическия агент, който им предлага систематично обхождане на упражнения и четива с цел цялостно изучаване на включените теми и тестване чрез всички възможни упражнения.

Четирите технологии за обработка на естествения език - Parasite, STyLE-Parasite, WebGen и модулът за оценка на релевантността на текстовете - са интегрирани в STyLE по невидим за потребителя начин [ASKYBV04]. От гледна точка на студента няма значение дали той изпълнява упражнения с фиксиран отговор или тестове с отговор на свободен английски език (освен че във втория случай има известно забавяне в реакцията на системата, особено ако са отворени няколко копия на Parasite и STyLE-Parasite в SICStus Prolog). Факти от базата се изобразяват пред студента от модула ОМС като концептуални графи в различни цветове. Части от йерархията също се показват като многоцветни изображения при диагностицирани грешки на обучаемия, с цветове откроряващи различните класификационни перспективи.