

V. ДИСКУРС

Подготвителни понятия за обработка на дискурса

Представяне на знания (ПЗ) - формализъм за представяне на факти + правила за извод. Вече говорихме за семантични мрежи, за йерархия на типовете. Знание за типовете съществено подпомага решението къде е “закачена” предложната група (PP) в следните две изречения с многозначна синтактична структура: *Четох книга за еволюцията през последните 10 минути. Четох книга за еволюцията през последните 2 млн. години.* Но има и по-сложни семантични случаи, когато е необходимо повече знание.

Когато формула P трябва да бъде истина при дадена база знания, казваме, че базата знания **влече** (entails) P. (\supset)

Импликацията е заключение, което обикновено може да се изведе в общия случай, но в частен случай може и да бъде отречено. Например, “Иван има две коли” влече, че Иван има една кола. Импликацията е, че няма три.

Видове извод:

(1) *дедуктивни* - логически следствия

(2) *недедуктивни:*

(2.1) *индуктивни* - напр. самообучение чрез общи характеристики в примери;

(2.2) *абдуктивни* - заключения, че е налице причината, щом е вярно следствието.

Нека $A \supset B$. По дедукция, по дадено A извеждаме B. По абдукция, при дадено B извеждаме A, понеже A е причината B да бъде вярно.

Монотонни представяния: добавяне на нови формули към базата знания само увеличава броя на формулите, които “се влекат” от базата знания.

Много системи ползват **правила по премълчаване** (default rules $A \Rightarrow B$), спрямо които са възможни изключения при определени екземпляри. В общия случай заключението се прави както следва: имаме правило $A \Rightarrow B$. Нека A е истина. Можем ли да докажем $\neg B$? Ако не можем, тогава $\neg B$ е лъжа и B е истина.

Пример: $\forall x \text{ BIRD}(x) \Rightarrow \text{FLIES}(x)$. Ако е дадено явно, че $\neg \text{FLIES}(\text{penguin})$, тогава пингвинът не лети. Иначе за всяка птица ще се изведе, че лети (по премълчаване).

Правилата по премълчаване въвеждат немонотонност. Става възможно с добавяне на нова формула да се намали броят на формулите, които се влекат от базата знания. Например, нека имаме базата KB1:

Котка(Пуфи). $\forall c \text{ Котка}(c) \Rightarrow \text{Мърка}(c)$.
 АнгорскаКотка(Пуфи).

От KB1 може да се изведе, че Пуфи мърка - понеже няма факти, които да попречат на извода. Да добавим сега нов факт получавайки KB2:

$\forall c \text{ АнгорскаКотка}(c) \Rightarrow \neg \text{Мърка}(c)$.

KB2 влече по-малко изрази, отколкото KB1.

Освен с правилата по премълчаване, друг полезен начин за въвеждане на немонотонност е предположението за затворен свят (closed world assumption, CWA). Предполага се, че базата знания съдържа пълна информация за дадени предикати; щом липсва информация да се изведе P - тогава не-P е истина.

Техники за извод - биват *процедурни* и *декларативни*.

Досега като език за представяне на знанията използвахме ПСПР с някои разширения, за да свържем този език с езика на ЛФ. В ЛФ имахме дискурсни променливи - например ?x, които се държат като квантор за всеобщност. Така $(P \text{ ?}x A)$ е еквивалентно на $\forall x P(x,A)$. Екзистенциално квантуваните променливи се третират чрез сколемизация - т.е. заместване на променливата с нова константа, която не е била използвана до момента. Например, $\exists y \forall x P(x,y) - (P \text{ ?}x Sk1)$, където Sk1 е несрещана константа на Сколем. Областта на квантора за съществуване се управлява чрез функция, която генерира различен обект за всяка необходима стойност:

$\forall y \exists x P(x,y) \rightarrow (P (Sk2 \text{ ?}y) \text{ ?}y)$.

По отношение на стила и изразителната сила на ЕПЗ - стремим се към ПСПР. Но кои са предикатите? Каква е тяхната грануларност? Имаме смесица от подходи измежду двата най-крайни:

- *всяко значение на дума е отделен предикат*. Така различаваме колкото искаме нюанси на значенията, но имаме твърде много значения и за всяко ще пишем аксиоми (което води до бавен извод).

- *семантични примитиви на предварително обобщени значения*. Всички други значения се изразяват в “координатната система” от семантични примитиви. Например *бягам, тичам, вървя, ходя, влача се, шляя се* всички могат да се опишат като MoveByFoot. Тогава ще имаме малко аксиоми - понеже те ще се пишат само за обобщени предикати като MoveByFoot, но пък ще загубим нюансите на значенията.

Както винаги, стремим се към златната среда. Въвеждаме *бягам, тичам, вървя, ходя, влача се, шля се* като подтипове на MoveByFoot. Аксиомите се пишат за обобщени типове, а за специфичните се задават само при нужда. Така системата може да обработва и непълна информация - при липса на данни за типа да го замести с над-типа му.

Трябва да отбележим, че колкото повече имената на предикатите се отдалечават от думите в ЕЕ, толкова по-трудно става да се генерира ЕЕ текст. Например:

$$\forall p (\text{MaleHuman}(p) \ \& \ \exists c \text{Parent}(p,c)) \ \& \ \text{MoveByCar}(p,L1) \ \& \ \text{Building}(L1) \ \& \ \text{UsedForTeaching}(L1)$$

“Всички хора от мъжки пол, които имат деца, се движат с кола към сграда, използвана на обучение” - т.е. всички бащи шофират към училище.

Счита се, че “доброто представяне на знанията” е да се избере такова множество предикати, което ще позволи ефективно използване на йерархичната организация. Добри попадения са концептуални обобщения, които са едновременно и важни лингвистични обекти - напр. класът OBJ/ACT, който разгледахме по-рано. В следващата лекция ще видим, че използването на тематичните роли като семантични връзки позволява генериране на очаквания, така че това е още една препоръка. Ако сравните две онтологии (Микрокосмос, правена за ОЕЕ и СуС - за изводи), веднага ще разпознаете по имената на категориите тази, която е ориентирана към ОЕЕ (<http://crl.nmsu.edu/Research/Projects/mikro/> и <http://www.cyc.com>).

Предствяне на стереотипна информация - фреймове

===== Декомпозиция на фрейм-ситуация на ТИПИЧНИ ограничения, условия, следствия и под-ситуации:

BUY(b):

Roles: Buyer, Seller, Object, Money

Constraints: Human(Buyer), SalesAgent(Seller), IsObject(Object), Value(Money), Price(Object)

Preconditions: OWNS(Buyer, Money)
OWNS(Seller, Object)

Effects: NOT OWNS (Buyer, Money)
NOT OWNS(Seller, Object)
OWNS(Buyer, Object)
OWNS(Seller, Money)

Decomposition: GIVE(Buyer, Seller, Money)
GIVE(Seller, Buyer, Object)

Кодиране на времето

Само ще споменем модела на Райхенбах (1947). По принцип е ясно, че ЕЕ-категорията *време* не тече линейно както във физиката например. Според теорията на Райхенбах, глаголното време (на английски) дава информация за три вида времена: R (reference time), S (speech time), E (event time). Те личат ясно в редица примери: *Когато Иван отвори вратата* (R), *Елена видя книгата* (E). или *Когато Иван отвори вратата* (R), *Елена беше видяла книгата* (E).

За английския език се предлага модел, свързващ глаголното време в изречението с някое от трите времена R, S и E. За простите глаголни времена, $E=R$. За така наречените постериорни времена, $R<E$. Получава се следната картина:

	Simple $E=R$	Perfect	Posterior $R<E$
Present	$\begin{array}{c} S \quad R \\ \hline \\ E \end{array}$ $S=R$ John sings.	$\begin{array}{c} E \quad S \\ \hline \quad \\ R \end{array}$ $E<R, R=S$ John has sung.	$\begin{array}{c} S \quad E \\ \hline \quad \\ R \end{array}$ $S=R$, John is going to sing
Past	$\begin{array}{c} E \quad S \\ \hline \quad \\ R \end{array}$ $R<S$ John sang	$\begin{array}{c} E \quad R \quad S \\ \hline \quad \quad \\ R \end{array}$ $E<R<S$ John had sung	$\begin{array}{c} R \quad S \quad E \\ \hline \quad \quad \\ R \end{array}$ $R<S$ John was going to sing
Future	$\begin{array}{c} S \quad E \\ \hline \quad \\ R \end{array}$ $R>S$ John will sing	$\begin{array}{c} S \quad E \quad R \\ \hline \quad \quad \\ R \end{array}$ $S<E<R$ John will have sung	$\begin{array}{c} S \quad R \quad E \\ \hline \quad \quad \\ R \end{array}$ $S<R$ John will be going to sing

В следващите лекции ще видим как се използва времето при обработка на контекста на дискурса (времето е част от контекста и във всички случаи се нуждаем от модел).

+++++

Дискурс (discourse): *семантично свързан фрагмент от ЕЕ (по-специално говорим език), обикновено по-дълъг от едно изречение, който се разглежда като взаимодействие между говорещи хора или между читател и писател.*

Има два важни вида дискурс: **текст** (където говорим за **изречение**, **писател**, **читател**) и **диалог** (там говорим за **изказване** /utterance/, **говорител** /speaker S/, **слушател** /hearer H/). Много модели се отнасят и за двата вида дискурс, така че ще използваме аналогичните понятия като взаимно-заменяеми.

При обработка на дискурса е полезно да различаваме четири вида контекст. Нека си представим, че в разговор между Иван и Мария тя казва: “кафето ми е студено”. Тогава:

	Специфичен/локален контекст	Общ/глобален контекст
Ситуационен контекст	Мария - S, Иван - Н. Часът е 3 следобед. Те са в барчето на ФМИ с по чаша кафе пред тях. Чашата е пластмасова.	Те са на Земята, 3-тата планета от Слънчевата система. Има едно слънце и една луна. Има живот. Птиците обикновено летят.
Дискурсен контекст	Синт. структура на последното изречение е (S (NP кафето ми) (VP е студено)). “Кафето ми” се отнася към течността кафе в чашата пред Мария. Тя включва към дискурсната ситуация твърдението, че к. й е студено.	Мария и Иван обсъждат специализацията по ИИ към ФМИ. После почват да говорят условията на живот на студентите, за цените в барчето и стигат до критики на храната и кафето.

=====

VI. ЛОКАЛЕН КОНТЕКСТ на ДИСКУРСА и референция

Начална дефиниция, която ще бъде уточнявана в следващите 1-2 лекции: **локалният контекст** се състои от синтактичната и семантична структура на последното изречение (или на главното изречение при анализ на подчинена клауза) плюс списък обекти (**дискурсни референти**), споменати в последното изречение, към които могат да реферират местоимения и членувани NP-групи.

Досега сме разглеждали различни примери за референции, нека дадем още:
 = *анафора*: Иван си загуби портмонето. Той го търси целия ден.
 = *елипса*: Иван си загуби портмонето и търсеше пари назаем. Петър правеше същото. (*същото* се отнася към последното VP, а не че и Петър си е загубил портмонето. При елипсата връщането е до последната VP-група).
 = *повърхнинна анафора*: Кажете ми успеха на Кристина в първи курс. А във втори? А този на Светлана? (референция към индиректно въведен обект).
 = *смесени примери*: Иван целуна жена си, Петко също.

Важна съставка на локалния контекст е списъка от дискурсни референти (ДР); това са множество от константи в базата знания, представящи обекти

споменати в главната клауза, към които може в последствие да се реферира с местоимение. Понякога може един ДР да не е явно споменат в предишното изречение, но да е въведен имплицитно в локалния контекст (като *жената на Петко* по-горе). Затова ще говорим за обектите, които дадено изречение поражда/предизвиква (evokes) явно или неявно.

Важно е да се отбележи, че референцията в ЕЕ работи дори когато нито говорителят, нито слушателят могат да идентифицират конкретния обект. Например: *Вчера Иван си купил кола. Тя била много скъпа.* При тази неопределена референция в ЛФ се генерира сколемова константа за неидентифицирания индивид.

1. Генериране на дискурсни референти

В процеса на интерпретацията на всяка клауза обикновено се генерира по един ДР за всяко NP. Различните видове NP-групи обикновено поставят различни изисквания към породените от тях ДР. Неопределените NP-групи генерират нов ДР, който често не се нуждае от по-нататъшна идентификация в базата. Имената са константи в базата знания. Членуваните NP-групи реферират често към ДР в локалния контекст или към еднократни обекти в глобалния, напр. *луната, папата*. Местоименията също често реферират към ДР в локалния контекст. NP-групи в множествено число реферират към множества обекти. Комплексни NP-групи като “Джон и Мери” пораждат 3 ДР: Джон1, Мери1, {Джон1, Мери1}.

ДР, предизвикани от неопределени NP-групи, са много важни, тъй като те въвеждат нов обект в локалния контекст. За да изчислим множеството от ДР, първо превръщаме всички квантори от ЛФ в явни индивиди или множества. По-долу ще илюстрираме този процес и в рамките на долните примери ще предполагаме, че собствените имена и членуваните NP-групи се заместват с константи от базата, които представят референтите им.

NP-групите в множествено число пораждат едно и също множество от ДР, независимо от колективното или дистрибутивно четене. В колективното четене множеството става аргумент на предиката, а в дистрибутивното е област на квантора за всеобщност. Примери за пораждане на ДР:

Пример	ДР - име	Ограничения в/у множество	Ограничения в/у индивидите (в множеството)
a man	M1	none	MAN1(M1)

three women	W1	W1 = 3	$W1 \subset \{w \mid \text{WOMAN1}(w)\}$
some black cats	C1	C1 > 1	$C1 \subset \{c \mid \text{CAT1}(c) \text{ BLACK1}(c)\}$

Неопределените NP-групи в област на действие на квантора за всеобщност предизвикват затруднения, тъй като такава NP-група трябва да породи не индивид, а множество. Пример:

Three boys each bought a pizza. They ate them in the park.

Тъй като them очевидно не реферира към they, от “a pizza” трябва да се породи $P1 = \{x \mid \text{Pizza}(x)\}$. От нечленуваната група “a pizza” със сколемизация се произвежда по един обект за всяко момче.

Началната интерпретация е:

$\exists B: B \subset \{x \mid \text{Boy}(x)\} \ \& \ |B| = 3 \ \forall b: b \in B \ \exists p: \text{Pizza}(p) \ \& \ \text{Buy}(b,p)$

Генерират се следните два ДР след първото изречение:

B1: $B1 \subset \{x \mid \text{Boy}(x)\} \ \& \ |B1| = 3$

P1: $P1 = \{x \mid \text{Pizza}(x)\} \ \& \ \exists y: y \in B1 \ \& \ x = \text{sk5}(y)$

Тогава семантичната интерпретация на първото изречение става:

$\forall b: b \in B1 \ \text{sk5}(b) \in P1 \ \& \ \text{Buy}(b, \text{sk5}(b))$

В локалния контекст се разрешават множество референции. (Досега в предишната лекция сме разгледали алгоритми за разрешаване на прономиналната анафора от вложено изречение към дискурсен референт от главното изречение - тоест алгоритми работещи в “едно изречение”). По-долу ще разгледаме още техники за разрешаване на референцията, този път в локалния контекст: според историята на споменаването, според теорията за фокуса и при контекстуална интерпретация с използване на знание за света.

Най-важният вид обработка, който се извършва /предимно/ в локалния контекст, е разрешаването на референцията между съседни изречения. Тук ще разгледаме три алгоритъма: за разрешаване на анафора:

(а) чрез “историята”,

(б) чрез теорията на фокуса и

(в) чрез налагане на очакванията и интерпретацията.

Тези три техники предполагат различно количество знание в системата, както и различна дълбочина на анализ. (А) работи след “плитък” анализ, така действат повечето системи за машинен превод, които имат много големи речници и се справят /донякъде/ с почти произволен вход, в реално време. (Б) изисква детайлно знание за синтактичната структура. (В) е най-сложна, но и най-мощна, и е приложима към много тесни затворени

светове. Тези системи са много успешни, дори с гласова връзка, но те правят “*closed world assumption*” и трудно се адаптират към друга област.

2. Разрешаване на анафора чрез списъци на историите

Става дума за местоименна анафора. В предишната лекция дискутирахме механизми за разрешаване на анафората в рамките на едно изречение. Тука се спираме накратко на техники за разрешаване в локалния контекст, тъй като най-често местоименната анафора се разрешава в локалния контекст.

Списък на историята СИ (history list): списък на ДР генерирани от предишните изречения, най-напред са най-скоро генерираните ДР. Алгоритъмът най-грубо работи както следва: като намериш местоимение, провери СИ за възможен antecedent (видяхме примерни 4-5 условия миналия път, съгласуваност по род, число, възвратност и т.н.). Ако не намериш кандидат в последния локален контекст - търси в пред-последния и т.н. Тази стратегия за търсене на antecedent е съобразена с ограничението за “случване напоследък” (recency constraint): antecedentът е най-скорошният ДР, който удовлетворява всички изисквания. Пример:

- a. Фирмите имат много пари и строят добри лодки.
- b. Момчетата, обаче, построиха лодка със съвсем ограничени средства.
- c. Те знаеха, че лесно ще спечелят надбягването.

СИ при изречение (c) изглежда както следва (подреден по старост):

- b. B2: $B2 \subseteq \{x \mid \text{Boat}(x)\}$
 B3: $\text{Boat}(B3) \ \& \ \text{BuiltBy}(B3, B2)$
 B4: $\text{Budget}(B4) \ \& \ \text{LimitedFunds}(B4, B3)$
- a. C1: $C1 \subseteq \{c \mid \text{Company}(c)\}$
 M1: $\text{Money}(M1)$
 B1: $\text{Boat}(B1) \ \& \ \text{OwnedBy}(B1, C1)$

При (c) и “те”, B2 е най-скорошен и подходящ antecedent. В този случай стратегията със СИ ще даде правилното разрешение на референцията. Не винаги обаче е така.

3. Център (фокус) на дискурса и анафора

Ограничението за “случване напоследък” подрежда локалните контексти по “старост”. А каква е наредбата на ДР вътре в един локален контекст, имат ли те различна тежест като кандидат-antecedenti? Отговорът е ДА,

например обектите в главното изречение са “по-важни” от обектите в подчиненото. Пример за неприятно звучащи изречения:

Мария пиеше водка на масата. Тя беше кръгла и кафява.

Много хора определят второто изречение като “неправилно” или “лошо”. Почти всички се затрудняват да го интерпретират.

Каква е причината за “лошата” референция? Може би подлогът е по-силен кандидат за антецедент? Понякога е така, например:

Иван видя Петър на купона. Той се върна на масата да си вземе още водка.

Тук Иван е най-вероятният кандидат-антецедент за *той*, понеже е подлог. Обаче има и други примери:

Иван видя Петър на купона. Той се беше почерпил добре.

Ако *той* не се отнася към *Петър*, не е ясно каква е връзката между двете изречения. Тъй като предположението за свързаност (кохерентност) на дискурса е много силно, човек приема за правдоподобна тази интерпретация, която осигурява свързаността. Тука подлог е *Иван*, но се предпочита другата интерпретация.

Приема се, че свързаният дискурс е организиран около някаква тема (обект) и я дискутира. Обектът се нарича **център** или **фокус** на дискурса. Тенденцията е центърът да остава непроменен в няколко съседни клаузи (прости изречения) и след това да се мести към друг обект. Най-често с местоимение се замества центърът. Теориите за центъра предлагат специфичен механизъм за разрешаване на местоименната анафора: ако няма указания за смяна на центъра, те сочат текущия център като най-вероятен кандидат-антецедент.

Пример	<i>Иван</i> отиде късно на купона.
за местене	Когато <i>той</i> пристигна, срещна <i>Петър</i> на вратата.
на центъра:	<i>Той</i> беше решил да си тръгне рано.

Нека U_n и U_{n+1} са две съседни изречения (или изказвания, utterances). В теориите за центъра се работи с две понятия, които се осмислят на границата между U_n и U_{n+1} :

- Минал център (backward-looking center на U_n - означен с $C_b(U_n)$ - дискурсен референт определен за център в U_n) и
- Потенциалните следващи центрове (potential next centres, или forward-looking centers на U_n - означени с $C_f(U_n)$). Това са всички ДР в U_n . Те се подреждат един спрямо друг (според синтактичната позиция, с изследване на поведението им в корпус) в преференциален списък, като най-отпред са най-вероятните кандидати за смяна на центъра:

подлог, пряко допълнение, непряко допълнение, и т.н.

По дефиниция $C_b(U_{n+1})$ е най-приоритетният елемент измежду $C_f(U_n)$, който се среща в U_{n+1} . За първото изречение в дискурса C_b не е дефиниран. Най-горният приоритетен ДР се нарича preferred next center, C_p .

Тук ще разгледаме алгоритъм за разрешаване референция на местоимения (публикуван през 1987). Кандидатът за антецедент се изчислява след разглеждане на отношенията между C_b и C_f в U_n и U_{n+1} . (Често C_b в U_n се замества с местоимение, но не винаги - дадохме примери). Дефинират се четири вида връзки между клаузите U_n и U_{n+1} , според отношението между $C_b(U_{n+1})$, $C_b(U_n)$ и $C_p(U_{n+1})$:

	$C_b(U_{n+1}) = C_p(U_n)$ или недефиниран $C_p(U_n)$	$C_b(U_{n+1}) \neq C_p(U_n)$
$C_b(U_{n+1}) = C_p(U_{n+1})$	Запазване на центъра (continue)	Превключване към предпочитан център (smooth-shift)
$C_b(U_{n+1}) \neq C_p(U_{n+1})$	Задържане на центъра (retain)	Превключване към непредпочитан (rough-shift)

Алгоритъмът работи с две правила:

Правило 1: Ако към някой ДР от $C_f(U_n)$ се реализира с местоимение в U_{n+1} , то и $C_b(U_{n+1})$ е проминализиран (т.е. реализиран с местоимение).

Правило 2: Състоянията за смяна на центъра са подредени по предпочитания, напр. запазването на центъра е предпочитаната ситуация в две съседни изречения, в сравнение със смяната на центъра. Редът е
Continue > Retain > Smooth-shift > Rough-Shift.

Алгоритъмът е:

1. Генерирай възможните комбинации C_b - C_f за две съседни изречения, при всички възможни разпределения на антецедентите и техните кореференти.
2. Приложи органиченията и филтрирай според правило 1 и според други ограничения, напр. съгласуваност по род, число и лице.
3. Приложи правило 2 и избери най-предпочитаната ситуация.

Пример.

U1: John saw a motor in the shop.

U2: He showed it to Bob.

U3: He bought it.

За изречение U_1 имаме (с преферциални граматични позиции):

$C_f(U_1) : \{\text{John, motor, shop}\}$

$C_p(U_1) : \text{John}$

$C_b(U_1) : \text{недефиниран}$

Изречение U_2 съдържа две местоимения: *he* и *it*. *He* реферира към *John* (лице), а *it* към *motor* или *shop*. (Този пример го цитираме на английски, понеже еквивалентът му на български няма да работи; на английски *he* е винаги референт към лице, докато на български *той* би могло да реферира към лице, към мотора или към магазина). Иван е по дефиниция $C_b(U_2)$, понеже е с най-висок ранг в $C_f(U_1)$ и се среща и в U_2 (заместен с личното местоимение *he*). За *it* в U_2 се генерират две възможности:

- или *it* реферира към мотора и тогава

$C_f(U_2) : \{\text{John, motor, Bob}\}$

$C_p(U_2) : \text{John}$

$C_b(U_2) : \text{John}$

Преход: *continue* понеже $C_b(U_2) = C_p(U_2)$ и $C_b(U_1)$ е недефиниран

- или *it* реферира към магазина и тогава

$C_f(U_2) : \{\text{John, shop, Bob}\}$

$C_p(U_2) : \text{John}$

$C_b(U_2) : \text{John}$

Преход: *continue* понеже $C_b(U_2) = C_p(U_2)$ и $C_b(U_1)$ е недефиниран

И двата прехода са *continue* и алгоритъмът ще ги разглежда като еднакво вероятни. Но моторът е пряко допълнение, а магазина - непряко. Така се решава, че *it* реферира към мотора.

Минава се към изречение U_3 . Както се каза по-горе, *he* реферира към лице и в този случай това е или *John*, или *Bob*. Поради липса на друга възможност, *it* отново реферира към мотора. Възможностите са пак две:

- или *he* реферира към John и тогава John е $C_b(U_3)$ и

$C_f(U_3) : \{\text{John, motor}\}$

$C_p(U_3) : \text{John}$

$C_b(U_3) : \text{John}$

Преход: continue понеже $C_b(U_3) = C_p(U_3) = C_b(U_2)$

- или *he* реферира към Bob и тогава Bob е $C_b(U_3)$ и

$C_f(U_3) : \{\text{Bob, motor}\}$

$C_p(U_3) : \text{Bob}$

$C_b(U_3) : \text{Bob}$

Преход: Smooth-Shift понеже $C_p(U_3) = C_b(U_3)$ и $C_b(U_3) \neq C_b(U_2)$

По правило 2 continue се предпочита пред smooth-shift и така John се взема за референт, тоест се решава, че John купува мотора.

Този алгоритъм за пресмятане на центъра и разрешаване на референцията неявно (1) използва граматическата роля на NP-то (тоест предполага се наличие на пълен синтактичен анализ като вход) и дори се предполага, че NP-тата могат да се наредят в приоритетен списък едно спрямо друго, (2) пази “история” на последно споменаване в предишно изречение и (3) дава приоритет на повтарящото се споменаване. През 1989 е направена ръчна оценка за поведението на алгоритъма над корпус от 281 примера от текстове в три жанра. Точността му е 77.6%.

Пример за “грубо местене” (rough-shift) към непредпочитан център са изреченията *Мария пиеше водка на масата. Тя беше кръгла и кафява.*

Обърнете внимание, че на български преференциалните четения са по-неясно изразени, например и двете интерпретации на референцията на (4-5) в долните изречения изглеждат еднакво вероятни:

Докато Иван(1) се разхождаше в парка(2), той(1) видя Петър(3).

Той(4) го(5) покани на забавата(6).

Това ни показва, че съставянето на еквивалентен алгоритъм за български език предполага детайлно изследване в корпус от текстове, модификация на правилата за определяне на предпочитано четене и т.н.

4. Членувани NP-описания

Те всъщност приличат малко на местоимения, понеже много от тях имат референти в предишния дискурс. Екзистенциалните четения могат и да въвеждат обект, затова тяхното разпознаване е важно:

Миялната машина е развалена (въвежда обекта и едно свойство).

Победителят на рали “Златни пясъци” е руснак.

Референциалните четения си имат референт или в СИ, или в един от 4-те вида контекст (напр. *Земята, Слънцето, президента*). Членуваните NP-групи могат да реферират и към обект в света на диалога, видим за слушателя и говорителя - което е част от локалния ситуационен контекст.

5. Контекстуална интерпретация и структура на дискурса

Досега сме разглеждали (на няколко пъти) знание за света, но не сме го свързвали с обработката на описваната чрез ЕЕ ситуация.

5.1. Кохерентност

Досега казахме: “Решението кое значение е адекватно ще се вземе при контекстната интерпретация”. Какво значи, обаче, дадено изречение да има смисъл в даден контекст? Очевидно *логическата смисленост* е необходимо условие за кохерентността, понеже съвсем безсмислените изречения не вървят в какъвто и да е контекст.

Дискурсът е кохерентен, ако можете лесно да определите как изреченията са свързани едно с друго. Дискурс от несвързани изречения ще бъде много неестествен. Предположението за кохерентност е водещо в процеса на човешката интерпретация на ЕЕ. Примери:

1a. Jack took out a match.

1b. He lit a candle. (LIT1=LIGHT1 и LIT2=ILLUMINATE1)

Общото знание за света позволява да се предположат връзки между думите и изреченията: *match* е кибритена клечка и тя е използвана не за *осветяване* (*illuminate*) на свещта, а за *запалването* (*light*) ѝ - понеже обикновено свещите се палят с кибр. клечки. В други случаи може да няма ясна връзка:

2a = 1a.

2b. The sun set.

Като няма друга видима връзка, предполагаме темпорална - т.е. последователност във времето. Дори такава минимална връзка е достатъчна, за да мотивира изказването на двете изречения едно след друго и да създаде кохерентна ситуация.

Знанието за света, обаче, поражда импликации - а не следствия тип “влече”. Така на някой по-късен етап могат да се наложат корекции. Налагането на възможните интерпретации на дадено изречение върху очакванията, генерирани от предишния дискурс (**matching against expectations**), е най-важната техника за контекстна интерпретация.

5.2. Налагане върху очакванията

Нека е дадено множество очаквания E_1, E_2, \dots, E_n и множество интерпретации I_1, I_2, \dots, I_k . Как бихме могли да определим дали E_i подхожда (match) на I_j ?

Нека предположим, че от изречение 1a получаваме очакването E_1 (по-долу ще обясним как):

E_1 (event): $LIGHT(E_1) \& (Agent(E_1)=Jack1) \& (Instrument(E_1)=Match33)$,
където Match33 е нова константа в базата от знание.

Интерпретацията на 1b е многозначна:

I1: $LIGHT1(E_2) \& (Agent(E_2)=Jack1) \& (Theme(E_2) = Candle1)$

I2: $ILLUMINATE1(E_3) \& (Agent(E_3)=Jack1) \& (Theme(E_3)=Candle1)$

Ако по принцип се стремим да използваме дедукция при налагане на интерпретациите върху очакванията, това би могло да стане по два начина: или от очакванията да доказваме интерпретациите, или обратното. Но горният пример показва, че това е невъзможно: тъй като те съдържат различни терми. В очакването E_1 имаме “инструмент” и Сколемова константа, а в интерпретациите I1 и I2 - “тема” и “свещ” вместо за “кибритена клечка”. Очевидно това е типичният случай, защото всяко изречение съдържа нова информация спрямо предишното - иначе няма защо да се говори. Значи само с дедукция няма да стане, понеже E_1 и I1-I2 не се съдържат пряко. Интуитивно, към тях трябва да се прибави такава информация, че те да почнат да се съдържат. Немонотонните техники за налагане и съвпадение са мотивирани чрез следната интуитивна идея: две формули “се налагат” една върху друга, ако има такова смислено множество от равенства, които могат да се предполагат за верни така, че $E_i \supset I_j$ за някои i и j .

За примера: ако имаме $E_1=E_2$ и $Theme(E_1)=Candle1$, т.е. ако предположим, че в двете свързани изречения се обсъждат едни и същи събития, тогава бихме могли да получим $E_1 \supset I_1$. Но при никое множество от предположения не бихме могли да получим $E_1 \supset I_2$.

5.3. Алгоритъм за “налагане” на графи

На всяка формула се съпоставя граф. За връх N в графа, нека:

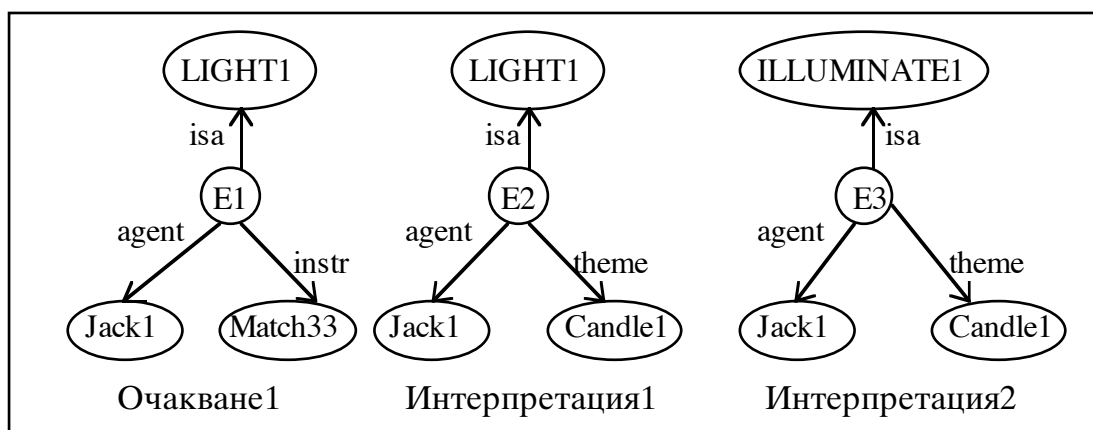
- $Label(N)$ е етикета на върха,
- $T(N)$ е типът (т.е. етикета на върха, в който сочи *is-a* връзката),
- $LL(N)$ е множеството от изходни стрелки от N .

За дъга A , нека $END(A)$ е етикетът на показвания от A връх.

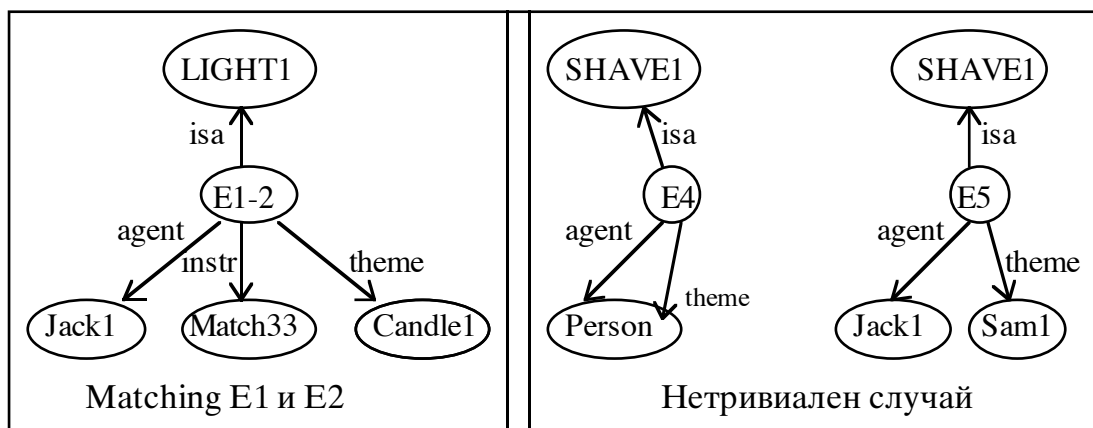
Два върха $N1$ и $N2$ “съвпадат”, тогава и само тогава, когато е изпълнено:

1. $T(N1)=T(N2)$ или единият е подтип на другия;
2. $Label(N1)$ и $Label(N2)$ не сочат към различни обекти, доколкото е известно.
3. За всяка връзка $L1$ в $LL(N1)$ (респ. $L2$ в $LL(N2)$):
 - или няма връзка със същия етикет, напускаща $N2$,
 - или има връзка $L2$ със същия етикет, напускаща $N2$, но $END(L1)$ и $END(L2)$ “съвпадат”.

Така “налагаме” графи, вж. примерни графи по-долу. По този алгоритъм, Очакване1 “съвпада” с Интерпретация1, но не и с Интерпретация2.



Когато налагаме събитията едно върху друго, има опасност да се получат и некоректни съвпадения - напр. че $Jack1=Sam1$ (долу вдясно). Затова алгоритъмът за налагане на графи трябва да проверява и глобалната консистентност. Очевидно не може да има идеален алгоритъм, това е евристична проверка дали две събития съвпадат. Хващат се типични случаи.



Да видим какви още ползи бихме имали, ако знаем как да генерираме очаквания.

5.4. Очаквания за “съвпадение” и референция

Досега говорихме, че местоименията и членуваните NP-групи се разрешават към техните референти в БЗ. Но може да се опитаме да ги разрешаваме и според очакванията за съвпадение и тогава успешното налагане ще дефинира референцията.

Да си припомним изречение 1b: He lit a candle. Ако местоимението “he” не е разрешено, при налагането на E1 и I1 ще се получи, че Agent (E1) = Agent(I1). Антецедентът на местоимението ще бъде идентифициран при налагането и в този случай референтът ще съвпадне с интерпретацията, предпочетена според теорията за центъра. Два примера показват как правенето на изводи и налагането на очаквания допълват теорията за центъра:

(3a). Иван отрови Петър. (3b). Той умря след седмица.

(4a). Иван отрови Петър. (4b). Той беше арестуван след седмица.

Центрирането ще покаже или един и същи антецедент в двата случая, или няма да може да избере между Иван и Петър. Обаче с налагане на интерпретацията върху очакванията можем да постигнем по-добър резултат. След (3a) и (4a) имаме очакванията E1, E2, E3:

E1: DIE(E4) & THEME(E4) = Peter1.

E2: IS-ILL(E5) & THEME(E5) = Peter1.

E3: ARRESTED(E6) & THEME(E6) = Ivan1.

Интерпретацията на 3b (без миналото време):
 DIE(E7) & THEME(E7)= HE1. (значи E7=E4 и HE1 съвпада с PETER1.

Интерпретацията на 4b:
 ARREST(E8) & (THEME(E8)=HE2) (значи E8=E6 и HE2 е IVAN1).

Същата техника върви и за членувани NP-групи:

(5a) Иван отрови Петър. (5b) Злодеят беше арестуван след седмица.

I1 на 5b: ARREST(E9) & (THEME(E9)=V1) & VILLAIN (V1)

PERSON

└───┬───┘

VILLAIN IVAN1

Щом знаем, че VILLAIN и IVAN1, то при налагане на интерпретацията върху очакванията ще получим V1=Ivan1. Така че налагането на очакванията е важна техника, но тя не замества другите техники, описани по-рано. Във всеки случай, очакванията са важни - но и структурата е важна. Да си припомним примера *Мария пиеше водка около масата. Тя беше кафява и кръгла*. Макар че според очакванията “масата” е очевидно единственият кандидат за antecedент и в крайна сметка точно по този принцип се разрешава референцията, този пример има структурни дефекти и затова звучи толкова лошо. Значи наличието на единствено очакване не компенсира дефектите на лошата структура.

Друг пример за важността на структурата:

7a. Иван отрови мъжа, от когото Петър открадна диамантите.

7b. Той беше арестуван след седмица.

8a. Петър открадна диамантите от мъжа, когото Иван отрови.

8b = 7b.

Съгласно очакванията, Иван и Петър са еднакво очаквани. Според теорията за центъра, в (7a-б) *той* е *Иван*, а в (8a-б) *той* е *Петър*. При равни очаквания се взимат преференциите според теорията за центъра в локалния контекст. Време е обаче да видим как се генерират очакванията.

5.5. Знание за действието и причинността, използвано при генериране на очаквания

Ще видим как знанието за действия може да се използва за генериране на очаквания. Ще се спрем на събитията, които се състоят в изпълняване на действия. Важното е, че тези действия причиняват следствия, които могат да се използват за генериране на очаквания. Например, ако системата види изречение “Иван отива към магазина”, след това реално е да се очаква

изречение “Иван е в магазина” (ефект на посоката на действието) и “купува нещо” (друго действие, предполагаемо поради типичната причина защо се ходи до магазин). Има много видове причинност и те влияят върху разбирането на дискурса. Накратко ще се спрем върху два вида важни причинно-следствени връзки, които свързват действия и състояния: ефект (effect) и условие (precondition).

Ефект: Всяко действие има множество следствия, които се причиняват от него и са в сила след изпълнение на действието (или докато то се изпълнява). Следствията са *съзнателни* (поради тях се изпълнява действието) или *несъзнателни* (те не са цел, но се случват). Действието не успява (fails), когато пропадат съзнателните ефекти. Неуспяло действие обаче може и да реализира страничните си ефекти.

Условие: Действие, което се изпълнява преди или по време на основното действие с цел осигуряване на успеха му.

Има важни връзки между две действия, които се използват при генериране на очаквания:

Осигуряване (enablement): Едно действие осигурява друго, ако ефектът на първото е предусловие за второто (или едно от предусловията).

Декомпозиция: Действие A1 е част от действие A2, ако A2 се декомпозира на стъпки. Например: ВЛИЗАНЕ-В-КОЛА е {ОТКЛЮЧВАНЕ-ВРАТА, ОТВАРЯНЕ-ВРАТА, ВЛИЗАНЕ-ВЪТРЕ}. Успехът на цялото изисква успех на всички компоненти.

Генерация: Изпълнение на A при дадени условия предизвиква B. Например, при работеща ел. инсталация, ЗАВЪРТАНЕ-НА-КЛЮЧА предизвиква ЗАПАЛВАНЕ-НА-ЛАМПАТА.

Често в ЕЕ има думи, които индицират тези връзки между действията: Запалих лампата, като завъртях ключа. Отидох до магазина с цел да купя мляко. Експлозията причини счупване на прозорците. Отидох на работа рано и това ми позволи да свърша всичко.

Повечето системи за представяне на знанията организират този вид информация около предиката, описва действието. Пример за BUY и PURCHASE-TICKET по-долу: ролите показват типичните участници-обекти, а ограниченията - типичните свойства на стойностите на ролите. Имаме предусловия и ефекти. Декомпозицията е поредица от поддействия. BUY е по-обща, а PURCHASE-TICKET е по-специфична.

ACTION BUY(b):

Roles: Buyer, Seller, Object, Money

Constraints:	Human(Buyer), SalesAgent(Seller), IsObject(Object), Value(Money, Price(Object))
Preconditions:	AT(Buyer, Loc(Seller)) OWNS(Buyer, Money) OWNS(Seller, Object)
Effects:	NOT OWNS (Buyer, Money) NOT OWNS(Seller, Object) OWNS(Buyer, Object) OWNS(Seller, Money)
Decomposition:	GIVE(Buyer, Seller, Money) GIVE(Seller, Buyer, Object)
<u>ACTION PURCHASE-TICKET(e):</u>	
Roles:	Agent, Clerk, Ticket, Booth, Money, Station
Constraints:	Human(Agent), Clerk(Clerk), IsTicket(Ticket), In(Booth, Station) TicketBooth(Booth), At(Clerk,Booth), Value(Money, Price(Ticket))
Preconditions:	OWNS(Agent, Money)
Effects:	OWNS(Agent, Ticket)
Decomposition:	GOTO(Agent, Booth), BUY(Agent, Clerk, Ticket)

Тази информация може да се използва по няколко начина. Първо, ако дойде за анализ изречението “Иван купи стерео”. Системата може да заключи, че Иван сега има стерео, а няма пари; имало е магазин, където е било стереото - но вече е у Иван. Така се създава екземпляр на BUY и всички неинстанцирани променливи в него са “очаквания”, понеже може да се заговори за тях. Но това са тривиални очаквания, понеже никой не казва: *Иван си купи стерео. Сега той го притежава*. По-интересни са очакванията, които се осигуряват от действието “купуване”: *Иван купи стерео. Сега той може да гърми по цяла нощ на съседите*.

Не е реалистично на всяка стъпка да се генерират всички възможни очаквания, има техники за контролиране на очакванията.

5.6. Scripts: разбиране на стереотипни ситуации

Сценариите (scripts) са по-големи единици от информация, идентифициращи стереотипни ситуации или сценарии в предметната област. TRAVEL-BY-TRAIN е пример за сценарии. Има два главни проблема при използване на сценарии в системи за ОЕЕ:

- системата да избере сценария, за който се говори в момента;
- да се помни коя точка на сценария се цитира в момента (now point)

Изреченията се “налагат” върху сценариите по различен начин. Изречения, описващи цели, се използват за избор на релевантен сценарий. Изречения, описващи действия, обновяват “now-points”.

Има три начина да започнем да обработваме сценарий, ако няма активен:

1. Появява се описание на действие като цел:

Иван трябваше да замине с влак за Пловдив, защото приятелката му го заряза и избяга там.

2. Появява се описание на състояние като цел: *Иван трябваше да бъде в Пловдив* . Съдържанието се съпоставя с наличните сценарии, за да се види кой би довел до тази цел.

3. Появява се описание на междинно действие: *Иван отиде до билетното гише*. Търси се сценарий, който има това действие като стъпка.

Веднъж активиран, сценарият се поддържа активен чрез преместване на точката “now point”.

Сценарии и инстанциация на роли

11a. Иван искаше да вземе влака за Пловдив.

11b. Той си купи билет на гарата.

След 11a се активира TRAVEL-BY-TRAIN с екземпляр T1.

TRAVEL-BY-TRAIN (e):

Roles: Actor, Clerk, SourceCity, DestCity, Train, Station, Booth, Ticket, Money

Constraints: Person(Actor), Person(Clerk), City(SourceCity), City(DestCity), TrainStation(Station), TicketBooth(Booth), In(Station, SourceCity), In(Booth,Station), At(Clerk, Booth), DepartCity(Train,SourceCity), Destination(Train,DestCity), TicketFor(Ticket, SourceCity, DestCity), Value(Money,Price(Ticket))

Preconditions: Owns(Actor, Money)
At(Actor, SourceCity)

Effects: Not Owns (Actor, Money)
Not At(Actor, SourceCity)
At(Actor, DestCity)

Decomposition: GoTo(Actor, Station)
Purchase-Ticket(Actor, Clerk, Ticket,Station)
GoTo(Actor, Loc(Train))
GetOn(Actor, Train)
Travel(Train, SourceCity, DestCity)
Arrive(Train, DestCity)
GetOff(Actor, Train)

GoTo(Ivan1, Station(T1))
PurchaseTicket(Ivan1,Clerk(T1),Ticket(T1),Station(T1))
GoTo(Ivan1,Loc(TR1))
GetOn(Ivan1,TR1)
Travel(TR1,SourceCity(T1), Plovdiv)
Arrive(TR1, Plovdiv)
GetOff(Ivan1,TR1)

