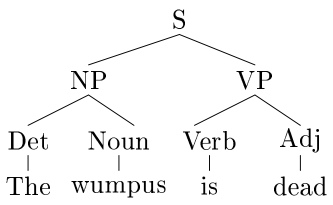
Chapter 10

Обработка на естествен език

10.1 Лекции

10.1.1 Комуникация между интелигентни агенти

1. Намерение $(a1) : knows(a2, \neg alive(wumpus, t_n))$.
2. Генериране на текст $(a1)$: The wumpus is dead.
3. Синтезиране на говор $(a1)$: "...thewampusisdead...".
4. Разпознаване на говор $(a2)$: The wumpus is dead.
5. Синтактичен анализ $(a2)$:



6. Семантична интерпретация $(a2)$: $\neg alive(wumpus, now) \vee tired(wumpus, now)$.
7. Прагматична интерпретация (свързване с текущия контекст на $a2$): $\neg alive(wumpus, t_n) \vee tired(wumpus, t_n)$.
8. Премахване на многозначността $(a2)$: $\neg alive(wumpus, t_n)$.
9. Използуване (проверка за съвместимост и включване към знанията на $a2$): $tell(KB, \neg alive(wumpus, t_n))$.

10.1.2 Синтактичен анализ

- Граматика $G = \langle S, T, N, R \rangle$:
  - Множество от терминални символи (думи от езика), $T = \{wumpus, dead, the, is\}$.
  - Множество от нетерминални символи (граматични категории), $N = \{S, NP, VP, Det, Noun, Adj\}$.
  - Начален символ $S \in N$.
  - Граматични правила (rewrite rules) $R : N \to N \cup T$. Пример (контекстно-свободна граматика - един символ отляво):
    $S \to NP\ VP$
    $NP \to Det\ Noun$
    $VP \to Verb\ Adj$
    $Det \to the$
    $Noun \to wumpus$
    $Verb \to is$
    $Adj \to dead$

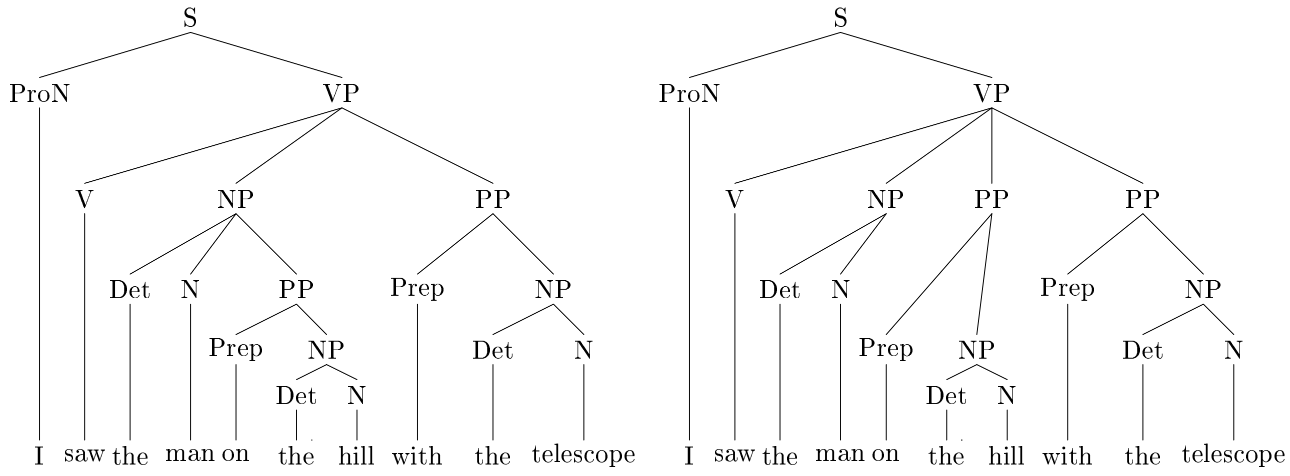


Figure 10.1: Дървета на разбора на многозначно изречение

- Дърво на разбора (фигура 10.1): `s(pn(I), vp(v(saw),np(det(the),n(man),pp(prep(on),np(det(the),n(hill)))), pp(prep(with),np(det(the),n(telescope))))).`
- Синтактична многозночност (фигура 10.1).
- Алгоритми за синтактичен анализ - отдолу нагоре (bottom-up parsing) и отгоре надолу (top-down parsing).

```
the wumpus is dead                         S
Det wumpus is dead   (Det -> the)          NP VP                 (S -> NP VP)
Det Noun is dead     (Noun -> wumpus)      Det Noun VP           (NP -> Det Noun)
NP is dead           (NP -> Det Noun)      the Noun VP           (Det -> the)
NP Verb dead         (Verb -> is)          the wumpus VP         (Noun -> wumpus)
NP Verb Adj          (Adj -> dead)         the wumpus Verb Adj   (VP -> Verb Adj)
NP VP                (VP -> Verb Adj)      the wumpus is Adj     (Verb -> is)
S                    (S -> NP VP)          the wumpus is dead    (Adj -> dead)
```

- Реализация на Пролог – логически граматики:
  - Недетерминиран append: `s(C) :- append(A,B,C),np(A),vp(B).`
  - Разлика на списъци. Нетерминали: `s(A,C) :- np(A,B),vp(B,C).` Терминали: `n([wumpus|X],X).`
  - Генериране на синтактично дърво: `s(A,C,s(X,Y)) :- np(A,B,X),vp(B,C,Y).`

10.1.3 Семантичен анализ

- Представяне на семантиката и семантични ограничения чрез case-frames. Например чрез съпоставяне на синтактичното дърво, `sem(Syntactic_tree,Semantics)`:

```
sem(s(np(Ag),vp(vp(Verb,np(_,Obj)),pp(P,NP))),[action=Act,agent=Ag,object=Obj,PP_type=NP]) :-
    person(Ag),
    object(Obj),
    meaning(Verb,P,Act,PP_type).

meaning(threw,to,propel,direction).
meaning(threw,for,sponsor,recipient).
```

- Премахване на многозначността:

```
[tom,threw,a,ball]             -> s(np(tom),vp(threw,np(a,ball)))
                               -> [action=propel,agent=tom,object=ball]
[tom,threw,a,ball,for,charity] -> s(np(tom),vp(vp(threw,np(a,ball)),pp(for,np(charity))))
                               -> [action=sponsor,agent=tom,object=ball,recipient=charity]
[tom,threw,a,ball,to,the,dog]  -> s(np(tom),vp(vp(threw,np(a,ball)),pp(to,np(the,dog))))
                               -> [action=propel,agent=tom,object=ball,direction=dog]
```

- Координация на частите на изречението:

```
[tom,takes,the,cake,from,the,table,and,eats,it] ->
   verb_action = taking               verb_action = eating
   subject_of(taking) = tom           subject_of(eating) = tom
   d_object_of(taking) = cake         d_object_of(eating) = cake

[tom,takes,the,cake,from,the,table,and,washes,it] ->
   verb_action = taking               verb_action = washing
   subject_of(taking) = tom           subject_of(washing) = tom
   d_object_of(taking) = cake         d_object_of(washing) = table
```

- Превод на изречения във формули от предикатното смятане (пример с програмата qa.pl):

```
?- sentence([every,man,that,lives,loves,a,woman],[],S).
S = all(_A,man(_A)&lives(_A)->exists(_B,woman(_B)&loves(_A,_B)))
```

10.1.4 Система за въпроси и отговори на естествен език

- Превод на изреченията във формули
- Превод в клаузна форма
- Дедуктивен извод чрез клаузна резолюция

10.2 Семинарни упражнения

- Реализация на граматики чрез недетерминиран append и чрез разлика на списъци (например граматика, която разпознава изреченията "john loves mary" и "john eats apples").
- Реализация на граматика, която генерира синтактично дърво. Например изречението "john loves mary" се превежда в терма s(np(john),vp(loves,np(mary))).
- Представяне на семантиката и семантични ограничения (например избягване на безмислени изречения като "john eats mary").

10.3 Лабораторни упражнения

- Програмиране на граматики чрез недетерминиран append и чрез разлика на списъци (например граматика за изреченията "john loves mary" и "john eats apples").
- Реализация на граматика, която генерира синтактично дърво. Добавяне на семантични проверки и генериране на семантика чрез съпоставяне на семантичното дърво.
- Експериметни с предикатите sentence и translate от програмата qa.pl. Кога изреченията се превеждат в повече от една клауза и кога в не-хорнова клауза?
- Експерименти с програмата qa.pl.