Úvod do počítačové lingvistiky

Aleš Horák

E-mail: hales@fi.muni.cz
http://nlp.fi.muni.cz/poc_lingv/

Obsah:

Organizace předmětu IB030

Hodnocení předmětu:

- závěrečná písemka (max 80 bodů)
  - dva řádné a jeden opravný termín
- průběžný úkol (max 20 bodů)
- hodnocení – součet bodů za písemku i úkol (max 100 bodů)
- rozdíly zk, k, z – různé limity
  např.:

A	80 – 100
B	73 – 79
C	65 – 72
D	58 – 64
E	50 – 57
F	0 – 49

K	45 – 100
Z	40 – 100

Základní informace

- přednáška je nepovinná
- cvičení – občas doporučené malé úkoly
- jeden hodnocený úkol (viz další slajd)
- web předmětu – http://nlp.fi.muni.cz/poc_lingv/
- slajdy – průběžně doplňovány na webu předmětu
- kontakt na přednášejícího – Aleš Horák <hales@fi.muni.cz> (Subject: IB030 . . .)

Samostatný hodnocený úkol

- letošní úkol – použít některou z dostupných jazykových knihoven pro češtinu:
  - NLTK – Natural Language Toolkit http://nltk.sourceforge.net
  - Field Linguist's Toolbox http://alphaworks.ibm.com/tech/lrw
  - FreeLing http://www.lsi.upc.es/~nlp/freeling
  - Stanford University Natural Language Software http://nlp.stanford.edu/software/
  - IBM LanguageWare Resource Workbench http://www.sil.org/computing/toolbox/
- k odevzdání je zapotřebí:
  - naprogramovaný vybraný algoritmus na češtině
  - dokumentace programu s ukázkami a návodem na instalaci/spuštění na serveru aurora.fi.muni.cz
  - vše odeslat v komprimovaném archivu e-mailem přednášejícímu (Subject: IB030 – odevzdani ukolu) do 26. května 2008
- hodnocení bude od 0 do 20 bodů podle:
  - složitosti vybraného algoritmus
  - kvality zpracování algoritmu i dokumentace

Literatura

Pala, Karel: Počítačové zpracování přirozeného jazyka, Brno FI MU, 2000. 190 s.

Allen, James: Natural language understanding, Redwood : Benjamin/Cummings Publishing, 1995, 654 s.

The Oxford handbook of computational linguistics, ed. by Ruslan Mitkov. Oxford University Press, 2003, 784 s.

Chomsky, Noam: Syntaktické struktury, Praha : Academia, 1966. 209 s.

Materna, Pavel - Štěpán, Jan: Filozofická logika: nová cesta?, Olomouc (Univerzita Palackého), 2000. 127 s.

slajdy na webu předmětu

Náplň předmětu

- počítačové zpracování přirozeného jazyka (*Natural Language Processing, NLP*)
- roviny jazyka
- reprezentace morfologických a syntaktických struktur
- analýza a syntéza: morfologická, syntaktická, sémantická
- formy reprezentace znalostí o lexikálních jednotkách
- porozumění jazyku: reprezentace významu věty, inference a reprezentace znalostí

Co je "počítačová lingvistika"

Lingvistika:

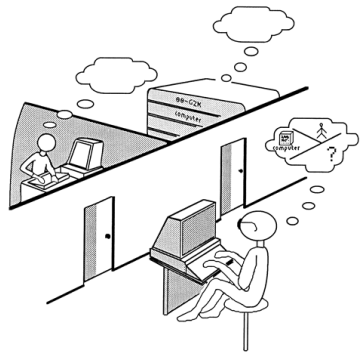
- jazykověda (*lingua* = lat. *jazyk*)
- věda o jazycích, jejich třídění, stavbě, zvukové i psané podobě
- zkoumá strukturu jazyka – slovotvorba, kombinace slov do vět, význam věty, ...

Počítačová lingvistika:

- od 60. let, *Computational linguistics*, často NLP (*Natural Language Processing*)
- spojení umělé inteligence (informatiky) a lingvistiky – jako jedna z kognitivních věd
- zkoumá problémy analýzy či generování textů nebo mluveného slova, které vyžadují určitou (ne absolutní) míru porozumění přirozenému jazyku strojem.
- tvoří jazykové modely – pojmy algoritmus, datová struktura, (formální) gramatika, ...

Turingův test

- z roku 1950, založen na tzv. imitační hře
- úkol – program komunikující jako člověk
- zahrnuje:
  - zpracování přirozeného jazyka (NLP)
  - reprezentaci znalostí (KRepresentation)
  - vyvozování znalostí (KReasoning)
  - strojové učení
  - (počítačové vidění)
  - (robotiku)



- od 1991 – Loebnerova cena (*Loebner Prize*) → každý rok $3.000 za "nejlidštější" program, nabízí $100.000 a zlatá medaile za složení celého Turingova testu

Eliza, Parry, ...

- Eliza – Joseph Weizenbaum, MIT, 1964–66
- program napodobující chování psychoterapeuta:

young woman:	*Men are all alike.*
eliza:	*In what way?*
young woman:	*They're always bugging us about something.*
eliza:	*Can you think of a specific example?*
young woman:	*Well, my boyfriend made me come here.*
eliza:	*Your boyfriend made you come here?*
young woman:	*He says I'm depressed much of the time.*
eliza:	*I am sorry to hear you are depressed.*
young woman:	*It's true. I'm unhappy.*

- Parry (Kenneth Colby, 70. léta), protějšek Elizy – počítačová simulace pacienta postiženého paranoiou
- oba využívají spíš "triky" než analýzu
- praktický význam – tzv. expertní systémy

Historie počítačové lingvistiky

- 1957 – rusko-anglický překlad
- Chomsky (60. léta) – generativní gramatika, vrozenost jazyka, ...
- strojový překlad není ani dnes dokonalý – potřebuje porozumět obsahu textu (Paretův zákon – pravidlo 80/20)
- problémy – víceznačnost, množství významů slov, různé způsoby užití slov k vyjádření významu, "Commonsense" a lidské uvažování
- Robert Wilensky: NLP je "AI-complete"
- 80. a 90. léta – rozvoj formalismů pro syntaktickou analýzu PJ (LFG, LTAG, HPSG)
- současně – zkoumání kvality statistických metod s rozsáhlými daty → srovnatelné výsledky!
- 90. léta až 200x – tvorba zdrojů vyšší úrovně (syntakticko-sémantické lexikony, wordnety, ...)
- stále není na obzoru splnění Turingova testu

Cíle počítačové lingvistiky

Významné úkoly v NLP:

- analýza přirozeného jazyka – morfologická, syntaktická, sémantická
- generování přirozeného jazyka
- syntéza a rozpoznávání řeči
- strojový překlad (*Machine translation*)
- odpovídání na otázky (*Question answering*)
- získávání informací (*Information retrieval*)
- korektura textu (*Spell-checking*, *Grammar checking*)
- extrakce informací (*Information extraction*)
- výtah z textu (*Text summarization*)
- určení typu dokumentu (*Text Classification/Clustering*)

Přednášky se vztahem k NLP na FI MU

- specializace Zpracování přirozeného jazyka, obor Umělá inteligence a zpracování přirozeného jazyka
- certifikát Euromasters in Speech and Linguistics
- vybrané přednášky:

IB030	Úvod do počítačové lingvistiky	Horák
IB047	Úvod do korpusové lingvistiky a počítačové lexikografie	Pala, Rychlý
IV029	Logická analýza přirozeného jazyka	Materna
PB016	Úvod do umělé inteligence	Horák
PB125	Řečová komunikace a dialogové systémy	Bártek, Kopeček
PV056	Vyhledávání znalostí v databázích	Popelínský
PV122	Formální struktura přirozeného jazyka	Peňáz

NLPlab – laboratoř ZPJ na FI MU

- sdružení lidí (studentů Bc., Mgr. a PGS i zaměstnanců) z oblasti NLP
- webový server `nlp.fi.muni.cz`
- fyzicky – 2 "skleníky" ve 2. patře budovy B:
  - 2 místnosti NLP – laboratoře zpracování přirozeného jazyka (doc. Pala)
  - část B203 pro LSD – laboratoř vyhledávání a dialogu (doc. Kopeček, prof. Zezula)
- vlastní laboratorní servery a stanice s OS Linux
- řeší několik velkých grantových projektů, pořádá mezinárodní konference (TSD, GWC, Lexicom, ...)
- práce studentů:
  - "malé projekty," které se využijí v rámci "velkých projektů"
  - bakalářské, diplomové i disertační práce
  - někdy i zaměstnanecký poměr
- PV173 Seminář Laboratoře zpracování přirozeného jazyka – pravidelná společná výměna informací

NLP projekty a SW na FI MU

Vybrané projekty:

- ajka – morfologický analyzátor
- i_par – editor morfologické databáze
- synt, klara, zuzana – syntaktické (a logický) analyzátory
- GDW (Grammar Development Workbench) – GUI pro vývoj gramatiky
- VisDic – editor wordnetů
- DEB – platforma pro XML databáze
- VerbaLex – slovník slovesných valencí
- bonito, manatee, Word Sketches – korpusový manažer
- demosthenes, text2phone (mbrola) – syntetizátory řeči
- UIO – inteligentní odpovídač
- Visual Browser – grafické znázornění (sémantických) sítí
- korpusy, slovníky, encyklopedie, ...

Roviny analýzy jazyka. Fonetika

Aleš Horák

E-mail: hales@fi.muni.cz
http://nlp.fi.muni.cz/poc_lingv/

Obsah:

Struktura jazyka

Struktura jazyka zahrnuje informace o:

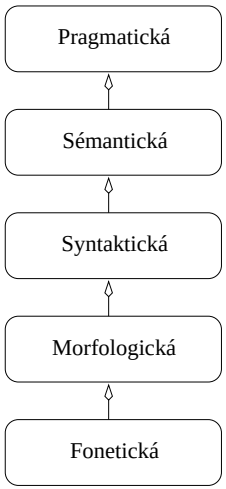
- co jsou slova (slovní tvary a jejich složky – morfémy)
- jak se slova (větné složky) kombinují do vět
- co slova označují, jaké jsou jejich lexikální významy
- jak se význam věty skládá z významů slov a slovních spojení (větných složek)

zpracování jazyka dále potřebuje:

- obecnou (encyklopedickou) znalost světa (ontologie)
- inferenční mechanismus
- znalost komunikační situace

Roviny analýzy jazyka

znalosti struktury jazyka jsou propojeny hierarchicky



jazykové roviny:

- fonetická
- morfologická
- syntaktická
- sémantická
- pragmatická
- kontextová
- znalost základní ontologie
- jazykové metaznalosti

Roviny analýzy jazyka – pokrač.

- fonetická – postihuje vztahy mezi zvuky používanými v (mluveném) jazyce, jejich skládání do slabik a slov
  foném – nejmenší jednotka jazyka, která může odlišit význam nadřazených jednotek

  kosit / *nosit* fonémy *k* a *n* odlišují dvě slova

  často odpovídají *znakům* → vždy ale označují *zvuky*

- morfologická – interní struktura slov, skládání slov z menších jednotek
  morfém – nejmenší jednotka, která může nést význam

  pří-lež-it- pří – prefix (*blízko*)
  -ost-n-ými: lež – lexikální kořen (*ležet*)
  it – adjektivní derivační sufix (*ten, který*)
  ost – substantivní derivační sufix (*ta skutečnost, že*)
  n – adjektivní derivační sufix (*charakteristický pro*)
  ými – gramatický afix (*instrumentál plurálu*)

Roviny analýzy jazyka – pokrač.

- syntaktická – struktura větných frází
  popisuje, jak vypadá gramaticky správná věta, většinou pomocí pravidel gramatiky
  syntaktický analyzátor – nástroj, který analyzuje vstup na základě gramatiky
  na výstup dává různé info, např. derivační stromy

- sémantická – význam výrazů přirozeného jazyka a jejich kombinací
  hodně závisí na zvolené sémantické reprezentaci
  logická analýza věty – strukturní část sémantické analýzy
- pragmatická – zkoumá vztah mezi výrazy přirozeného jazyka a kontextem
  často se do ní řadí znalost komunikační situace, základní ontologie a jazykových metaznalostí

Fonetika a fonologie

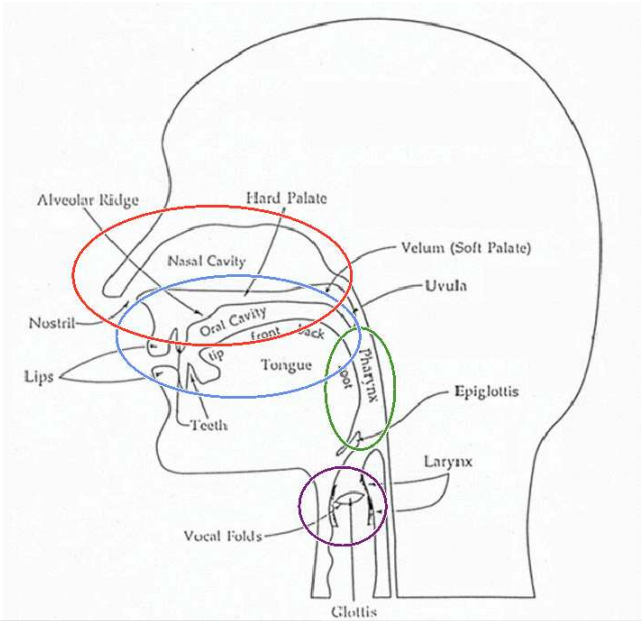
Fonetika:

- studuje produkci, přenos a příjem jazykových zvuků
- má klíčový význam např. pro oblast automatického rozpoznávání a syntézy řeči
- není tradičně chápána jako součást gramatiky jazyka

Fonologie:

- fonologický systém jazykových zvuků v *určitém jazyce*
- pracuje s gramatikou řečových zvuků
- pomocí gramatických pravidel popisuje historické změny i současné alternace

Kde vznikají jazykové zvuky?



Členění řečového proudu

řečový proud:

- nejsou mezery mezi slovy
- nejsou žádné izolované zvuky
- přesto všechny jazyky pracují s lingvistickými jednotkami jako separátními

orofón – fráze, které zní stejně/podobně, ale mají jiný obsah

> It's not easy to recognize speech.
> It's not easy to wreck a nice beach.

Fonetické jednotky

- foném (*phoneme*)
  - základní jednotka zvukového systému jazyka
  - foném je *abstraktní věc*, konkretizuje se pomocí *fónů* (viz dále)
  - např. v češtině – 37 fonémů:

    a, a:, b, ts, tS, d, d', dz, dZ, e, e:, f, g, h\, x, i, i:, j, k, l, m, n, n', o, o:, p, r, r', s, S, t, t', u, u:, v, z, Z

- fón (*phone*)
  - řečový zvuk z hlediska jeho fyzikálních charakteristik (zvuková vlna určitého tvaru)
  - bez zařazení k zvukovému systému jazyka
  - jeden foném odpovídá množině fónů
  - alofón určitého fonému = jeden z množiny fónů tohoto fonému
    např. nosit, banka

Fonetická transkripce

- jeden z nejpoužívanějších nástrojů fonetiky
- převod řečového proudu do oddělených, lingvisticky významných symbolických jednotek
- používá se standardních fonetických abeced (viz dále)
- široká × úzká (broad/narrow) transkripce = převod *do fonémů*/*fónů*
- důvody pro tento převod:
  - nedostatečnost písmenného zápisu
    - jedno písmeno → různý zvuk vypít [v] / vpustit [f]
    - jeden zvuk → různá písmena chovat [x] / sháněť [x]
  - mezijazykové variace v písmenném zápisu
    - 'k' → 'c' v latinském canis, 'ch' v italském Chianti
    - 'č' → 'ch' v anglickém cheat, 'ci' v italském ciao
  - jeden foném může být zaznamenán více písmeny

    např. 'f': → 'f' v českém fyzika
    → 'gh' v anglickém laugh
    → 'ph' v řeckém philosophia

Fonetické abecedy IPA a SAMPA

IPA:

- *International Phonetic Alphabet*
- vznikla v roce 1886 v Paříži, od té doby mnoho revizí (poslední 1996)
- speciální znak pro vyjádření každého fónu
- mezinárodně standardní zápis – jsou k dispozici tabulky a fonty
- *Unicode* – speciální IPA znaky v rozsahu U+0250–02AD

SAMPA:

- *Speech Assessment Methods Phonetic Alphabet*
- vznikla v projektu SAM (Speech Assessment Methods) v letech 1987–89
- strojově čitelná fonetická abeceda
- http://www.phon.ucl.ac.uk/home/sampa/

IPA – souhlásky

v americké angličtině

	labial		labio-dental		interdental		alveolar		palatal		velar		glottal	
stops	p	b					t	d			k	g		
fricatives			f	v	θ	ð	s	z	ʃ	ʒ			h	
affricates									tʃ	dʒ				
nasals		m						n				ŋ		
liquids lateral								l						
liquids retroflex								r						
glides		w								j				

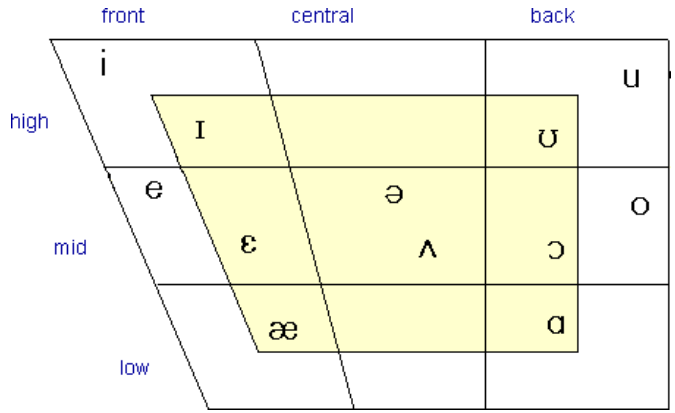
IPA – souhlásky ve slovech

p plate, piece, spin, capital, stop, tramp
t trip, time, winter, retire, wait, front
k kite, climb, character, rocket, back, sink
b bill, brush, sober, ramble, sob, bulb
d dark, drive, redden, ponder, head, hard
g go, grease, rigor, anger, log, iceberg
m man, mile, remorse, ample, climb, harm
n nice, know, enough, cunning, sign, burn
ŋ finger, singer, drunk, rang, thing
θ thank, three, ether, panther, path, birth
ð then, these, feather, breathe
f fit, fly, effort, perform, enough, Ralph
v very, view, every, prevail, love, starve

s ceiling, slim, psychology, Pacific, nasty, pass
z zoo, zipper, hazard, prison, cares, breeze
ʃ shore, sugar, nation, rash, Porche
ʒ (genre), visual, measure, decision, massage
h hat, who, ahead, perhaps
tʃ China, cheap, ritual, teaching, beach, punch
dʒ jump, pidgeon, reject, individual, ridge, engine
l light, look, pillow, applaud, salt, ball, girl
r real, row, around, part, care, hear
w wind, was, await, swim, queen
j yes, use, beyond, beauty, punitive

IPA – samohlásky

v americké angličtině



IPA – samohlásky ve slovech

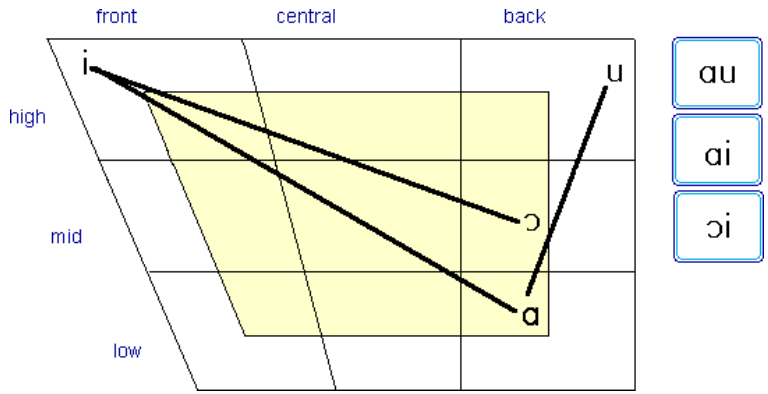
i heed, beat, believe, people, scary
ɪ hid, bit, injure, resist, finish
e hate, bait, great, they, say, neighbor
ɛ head, bet, friend, says, guest
æ had, bat, laugh, calf, language

u food, boot, pool, through, who, sewer
ʊ hood, book, pull, put, would
o hole, boat, sew, know, so
ɔ bought, law, wrong, stalk
ɑ pot, "la", stocking, father, rob

ə above, around, sofa, police
ʌ bus, rush, under, other

IPA – dvojhlásky

v americké angličtině



aɪ find, high, aisle, quiet, ride
aʊ house, crown, around, flower, how
ɔɪ boy, enjoy, Freud, avoid, join

Prozodie

- tzv. suprasegmentální rysy
- popisuje řečový proud spolu s přepisem do fonémů
- vyjádřena pomocí dalších lokálních fyzikálních charakteristik výsledné zvukové vlny:
  - délka fonému
  - intonace věty – vzor pro hladinu základní frekvence (*pitch*)
  - tón – v některých (tzv. tónových) jazycích určuje význam
  - přízvuk – v přízvukových jazycích ovlivňuje délku, hlasitost a tón slov
- kvalitní výpočet prozodie = přirozenost syntetizované řeči

Text-to-Speech systémy

- syntéza řeči – převod psaného textu na (digitální) zvuk
- TTS, *Text-to-Speech*
- dvě hlavní části
  1. jazykový modul, NLP modul
     vstup = text
     výstup = fonémy + prozodická informace
     označována také jako TTP, *Text-to-Phoneme*
  2. modul zpracování signálu, DSP (Digital Signal Processing) modul
     vstup = výstup z NLP modulu
     výstup = zvukový soubor

Příklady dat pro českou syntézu s MBROLA

- pravidla pro přepis do fonémů

```
CLASS SA    [aáeéěiíoóuúůyý]   # samohlásky
CLASS ZPS   [bdďgvzžhcč]       # znělé párové souhlásky
CLASS NPS   [ptťkfsšhcč]       # neznělé párové souhlásky
[[ dě ]] → ď e
[[ b ]] (_|NPS|ZPS_) → p
[[ p ]] ZPS → b
```

- vstup pro MBROLu – text "shání tě též muž"

```
_ 200 0 132          i: 93 0 114          S 81 0 114
z 57 0 115           t' 27 0 120          m 43 0 120
h 45                 e 50 0 114           u 61
a: 137               t 31 0 120           S 110
n' 75 0 132          e: 102               #
```

- zvuková databáze cz2 – 37 fonémů, 1442 difónů
  nutné ručně "nařezat" všechny difóny

Příklady TTS systémů

- české
  - Epos – z 90. let, Karlova univerzita a ČAV, nejlepší český open source
  - Demosthenes – FI MU Brno, laboratoř LSD
    slabiková syntéza, základní prozodie
  - ARTIC (ARtificial Talker In Czech) – ZČU Plzeň, DEMO
    obsahuje i "Talking head" vizuální část
  - CS-Voice 97 – komerční, Frog Systems, pro Windows
- zahraniční
  - Festival – z Edinburghu, GPL, hodně jazyků, projet Festival Czech
  - MBROLA – difónová syntéza MBR-PSOLA, řeší DSP část
    Mikuláš Piňos, DP 2000 – česká DB pro MBROLu, `text2phone` v Perlu
  - mnohé další – HADIFIX, SVOX, Bell Labs, AT&T, ...

Syntéza a rozpoznávání řeči

Pavel Cenek, Aleš Horák

E-mail: hales@fi.muni.cz
http://nlp.fi.muni.cz/poc_lingv/

Obsah:

Syntéza řeči

- Text to Speech, TTS
- Konverze textu do mluvené podoby
- V ideálním případě by měla syntetizovaná řeč znít tak, jako kdyby daný text přečetl člověk
- Probíhá obvykle ve 4 fázích
  - Normalizace textu
  - Fonetický přepis
  - Prozodický přepis
  - Akustické modelování

Normalizace textu

- Rozčlenění textu na věty
- Rozvinutí zkratek, měrných jednotek, čísel apod.

"130895" {
- číslo
- telefonní číslo
- datum
- ...
}

Fonetický přepis

- Převede předzpracovaný text do fonetické podoby (tj. do tvaru, který popisuje výslovnost daného textu)
- Mezinárodní fonetická abeceda (IPA) – v češtině cca 40 fonémů
- Fonetický přepis češtiny musí zohlednit např.
  - Spodoba znělosti (`včela`/`fčela`, `dub`/`dup`)
  - Krajové zvyky (např. `shoda`/`zhoda` nebo `schoda`).
- Problémy přináší přepis cizích vlastních jmen a cizích slov obecně (např. `faux pas` nebo francouzská vlastní jména)
- Dvě základní metody
  - Fonetický přepis založený na pravidlech (např. pro češtinu funguje dobře)
  - Fonetický přepis pomocí výslovnostních lexikonů
- Obě metody lze kombinovat

Prozodický přepis

- Prozodie je důležitá pro přirozenost řeči, ale např. u *tonálních jazyků* silně ovlivní i porozumění
- Obohacení textu o informace (viz SSML dále), které zajistí, že výsledná řeč bude znít přirozeně
- Zejména popis intonace, tempa řeči, pauz a informace o lexikálním přízvuku
- Emoce
  - člověk je při projevu používá
  - výzkum syntézi s emocemi je o dost složitější

Speech Synthesis Markup Language (SSML)

- Doporučení W3C (jako HTML, XML, ...) – standardní způsob pro doplnění fonetiky a prozodie do textu
- Pokrývá první 3 fáze syntézy řeči (normalizace, fonetický přepis, prozodie)
- `<say-as>` – explicitní určení typu dat (viz Normalizace)
- `<phoneme>` – fonetický přepis textu
- `<voice>` – změna hlasu (atributy *věk*, *muž/žena*, . . .)
- `<emphasis>` – přidání/odebrání důrazu
- `<break>` – vložení/zrušení pauzy
- `<prosody>` – ovlivnění prozodie (výška hlasu, kontura, rychlost, hlasitost atd.)

Speech Synthesis Markup Language (SSML) – příklad

```
<?xml version="1.0" encoding="UTF-8"?>
<vxml version="2.0" xmlns="http://www.w3.org/2001/vxml">
<form>
 <block>
  <prompt>
   <voice gender="male"><emphasis>Hello</emphasis> Jane.</voice>
   <voice gender="female"><emphasis>Hello</emphasis> Mike,
        how <emphasis>are</emphasis> you?</voice>
   <voice gender="male">I am fine. And how are
        <emphasis>you</emphasis> Jane?</voice>
   <voice gender="female">Not bad.</voice>
   <voice gender="male">OK, Goodbye.</voice>
   <voice gender="female"><emphasis>Goodbye</emphasis>
        Mike.</voice>
  </prompt>
 </block>
</form>
</vxml>
```


Akustické modelování

- Generování výsledného akustického signálu z předzpracovaného textu
- Dva základní přístupy
  - syntéza řeči v časové oblasti
  - syntéza řeči ve frekvenční oblasti

Syntéza řeči v časové oblasti

- = konkatenativní syntéza
- Výsledná řeč se skládá z vybraných, dopředu namluvených segmentů řeči (difónů, trifónů, slabik apod.)
- Relativně jednoduché na implementaci
- Nutnost vytvoření rozsáhlé databáze segmentů (koartikulace, např. 'á' zní jinak v `táta` a `máma`):
  - difóny – `t á t a`
  - trifóny – `t á t a`
  - kombinace – heterogenní segmenty (někdy difóny, trifóny i celá slova)
- Dochází k deformaci segmentů jejich spojováním a aplikací prozodických pravidel – "tajemství" komerčních aplikací

Syntéza řeči ve frekvenční oblasti

2 hlavní přístupy:

- Modelování hlasového ústrojí
  - Generovaný zvuk závisí na parametrech tohoto hlasového ústrojí.
  - ⊕ Velká flexibilita (nový hlas lze vytvořit pouhou změnou parametrů)
  - ⊖ Velmi náročné výpočty (řeší se fyzikální rovnice modelující situaci ve vokálním traktu, diferenciální rovnice, větš. degradují na válce/koule, ale stejně moc náročné) ⇒v praxi se téměř nepoužívá
- Formantová syntéza
  - Modelování (jen) *hlavních* akustických rysů řečového signálu
  - Zdroj/filtr model – zdroj generuje základní tón pro znělé části řeči a šum pro neznělé části řeči a filtry modifikují zvukové spektrum a napodobují tak hlavní funkce lidského vokálního traktu
  - Zdroj i filtr jsou řízeny množinou fonetických pravidel → syntéza založená na pravidlech
  - Lze počítat v reálném čase
  - Mnohem menší data než u konkatenativní syntézy → vhodné i pro PDA

TTS systémy ve světě

nejčastější použití – telefonní systémy

- ©Nuance (`http://www.nuance.com/`) ▸ DEMO
- ©Loquendo (`http://www.loquendo.com/`) ▸ DEMO
- ©Acapela group (`http://www.acapela-group.com/`) ▸ DEMO
  - založena v roce 2004 třemi společnostmi, jedna z nich autor Mbroly
- ©IBM (`http://www.research.ibm.com/tts/`)
- ©AT&T (`http://www.research.att.com/projects/tts/`)
- Festival (`http://www.cstr.ed.ac.uk/projects/festival/`)
- Mbrola (`http://tcts.fpms.ac.be/synthesis/mbrola.html`)
- FreeTTS (`http://freetts.sourceforge.net/`)

České TTS systémy

- EPOS TTS (`http://sourceforge.net/projects/epos`) ▸ DEMO
  - Česká akademie věd + Karlova univerzita
- Demosthenes, Popokatepetl
  - LSD FI
- ERIS TTS (`http://www.speechtech.cz/`), heterogenní segmenty ▸ DEMO
  - SpeechTech, s.r.o. + katedra kybernetiky FAV ZČU
    © verze je nejlepší český
- Český hlas pro Mbrolu
  - Mikuláš Piňos, NLP lab FI

Rozpoznávání řeči

- Automatic Speech Recognition, ASR
- Konverze řeči na text
  - Výstupem je většinou množina hypotéz spolu s pravděpodobností správnosti dané hypotézy. K výběru správné hypotézy se běžně využívají jazykové modely
- Lze zhruba rozdělit na
  - Rozpoznávání izolovaných slov – slyšitelná pauza mezi slovy
  - Rozpoznávání kontinuální řeči – plynulá řeč (řeč školeného mluvčího nebo čtený text)
  - Rozpoznávání spontánní řeči – přeřeky, pauzy, začátky vět (*false-starts*)

Rozpoznávání řeči pokrač.

- Diktovací stroje (např. Dragon Naturally Speaking)
  - Schopné rozpoznat cokoliv
  - *N*-gramové statistické jazykové modely
  - Závislé na mluvčím (je potřeba je natrénovat)
- Rozpoznávače založené na gramatikách
  - Rozpoznají jen fráze popsané (regulární) gramatikou (gramatika = jazykový model)

    S → "Jedu do " MESTO
    MESTO → "Prahy" | "Brna"
  - Nezávislé na mluvčím – telefonní aplikace
  - Speech Recognition Grammar Specification (SRGS)
    - standard W3 konzorcia, à la BNF
    - existují 2 notace – XML a šipková pro čtení
    - dá se do ní dát i "význam" vstupu

Rozpoznávání řeči pokrač.

Probíhá obvykle ve 3 fázích:

1. Vstup signálu
   - Amplituda akustického vlnění je snímána v pravidelných intervalech a uložena ve formě celého čísla (digitalizace a vzorkování signálu)
2. Vytvoření akustických charakteristik signálu (akustické vektory)
   - Snižuje variabilitu a odstraňuje redundanci (řeč 300 000× redundantní)
   - Počítají se rozdělením na segmenty 10–40 ms, ze kterých se odečítají charakteristiky jako je počet průchodů nulou nebo prvních 12 koeficientů FFT (cca 40 čísel, není přesně dané které, ale výběr velice ovlivní výsledek)
3. Porovnávání vektorů parametrů
   - K získané sekvenci vektorů parametrů se hledá co nejpodobnější sekvence známých, předem naučených, vektorů reprezentující např. fonémy, trifóny, slabiky, celá slova apod.

Porovnávání vektorů parametrů

- Algoritmus borcení časové osy (dynamic time warping, DTW)
  - odstraňuje časové nerovnoměrnosti v akustickém signálu
- Skryté Markovovy modely (*Hidden Markov Models, HMM*)
  - Pravděpodobnostní konečné automaty
  - V každém okamžiku je hlasové ústrojí v určitém stavu a může s určitou pravděpodobností přejít do jednoho z následujících stavů
  - Jako doplněk se mohou využít neuronové sítě
  - Je nejprve potřeba natrénovat za pomocí dat z řečového korpusu

ASR systémy ve světě

- ©Nuance (`http://www.nuance.com/`)
- ©Loquendo (`http://www.loquendo.com/`)
- ©LumenVox (`http://www.lumenvox.com/`)
- ©IBM ViaVoice (`http://www306.ibm.com/software/voice/viavoice/`)
- Sphinx (`http://cmusphinx.sourceforge.net/`)

České ASR systémy

- Laboratoř počítačového zpracování řeči na Fakultě mechatroniky Technické univerzity v Liberci (`http://itakura.kes.vslib.cz/kes/`)
- ERIS ASR (`http://www.speechtech.cz/`)
  - SpeechTech, s.r.o. + katedra kybernetiky FAV ZČU
- Speech@FIT VUT Brno (`http://www.fit.vutbr.cz/research/groups/speech/`)
  - keyword spotting – jestli se vyskytlo dané slovo v běžné řeči

Související technologie

- Dialogové systémy
  - Počítačové systémy komunikující s uživatelem pomocí přirozeného jazyka
  - Využívají ASR a TTS jako své komponenty
- Rozpoznávání mluvčího
  - identifikace mluvčího – určení, který z registrovaných mluvčích pronesl danou větu
  - verifikace mluvčího – akceptování nebo odmítnutí identity mluvčího
- Identifikace mluveného jazyka
  - fonémicko-fonetický rozpoznávač pro každý rozpoznávaný jazyk – sledují se fonémy specifické pro každý jazyk
  - daná promluva je zpracována všemi rozpoznávači a jako jazyk dané promluvy je zvolen jazyk, jehož rozpoznávač dosáhl nejvyššího skóre

TTS Demo

- `http://www.nuance.com/realspeak/demo/`
- `http://actor.loquendo.com/actordemo/default.asp?language=en`
- `http://demo.acapela-group.com/`
- `http://epos.ure.cas.cz/`
- `http://speechtech.cz/demo.php`

Morfologie, morfologická analýza

Aleš Horák

E-mail: hales@fi.muni.cz
http://nlp.fi.muni.cz/poc_lingv/

Obsah:

Morfologie

- nauka o stavbě a tvorbě slov (v daném jazyce)
- morfém – nejmenší jednotka, která může nést význam

pří-lež-it-ost-n-ými

základní tvar = příležitostný
příd. jméno, rod muž. živ., neživ., žen. nebo stř., 7. pád, mn. č.

pří – prefix (*blízko*)
lež – lexikální kořen (*ležet*)
it – adjektivní derivační sufix (*ten, který*)
ost – substantivní derivační sufix (*ta skutečnost, že*)
n – adjektivní derivační sufix (*charakteristický pro*)
ými – gramatický afix (*instrumentál plurálu*)

Dělení morfémů

dělení používané zejména v analytických jazycích (angličtina):

- morfémy obsahové (*content*) × funkční (*function*)
- morfémy volné (*free*) × vázané (*bound*)

dělení používané zejména ve flektivních jazycích (čeština):

- kořeny – nesamostatné morfémy nesoucí elementární lexikální významy
- afixy, které se dále dělí
  - podle funkce:
    - *gramatické/flekční*
    - *slovotvorné/derivační*
  - podle postavení vzhledem ke kořeni:
    - *prefixy* – morfémy stojící před kořenovým morfémem (pod-, anti-, v-)
    - *sufixy* – morfémy připojované za kořenové morfémy (-ík, -izmus, ...)
    - *postfixy* – slovotvorné morfémy připojované až za gramatický sufix (kdosi, kohokoli, ...)
    - *circumfix* – morfémy připojované "kolem" základu, není v češtině
    - *infix*, *interfix* – morfémy vsazované dovnitř slova (mal-il-inký, velk-o-město, ...)

Základní lingvistické termíny v morfologii

- slovní druh – podstatné jméno (*substantivum*), přídavné jméno (*adjektivum*), sloveso (*verbum*), příslovce (*adverbium*), ...
- pád – *nominativ*, *genitiv*, *dativ*, *akuzativ*, *vokativ*, *lokál*, *instrumentál*
- číslo – *singulár*, *plurál*
- rod – 4 rody, mužský (*masculinum*) životný a neživotný (*animativní* a *inanimativní*), ženský (*femininum*) a střední (*neutrum*)
- slovotvorba – předpona (*prefix*), přípona (*sufix*), předpona nebo přípona (*afix*)
- základní tvar slova – *lemma* (mn.č. *lemmata*)
- ohýbání slov (*flexe*) – skloňování (*deklinace*) a časování (*konjugace*)
- odvozování – *derivování*

Procesy tvoření slov

dělení podle třech základních procesů tvoření slov:

- flektivní morfologie – popisuje strukturu slovních tvarů pomocí flexe (ohýbání – skloňování a časování)

1	pes	2	psa	3	psovi, psu	4	psa
5	pse	6	psovi, psu	7	psem		
1	*psové*, psi	2	psů	3	*psum*, psům	4	psy
5	*psové*, psi	6	*psách*, psech	7	psy, *psama*		

- derivativní (derivační) morfologie – zkoumá odvozování slov

  mýdlo: mydl-ář, mydl-ina, mýdel-ný, mydl-it, mýdél-ko

- kompozicionální (kompoziční) morfologie – zachycuje tvoření slov pomocí skládání

  ohni-vzdorný, pravdě-podobný, oka-mžik
  tlako-měr, vodo-pád, děje-pis
  samo-obsluha, malo-město, bylo-žravý

Derivační morfologie – vztah fundace

fundace – základní slovotvorný vztah

- slova neutvořená, prvotní, fundující – nemůžeme vysvětlit pomocí jiných slov jazyka

  voda, hlava, vejce

- slova utvořená, fundovaná – opírají se o slova základová

  trávník, růžový, učitel

- fundace – spojení slova základového se slovem utvořeným

  mladý → mladík

- slovotvorná řada – opakované odvození až k prvotnímu slovu

  rybníkářský → rybníkář → rybník → ryba

Derivační morfologie – vztah fundace

- slovotvorný svazek/hnízdo – souhrn slov fundovaných jedním slovem

  mýdlo → mydl-ář, mydl-ina, mýdel-ný, mydl-it, mýdél-ko

- slovotvorná čeleď – souhrn všech příbuzných slov (se stejným kořenem)

  les

  - pra-les → pra-les-ní
  - les-ní
    - lesn-ík → lesnic-ký → lesnic-tví
    - lesn-ice
    - nad-lesní
  - les-ík → lesíč-ek

Morfologická analýza

- rozpoznávání slovních tvarů
- nástroj se nazývá morfologický analyzátor (*Part-of-Speech tagger*)
- provádí lemmatizaci – přiřazuje k rozpoznaným slovním tvarům základní tvar (lemma)
- charakterizuje morfo-syntaktické vlastnosti nalezených slovních tvarů:

```
příležitostného
1. <s> příležitostn-ého (mladý GcAa)
   <l> příležitostný
   <c> adje      Man sg #4
   <c> adje      Man,Min,Neu sg #2
```

- kvalita morfologické analýzy ovlivňuje všechny následující analytické roviny

Lexikální a gramatické kategorie

Morfologie klasifikuje (značkuje, *tag*) slovní tvary jednotlivých kategorií.
Kategorie pro účely analýzy můžeme dělit na dvě skupiny:

- lexikální kategorie – pojmenovávají věci, akce, myšlenky
  podstatná jména, slovesa, přídavná jména, příslovce, ...
- gramatické kategorie – vyjadřují vztahy mezi ostatními větnými členy
  předložky, spojky, částice, anglické členy, ...

jazyky s
- jednoduchou morfologií (angličtina) – několik desítek kategorií (*POS – Part of Speech* – slovní druhy)
- bohatou morfologií – hierarchický systém, kde vedle základních slovních druhů určujeme nejrůznější subklasifikace (pád, číslo, rod, osoba, druhy příslovcí,...) – celkově tisíce značek

Anglické gramatické morfémy

`-s`	3. osoba, jedn.č., přítomný čas
`-ed`	minulý čas
`-ing`	průběhový
`-en`	příčestí minulé trpné
`-s`	množné číslo
`-'s`	přivlastnění
`-er`	2. stupeň přídavného jména (komparativ)
`-est`	3. stupeň přídavného jména (superlativ)

Brillův značkovač

- učí se podle trénovacích dat:
  1. přiřaď nejčastější značku
  2. zkontroluj, kde jsou chyby (podle trénovacích dat)
  3. ohodnoť pravidla pro opravu chyb → vyber nelepší → oprav zpětně chybné značky
  4. opakuj, dokud se daří odvozovat dobrá pravidla
- používá učení založené na transformacích (*transformation-based learning*)
- analogie – malování obrazu: nejprve pozadí a pak přes něj stále drobnější detaily
- značkuje 36 různých POS značek
- úspěšnost – přes 90 %

Brillův značkovač – příklad

věta:	zlatý standard:	podle frekvence:	P1:	P2:
The	at	at		
President	nn-tl	nn-tl		
said	vbd	vbd		
he	pps	pps		
will	md	md		
ask	vb	vb		
Congress	np	np		
to	to	to		
increase	vb	nn	vb	
grants	nns	nns		
to	in	to	to	in
states	nns	nns		
for	in	in		
vocational	jj	jj		
rehabilitation	nn	nn		
.	.	.		

P1: Replace `nn` with `vb` when the previous word is `to`
P2: Replace `to` with `in` when the next tag is `nns`

Brillův značkovač – příklad

```
Loading tagged data...
Training unigram tagger: [accuracy: 0.820940]
Training Brill tagger on 37168 tokens...

Iteration 1: 1482 errors; ranking 23989 rules;
  Found: "Replace POS with VBZ if the preceding word is tagged PRP"
  Apply: [changed 39 tags: 39 correct; 0 incorrect]

Iteration 2: 1443 errors; ranking 23662 rules;
  Found: "Replace VBP with VB if one of the 3 preceding words is tagged MD"
  Apply: [changed 36 tags: 36 correct; 0 incorrect]

Iteration 3: 1407 errors; ranking 23308 rules;
  Found: "Replace VBP with VB if the preceding word is tagged TO"
  Apply: [changed 24 tags: 23 correct; 1 incorrect]
...
Iteration 21: 1128 errors; ranking 20569 rules;
  Found: "Replace VBD with VBN if the preceding word is tagged VBD"
  [insufficient improvement; stopping]

Brill accuracy: 0.835145
```


Algoritmický popis české formální morfologie

v češtině nestačí pravidla podle obecných morfémů – je potřebné mít lexikon, který ke každému *kmenu* obsahuje jeho přiřazení ke *vzoru*

morfologické (tvaroslovné) paradigma – soubor tvarů ohebného slova vyjadřující systém jeho mluvnických kategorií
vzor – reprezentace tvaroslovného paradigmatu paradigmatem určitého konkrétního slova

Algoritmický popis:

1. definice koncovkových množin
2. definice vzorů prostřednictvím vzorových slov rozdělených na:
   - neměnná část vzorového slova – kmenový základ
   - proměnlivé části vzorového slova – intersegmenty
   - koncovkové množiny obsahující utříděné seznamy všech přípustných koncovek vzorového slova spolu s jejich gramatickými významy

popis vzoru = formální pravidlo, které specifikuje přípustné kombinace těchto komponent (segmentů) ohebného slova

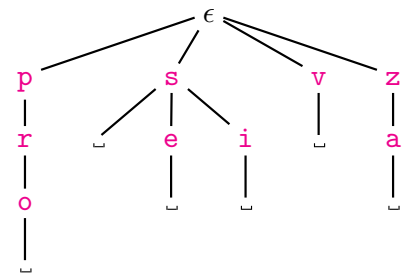
Segmentace slova pro potřeby algoritmického popisu

- segmentace od začátku slova
  a) segmenty se snadno formalizovatelným výskytem vázaným gramaticky:
     - negativní prefix `ne-`
     - superlativní prefix `nej-`
     - futurální slovesný prefix `po-`
  b) segmenty s nesnadno formalizovatelným výskytem vázáným sémanticky:
     - prefixy
     - první členy kompozit
     - prefixy `ni-`, `ně-` zájmen neurčitých a záporných
- segmentace od konce slova
  a) rozdělení slovního tvaru na kmen a koncovku
  b) další segmentace kmene na kmenový základ a intersegment

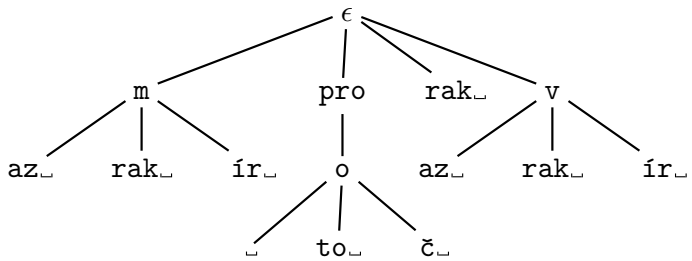
Efektivní implementace morfologického lexikonu – trie

struktura trie:

- uspořádaný strom nad danou abecedou A
- v každém uzlu je různé písmeno z abecedy A
- klíč je v trie uložen jako cesta od kořene
- výhody:
  - sdílení společných prefixů
  - v každém případě nalezení nejdelšího shodného prefixu



Eliminace cest v trie



Jiná efektivní implementace ML – konečný automat

- BP, Radovana Štancela
- použití mírně pozměněných knihoven pro práci s KA od Jana Daciuka
- vstupní data se generují ze slovníku ajky převedeného do tvaru "`slovo<TAB>lemma<TAB>značka`" (cca 33 mil. řádků)

Abcházce	Abcházec	k1gMnPc4
Abcházce	Abcházec	k1gMnSc2
Abcházce	Abcházec	k1gMnSc4
Abcházcem	Abcházec	k1gMnSc7
Abcházci	Abcházec	k1gMnPc1
Abcházci	Abcházec	k1gMnPc5
Abcházci	Abcházec	k1gMnPc7
Abcházci	Abcházec	k1gMnSc3
Abcházci	Abcházec	k1gMnSc6
...		

Jiná efektivní implementace ML – konečný automat

- data se dále upravují pro KA – `slovo+zkr.lemma+značky`:

  Abcházce+ACec+k1gMnPc4, k1gMnSc2, k1gMnSc4
  Abcházcem+ADec+k1gMnSc7
  Abcházci+ACec+k1gMnPc1, k1gMnPc5, k1gMnPc7, k1gMnSc3, ...
  ...

- v lemmatu – 1. písmeno je počet znaků, které se odtrhnou jako předpona, 2. písmeno je počet znaků, které se trhají od konce a ostatní znaky se přidají
- tím se sníží počet řádků na 6.7 mil. řádků, ze kterých se přímo generuje (a minimalizuje) konečný automat
- výsledný slovník má 4.3MB
- rychlost je cca o 1/4 lepší než u trie, velikost řádově srovnatelná

České morfologické analyzátory

- ajka
  - Radek Sedláček, FI MU Brno
  - `http://nlp.fi.muni.cz/projekty/ajka/`
  - značky jsou řetězce dvojic atribut–hodnota
  - napsaný v C
  - využívá struktury trie
  - 390 000 základních tvarů, 6 300 000 různých slovních tvarů, 15 000 různých značek, slovník 3.13MB
  - rychlost analýzy – cca 18 000 slov/s
  - data se v současnosti editují pomocí nástroje `i_par` od Marka Vebera
- pražský morfologický analyzátor
  - Barbora Hladká, Jan Hajič a jeho tým, ÚFAL MFF UK Praha
  - `http://ufal.mff.cuni.cz/czech-tagging/`
  - používá poziční značky
  - "free" část napsaná v Perlu, menší slovník (cca 76 000 základních tvarů, 6 000 koncovek)

Pražský morfologický analyzátor – poziční značky

pozice	kategorie	anglicky	česky
1	POS	Part of Speech	Slovní druh
2	SUBPOS	Detailed Part of Speech	Slovní poddruh
3	GENDER	Agreement Gender	Rod
4	NUMBER	Agreement Number	Číslo
5	CASE	Case	Pád
6	POSSGENDER	Possessor's Gender	Rod vlastníka
7	POSSNUMBER	Possessor's Number	Číslo vlastníka
8	PERSON	Person	Osoba
9	TENSE	Tense	Čas
10	GRADE	Degree of Comparison	Stupeň
11	NEGATION	Negation (by prefix)	Negace
12	VOICE	Voice	Slovesný rod
13	RESERVE1	Reserved for future use	Rezerva
14	RESERVE2	Reserved for future use	Rezerva
15	VAR	Variant, Style, Register	Varianta, styl

Pražský morfologický analyzátor – příklad

- vstup:

  Prezident rezignoval na svou funkci.

- výstup:

```
<csts>
<f cap>Prezident<MMl>prezident<MMt>NNMS1-----A----
<f>rezignoval<MMl>rezignovat_:T<MMt>VpYS---XR-AA---
<f>na<MMl>na<MMt>RR--4----------<MMt>RR--6----------
<f>svou<MMl>svůj-1_^(přivlast.)<MMt>P8FS4---------1
   <MMt>P8FS7---------1
<f>funkci<MMl>funkce<MMt>NNFS3-----A----
   <MMt>NNFS4-----A----<MMt>NNFS6-----A----
<D>
<d>.<MMl>.<MMt>Z:-------------
</csts>
```


Značky morfologického analyzátoru `ajka`

značka = řetězec dvojic *atributHodnota*: `k1gNnSc3`

`k`	slovní druh	`1` – podst.jméno, `2` – př.jméno, ...
`g`	rod	`M` – muž.životný, `I` – muž.neživotný, ...
`n`	číslo	`S` – jednotné, `P` – množné, `D` – duál
`c`	pád	`1`, `2`, ..., `7`
`p`	osoba	`1`, `2`, `3`
`m`	slovesný způsob	`F` – infinitiv, `R` – imperativ, ...
`a`	slovesný vid	`P` – dokonavý, `I` – nedokonavý
`t`	typ příslovcí	`T` – času, `L` – místa, `M` – způsobu, ...
`x`	typ spojky	`C` – souřadící, `S` – podřadící

Morfologický analyzátor `ajka` – příklad

- dávkově

```
Prezident <l>prezident <c>k1gMnSc1
rezignoval <l>rezignovat <c>k5eApMnStMmPaI <c>k5eApInStMmPaI
na <l>na <c>k7c4 <c>k7c6
svou <l>svůj <c>k3xOgFnSc4p3 <c>k3xOgFnSc7p3
funkci <l>funkce <c>k1gFnSc3 <c>k1gFnSc6 <c>k1gFnSc4
```

- interaktivně

```
<s> ne=snesiteln=ého== (1023)
<l>snesitelný
<c>k2eNgMnSc2d1
<c>k2eNgMnSc4d1 ...
```

- všechny tvary (`ajka -a`)

```
<s> =p=es== (1148)
<l>pes
<c>k1gMnSc1
pes psům psů psovi psem psa psu psy psech pse psi psové
```

Morfologický analyzátor ajka – webové rozhraní

http://nlp.fi.muni.cz/projekty/wwwajka/

Výsledek morfologické analýzy - interaktivní režim

(*) - Vypiš všechny odvozené tvary

Analyzovaný tvar: stát

Základní tvar	Segmentace	Číslo vzoru	Kategorie
stát (*)	=st=á=t=	1422-stát	k5eAaImP
stát (*)	=st=á=t=	1587-vstát	k5eAaPmP
stát (*)	=stát===	874-most	k1gInSc1
			k1gInSc4

Analyzuj text: Analyzuj

Morfologická analýza - interaktivní režim Morfologická analýza - dávkový režim

Syntaxe – gramatiky a syntaktické struktury

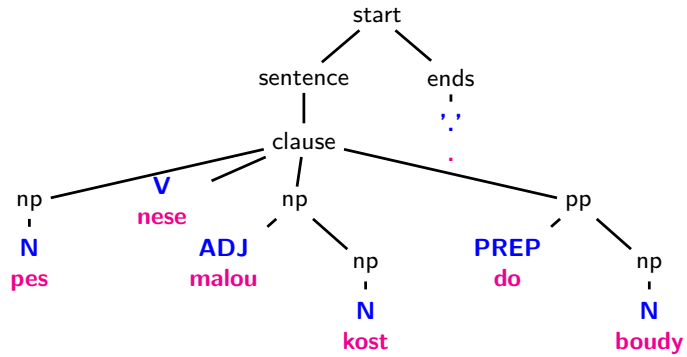
Aleš Horák

E-mail: hales@fi.muni.cz
http://nlp.fi.muni.cz/poc_lingv/

Obsah:

Syntaxe, syntaktická analýza

- syntaxe – charakterizace dobře utvořených kombinací slovních tvarů do věty nebo fráze
- pomocí gramatických pravidel
- výstup ze syntaktické analýzy (např. derivační strom) tvoří často vstup pro analýzu sémantickou



Typy gramatik

gramatiky:

- regulární (regular) neterminál → **terminál**[neterminál]

  S → **a**S
  S → **b**

  ekvivalentní síle konečných automatů, neumí $a^n b^n$

- bezkontextové (context-free) neterminál → cokoliv

  S → **a**S**b**

  ekvivalentní síle zásobníkových automatů, umí $a^n b^n$, neumí $a^n b^n c^n$

- kontextové (context-sensitive) – víc neterminálů na levé straně; na levé straně se jejich počet "zmenšuje"

  ASB → AA**a**BB

  umí $a^n b^n c^n$

- rekurzivně vyčíslitelné (recursively enumerable) – bez omezení ekvivalentní síle Turingova stroje

přirozený jazyk byl dlouho pokládán za bezkontextový → nyní prokázáno, že obsahuje kontextové prvky

Syntaktická analýza programovacích × přirozených jazyků

- počítačové programy a přirozené jazyky sdílí teorii formálních jazyků a praktický zájem o efektivní algoritmy analýzy
- ALGOL 60 – první programovací jazyk popsaný pomocí Backus-Naurovy formy (BNF)

```
<if_statement> ::= if <boolean_expression> then
                       <statement_sequence>
                   [ else
                       <statement_sequence> ]
                   end if ;
```

- dokázalo se, že BNF je ekvivalentní CFG (1962) → podnítilo výzkum formálních jazyků z hlediska jazyků přirozených

Gramatiky přirozeného jazyka

- konkrétní popis gramatiky přirozeného jazyka je velmi složitým úkolem
- kontrast s faktem, že rodilí mluvčí nemívají potíže s pochopením významu vět
- asi nejstarší formální popis jazyka – gramatika sanskrtu od indického učence Paniniho

संस्कृत भारती

- vznikla cca 400 př.n.l.
- dochovaná v rituálních védických textech
- gramatika podobná BNF (Backus-Naurově formě)
- používala bezkontextových i kontextových pravidel, obsahovala asi 1700 termů
- zabývala se z větší části morfologií, nikoliv syntaxí, neboť pořádek slov je v sanskrtu dosti volný
- toto dílo bylo evropské škole obecné lingvistiky, která má kořeny v řecké a římské tradici, neznámé až do 19. století

Základní termíny

- fráze (*phrase*) – jednotka jazyka větší než slovo, ale menší než věta např. *jmenná fráze*, *slovesná fráze*, *adjektivní fráze* nebo *příslovečná fráze*
- lexikální symbol, lexikální kategorie (*lexical category*) tzv. pre-terminál speciální neterminál gramatiky, který se přímo přepisuje na terminálový řetězec znaků, tj. pravidla tvaru $X \to w$

N	→	pes	|	člověk	| dům ...
V	→	nese	|	chodit	| psal ...
ADJ	→	...			
PREP	→	...			
ADV	→	...			

označuje všechny slova, která odpovídají určitému lexikálnímu symbolu (všechna podstatná jména, přídavná jména, ...)

Základní termíny – pokrač.

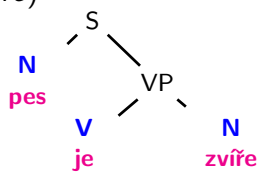
- frázová kategorie (*phrasal category*)
  neterminální symbol gramatiky, který nevyjadřuje lexikální kategorii

ADJP	→	ADJP	ADJ
NP	→	ADJP	N
VP	→	V	NP
S	→	NP	VP

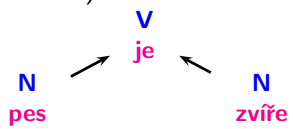
- větný člen (*constituent*) lexikální nebo frázová kategorie

Základní termíny – pokrač.

- větná struktura (*sentence structure*) – strukturovaný popis větných členů
  - povrchová struktura (*surface structure*)
    derivační/složkový strom jako výsledek bezkontextové (CF) analýzy



  - závislostní struktura (*dependency structure*)
    zobrazuje závislosti mezi větnými členy



- hloubková struktura (*deep structure*) – sémantická interpretace fráze. Popisuje role větných členů (agens, patiens, donor, cause, ...)

Složkový a závislostní přístup

dva základní způsoby zadávání gramatik

složkový přístup:

- skupiny slov tvoří větné jednotky, které jsou označovány jako fráze, a jako větné členy (*složky*, *constituents*) formují větu
- např.

  podstatné jméno – součást jmenné fráze (noun phrase – NP)
  jmenná fráze spolu s předložkou – tvoří předložkovou frázi (prepositional phrase – PP)

- syntaktická struktura věty je zachycována jako složkový strom

Složkový a závislostní přístup – pokrač.

závislostní přístup:

- jeden člen vazby je označován jako řídící, druhý jako závislý
- např.

  přídavné jméno závisí na řídícím podstatném jménu

- syntaktická struktura věty je zachycována pomocí závislostního stromu:
  - *uzly* odpovídají elementárním jednotkám vstupu (často slovům)
  - *hrany* označují vztahy závislosti mezi elementárními jednotkami
- závislost není relací mezi jednotlivými slovy, ale obecně relací mezi jedním slovem a frází řízenou druhým slovem. např.

  vazba mezi konkrétním slovesem a podmětem
  nebo vazba mezi slovesem a předmětem věty

  technicky vzato, závislostní relace je vztahem mezi uzly a podstromy (uzlem a všemi uzly, které na tomto uzlu závisí)

Složkový a závislostní přístup – pokrač.

- jen zřídka se používá čistě složkový či striktně závislostní přístup
- ve složkovém jsou závislosti zpravidla vyjádřeny přidáním označení, která složka je řídící pro danou frázi
- závislostní strom bývá doplněn o informaci určující lineární precedenci
- je možné pak mezi těmito přístupy výsledek převádět

Uzly syntaktického stromu

označení uzlu (název neterminálu):

- gramatická role (gramatická funkce)
  - charakterizují vztahy mezi větnými složkami na povrchové úrovni
  - určujeme, zda daný větný člen je NP v roli podmětu, NP v roli předmětu, ADVP určující lokaci atd.
  - v češtině (a jazycích se systémem gramatických pádů) pomáhá k určení gramatické role právě informace o pádu
  - ovšem přiřazení gramatických rolí ke gramatickým pádům a naopak není zdaleka jednoznačné.
- tematická role (též hloubkový/sémantický pád)
  - na rozdíl od gramatické role se jedná o sémantickou kategorii
  - určujeme např.:
    - Agens – kdo je životným *původcem* nějaké cílevědomé činnosti
    - Patiens – co hraje roli entity, na kterou *se působí*
    - Donor – osoba, která *dává*
    - Cause – entita, která *způsobuje*, že je něco děláno
  - opět neexistuje jednoznačná vazba mezi gramatickými a tematickými rolemi (viz např. aktivní a pasivní konstrukce, kdy je stejná tematická role realizována podmětem i předmětem)

Příznaky a příznakové struktury

informace v uzlu syntaktického stromu:

- příznaky/rysy (*features*) – zaznamenávají syntaktické nebo sémantické informace o slovu nebo frázi.
  např. test na shodu:

  `Malý Petr přišel domů.`

  podmět (`Petr`) je ve shodě s přísudkem (`přišel`) v čísle a rodě
  přídavné jméno (`malý`) a podstatné jméno (`Petr`) se shodují v pádě, čísle a rodě

  S(n, g) → NP(_, n, g) VP(n, g)
  NP(c, n, g) → ADJ(c, n, g) N(c, n, g)

Příznaky a příznakové struktury – pokrač.

- gramatické znaky (slovní druh, gramatický pád, rod, číslo, osoba, . . .) je výhodné začlenit do gramatiky ve formě dvojic atribut–hodnota
- potom je možné zobecňovat, např. vyjádřit shodu v pádě, čísle a rodě výhradně pomocí atributů
- aplikace – v mnoha gramatických formalismech jazykové objekty jsou zde modelovány jako příznakové struktury (*feature structures*), tedy právě matice dvojic atribut–hodnota.
- u složitějších struktur – nestačí pak běžné porovnání instanciace jde oběma směry → použije se unifikace

Pořádek slov ve větě

syntaktická pozice – standardní pozice větných členů ve větě

angličtina: S V O M P T
Subject, Verb, Object, Modus, Place, Temp

- avšak např. předmět se může přesunout na první pozici – topikalizace

  The book I read.

- v češtině – téměř libovolné přesuny syntaktických elementů souvisí s tzv. aktuálním větným členěním

Možnosti zadávání gramatik

- nejčastější formát specifikace gramatik – produkční pravidla
  gramatika se skládá z pravidel generujících správně utvořené řetězce
- cíl analyzátoru – najít odvození vstupního řetězce ze zadaného neterminálu (označovaného obyčejně velkým písmenem *S* z anglického *sentence* – věta) na základě daných pravidel
- pokud je tohoto cíle dosaženo, vstup je akceptován a je mu přiřazena odpovídající struktura
- v minulosti rovněž populární – přechodové sítě (*transition networks*)
  přechody sítě = lingvistické jednotky, uzly sítě = stavy analyzátoru v procesu analýzy vstupu. Přechody jsou označeny symboly definujícími, za jakých podmínek se analyzátor může přesunout z jednoho stavu do stavu druhého.
  rozšířené přechodové sítě (*ATN – Augmented TN*) jsou doplněny o podmínky a procedury – ekvivalentní deklarativním gramatikám

Standardní teorie syntaxe

- 50. léta 20. stol. – Noam Chomsky vytvořil formální teorii syntaxe
- jedna ze základních tezí – autonomie syntaxe
  ⇐ k ověření syntaktické správnosti věty nepotřebujeme znát její význam

  Bezbarvé zelené myšlenky zuřivě spí.
- syntaktické principy mají univerzální platnost pro různé přirozené jazyky

Chomského standardní teorie syntaxe

znalost jazyka = gramatika
Chomského předpoklady o rozumu:

- rozum má *vrozenou strukturu*
- rozum je *modulární*
- rozum obsahuje speciální modul pro *jazyk*
  porozumění jazyku je oddělitelné od jiných aktivit
- syntaxe je *formální*
  nezávislá na významu a komunikačních funkcích
- znalost jazyka je *modulární*
  obsahuje moduly pro jednotlivé fáze analýzy jazyka

Standardní teorie syntaxe – pokrač.

- Noam Chomsky, Aspects of the Theory of Syntax, 1965 – standardní teorie syntaxe – transformační generativní gramatika (TGG)
- snaží se řešit i zachycení sémantických vztahů v hloubkové struktuře
- postupně se vyvinula:
  - v rozšířenou standardní teorii (1968)
  - později tzv. Government & Binding Theory (teorie nadřazení a vázání, 1981), která zakládá na pojmu *univerzální gramatiky*
  - 90. léta – teorie minimalismu (snaha po úspornosti popisného aparátu)

Standardní teorie syntaxe – pokrač.

základní části standardní teorie:

- bázová komponenta
  - bezkontextová pravidla a schémata pravidel generují základní strukturu větných členů
  - lexikon popisuje lexikální kategorie a syntaktické rysy lexikálních položek
- transformační pravidla – vložení, smazání, přesun, změna-rysu, kopie-rysu
  transformace převádí hloubkové struktury na struktury povrchové

Příklad bázové komponenty

pravidla:

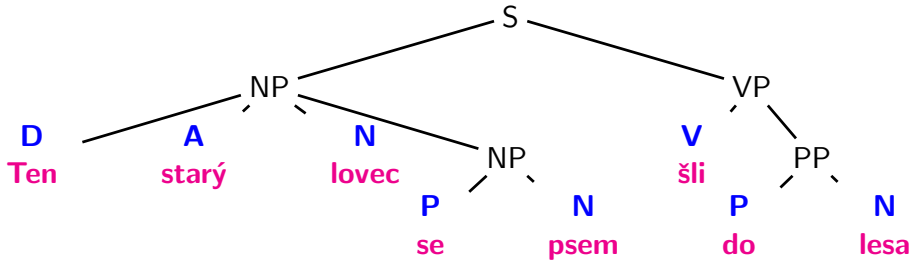
S → NP VP
NP → (D) A* N PP*
VP → V (NP) (PP)
PP → P NP

lexikon:

D: ten, ta
A: velký, hnědý, starý
N: pták, psem, lovec, já, lesa
V: loví, jí, šli
P: se, do

věta: Ten starý lovec se psem šli do lesa.

syntaktický strom:



Příklad transformačních pravidel

např. pasivizace (v angličtině):

John chose a book.

NP1 – Aux – V – NP2
1 – 2 – 3 – 4 → 4 – 2+be+en – 3 – by+1
přesuny + vložení + změny-rysu

- transformace:
  - obligatorní – např. přesun slovesné koncovky za sloveso
  - fakultativní – např. pasivizace, tvorba otázek, negace (změna významu)
- pravidla bázové komponenty – popisují strom hloubkové struktury v obvyklém pořadí
- transformace umožňují jeho změny na různé povrchové varianty (trpný rod, otázka, . . .)
- stopa (*trace*) – ukazuje, kde byl prvek před přemístěním

Návrh podkladů a datových struktur

- syntaktický (odvozovací, derivační) frázový strom – kompletní hierarchický popis struktury věty
- úkol syntaktické analýzy = pro danou gramatiku a daný vstup (větu) dát všechny odvozovací stromy
- existují techniky pro kompaktní uložení lesa takových stromů (chart parsing)
- jelikož se zabýváme výhradně syntaktickou strukturou a nevylučujeme a priori derivační stromy s absurdní interpretací, má většina vět mnoho různých syntaktických stromů

Obehnat Šalounův pomník mistra Jana Husa na pražském Staroměstském náměstí živým plotem z hustých keřů s trny navrhuje občanské sdružení Společnost Jana Jesenia.

`Pocet uspesnych stromu = 57 102 672`

Návrh podkladů a datových struktur – pokrač.

Automatická analýza syntaxe musí vždy projít třemi fázemi:

1. musí být zvolena notace pro zápis gramatiky – gramatický formalismus
2. musí být ve zvoleném formalismu napsána gramatika pro každý jazyk, který bude zpracováván
3. musí být vybrán nebo navržen algoritmus, který určí, zda daný vstup odpovídá gramatice, a pokud ano, jaký popis mu odpovídá

Gramatické formalismy pro ZPJ

Aleš Horák

E-mail: hales@fi.muni.cz
http://nlp.fi.muni.cz/poc_lingv/

Obsah:

Gramatické formalismy

- existuje velké množství různých přístupů k formální specifikaci gramatik, různé gramatické formalismy
- popíšeme několik nejrozšířenějších formalismů:
  - kategoriální gramatiky – categorial grammars, CG
  - závislostní gramatiky – dependency grammars
  - stromové gramatiky – (Lexicalized) Tree Adjoining Grammar, (L)TAG
  - lexikální funkční gramatiky – Lexical Functional Grammar, LFG
  - gramatiky příznakových struktur – Head Phrase Structure Grammar, HPSG
- soustředíme se jen na zápis gramatiky (notaci)

Kategoriální gramatiky

- kategoriální gramatika (categorial grammar, CG) – skupina teorií syntaxe a sémantiky PJ s velkým důrazem na lexikon
- neobsahuje *pravidla* pro kombinování slov → lexikální kategorie slov tvoří funkce, které určují, jak se dané kategorie kombinují s jinými výraz je výsledkem aplikace podvýrazů na sebe

  pěkný := NP/N ... funkce, která má argument N a vrací NP

- všechny verze CG se opírají o princip kompozicionality:
  Význam složeného výrazu je jednoznačně určen významy částí tohoto výrazu a způsobem, jakým jsou tyto části složeny dohromady.
- zakladatelé generativních gramatik – Leśniewski (publ. 1929) a Ajdukiewiczem (publ. 1935) ve vazbě na Husserlovu a Russellovu teorii kategorií a teorii typů
- první použití kategoriálních gramatik pro popis přirozeného jazyka – Bar-Hillel, Yehoshua 1953

Notace kategoriálních gramatik

- existuje několik různých variant notace

$$\dfrac{\dfrac{\underline{\text{šikovní}}\ \underline{\text{psi}}}{\dfrac{NP/N\quad N}{NP}>}\quad \dfrac{\underline{\text{mají rádi}}\ \underline{\text{kočky}}}{\dfrac{(S\backslash NP)/NP\quad NP}{S\backslash NP}>}}{S}<$$

- jiný rozšířený zápis – výsledek na vrcholku (result on top) Lambek 1958

$$\dfrac{\dfrac{\underline{\text{šikovní}}\ \underline{\text{psi}}}{\dfrac{NP/N\quad N}{NP}>}\quad \dfrac{\underline{\text{mají rádi}}\ \underline{\text{kočky}}}{\dfrac{(NP\backslash S)/NP\quad NP}{NP\backslash S}>}}{S}<$$

Notace kategoriálních gramatik – pokrač.

kategoriální gramatika je šestice $\langle \Sigma, C_{base}, C, Lex, RS, C_{complete} \rangle$, kde

1. Σ je konečná množina slov
2. C_{base} je konečná množina základních kategorií (funkčních typů)
3. C je množina kategorií definovaná induktivně takto:
   a) $C_{base} \subseteq C$
   b) pokud $X, Y \in C$, potom i $(X/Y) \in C$ a $(X\backslash Y) \in C$
   c) C obsahuje pouze prvky dané výše uvedenými body a) a b)
4. $Lex \subseteq \Sigma \times C$ je konečná množina – lexikon (zapisujeme v indexovém tvaru $\texttt{slovo}_{kategorie}$)
5. RS je množina následujících schémat pravidel:
   a) $\alpha_{(X/Y)} \circ \beta_{(Y)} \rightarrow \alpha\beta_{(X)}$
   b) $\beta_{(Y)} \circ \alpha_{(X\backslash Y)} \rightarrow \beta\alpha_{(X)}$,
   kde $\alpha, \beta \in \Sigma$ a $X, Y \in C$
6. $C_{complete} \subseteq C$ je množina dokončených (kompletních) kategorií

Notace kategoriálních gramatik – pokrač.

- daná schémata umožňují 2 způsoby kombinace:
  - argument vpravo (/) – $\alpha_{(X/Y)} \circ \beta_{(Y)} \rightarrow \alpha\beta_{(X)}$
  - argument vlevo (\) – $\beta_{(Y)} \circ \alpha_{(X\backslash Y)} \rightarrow \beta\alpha_{(X)}$
- tento typ kategoriální gramatiky označoval Bar-Hillel jako obousměrný (bidirectional CG)

`Karel miluje Marii`:

- bázové kategorie $= \{NP, S\}$
- kategorie z lexikonu: $\texttt{Karel}_{(NP)}$, $\texttt{Marii}_{(NP)}$, $\texttt{miluje}_{((S\backslash NP)/NP)}$
- $C_{complete} = \{S\}$

- v tomto tvaru je odvození ekvivalentní derivačním stromům CFG
- existují ale rozšíření kategoriálních gramatik, která vedou k systémům s vyšší vyjadřovací silou, než mají standardní CFG

Rozšíření kategoriálních gramatik

- klíčový problém – nespojité větné části, tzv. neprojektivity
- řešení pomocí rozšíření CG – přídavné kombinatorické operátory založené na typech
- dva možné přístupy:
  - pravidlově orientovaný přidává pravidla odpovídající jednoduchým operacím nad kategoriemi, jako jsou:
    - wrap – komutace argumentů
    - type-raising – aplikace typů podobná aplikaci tradičních pádů na jmenné fráze
    - comp – kompozice funkcí

    k nejpropracovanějším systémům tohoto typu patří kombinatorické kategoriální gramatiky (CCG).
  - deduktivní přístup vychází z Lambekova syntaktického kalkulu
    - pohled na kategoriální lomítko (slash) jako formu logické implikace
    - axiomy a inferenční pravidla potom definují teorii důkazu
      např. *aplikace funkce* $\approx$ pravidlo *modus ponens* $P \wedge (P \Rightarrow Q) \Rightarrow Q$

Závislostní gramatiky

- blízko ke kategoriálním gramatikám – vztah závislosti mezi řídícími a závislými větnými členy
- vhodné pro popis jazyků s volným slovosledem
- používají výhradně lexikalizovaných uzlů (v závislostním stromu) – neexistují žádné neterminály
  $\rightarrow$ závislostní analýza se jeví *jednodušší*
- využívá valence či subkategorizace – vztah mezi jedním slovem a jeho argumenty
  typicky vztah mezi slovesem a jeho možnými doplněními:

```
nosit
= koho|co
= komu & koho|co
```

Závislostní gramatiky – pokrač.

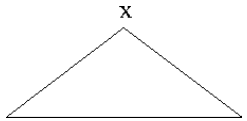
hlavní přístupy:

- navazuje na evropskou lingvistickou tradici – až k antice
- nejstarší užití – Tesnière 1959
- funkční generativní popis (*Functional Generative Description*, FGD) – jeden z nejpropracovanějších závislostních systémů, pražská lingvistická škola (Sgall, Hajičová, Panevová)
- UDG, *Unification Dependency Grammar* – Maxwell
- MTT, *Meaning-Text Theory* – Mel'čuk
- WG, *Word Grammar* – Hudson
- Lexicase – Starosta
- FG, *Functional Grammar* – Dik
- LG, *Link Grammar* – Temperley, Carnegie Mellon University `http://www.link.cs.cmu.edu/link/`
- DUG, *Dependency Unification Grammar* – Halliday

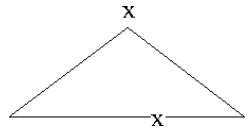
Stromové gramatiky TAG a LTAG

- Tree Adjoining Grammar – Joshi, Levy a Takahashi: *TAG Formalism*, 1975
  Lexicalized TAG – Joshi a Schabes: *Parsing with Lexicalized TAG*, 1991
- pracují přímo se stromy a ne s řetězci slov
- množina počátečních stromů – základní stavební prvky
- složitější věty odvozovány s použitím pomocných stromů

počáteční (*initial*) strom:



pomocný (*auxiliary*) strom:



TAG – počáteční a pomocné stromy

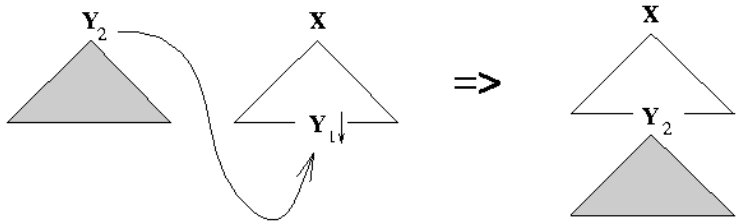
- počáteční stromy – neobsahují rekurzi → popisují složkovou strukturu jednoduchých vět, jmenných skupin, předložkových skupin, ...
  1. všechny nelistové uzly odpovídají *neterminálům*
  2. všechny listové uzly odpovídají *terminálům* nebo *neterminálním* uzlům určeným k *substituci*

  počáteční strom typu X = jeho kořen je označen termem X
- pomocné stromy – reprezentují *rekurzivní struktury*
  popisují větné členy, které se připojují k základním strukturám (např. příslovečné určení)
- charakterizace:
  1. všechny nelistové uzly odpovídají *neterminálům*
  2. všechny listové uzly odpovídají *terminálům* nebo *neterminálním* uzlům určeným k *substituci* kromě právě jednoho neterminálního uzlu (patový uzel, *foot node*)
  3. patový uzel má stejné označení jako kořenový uzel

  patový uzel – slouží k připojení stromu k jinému uzlu

TAG – operace

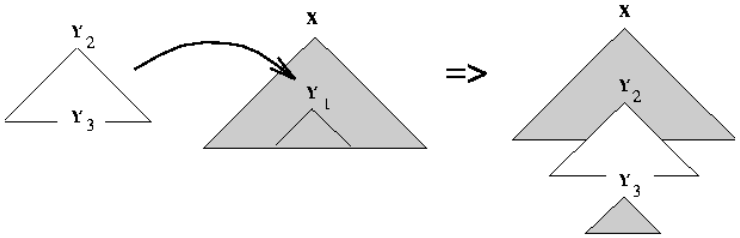
dvě operace – substituce a připojení (*adjunction*)
operace substituce – nahrazuje označený neterminál v listech nějakého stromu stromem, jehož kořen nese stejné označení



$Y_1\downarrow$ – označený pro substituci

TAG – operace připojení

operace připojení – vložení pomocného stromu, popisujícího rekurzi neterminálu X, se stromem, který obsahuje uzel označený rovněž X



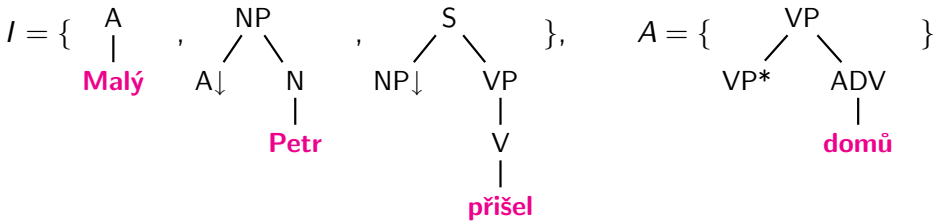
Definice TAG

- TAG $G = (I, A, S)$ je:
  - množina I konečných počátečních stromů
  - množina A pomocných stromů
  - typ stromu S – neterminál označující větu
- množina stromů $\mathcal{T}(G)$ TA gramatiky G = množina všech stromů odvoditelných z počátečních stromů typu S z I, jejichž spodní okraj sestává čistě z terminálních uzlů (všechny substituční uzly byly doplněny)
- jazyk řetězců $\mathcal{L}(G)$ generovaných TA gramatikou G = množina všech terminálních řetězců na spodním okraji stromů v $\mathcal{T}(G)$.

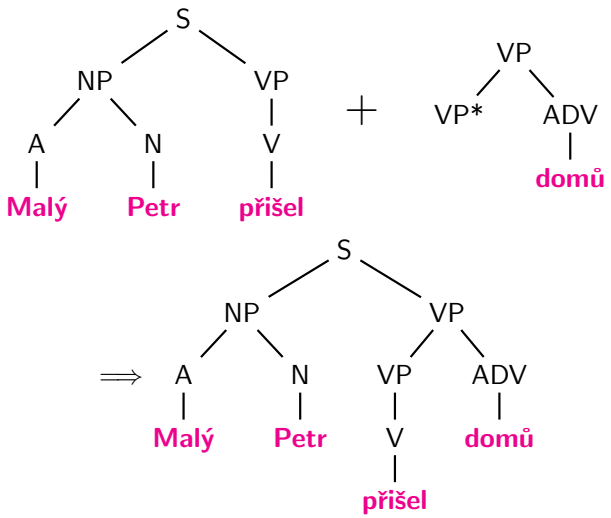
LTAG – lexikalizace

LTAG je lexikalizovanou variantou formalismu TAG
→ počáteční i pomocné stromy obsahují v listech jednu nebo více tzv. lexikálních kotev – uzly, které jsou přiřazeny (ukotveny) k určitým slovům lexikonu

lexikalizované stromy (*substituční uzly* – ↓, *patové uzly* – *):



LTAG – lexikalizované připojení



TAG a LTAG – generované jazyky

- díky použití operace připojení mají TAG a LTAG větší generativní sílu než bezkontextové gramatiky (CFG $\subset$ MCSL) $\rightarrow$ generují mírně kontextové jazyky (*mildly context-sensitive languages*)
  MCSL:
  - vlastnost konstantního růstu – pokud uspořádáme řetězce jazyka vzestupně podle délky, potom rozdíl dvou po sobě jdoucích délek nemůže být libovolný (každá délka je lineární kombinací konečného počtu pevných délek).
  - analyzovatelnost v polynomiálním čase $O(n^6)$ vzhledem k délce vstupu
- i jiné formalismy umí MCSL (jsou ekvivalentní s (L)TAG):
  - LIG, *Linear Indexed Grammars* – Gazdar, 1985
  - HG, *Head Grammars* – Pollard, 1984
  - CCG, kombinatorické kategoriální gramatiky

The XTAG Project – `http://www.cis.upenn.edu/~xtag/`

Lexikální funkční gramatiky LFG

- LFG, *Lexical Functional Grammar* – Kaplan a Bresnan, 1982
- dva typy syntaktických struktur
  - vnější, c-struktura – viditelná hierarchická organizace slov do frází
  - vnitřní, f-struktura – abstraktnější struktura gramatických funkcí, které tvoří hierarchii komplexních funkčních struktur

  důvod:
  - různé přirozené jazyky se významným způsobem odlišují v organizaci fráze, v pořadí a způsobech realizace gramatických funkcí
  - abstraktnější, funkcionální organizace jazyků se odlišuje mnohem méně v mnoha jazycích se např. objevují gramatické funkce *podmětu*, *předmětu* atd.

Lexikální funkční gramatiky LFG – pokrač.

- L = vztahy mezi jazykovými formami, např. mezi aktivními a pasivními formami slovesa, jsou zobecněním struktury lexikonu, ne transformačními operacemi, derivujícími jednu formu z druhé
- F = funkcionální teorie – gramatické vztahy, jako je podmět, předmět atd., jsou základními konstrukty, a nejsou definovány pomocí konfigurace frázové struktury, nebo sémantických pojmů typu Agent a Patient
- v LFG – pro reprezentaci funkcionální syntaktické informace je vhodné definovat hierarchickou strukturu jazykových jednotek, avšak vynucená linearizace pořádku těchto struktur není vhodná

Syntaktické úrovně LFG

- dvě syntaktické úrovně:
  - složková struktura (*c-structure*, *constituent structure*) – zachycuje frázovou dominanci a prioritu a je reprezentována jako strom frázové struktury (CFG strom)
  - funkcionální struktura (*f-structure*) – zachycuje syntaktickou strukturu typu predikát-argumenty a je reprezentována *maticí* dvojic *atribut-hodnota*
    nabízí jednotnou reprezentaci syntaktické informace abstrahující od detailů struktury fráze a lineárního pořádku

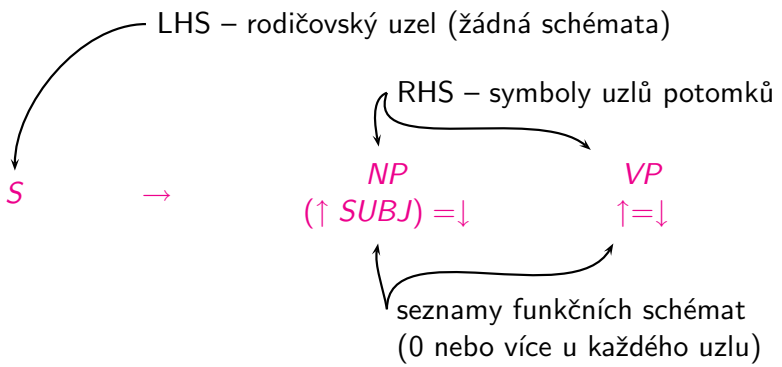
  f-struktura obsahuje soubor atributů:
  - příznaky – čas, rod, číslo, ...
  - funkce – PRED, SUBJ, OBJ, jejichž hodnoty mohou být jiné f-struktury
- vztah mezi c-strukturami (stromy) a odpovídajícími f-strukturami:

  projekce ϕ : {uzly stromu c-struktury} $\rightarrow$ {f-struktury}

LFG – c-struktura

LFG pravidla:

- klasická CF pravidla
- plus funkční schémata – výrazy pracující se symboly na pravé straně pravidel (za →, RHS)



LFG – pravidla

příklady:

$$\begin{array}{lccc} S \rightarrow & NP & & VP \\ & (\uparrow \text{SUBJ}) = \downarrow & & \uparrow = \downarrow \end{array}$$

$$\begin{array}{lccc} VP \rightarrow & V & & (NP) \\ & \uparrow = \downarrow & & (\uparrow \text{OBJ}) = \downarrow \end{array}$$

$$\begin{array}{lccc} NP \rightarrow & (DET) & & N \\ & \uparrow = \downarrow & & \uparrow = \downarrow \end{array}$$

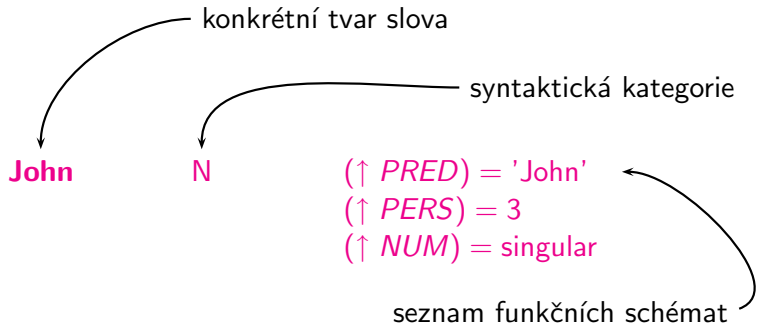
výrazy $(\uparrow \text{SUBJ}) = \downarrow$, $\uparrow = \downarrow$ a $(\uparrow \text{OBJ}) = \downarrow$ jsou *funkční schémata*

LFG – lexikon

lexikon také obsahuje funkční schémata

položka lexikonu:

1. konkrétní tvar slova
2. syntaktickou kategorii
3. seznam funkčních schémat



LFG – lexikon – pokrač.

příklady:

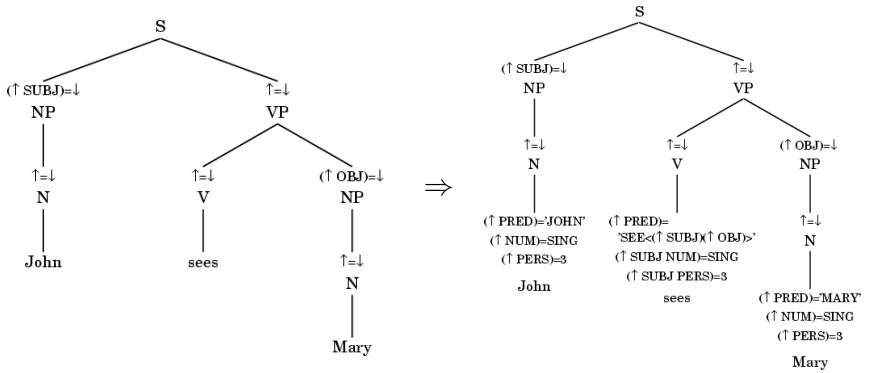
John	N	(↑ PRED)	=	'JOHN'
		(↑ NUM)	=	SING
		(↑ PERS)	=	3
sees	N	(↑ PRED)	=	'SEE<(↑SUBJ)(↑OBJ)>'
		(↑ SUBJ NUM)	=	SING
		(↑ SUBJ PERS)	=	3
Mary	N	(↑ PRED)	=	'MARY'
		(↑ NUM)	=	SING
		(↑ PERS)	=	3

XLE web interface –
`http://decentius.aksis.uib.no/logon/xle.xml`

LFG – konstrukce c-struktury

informace v c-struktuře:

- hierarchická struktura větných členů
- funkční anotace (funkční schémata převedená do stromu) – po jejich *interpretaci* získáme výslednou f-strukturu



LFG – f-struktura

$$f_n \begin{bmatrix} A & f_m \begin{bmatrix} B & C \\ D & E \end{bmatrix} \\ F & G \\ H & I \end{bmatrix}$$

grafický zápis:

matice atribut-hodnota (*attribute-value matrix*, AVM) – levé sloupce jsou atributy, pravé sloupce hodnoty (symboly, podřazené f-struktury nebo sémantické formy)

funkční rovnice a f-struktury:

$(f_p \text{ ATT}) = \text{VAL}$

v f-struktuře f_p je řádek, kde
atribut je ATT
a jeho hodnota je VAL

funkční rovnice mohou být splněny nebo nesplněny (*true*/*false*)

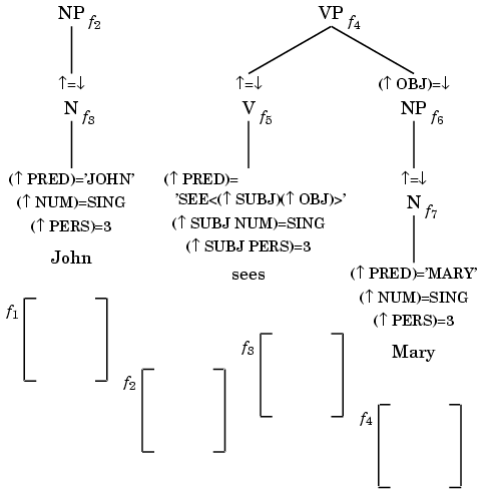
LFG – instanciace hodnot

Instanciace hodnot

- doplňuje hodnoty metaproměnných ↑ a ↓
- transformuje schémata na funkční rovnice – výrazy získané z f-struktur

grafický zápis – f-struktura v hranatých závorkách []

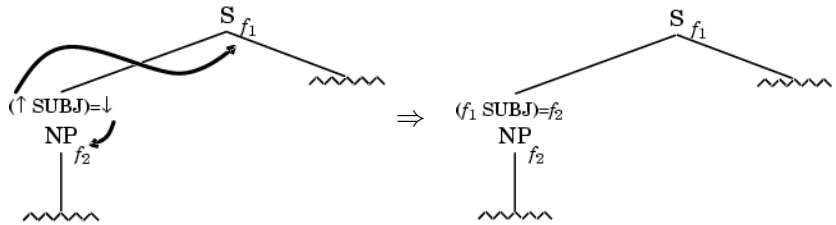
každý *uzel c-struktury* má k sobě připojenou *matici f-struktury*, které se označují indexy f_i

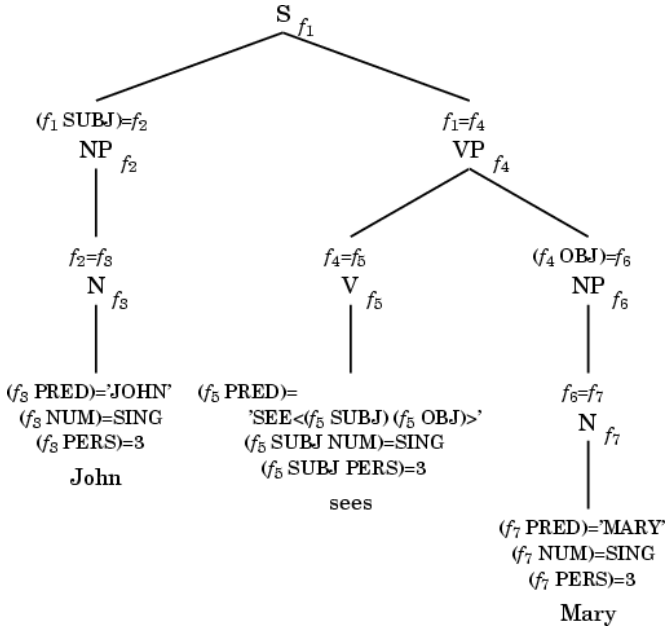


LFG – doplnění hodnot metaproměnných

↑ a ↓ (metaproměnné) se odkazují na f-struktury
je potřeba najít správné proměnné f_i na místa šipek

- ↓ – metaproměnná EGO nebo SELF – odkazuje na f-strukturu uzlu nad schématem
- ↑ – metaproměnná MOTHER – odkazuje na f-strukturu rodičovského uzlu vzhledem k uzlu nad schématem





LFG – funkční popis

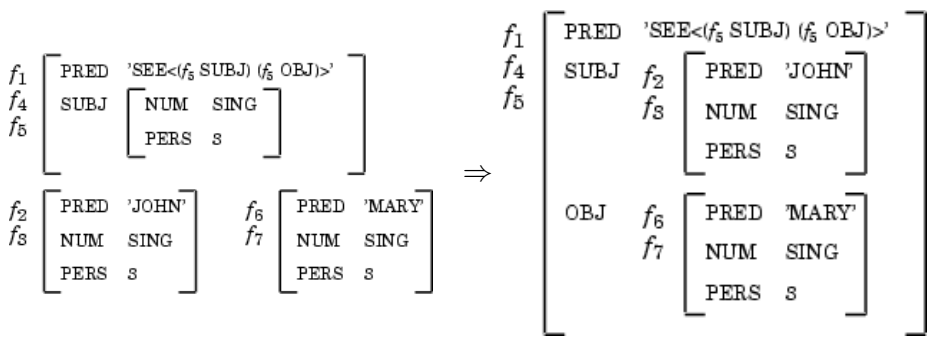
funkční popis = množina všech instanciovaných funkčních rovnic stromu
vlastní konstrukce f-struktury pracuje pouze s tímto funkčním popisem
funkční popis předchozí věty:

a. $(f_1$ SUBJ$) = f_2$
b. $f_3 = f_2$
c. $(f_3$ PRED$)$ = 'JOHN'
d. $(f_3$ NUM$)$ = SING
e. $(f_3$ PERS$) = f_3$
f. $f_1 = f_4$
g. $f_4 = f_5$
h. $(f_5$ PRED$)$ = 'SEE<$(f_5$ SUBJ$)(f_5$ OBJ$)$>'
i. $(f_5$ SUBJ NUM$)$ = SING
j. $(f_5$ SUBJ PERS$) = f_3$
k. $(f_4$ OBJ$) = f_6$
l. $f_6 = f_7$
m. $(f_7$ PRED$)$ = 'MARY'
n. $(f_7$ NUM$)$ = SING
o. $(f_7$ PERS$) = f_3$

LFG – konstrukce f-struktury

f-struktura se tvoří z funkčního popisu tak, aby všechny funkční rovnice byly splněny

výsledná f-struktura musí být minimální taková f-struktura



Gramatické formalismy pro ZPJ II

Aleš Horák

E-mail: hales@fi.muni.cz
http://nlp.fi.muni.cz/poc_lingv/

Obsah:

HPSG – Head-driven Phrase Structure Grammar

- HPSG, Head-driven Phrase Structure Grammar – Pollard & Sag, 1994
- navazuje na Gazdar, Generalized Phrase Structure Grammar, 1985
- lexikalizovaná teorie generativní gramatiky přirozeného jazyka
- *neterminály* CFG jsou nahrazeny příznakovými strukturami
- založená na omezeních (constraints)
- modeluje jazyk pomocí deklarativních omezení typovaných struktur
- příznaky jsou propojeny pomocí strukturního sdílení, tedy předáváním proměnných mezi podstrukturami dané struktury
- HPSG je nederivační, na rozdíl od jiných formalismů, kde jsou různé úrovně syntaktické struktury sekvenčně odvozovány pomocí transformačních operací

HPSG – Head-driven Phrase Structure Grammar – pokrač.

- gramatika je v HPSG modelována pomocí uspořádaných příznakových struktur, které korespondují s typy výrazů přirozeného jazyka a jejich částmi
- cílem teorie je detailní specifikace, které příznakové struktury jsou přípustné
- příznakové struktury definují omezení
  hodnoty příznaků mohou být jednoho ze čtyř typů
  - atomy
  - příznakové struktury
  - množiny příznakových struktur ($\{\ldots\}$)
  - nebo seznamy příznakových struktur ($<\ldots>$)

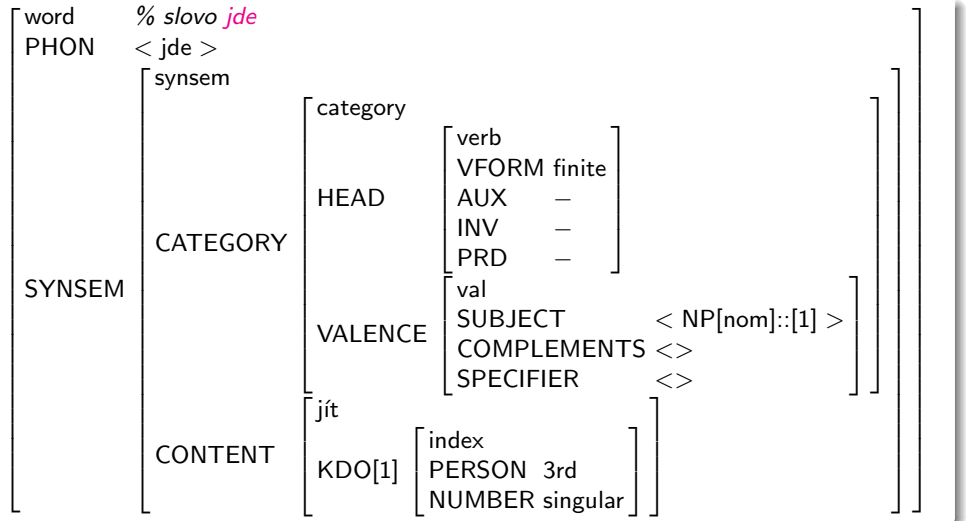
HPSG – lexikální hlava

- slova (lexikální položky) obsahují hodně informací – podle psycholingvistiky se podobá *zpracování v lidském mozku*
- lexikální hlava – základní prvek frázové struktury HPSG
  lexikální hlava = jedno slovo, jehož položka specifikuje informace, které určují základní gramatické vlastnosti fráze, kterou hlava zastupuje
  gramatické vlastnosti zahrnují:
  - morfologické informace (part-of-speech, POS)

    N zastupuje NP, VP zastupuje S, V zastupuje VP
  - relace závislosti (např. valenční rámec slovesa)
- lexikální hlava obsahuje také klíčové sémantické informace, které sdílí se zastupovanou frází

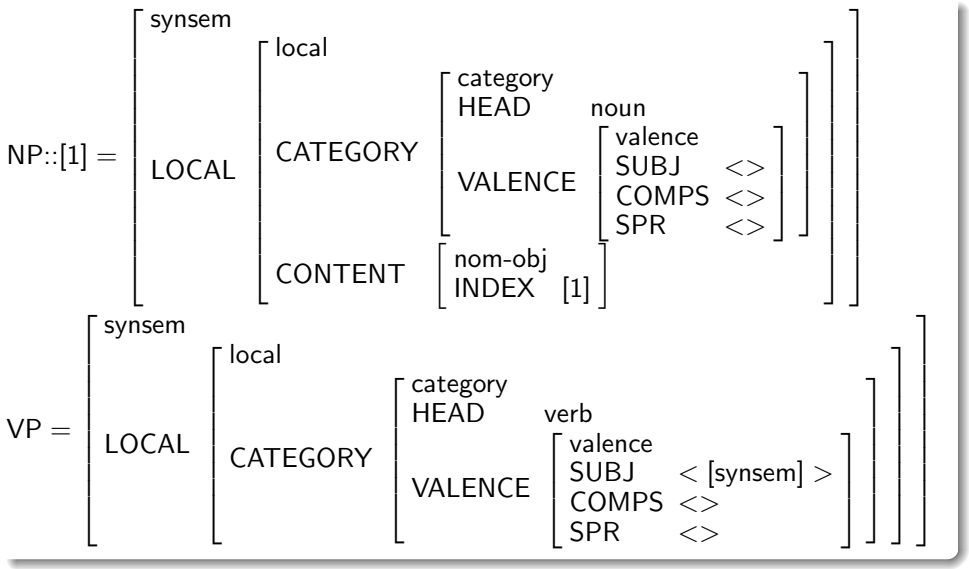
HPSG – struktury

HPSG struktury jsou typované příznakové struktury
zapisují se pomocí AVM – příznaky velkými písmeny, typy malými



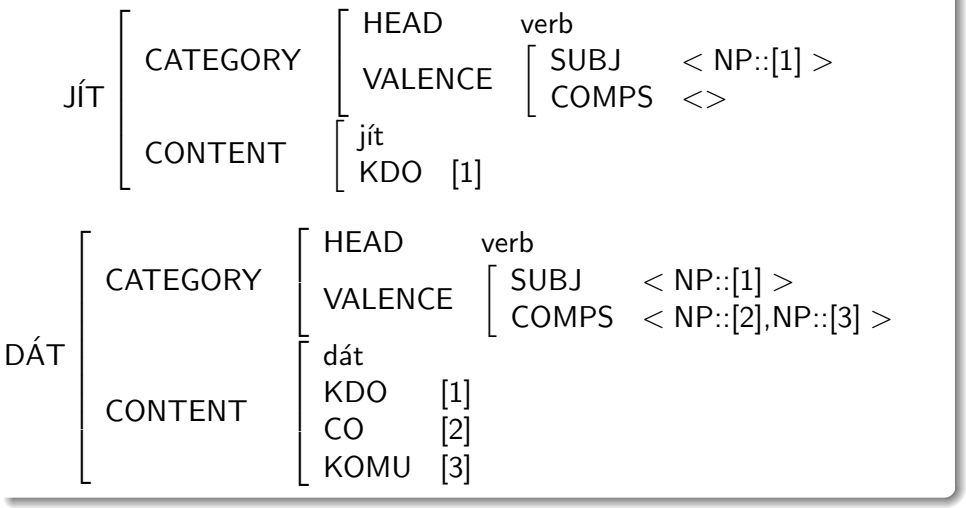
HPSG – syntaktické kategorie

symboly syntaktických kategorií – zkratky určitých příznakových popisů:



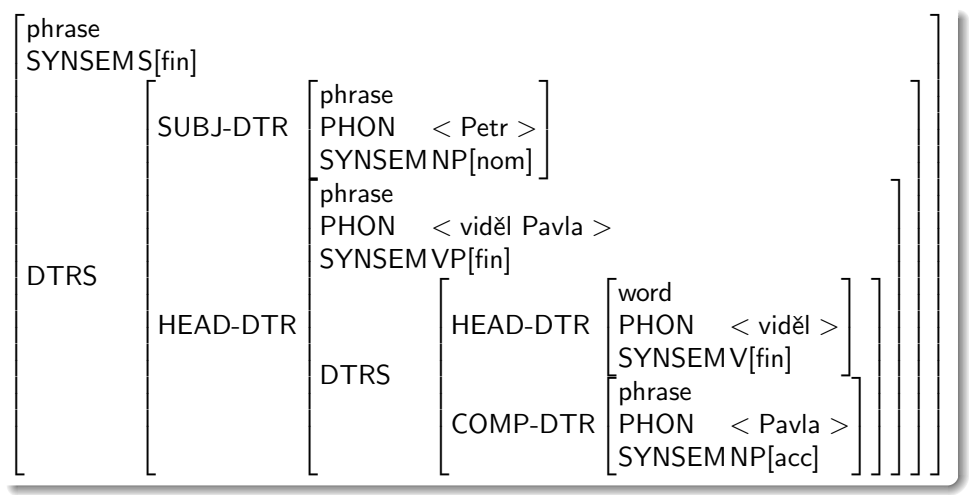
HPSG – lexikální položky

velké množství akcí je v lexikonu:



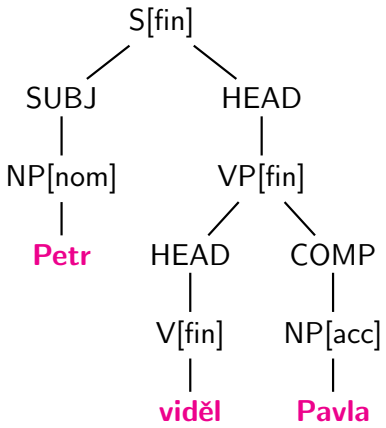
HPSG – fráze

reprezentace frází – v HPSG obdoba reprezentace slov
navíc příznak DAUGHTERS – struktura členů fráze



HPSG – fráze – pokrač.

pro snazší čtení popisů frází používáme stromový zápis:



ve skutečnosti se ovšem jedná o příznakovou strukturu, ne strom!

HPSG – dobře utvořené příznakové struktury

dobře utvořené příznakové struktury musí splňovat omezení daná gramatikou

příznaková struktura je dobře utvořená ⇔:

- každý uzel splňuje omezení geometrie příznaku
- každá uzel vstupního slova splňuje omezení některé lexikální položky
- každý frázový uzel splňuje frázová omezení – *omezení přímé dominance* (immediate dominance, viz dále), *omezení hlavových příznaků* (head feature), *valenční omezení*, ...

omezení geometrie příznaku specifikují:

- s jakými typy se pracuje
- jaká je použitá typová hierarchie – který typ je podtypem jiného typu
- pro každý typ – jaké příznaky přísluší tomuto typu
- pro každý typ a každý příznak – jakých typů mohou být hodnoty tohoto příznaku

HPSG – deklarace typu

pro popis omezení geometrie příznaku se používají typové deklarace:

```
category: [HEAD: head, VALENCE: valence]

head     # příznaková struktura složená z příznakových struktur
   noun: [CASE: case]
   verb: [VFORM: vform, AUX: boolean, INV: boolean]
   prep: [PFORM: pform]
   ...

vform    # jednoduchý příznak, forma slovesa – možné hodnoty:
   fin    # určitý tvar slovesa
   inf    # neurčitý tvar slovesa – infinitive
   ...

case     # jednoduchý příznak, gramatický pád
   nom    # 1. pád, nominativ
   acc    # 4. pád, akuzativ
   ...
```


HPSG – dobře utvořená slova a fráze

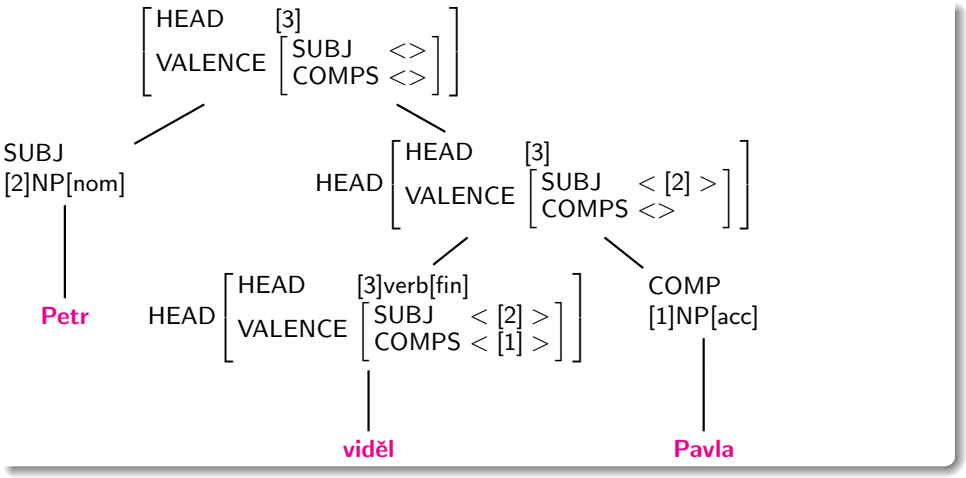
- každé vstupní slovo musí splňovat některou lexikální položku
- fráze musí splňovat frázová omezení (constraints):
  - omezení přímé dominance – každá fráze musí odpovídat jednomu ze schémat – schéma *head-subject*, schéma *head-specifier*, schéma *head-complement*, ...

```
   [SUBJ <>]              [SPR <>]              [COMPS <>]
      /\                     /\                    /\
  SUBJ/  \HEAD           SPR/  \HEAD           HEAD/  \COMPS
    /      \               /      \              /      \
phrase    phrase       phrase    phrase       word    list of phrases
          [COMPS <>, SPR <>]       [COMPS <>]
```

  - omezení hlavových příznaků – pro každou frázi, která má hlavu, musí být hlavové příznaky fráze shodné s hlavovými příznaky potomka, který je hlavou
  - valenční omezení – pro každý z valenčních příznaků (SUBJECT, COMPLEMENTS, ...) – hodnota příznaku na hlavové frázi musí odpovídat hodnotě na potomku, který je hlavou, mínus ty příznaky, které jsou splněny některým z nehlavových potomků

HPSG – dobře utvořené příznakové struktury

omezení ve větě 'Petr viděl Pavla.':



DEMO: **GG** – HPSG pro němčinu, DFKI Language Technology Lab, Saarbrücken
`http://www.cl.uni-bremen.de/~stefan/Babel/Interaktiv/`

Metagramatika systému synt

3 formy (meta)gramatiky:

- metagramatika (G1)
  - pravidla s kombinatorickými konstrukty + globální omezení pořadí
  - akce (= gramatické testy + kontextové akce)
  - česká lingvistická tradice – závislostní struktury, kontrola shody, pravidla pro pořadí slov, ...
- generovaná gramatika (G2)
  - bezkontextová pravidla
  - akce
- expandovaná gramatika (G3)
  - jen bezkontextová pravidla

Metagramatika – kombinatorické konstrukty

kombinatorické konstrukty se používají pro generování variant pořadí daným terminálů a neterminálů

hlavní kombinatorické konstrukty:

- `order()` generuje všechny možné permutace zadaných komponent
- `first()` argument musí být na prvním místě
- `rhs()` doplní všechny pravé strany svého argumentu

```
/* budu se ptát */
clause ===> order(VBU,R,VRI)

/* který ... */
relclause ===> first(relprongr) rhs(clause)
```


Metagramatika – typy pravidel

- `->` normální CF pravidlo
- `-->` vložit `intersegment` mezi každé dva prvky
- `==>` + kontrola správného pořadí příklonek
- `===>` intersegmenty na začátku a konci RHS, spojky, ...

```
ss -> conj clause
/* budu muset číst */
futmod --> VBU VOI VI
/* byl bych býval */
cpredcondgr ==> VBL VBK VBLL
/* musím se ptát */
clause ===> VO R VRI
```

`clause` pravidla se zadávají pomocí pravidlových vzorů

Metagramatika – globální omezení pořadí

globální omezení pořadí zakazuje některé kombinace pořadí preterminálů

%enclitic – které preterminály jsou brány jako příklonky
%order – zajišťuje dodržení precedence zadaných preterminálů

```
/* jsem, bych, se */
%enclitic = (VB12, VBK, R)

/* byl — četl, ptal, musel */
%order VBL = {VL, VRL, VOL}
```


Metagramatika – generativní konstrukty

skupina výrazů %list_* – produkují nová pravidla pro seznamy (s oddělovači/bez oddělovačů, s různými testy na shody, . . .)

```
/* (nesmim) zapomenout udelat - to forget to do */
%list_nocoord vi_list
vi_list -> VI

%list_coord_case np
%list_coord_case_number_gender left_modif
/* krasny velky pes a mala kocka - beautiful dog and small cat */
np -> left_modif np
```

koncovky *_case, *_number_gender and *_case_number_gender určují typ shody

Metagramatika – pravidlové vzory

pravidla pro slovesné skupiny – cca 40% všech pravidel metagramatiky
pravidlové vzory %group – definují časté skupiny konstrukcí v pravidlech

```
%group verbP={
  V:    verb_rule_schema($@,"(#1)")
        groupflag($1,"head"),
  VR R: verb_rule_schema($@,"(#1 #2)")
        groupflag($1,"head"),
}

%template clause ====> order(RHS)

/* ctu/ptam se - I am reading/I am asking */
clause %> group(verbP) vi_list
  verb_rule_schema($@,"#2")
  depends(getgroupflag($1,"head"), $2)
```


Metagramatika – pravidlové vzory – pokrač.

- předchozí příklad – skupina verbP = dvě skupiny preterminálů (V a VR R) s příslušnými akcemi
- při použití v clause vytvoří postupně dvě různé pravé strany
- (get)groupflag – odkaz na prvek uvnitř %group
- vzor celého pravidla – speciální pravidlová šipka %>
  %template definuje vzor každého pravidla s %>

Metagramatika – úrovně pravidel

- používá se pro ohodnocení výstupních stromů pro jejich třídění
- doplněk trénování na velkých stromových korpusech (zatím jen 5.000 vět)
- zadané lingvistou – specialistou na vývoj gramatiky
- základní úroveň – 0, vyšší úrovně – méně frekventované fenomeny
- pravidla vyšších úrovní mohou být v průběhu analýzy zapnuté/vypnuté

```
3:np -> adj_group
  propagate_case_number_gender($1)
```


Gramatika G2 – kontextové akce

- gramatické testy na shody – pád, rod, číslo
- testy na zanoření vedlejších vět – `test_comma`
- akce pro specifikaci závislostních hran
- akce typové kontroly logických konstrukcí

```
np -> adj_group np
  rule_schema($@, "lwtx(awtx(#1) and awtx(#2))")
  rule_schema($@, "lwtx([[awt(#1),#2],x])")
```

`rule_schema` – schéma pro tvorbu logické konstrukce ze subkonstrukcí
projdou jenom kombinace, které typově vyhovují danému schématu

Expandovaná gramatika G3

- překlad testů na shody do CF pravidel
- v češtině – 7 gramatických pádů, dvě čísla a 4 rody → 56 možných variant pro plnou shodu mezi dvěma prvky

počty pravidel

metagramatika G1	253
gramatika G2	3091
expandovaná gramatika G3	11530

DEMO: **wwwsynt** – webové rozhraní k syntu
`http://nlp.fi.muni.cz/projekty/wwwsynt/`
manuál ke **GDW** – Grammar Development Workbench
`http://nlp.fi.muni.cz/projekty/grammar_workbench/manual/`

Systém synt – příklad logické analýzy

vyhodnocení `rule_schema` pro `np` 'pečené kuře'

```
4, 6, -npnl -> . left_modif  np .: k1gNnSc145
agree_case_number_gender_and_propagate OK
rule_schema: 2 nterms, 'lwtx(awtx(#1) and awtx(#2))'
And constrs, Abstr and Exi vars are just gathered
1 (1x1) constructions:
    λw2λt3λx4([pečený_{w2t3}, x4] ∧ [kuře_{w2t3}, x4]) ... (oι)_{τω}
And constrs: none added
Exi vars: none added
```

Systém synt – příklad logické analýzy – pokrač.

vyhodnocení `verb_rule_schema` pro celou `clause`

`verb_rule_schema: 3 groups`
`no acceptable subject found: supplying an inexplicit one`
`inexplicit subject: k3xPgMnSc1,k3xPgInSc1:` $On \dots \iota$
`Clause valency list:  jíst <v>#1:(1)hA-#2:(2)hPTc1,  ...`
`Verb valency list:  jíst <v>#2:hH-#1:hPTc4ti`
`Matched valency list:  jíst <v>#2:(1)hH-#1:(2)hPTc4ti`
`time span:` $\lambda t_{12}\mathbf{dnes}_{tt_{12}} \dots (o\tau)$
`frequency:` $\mathbf{Onc} \dots ((o(o\tau))\pi)_{\omega}$
`verbal object:` $x_{15} \dots (o(o\pi)(o\pi))$
`present tense clause:`
$\lambda w_{17}\lambda t_{18}(\exists i_{10})(\exists x_{15})(\exists i_{16})([\mathbf{Does}_{w_{17}t_{18}}, On, [\mathbf{Imp}_{w_{17}}, x_{15}]] \wedge [\mathbf{večeře}_{w_{17}t_{18}}, i_{10}] \wedge [\mathbf{pečený}_{w_{17}t_{18}}, i_{16}] \wedge [\mathbf{kuře}_{w_{17}t_{18}}, i_{16}] \wedge x_{15} = [\mathbf{jíst}, i_{16}]_{w_{17}} \wedge [[\mathbf{k}_{w_{17}t_{18}}, i_{10}]_{w_{17}}, x_{15}]) \dots \pi$
`clause:`
$\lambda w_{19}\lambda t_{20}[\mathbf{P}_{t_{20}}, [\mathbf{Onc}_{w_{19}}, \lambda w_{17}\lambda t_{18}(\exists i_{10})(\exists x_{15})(\exists i_{16})([\mathbf{Does}_{w_{17}t_{18}}, On, [\mathbf{Imp}_{w_{17}}, x_{15}]] \wedge [\mathbf{večeře}_{w_{17}t_{18}}, i_{10}] \wedge [\mathbf{pečený}_{w_{17}t_{18}}, i_{16}] \wedge [\mathbf{kuře}_{w_{17}t_{18}}, i_{16}] \wedge x_{15} = [\mathbf{jíst}, i_{16}]_{w_{17}} \wedge [[\mathbf{k}_{w_{17}t_{18}}, i_{10}]_{w_{17}}, x_{15}])], \lambda t_{12}\mathbf{dnes}_{tt_{12}}] \dots \pi$

Algoritmy syntaktické analýzy (pomocí CFG)

Vladimír Kadlec, Aleš Horák

E-mail: hales@fi.muni.cz
http://nlp.fi.muni.cz/poc_lingv/

Obsah:

Základní postupy pro syntaktickou analýzu obecných bezkontextových gramatik

- obecná CFG – rozsáhlá, (silně) víceznačná, s ϵ-pravidly
- všechny uvedené algoritmy pracují s *polynomiální časovou a prostorovou složitostí*
- algoritmus CKY – *Cocke*, *Kasami*, *Younger*;
- tabulková (chart) analýza (neplést s LR tabulkou):
  - shora dolů (*top-down*);
  - zdola nahoru (*bottom-up*);
  - analýza řízená hlavou pravidla (*head-driven*);
- Tomitův zobecněný algoritmus LR

Syntaktická analýza

- Vstupy:
  - řetězec lexikálních kategorií (preterminálních symbolů) $a_1\ a_2\ \ldots\ a_n$

    např.: ADJ CONJ ADJ N V PREP N '.'
  - bezkontextová gramatika $G = \langle N, \Sigma, P, S \rangle$.
- Výstup:
  - efektivní reprezentace derivačních stromů.

Algoritmus CKY

- Gramatika musí být v Chomského normální formě.

  CNF (každá CFG jde do ní převést): $A \rightarrow BC$, $D \rightarrow$ 'd'
- Pro daný vstup délky n derivujeme podřetězce symbolů délky q na pozici p, značíme $w_{p,q}$, $1 \leq p, q \leq n$.
- Derivace řetězců délky 1, $A \Rightarrow w_{p,1}$, je prováděno prohledáváním terminálních pravidel.
- Derivace delších řetězců $A \Rightarrow^* w_{p,q}$, $q \geq 2$ vyžaduje aby platilo $A \Rightarrow BC \Rightarrow^* w_{p,q}$. Tedy z B derivujeme řetězec délky k, $1 \leq k \leq q$, a z C derivujeme zbytek, řetězec délky $q - k$. Tzn. $B \Rightarrow^* w_{p,k}$ a $C \Rightarrow^* w_{p+k,q-k}$. Kratší řetězce máme tedy vždy "předpočítané."

Algoritmus CKY pokrač.

```
program CKY Parser;
begin
    for p := 1 to n do V[p,1] := {A|A → a_p ∈ P };
    for q := 2 to n do
        for p := 1 to n − q + 1 do
            V[p,q] = ∅;
            for k :=1 to q − 1 do
                V[p,q] =
                    V[p,q] ∪
                    ∪ {A|A → BC ∈ P, B ∈ V[p,k], C ∈ V[p + k, q − k]};
            od
        od
    od
end
```

složitost CKY je $O(n^3)$

Algoritmus CKY, příklad – zadání

- vstupní gramatika je:

  $S \to AA|BB|AX|BY|a|b$
  $X \to SA$
  $Y \to SB$
  $A \to a$
  $B \to b$

- vstupní řetězec je $w = abaaba$.

Algoritmus CKY, příklad – řešení (matice V)

a b a a b a

$S \to AA|BB|AX|BY|a|b$
$X \to SA$
$Y \to SB$
$A \to a$
$B \to b$

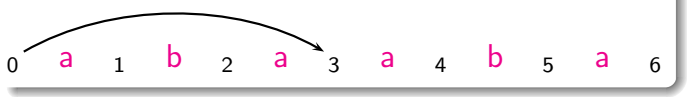
p – pozice, q – délka

q \ p	1	2	3	4	5	6
1	S, A	S, B	S, A	S, A	S, B	S, A
2	Y	X	S, X	Y	X	
3	S	$\emptyset$	Y	S		
4	X	S	$\emptyset$			
5	$\emptyset$	X				
6	S					

Tabulkové (chart) analyzátory

- Rozlišujeme tři základní typy tabulkových analyzátorů:
  - shora dolů;
  - zdola nahoru;
  - analýza řízená hlavou pravidla.
- Mnoho dalších variant je popsáno v:
  Sikkel Klaas: Parsing Schemata: A Framework for Specification and Analysis of Parsing Algorithm, 1997.
- Neklade se žádné omezení na gramatiku.
- Analyzátory typu "chart" v sobě většinou obsahují dvě datové struktury chart a agendu. Chart a agenda obsahují tzv. *hrany*.
- Hrana je trojice $[A \to \alpha \bullet \beta, i, j]$, kde:
  - i, j jsou celá čísla, $0 \leq i \leq j \leq n$ pro n slov ve vstupní větě
  - a $A \to \alpha\beta$ je pravidlem vstupní gramatiky.

$[A \to BC \bullet DE, 0, 3]$



Obecný analyzátor typu "chart"

```
program Chart Parser;
begin
    inicializuj (CHART);
    inicializuj (AGENDA);
    while (AGENDA není prázdná) do
        E := vezmi hranu z AGENDA;
        for each (hrana F, která může být vytvořena pomocí
          hrany E a nějaké jiné hrany z CHART) do
          if ((F není v AGENDA) and (F není v CHART) and
            (F je různá od E)
            then přidej F do AGENDA;
          fi;
        od;
        přidej E do CHART;
    od;
end;
```


Varianta shora dolů

Inicializace:

- $\forall\ p \in P \mid p = S \rightarrow \alpha$ přidej hranu $[S \rightarrow \bullet\alpha,\ 0,\ 0]$ do agendy.
- počáteční chart je prázdný.

Iterace – vezmi hranu E z agendy a pak:

a) (*fundamentální pravidlo*) pokud je E ve tvaru $[A \rightarrow \alpha\bullet,\ j,\ k]$, potom pro každou hranu $[B \rightarrow \gamma\bullet\ A\ \beta,\ i,\ j]$ v chartu vytvoř hranu $[B \rightarrow \gamma\ A\ \bullet\beta,\ i,\ k]$.

b) (*uzavřené hrany*) pokud je E ve tvaru $[B \rightarrow \gamma\bullet\ A\ \beta,\ i,\ j]$, potom pro každou hranu $[A \rightarrow \alpha\bullet,\ j,\ k]$ v chartu vytvoř hranu $[B \rightarrow\ \gamma\ A\ \bullet\beta,\ i,\ k]$.

c) (*terminál na vstupu*) pokud je E ve tvaru $[A \rightarrow \alpha\bullet a_{j+1}\beta,\ i,\ j]$, vytvoř hranu $[A \rightarrow\ \alpha\ a_{j+1}\bullet\beta,\ i,\ j{+}1]$.

d) (*predikce*) pokud je E ve tvaru $[A \rightarrow \alpha\bullet\ B\ \beta,\ i,\ j]$ potom pro každé pravidlo $B \rightarrow \gamma \in P$, vytvoř hranu $[B \rightarrow\ \bullet\ \gamma,\ i,\ i]$.

Příklad – tabulkové analýzy (typu chart)

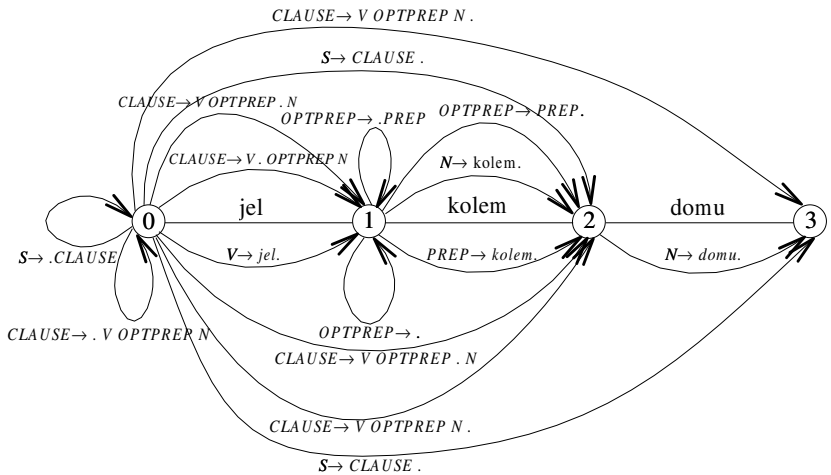
Gramatika:

$$
\begin{aligned}
S &\rightarrow \mathit{CLAUSE}\\
\mathit{CLAUSE} &\rightarrow \mathit{V\ OPTPREP\ N}\\
\mathit{OPTPREP} &\rightarrow \epsilon\\
\mathit{OPTPREP} &\rightarrow \mathit{PREP}\\
V &\rightarrow \mathit{jel}\\
\mathit{PREP} &\rightarrow \mathit{kolem}\\
N &\rightarrow \mathit{domu}\\
N &\rightarrow \mathit{kolem}
\end{aligned}
$$

Věta:

"jel kolem domu" (a_1=*jel*, a_2=*kolem*, a_3=*domu*).

Příklad – chart po analýze shora dolů



Varianta zdola nahoru

Inicializace:

- $\forall\ p \in P \mid p = A \rightarrow \epsilon$ přidej hrany $[A \rightarrow \bullet, 0, 0]$, $[A \rightarrow \bullet, 1, 1]$, ..., $[A \rightarrow \bullet, n, n]$ do agendy.
- $\forall\ p \in P \mid p = A \rightarrow a_i\alpha$ přidej hranu $[A \rightarrow \bullet a_i\alpha, i\text{-}1, i\text{-}1]$ do agendy.
- počáteční chart je prázdný.

Iterace – vezmi hranu E z agendy a pak:

a) (*fundamentální pravidlo*) pokud je E ve tvaru $[A \rightarrow \alpha\bullet, j, k]$, potom pro každou hranu $[B \rightarrow \gamma\bullet\ A\ \beta, i, j]$ v chartu vytvoř hranu $[B \rightarrow \gamma\ A\ \bullet\beta, i, k]$.

b) (*uzavřené hrany*) pokud je E ve tvaru $[B \rightarrow \gamma\bullet\ A\ \beta, i, j]$, potom pro každou hranu $[A \rightarrow \alpha\bullet, j, k]$ v chartu vytvoř hranu $[B \rightarrow\ \gamma\ A\ \bullet\beta, i, k]$.

c) (*terminál na vstupu*) pokud je E ve tvaru $[A \rightarrow \alpha\bullet a_{j+1}\beta, i, j]$, potom vytvoř hranu $[A \rightarrow\ \alpha\ a_{j+1}\bullet\beta, i, j{+}1]$.

d) (*predikce*) pokud je E ve tvaru $[A \rightarrow \alpha\bullet, i, j]$, potom pro každé pravidlo $B \rightarrow A\gamma$ vstupní gramatiky vytvoř hranu $[B \rightarrow \bullet A\gamma, i, i]$.

Analýza řízená hlavou pravidla

- *head-driven chart parsing*
- Hlava pravidla – libovolný (určený) symbol z pravé strany pravidla.

  Například pravidlo *CLAUSE* → *V* *OPTPREP* *N* může mít hlavy *V*, *OPTPREP*, *N*.
- Epsilon pravidlo má hlavu ϵ.
- Hrana v analyzátoru řízeném hlavou pravidla – trojice $[A \rightarrow \alpha\bullet\beta\bullet\gamma, i, j]$, kde i, j jsou celá čísla, $0 \leq i \leq j \leq n$ pro n slov ve vstupní větě a $A \rightarrow \alpha\beta\gamma$ je pravidlo vstupní gramatiky a hlava je v β.
- Algoritmus vlastní analýzy (varianta zdola nahoru) je podobný jednoduchému přístupu. Analýza neprobíhá zleva doprava, ale začíná na hlavě daného pravidla.

Analyzátor řízený hlavou pravidla

Inicializace:

- $\forall\ p \in P \mid p = A \rightarrow \epsilon$ přidej hrany $[A \rightarrow \bullet\bullet, 0, 0]$, $[A \rightarrow \bullet\bullet, 1, 1]$, ..., $[A \rightarrow \bullet\bullet, n, n]$ do agendy.
- $\forall\ p \in P \mid p = A \rightarrow \alpha\underline{a_i}\beta$ (a_i je hlavou pravidla) přidej hranu $[A \rightarrow \alpha\bullet a_i\bullet\beta, i\text{-}1, i]$ do agendy.
- počáteční chart je prázdný.

Je tato inicializace v pořádku?

Analyzátor řízený hlavou pravidla pokrač.

Iterace – vezmi hranu E z agendy a pak:

a_1) pokud je E ve tvaru $[A \rightarrow \bullet\alpha\bullet, j, k]$, potom pro každou hranu: $[B \rightarrow \beta\bullet\gamma\bullet A\delta, i, j]$ v chartu vytvoř hranu $[B \rightarrow \beta\bullet\gamma A\bullet\delta, i, k]$.

a_2) $[B \rightarrow \beta A\bullet\gamma\bullet\delta, k, l]$ v chartu vytvoř hranu $[B \rightarrow \beta\bullet A\gamma\bullet\delta, j, l]$.

b_1) pokud je E ve tvaru $[B \rightarrow \beta\bullet\gamma\bullet A\delta, i, j]$, potom pro každou hranu $[A \rightarrow \bullet\alpha\bullet, j, k]$ v chartu vytvoř hranu $[B \rightarrow \beta\bullet\gamma A\bullet\delta, i, k]$.

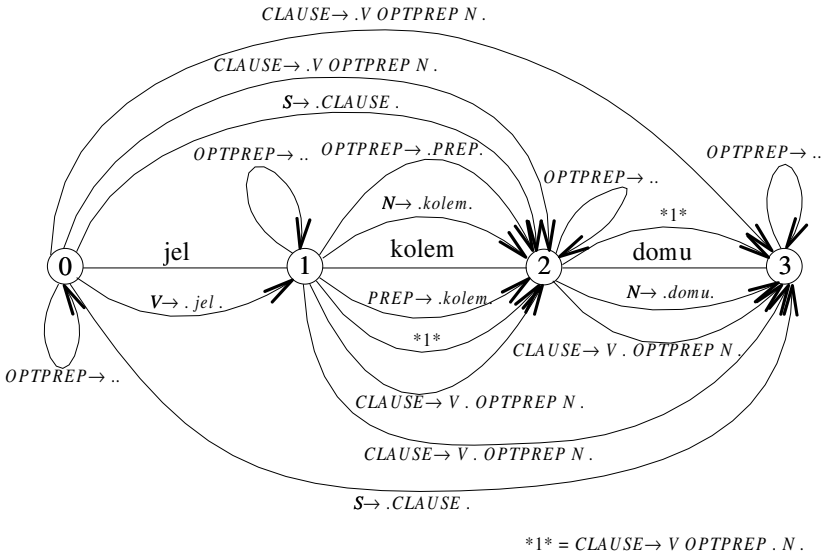
b_2) pokud je E ve tvaru $[B \rightarrow \beta A\bullet\gamma\bullet\delta, k, l]$, potom pro každou hranu $[A \rightarrow \bullet\alpha\bullet, j, k]$ v chartu vytvoř hranu $[B \rightarrow \beta\bullet A\gamma\bullet\delta, j, l]$.

c_1) pokud je E ve tvaru $[A \rightarrow \beta a_i\bullet\gamma\bullet\delta, i, j]$, potom vytvoř hranu $[A \rightarrow \beta\bullet a_i\gamma\bullet\delta, i\text{-}1, j]$.

c_2) pokud je E ve tvaru $[A \rightarrow \beta\bullet\gamma\bullet a_{j+1}\delta, i, j]$, potom vytvoř hranu $[A \rightarrow \beta\bullet\gamma a_{j+1}\bullet\delta, i, j{+}1]$.

d) pokud je E ve tvaru $[A \rightarrow \bullet\alpha\bullet, i, j]$, potom pro každé pravidlo $B \rightarrow \beta\ \underline{A}\ \gamma$ ve vstupní gramatice vytvoř hranu $[B \rightarrow \beta\bullet A\bullet\gamma, i, j]$ (symbol A je hlavou pravidla).

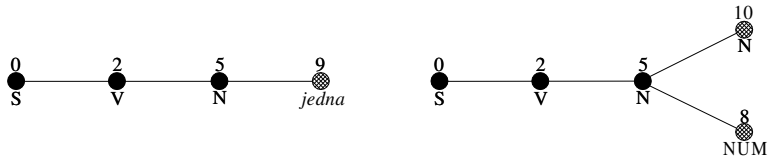
Příklad – chart po analýze řízené hlavou pravidla



Tomitův zobecněný analyzátor LR

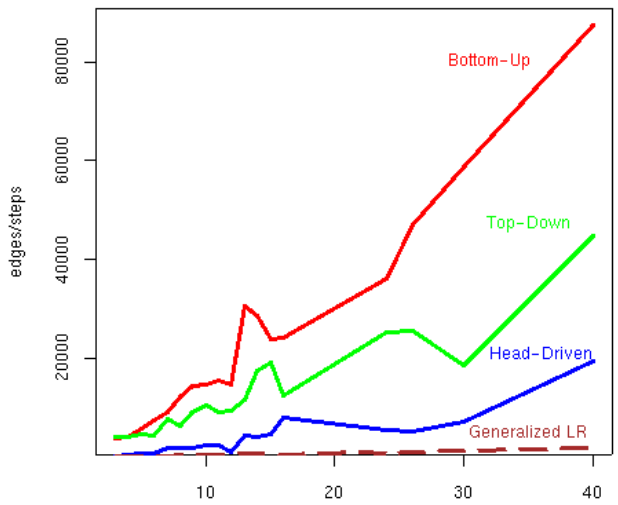
- *generalized LR parser (GLR)*
- Masaru Tomita: Efficient parsing for natural language, 1986
- standardní LR tabulka, která může obsahovat konflikty;
- zásobník je reprezentován acyklickým orientovaným grafem (*DAG*);
- derivační stromy jsou uloženy ve sbaleném "lese" stromů.
- v podstatě stejný, jako algoritmus LR;
- udržujeme si seznam aktivních uzlů zásobníku (grafu);
- akce redukce provádíme vždy před akcemi čtení;
- akci čtení provádíme pro všechny aktivní uzly najednou;
- kde je to možné, tam uzly slučujeme.

Příklad konfliktu redukce/redukce



stav	položka	akce	symbol	další stav
5	*CLAUSE→V N•NUM*	shift	*NUM*	8
	NN→ N•N		*N*	10
	NUM→ •jedna		*jedna*	9
	N→ •tramvaj		*tramvaj*	7
	N→ •jedna			
9	*NUM→jedna•*	reduce (6)		
	N→jedna•	reduce (5)		

Porovnání jednotlivých algoritmů



Korpusy textů a jejich využití

Pavel Rychlý, Aleš Horák

E-mail: hales@fi.muni.cz
http://nlp.fi.muni.cz/poc_lingv/

Obsah:

Co to je korpus?

- Co to je text, dokument?
  - lecos
- Různé typy korpusů
  - textové
  - mluvené
- Pro potřeby NLP
  - textový korpus

Textový korpus

- soubor textů
- charakteristiky
  - rozsáhlý (stovky mil. až mld. pozic/slov)
  - v jednotném formátu
  - stukturovaný
  - v elektronické podobě

Typy korpusů

- vždy záleží na účelu a způsobu použití
- možnosti
  - jazyk
  - typy textů
  - zdroj dat
  - značkování
  - ...

První korpus

Brown

- americká angličtina (1961)
- Brown University, 1964
- gramatické značkování, 1979
- 500 textů, 1 mil. slov
- W. N. Francis & H. Kučera
  - první statistické charakteristiky angličtiny
  - relativní četnosti slov a slovních druhů

SUSANNE

SUSANNE

- autor Geoffrey Sampson, Sussex University
- kniha *English for the Computer*
- část korpusu Brown ($\frac{1}{4}$)
- nové gramatické značkování
- syntaktické značkování

BNC

British National Corpus

- britská angličtina, 10% mluva
- první velký korpus pro lexikografy
- vydavatelé slovníků (OUP) + univerzity
- 1991–1994, World Edition 2000
- ≈3000 textů, 100 mil. slov
- gramatické značkování automatickým nástrojem

BoE

Bank of English

- britská angličtina
- COBUILD (HarperCollins), University of Birmingham
- 1991, stále rozšiřován
- 2002, ≈450 mil. slov

Další národní korpusy

- Český národní korpus
  - ÚČNK, FF UK
  - SYN2000: 100 mil. slov
  - Litera, Synek, BMK, ...
- Slovenský, Maďarský, Chorvatský, ...
- Americký

Korpusy na FI

vytvořené na FI, příklady:

- Desam
  - 1996, ručně značkovaný (desambiguovaný)
  - ≈1 mil. slov
- WWW
  - periodika z webu, z let 1996–1998
  - ≈100 mil.
- Chyby
  - práce studentů předmětu Základy odb. stylu s vyznačenými chybami
  - ≈400 tis.

Korpusy na FI

spolupráce

- Dopisy
- Mluv
- Kačenka
- ČNPK
- 1984
- Otto
- Italian
- Giga Chinese
- Francouzský, Slovinský, Britská angličtina, ...

Formáty korpusů

- archiv/kolekce
  - různé formáty, podle zdroje/typu
- textové banky
  - jednotný formát a základní struktura
  - dokumenty/texty, základní metainformace
- vertikální text
- binární data v aplikaci
  - pomocná data pro rychlejší zpracování
    - indexy
    - statistiky

Kódování znaků

- 8 bitů ≈ 256 znaků
  - ASCII – základ 7 bitů
  - kódování pro češtinu
    - ISO-Latin-2, Windows-1250, 852
- Unicode
  - 32bitů na znak
  - UTF-8
    - 1 až 4 byty na znak
  - UTF-16
    - 2 až 4 byty na znak

Kódování metainformací

- escape-sekvence
  - speciální znak mění význam následujících znaků
  - `\n, \t, &, <tag>`
- SGML
  - Standard Generalised Markup Language
  - ISO 8879:1986(E)
- XML
  - Extensible Markup Language
  - W3C, 1998

XML

- struktura popsána v DTD
- elementy
  - počáteční, koncová značka
  - `<doc>, <head>, </head>, <g/>`
- atributy elementů/značek
  - `<doc title="Jak pejsek ..." author="Čapek">`
  - `<head type="main">`
- entity
  - `&gt;, &lt;, &amp;, &eacute;`

Standardy pro ukládání

- SGML/XML
- TEI
  - Text Encoding Initiative
  - TEI Guidelines for Electronic Text Encoding and Interchange
- CES, XCES
  - Corpus Encoding Standard

Obsah korpusu

Co je v korpusu uloženo?

- text
- metainformace
- struktura dokumentu
  - odstavce, nadpisy, verše, věty
- značkování
  - informace o slovech
  - morfologie, základní tvary

Tokenizace

Rozdělení textu do pozic

- token (pozice) = základní prvek korpusu
- většinou slovo, číslo, interpunkce
  - bude-li, don't
- může silně ovlivnit výsledky

Vertikální text

- jednoduchý formát i jeho zpracování
  - každý token na samostatném řádku
  - struktury formou XML elementů
  - značkování odděleno tabulátorem
- podrobnosti
  - `http://www.fi.muni.cz/nlp/`
  - Informace pro současné a potenciální spolupracovníky
  - Textové korpusy
  - Popis vertikálů

Zpracování textů na UNIXu

- coreutils
  - cat, head, tail, wc, sort, uniq, comm
  - cut, paste join, tr
- grep
- awk
- sed / perl

Příklady použití coreutils

- slovník z vertikálního textu

```
cut -f 1 -s desam.vert |sort |uniq -c \
|sort -rn >desam.dict
```

- jednoduchá tokenizace

```
tr -cs 'a-zA-Z0-9' '\n' <GPL >GPL.vert
cat GPL.vert |sort |uniq -c |sort -rn >GPL.dict
```


Korpusové manažery

nástroje na zpracování korpusů

- uložení textu
- editace/příprava textu
- značkování
- rozdělení do pozic (tokenizace)
- vyhledávání (konkordance)
- statistiky

Systém Manatee

- korpusový manažer
- přímo podporuje
  - uložení textu
  - vyhledávání (konkordance)
  - statistiky
- externí nástroje
  - značkování
  - rozdělení do pozic

Systém Manatee

hlavní zaměření

- velké korpusy
- rozsáhlé značkování
  - morfologické, syntaktické, metainformace
- návaznost na další aplikace/nástroje
  - korpusový editor, tvorba slovníků
- univerzálnost
  - různé jazyky, kódování, systémy značek

Klíčové vlastnosti

- modulární systém
- přístup z různých rozhraní
  - grafické uživatelské rozhraní (Bonito)
  - aplikační programové rozhraní (API)
  - příkazový řádek
- rozsáhlá data
  - až 2 mld. pozic
  - neomezeně atributů a metainformací
- rychlost
  - vyhledávání, statistiky

Klíčové vlastnosti

- multihodnoty
  - zpracování víceznačných značkování
- dynamické atributy
  - vyhledávání a statistiky na počítaných datech
- subkorpusy
- silný dotazovací jazyk
  - dotazy na všechny atributy, metainformace
  - pozitivní/negativní filtry

Klíčové vlastnosti

- frekvenční distribuce
  - víceúrovňová
  - všechny atributy a metainformace
- kolokace
  - různé statistické funkce

Sémantika a základní sémantické reprezentace

Aleš Horák

E-mail: hales@fi.muni.cz
http://nlp.fi.muni.cz/poc_lingv/

Obsah:

Sémantika

studium významu – rozdílné, i když překrývající se přístupy různých vědeckých disciplín:

- filosofie – Jak je možné, že něco vůbec něco znamená? Jaký typ relace musí být mezi X a Y, aby X znamenalo Y? (filosofie jazyka)
- psychologie – psycholingvistika – experimentální studie, jak jsou významy reprezentovány v mysli a jaké mechanismy ovlivňují při kódování a dekódování zpráv (délka odezvy u konkrét a abstrakt se liší)
- neurologie – jak jsou psychologické stavy a procesy *implementovány* na úrovni neuronů

Význam v jazyce

Rozdělení studia významu v jazyce:

- lexikální sémantika
- gramatická sémantika – větné fráze, slovotvorba
- logická sémantika – výroková, predikátová a vyšší logiky
- lingvistická pragmatika

entail = znamenat, vyplývat; nutnost a očekávanost

1. X přestal zpívat ?→? X nepokračoval ve zpěvu
2. X je kočka ?→? je zvíře
3. X je v jiném stavu ?→?X je žena
4. X je fyzikální objekt ?→? X má hmotnost
5. X je čtyřnožec ?→? X má čtyři nohy
6. X je žena Y ?→? X není dcera Y

Princip kompozicionality

> *Význam složeného tvrzení je funkcí významu jednotlivých komponent.*

(je určován, je odhadnutelný, každá složka hraje význam?)
omezení PK: idiomy, ustrnulé metafory, kolokace, klišé

listém je jazykový výraz, jehož význam není určen významy jeho částí (pokud existují), a který si tedy uživatel jazyka musí zapamatovat jako kombinaci formy a významu.

Problémy při analýze přirozeného jazyka

- víceznačnost
- anaforické výrazy
- indexické výrazy
- nejasnost
- nekompozicionalita
- struktura promluvy
- metonymie
- metafory

Víceznačnost

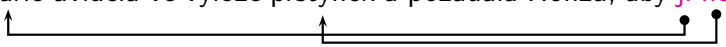
- *ambiguity*
- víceznačnost může být lexikální, syntaktická, sémantická a referenční
- lexikální – "stát," "žena," "hnát"
- syntaktická – "Jím špagety s masem."
  "Jím špagety se salátem."
  "Jím špagety s použitím vidličky."
  "Jím špagety se sebezapřením."
  "Jím špagety s přítelem."
- sémantická – "Jeřáb je vysoký." "Viděli jsme veliké oko."
- referenční – "Oni přišli pozdě." "Můžeš mi půjčit knihu?"
  "Ředitel vyhodil dělníka, protože (on) byl agresivní."

Anaforické a indexické výrazy

anaforické výrazy:

- *anaphora*
- používají zájmena pro odkazování na objekty zmíněné dříve
- "Poté co se Honza s Marií rozhodli se vzít, (oni) vyhledali kněze, aby je oddal."

- "Marie uviděla ve výloze prstýnek a požádala Honzu, aby jí ho koupil."



indexické výrazy:

- *indexicals*
- odkazují se na údaje v jiných částech promluvy
- "Já jsem tady."
- "Proč jsi to udělal?"

Metafora a metonymie

metafora:

- *metaphor*
- použití slov v přeneseném významu (na základě podobnosti), často systematicky
- "Zkoušel jsem ten proces zabít, ale nešlo to."
- "Bouře se vzteká."

metonymie:

- *metonymy*
- používání jména jedné věci pro (často zkrácené) označení věci jiné
- "Čtu Shakespeara."
- "Chrysler oznámil rekordní zisk."
- "Ten pstruh na másle u stolu 3 chce další pivo."

Nekompozicionalita

- *noncompositionality*
- příklady porušení pravidla kompozicionality u ustálených termínů nebo přednost jiného možného významu při určitých spojeních
- "aligátoří boty," "basketbalové boty," "dětské boty"
- "pata sloupu"
- "červená kniha," "červené pero"
- "bílý trpaslík"
- "dřevěný pes," "umělá tráva"
- "velká molekula"

Slovníky a encyklopedie

Slovníky typicky obsahují:

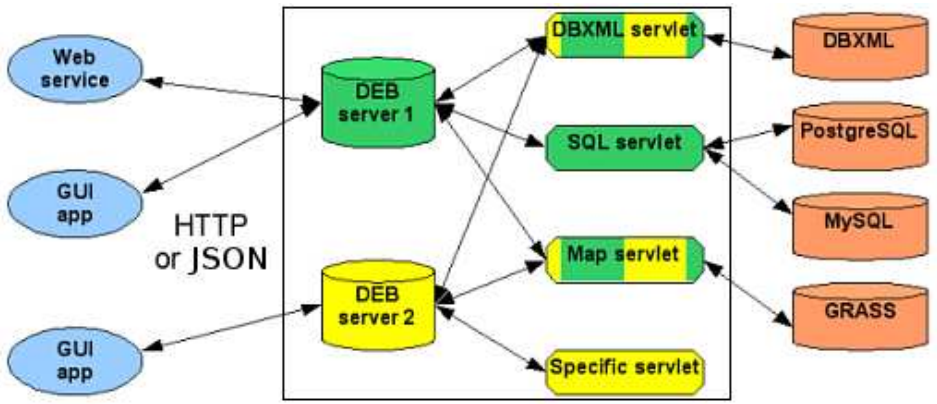
- specifikace formy:
  - grafická podoba – alternativy, dělení, velká počáteční písmena
  - zvuková podoba – výslovnost a její alternativy, slabiky, přízvuk, výška
- gramatické (morfo-syntaktické) informace – slovní druh a příslušné gramatické kategorie, morfologický vzor?
- specifikace významu – hierarchie

slovník uvádí významy listémů, encyklopedie informace o jejich denotátech

specializované lexikony a encyklopedie (znalost odborníků a rozdílné předpoklady a pohledy)

DEB – platforma pro vývoj slovníků

- platforma pro vývoj systémů na psaní slovníků
  - `http://deb.fi.muni.cz/`
  - pracuje s hesly ve formě XML struktury
- striktní klient-server architektura
- server
  - specializované moduly – *servlety*
  - databázové úložiště
- klient
  - jen jednoduchá funkcionalita
  - GUI i web rozhraní – postavený na *Mozilla Engine*

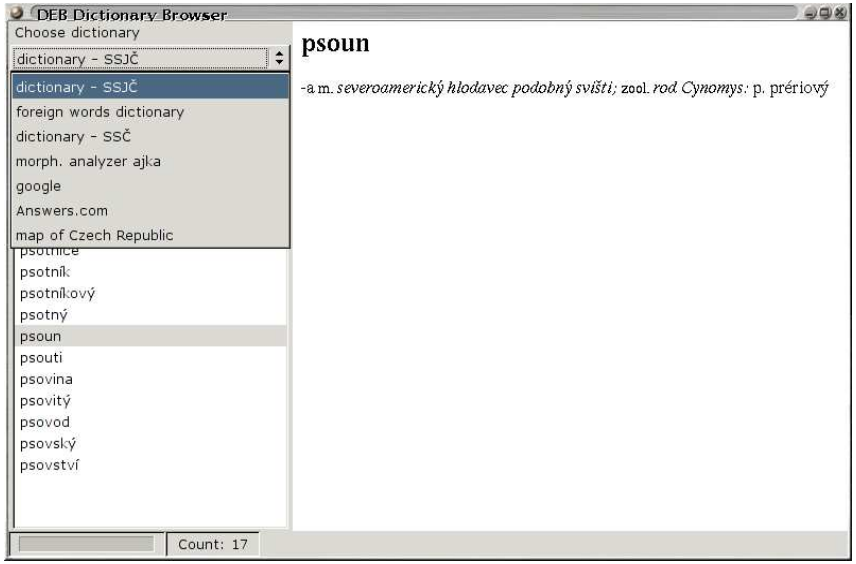


DEB používá komunikaci typu AJAX

DEBDict – příklad DEB klienta

jednoduchý klient původně určený pro demo základních funkcí

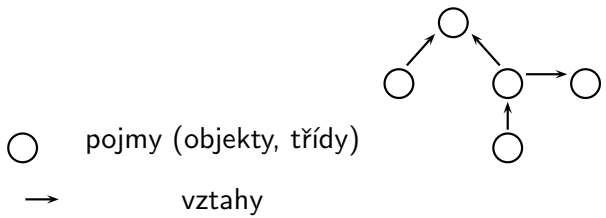
- dostupný jako instalovatelné rozšíření Firefoxu i jako vzdálená webová služba
- vícejazyčné uživatelské rozhraní (angličtina, čeština, další lze snadno doplnit)
- dotazy do několika XML slovníků s různou strukturou, výsledky jsou zpracovány XSLT transformací
- napojení na český morfologický analyzátor
- napojení na externí webové stránky (Google, Answers.com, Wikipedia)
- napojení na geografický informační systém – zobrazení geografických odkazů přímo na mapě



Sémantické sítě

sémantické sítě – reprezentace faktových znalostí (pojmy + vztahy)

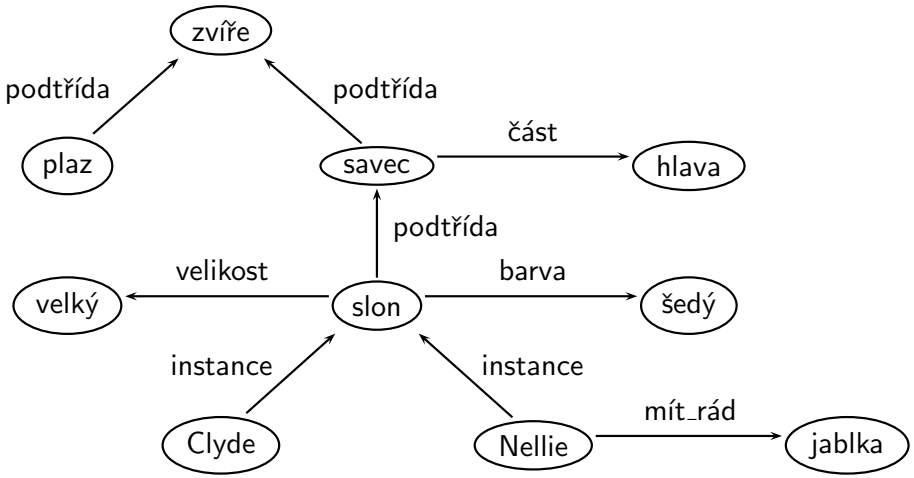
- vznikly kolem roku 1960 pro reprezentaci významu anglických slov
- znalosti jsou uloženy ve formě grafu



- nejdůležitější vztahy:
  - podtřída (*subclass*) – vztah mezi třídami
  - instance – vztah mezi konkrétním objektem a jeho rodičovskou třídou

jiné vztahy – část (has-part), barva, ...

Sémantické sítě – příklad



Dědičnost v sémantických sítích

- pojem sémantické sítě *předchazí* OOP
- dědičnost:
  - jestliže určitá vlastnost platí pro třídu→ platí i pro všechny její podtřídy
  - jestliže určitá vlastnost platí pro třídu→ platí i pro všechny prvky této třídy
- určení hodnoty vlastnosti – rekurzivní algoritmus
- potřeba specifikovat i výjimky – mechanizmus vzorů a výjimek (*defaults and exceptions*)
  - vzor – hodnota vlastnosti u třídy nebo podtřídy, platí ta, co je blíž objektu
  - výjimka – u konkrétního objektu, odlišná od vzoru

Dědičnost vztahů část/celek

- "krávy mají 4 nohy."
  - každá noha je částí krávy
- "Na poli je (konkrétní) kráva."
  - všechny části krávy jsou taky na poli
- "Ta kráva (na poli) je hnědá (celá)."
  - všechny části té krávy jsou hnědé
- "Ta kráva je šťastná."
  - ~~všechny části té krávy jsou šťastné~~ – neplatí
- lekce: některé vlastnosti jsou děděny částmi, některé nejsou explicitně se to vyjadřuje pomocí pravidel jako

$$\textit{part-of}(x, y) \wedge \textit{location}(y, z) \Rightarrow \textit{location}(x, z)$$

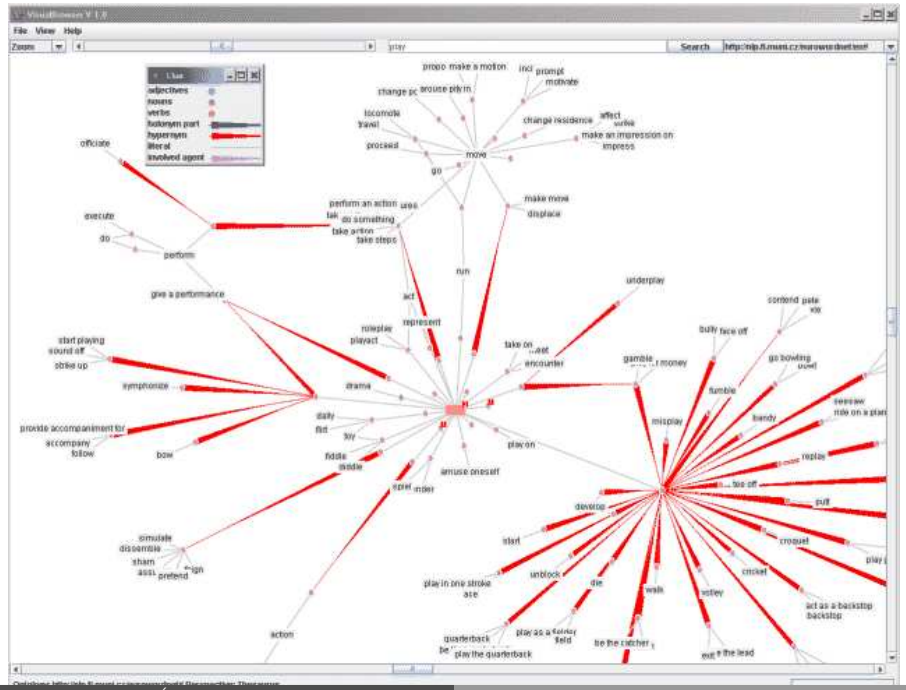
Vzory a výjimky – příklad

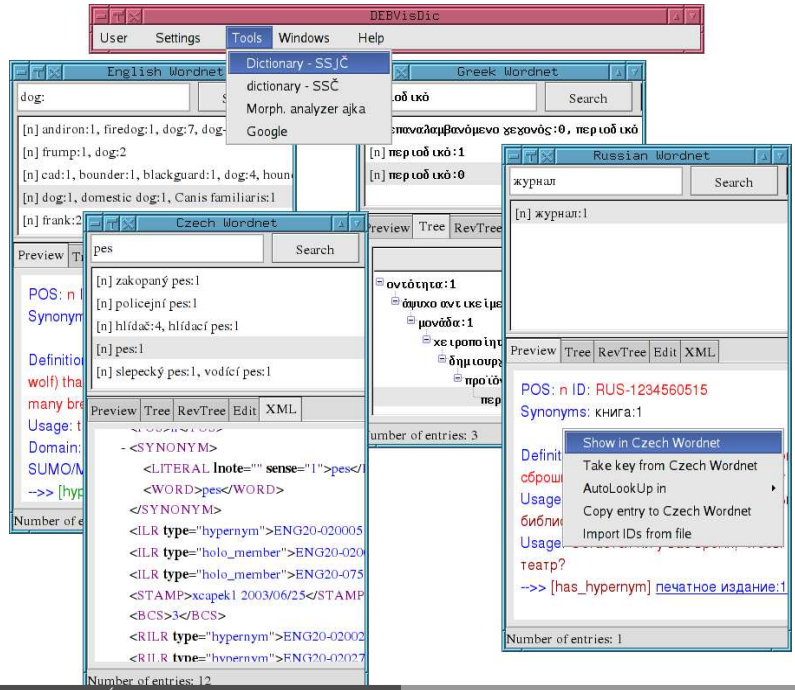
- "všichni ptáci mají křídla."
- "všichni ptáci umí létat."
- "ptáci se zlomenými křídly jsou ptáci, ale neumí létat."
- "tučnáci jsou ptáci, ale neumí létat."
- "kouzelní tučňáci jsou tučňáci, kteří umí létat."
- kdo umí létat:
  - "Tweety je pták."
  - "Petřík je tučnák."
  - "Penelope je kouzelný tučnák."
- všimněte si, že víra v hodnotu vlastnosti objektu se může měnit s příchodem nových informací o klasifikaci objektu

Aplikace sémantických sítí

(Princeton) WordNet – `http://wordnet.princeton.edu/`

- sématická síť 100.000 (anglických) pojmů, zachycuje:
  - synonyma, antonyma (významově stejná/opačná)
  - hyperonyma, hyponyma (podtřídy)
  - odvozenost a další jazykové vztahy
- tvoří se národní wordnety (navázané na anglický WN)
  český wordnet – cca 30.000 pojmů
- nástroj na editaci národních wordnetů – DEBVisDic, vyvinutý na FI MU
- VisualBrowser – `http://nlp.fi.muni.cz/projekty/visualbrowser/`
  nástroj na vizualizaci (sémantických) sítí, vznikl jako DP na FI MU





České valenční lexikony

zdroje (lexikony) slovesných valencí:

- syntaktické valenční rámce Brief (FI MU, od 1997) cca 15,000 sloves:

```
lámat <v>hPTc4,hPTc4-hTc7,hPc3-hTc4
```

- valenční rámce v českém wordnetu (FI MU 2000), cca 3,000 slovesných literálů (sloveso+význam):

```
synset: lámat:3, dobývat:1, těžit:2
valence: kdo1*AG(person:1)=co4*SUBS(substance:1)
valence: co1*AG(institution:1)=co4*SUBS(substance:1)
```

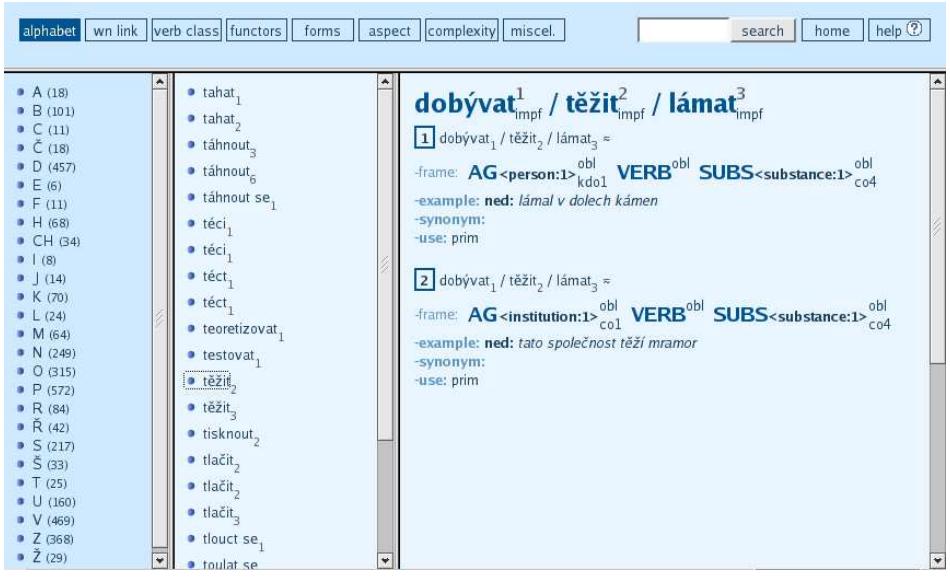
- pražský lexikon Vallex 1.0, na začátku roku 2005 cca 1,000 sloves (teď snad až 4,000):

```
~ impf: lámat
+ ACT(1;obl) PAT(4;obl)
```


Valeční lexikon VerbaLex

- vznikl na začátku roku 2005, využívá všech dostupných zdrojů aktuálně se do něj doplňují slovesa z Briefu
- edituje se v jednoduchém textovém formátu, který se pro další zpracování převádí do XML
- vlastnosti:
  - dvouúrovňové sémantické role
  - odkazy na hypero/hyponymickou hierarchii v českém wordnetu
  - odlišení životnosti a neživotnosti větných členů
  - implicitní pozice slovesa
  - valenční rámce se odkazují na číslované významy sloves
- exporty z XML do HTML pro prohlížení a PDF pro tisk

VerbaLex v HTML



Intenzionální sémantika, reprezentace znalostí

Aleš Horák

E-mail: hales@fi.muni.cz
http://nlp.fi.muni.cz/poc_lingv/

Obsah:

- Intenzionální sémantika
- Reprezentace znalostí

Logická analýza přirozeného jazyka

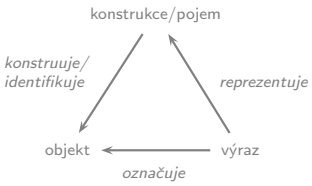
logická analýza PJ – analýza významu výrazů (vět) PJ
přirozený jazyk = nástroj *pojmového* uchopení reality
pojem – kritéria/procedury umožňující identifikovat různé konkrétní a abstraktní objekty

např. "planeta" – třída nebeských těles s určitými charakteristikami – obíhá po oběžné dráze kolem stálice, není zdrojem světla, ...

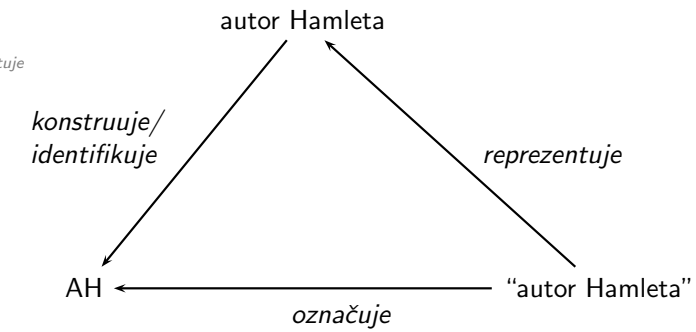
- pojem $\neq$ *výraz* – např. výrazy v různých jazycích často reprezentují stejný pojem (pojem("prvočíslo") $\equiv$ pojem("prime number"))
- pojem $\neq$ *představa* – představa je *subjektivní*, pojem je objektivní
- pojmy mohou identifikovat různé objekty:
  - jedno individuum – individuální pojmy (např. Petr, Pegas, prezident ČR)
  - třídu objektů – vlastnost (např. červený, šelma, hora)
  - n-člennou relaci – vztah (např. otec (někoho), křivdit (někdo někomu))
  - pravdivostní hodnotu – propozice (např. v Brně prší)
  - funkcionální přiřazení – empirické funkce (např. rychlost)
  - číslo – (fyzikální) veličiny (např. rychlost světla)

Vztah pojmu a výrazu

ve zjednodušené podobě: pojem odpovídá logické konstrukci



složený z pojmů 'autor' a 'Hamlet'



funkce ukazující v našem světě na Williama Shakespeara

Omezenost predikátové logiky 1. řádu

dva omezující rysy:

- nedostatečná expresivita
- extenzionalismus

Expresivita: vyjadřovací síla jazyka

"*Je-li barva stropu pokoje č. 3 uklidňující, je pokoj č. 3 vhodný pro pacienta X a není vhodný pro pacienta Y.*"

analýza ve výrokové logice:

$P \Rightarrow (Q \wedge \neg R)$

P	"Barva stropu pokoje č. 3 je uklidňující."
Q	"Pokoj č. 3 je vhodný pro pacienta X."
R	"Pokoj č. 3 není vhodný pro pacienta Y."

analýza v PL1:

$U(B) \Rightarrow (V(P, X) \wedge \neg V(P, Y))$

U	třída uklidňujících objektů
B	individuum 'barva stropu pokoje č. 3'
V	relace mezi individuy 'být vhodný pro'
P	individuum 'pokoj č. 3'
X, Y	individua 'pacient X' a 'pacient Y'

Nedostatečná expresivita PL1 – pokrač.

Červená barva je krásnější než hnědá barva. *Kostka je červená.*

analýza v PL1:

$$Kr(\check{C}_1, H) \qquad \check{C}_2(Ko)$$

$\check{C}_1$ individuum 'červená barva'
$\check{C}_2$ vlastnost individuí 'být červený' (třída červených objektů)

nelze vyjádřit $\check{C}_1 \equiv \check{C}_2$

Extenzionalismus PL1

Varšava *hlavní město Polska*

Varšava – jméno individua, jasně identifikovatelné a odlišitelné
hlavní město Polska – individuová role, momentálně identifikuje Varšavu, ale dříve to byl i Krakov

'hlavní město Polska':

- závisí na světě a čase
- pochopení významu, ale není vázané na znalost obsahu – tj. význam na světě a čase nezávisí

číslo X je větší než číslo Y *budova X je větší než budova Y*

matematické větší než – relace dvojic čísel, pevně daná
empirické větší než – vztah dvou individuí, který se může měnit v čase (otec a syn)

Extenzionalismus PL1 – pokrač.

ano *V Brně prší*

ano – pravdivostní hodnota *true*
V Brně prší – propozice – označuje pravdivostní hodnotu, která se mění (alespoň) v čase

i když hodnota někdy závisí na světě a čase, samotný význam na nich nezávisí

Extenze a intenze

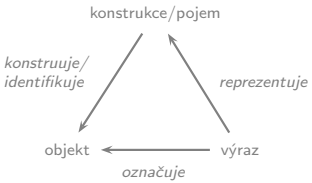
Definujeme:

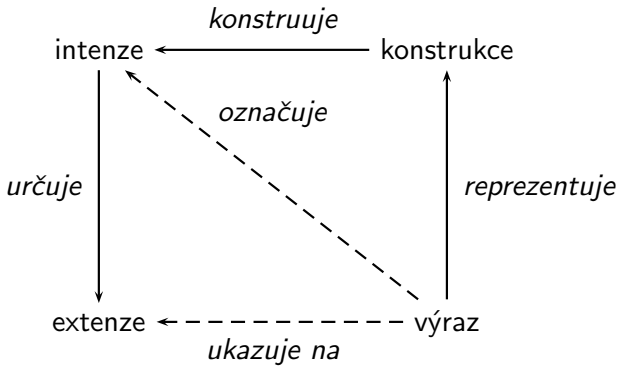
- intenze – objekty typu funkcí, jejichž hodnoty závisí na světě a čase
- extenze – ostatní objekty (na světě a čase nezávislé)

časté extenze a intenze:

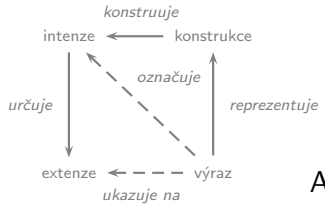
extenze	*intenze*
individua	individuové role
třídy	vlastnosti
relace	vztahy
pravdivostní hodnoty	propozice
funkce	empirické funkce
čísla	veličiny

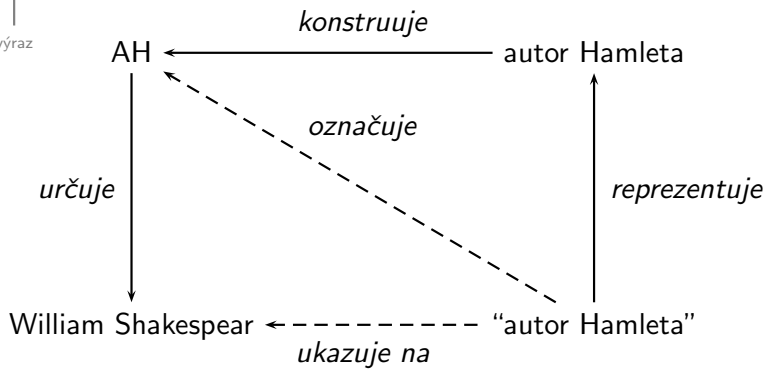
Rozšířený vztah výrazu a významu u intenzí





Rozšířený vztah výrazu a významu u intenzí





Transparentní intenzionální logika

- *Transparent Intensional Logic*, TIL
- logický systém speciálně navržený pro zachycení významu výrazů PJ
- autor Pavel Tichý: *The Foundations of Frege's Logic*, de Gruyter, Berlin, New York, 1988.
- obdobná teorie – *Montagueho intenzionální logika* – Tichý ukazuje její nedostatky
- Tichý vychází z myšlenek – *Gottlob Frege* (1848 – 1925, logik) a *Alonzo Church* (1903 – 1995, teorie typů)
- vlastnosti:
  - rozvětvená typová hierarchie (s typy vyšších řádů)
  - temporální
  - intenzionální (intenze × extenze)
- transparentost:
  1. nositel významu (konstrukce) není prvek formálního aparátu, tento aparát pouze *studuje* konstrukce
  2. zachycení intenzionality je přesně popsáno z matematického hlediska

otázka:

Jak zapíšeme znalosti o problému/doméně?
Když je zapíšeme, můžeme z nich mechanicky odvodit nová fakta?

- reprezentace znalostí (*knowledge representation*) – hledá způsob vyjádření znalostí počítačově zpracovatelnou formou (za účelem odvozování)
- vyvozování znalostí (*reasoning*) – zpracovává znalosti uložené v bázi znalostí (*knowledge base, KB*) a provádí odvození (inference) nových závěrů:
  - odpovědi na dotazy
  - zjištění faktů, které vyplývají z faktů a pravidel v KB
  - odvodit akci, která vyplývá z dodaných znalostí, ...

Reprezentace znalostí

proč je potřeba speciální reprezentace znalostí?

vnímání lidí × *vnímání počítačů*

- člověk
  - když dostane novou věc (třeba pomeranč) – prozkoumá a zapamatuje si ho (a třeba sni)
  - během tohoto procesu člověk zjistí a uloží všechny základní vlastnosti
  - později, když se zmíní daná věc, vyhledají se a připomenou uložené informace
- počítač
  - musí se spolehnout na informace od lidí
  - jednodušší informace – přímé *programování*
  - složité informace – zadané v symbolickém jazyce

Volba reprezentace znalostí

která reprezentace znalostí je nejlepší?

> *Pro řešení skutečně obtížných problémů musíme použít několik různých reprezentací. Důvodem pro to je to, že každý typ datových struktur má své přínosy i nedostatky a žádná z nich není adekvátní pro všechny různé funkce používané v tom, čemu říkáme "zdravý rozum" (common sense).*
>
> *– Marvin Minsky*

Reprezentace znalostí pomocí logiky nebo sém. sítí

Logika:

- znalosti uloženy ve formě logických formulí
- vyvozování nových znalostí = hledání důkazu

Sémantické sítě:

- reprezentace faktových znalostí (pojmy + vztahy)
- znalosti jsou uloženy ve formě grafu
- nejdůležitější vztahy:
  - podtřída (*subclass*) – vztah mezi třídami
  - instance – vztah mezi konkrétním objektem a jeho rodičovskou třídou

jiné vztahy – část (has-part), barva, ...

Rámce

Rámce (*frames*):

- varianta sémantických sítí
- velice populární pro reprezentaci znalostí v expertních systémech
- všechny informace relevantní pro daný pojem se ukládají do univerzálních struktur – rámců
- stejně jako sémantické sítě, rámce podporují dědičnost
- OO programovací jazyky vycházejí z teorie rámců

Rámce – příklad

rámec obsahuje objekty, *sloty* a hodnoty slotů
příklady rámců:

savec:
- *podtřída*: zvíře
- *část*: hlava
- **má_kožich*: ano

slon:
- *podtřída*: savec
- **barva*: šedá
- **velikost*: velký

Nellie:
- *instance*: slon
- *mít_rád*: jablka

'*' označuje vzorové hodnoty, které mohou měnit hodnoty u podtříd a instancí

Sémantické sítě × rámce

sémantické sítě	rámce
uzly	objekty
spoje	sloty
uzel na druhém konci spoje	hodnota slotu

deskripční logika – logický systém, který manipuluje přímo s rámci

Pravidlové systémy

- snaha zachytit produkčními pravidly znalosti, které má expert
- obecná forma pravidel

  IF *podmínka*
  THEN *akce*

  - podmínky – booleovské výrazy, dotazy na hodnoty proměnných
  - akce – nastavení hodnot proměnných, příznaků, ...
- důležité vlastnosti:
  - znalosti mohou být strukturovány do modulů
  - systém může být snadno rozšířen přidáním nových pravidel beze změny zbytku systému

Pravidlová báze znalostí – příklad

pravidla pro oblékání:

pravidlo 1
IF X je seriózní
AND X bydlí ve městě
THEN X by měl nosit sako

pravidlo 2
IF X je akademik
AND X je společensky aktivní
AND X je seriózní
THEN X by měl nosit sako a kravatu

pravidlo 3
IF X bydlí ve městě
AND X je akademik
THEN X by měl nosit kravatu

pravidlo 4
IF X je podnikatel
AND X je společensky aktivní
AND X je seriózní
THEN X by měl nosit sako, ale ne kravatu

společenská pravidla:

pravidlo 5
IF X je podnikatel
AND X je ženatý
THEN X je společensky aktivní

pravidlo 6
IF X je akademik
AND X je ženatý
THEN X je seriózní

profesní pravidla:

pravidlo 7
IF X učí na univerzitě
OR X učí na vysoké škole
THEN X je akademik

pravidlo 8
IF X vlastní firmu
OR X je OSVČ
THEN X je podnikatel

Expertní systémy

- aplikace pravidlových systémů
- zaměřeny na specifické oblasti – medicínská diagnóza, návrh konfigurace počítače, expertíza pro těžbu nafty, ...
- snaha zachytit znalosti experta pomocí pravidel
  ale znalosti experta zahrnují – postupy, strategie, odhady, ...
- expertní systém musí pracovat s procedurami, nejistými znalostmi, různými formami vstupu
- vhodné oblasti pro nasazení expertního systému:
  - diagnóza – hledání řešení podle symptomů
  - návrh konfigurace – složení prvků splňujících podmínky
  - plánování – posloupnost akcí splňujících podmínky
  - monitorování – porovnání chování s očekávaným chování, reakce na změny
  - řízení – ovládání složitého komplexu
  - předpovědi – projekce pravěpodobných závěrů z daných skutečností
  - instruktáž – inteligentní vyučování a zkoušení studentů

Nejistota a pravděpodobnost

definujme akci A_t jako "Vyrazit na letiště t hodin před odletem letadla."
jak najít odpověď na otázku "*Dostanu se akcí A_t na letiště včas?*"

problémy:

1. částečná pozorovatelnost (stav vozovky, záměry ostatních řidičů, ...)
2. šum v senzorech (hlášení o dopravní situaci)
3. nejistota výsledků akcí (píchnutí kola, ...)
4. obrovská složitost modelování a předpovědi dopravní situace

čistě logický přístup tedy:

- riskuje chybu – "A_5 mě tam dostane včas."
- vede k závěrům, které jsou příliš slabé pro rozhodování: "A_5 mě tam dostane včas, pokud nebude na dálnici nehoda a pokud nebude pršet a jestli nepíchnu kolo ..."

Metody pro práci s nejistotou

- defaultní/nemonotónní logika
  Předpokládejme, že nepíchnu cestou kolo.
  Předpokládejme, že A_5 bude OK, pokud se nenajde protipříklad.
- pravidla s faktory nejistoty
  $A_5 \mapsto_{0.3}$ dostat se na letiště včas.
  zalévání $\mapsto_{0.99}$ mokrý trávník
  mokrý trávník $\mapsto_{0.7}$ déšť
- pravděpodobnost
  Vzhledem k dostupným informacím, A_3 mě tam dostane včas s pravděpodobností 0.05.

Vybrané aktuální projekty Laboratoře NLP

Pavel Šmerk, Miloš Husák

E-mail: xsmerk@fi.muni.cz,xhusak@fi.muni.cz
http://nlp.fi.muni.cz/poc_lingv/

Obsah:

- Guesser – značkování českých neznámých slov
- GDEX – Good Dictionary Examples

Guesser

Značkování českých neznámých slov

Pavel Šmerk
xsmerk@fi.muni.cz

Problém

Není možné mít všechna slova, která lze při zpracování přirozeného jazyka "potkat", předem ve slovníku. Je tedy potřeba umět řešit situaci, kdy slovo ve slovníku není.

Potřebné informace o neznámém slovu mohu odhadnout buď to z jeho větného kontextu, nebo z jeho podoby, konkrétně z jeho konce. Optimální řešení bude pochopitelně kombinovat oba přístupy.

Dále bude řeč o druhé variantě. Tedy dostanu slovo *napipetovaná*, řeknu si, aha, to je jako *malovaná*, tedy lemma *napipetovaný* podle *malovaný*, značky stejné.

Produktivita

Čeština má relativně velké množství vzorů pro ohýbání i odvozování slov. Velká část z nich ale není v současnosti "produktivních", jsou to jen pozůstatky z minulosti (*den, husa, myš, kost; dřevěný vs. umakartový, ...*), případně jsou produktivní jen nějak omezeně (*pařba x *krvácba*).

Nově se objevující *česká* slova budou tedy patřit k nějaké podmnožině produktivních vzorů. Při odhadování konkrétního vzoru budou neproduktivní vzory nadbytečné, naopak dokonce budou tvořit šum a "překážet". My ale (zatím) nemáme informaci, které vzory jsou produktivní.

Předpoklad

Můžeme předpokládat, že slovníky zahrnují všechna nepravidelně se chovající slova, která lze reálně "potkat" (pokud bychom se dostali k nějakému historickému textu, nemuselo by to tak být). Zároveň byly jistě tvořeny se snahou o co největší pokrytí reálného jazyka. To znamená, že se přidávala "nová", pravidelně se chovající slova, např. z různých oborů.

Jde si to představit i jinak: máme-li 390 000 lemmat, není možné, aby naprostá většina z nich nebyla pravidelná, to by člověk nezvládl udržet v hlavě.

Pak si lze myslet, že to, co je produktivní, poznáme podle četnosti ve stávajících datech. Cílem je tedy z dat extrahovat jen jejich produktivní (častou) část a pouze na jejím základě pak odhadovat morfologické vlastnosti neznámých slov.

Konstrukce dat pro guesser

- ze slovníku ajky (.dic) vyhodím lemmata od vzorů, u kterých je méně než 20 lemmat (tedy cca 6000 lemmat z 390 000, 55 000 slovních tvarů)
- pro všechny tvary každého lemmatu vygeneruju trojice word-lemma-tag ve formátu `abaraks:a::k1gMnSc2,k1gMnSc4`, kde nejprve je obrácený slovní tvar, dále, co musím utrhnout, a za další dvojtečkou, co musím přidat, abych dostal lemma, nakonec značky
- beru jen slova délky alespoň 7, vyhazuju eN, d3 a věci mimo k[1256], neberu wH a pod.
- setřídím podle abecedy (nikoli česky)

Konstrukce dat pro guesser

- procházím a zaznamenávám do pole, min. kořen 3 znaky, max. konec 10 znaků
- pokud v daném místě pole bylo něco jiného, zkusím vypsat. Pro výpis to musí mít aspoň 20 výskytů, jednotlivé značky aspoň 10x, pokud je něčeho víc než 20x méně než zbytku, ignoruju.
- vypsané odečítám od levých podřetězců
- setřídím podle abecedy
- slučuju stejné věci (s ohledem na počty), ignoruju vid
- setřídím retrográdně kvůli snadnému nacpání do `$re (...|ab|a)`, viz dále

Použití

- hádaná slova otáčím a zkouším `=~ /($re).{$minroot}/`
  `# $minroot == 3`
- díky "uspořádání" `/^(ab|a)/` se pro *abxyz* namatchuje *ab*, a nikoli *a*, tedy obecně nejdelší možný konec
- k nalezenému konci slova mám v hashi info o změně konce na lemma a příslušných značkách

`/nlp/projekty/rule_ind/stat/guesser.pl`, očekává vertikál, případně označkovaný něčím jako `ajka -c -` (existující značkování ponechává beze změny)

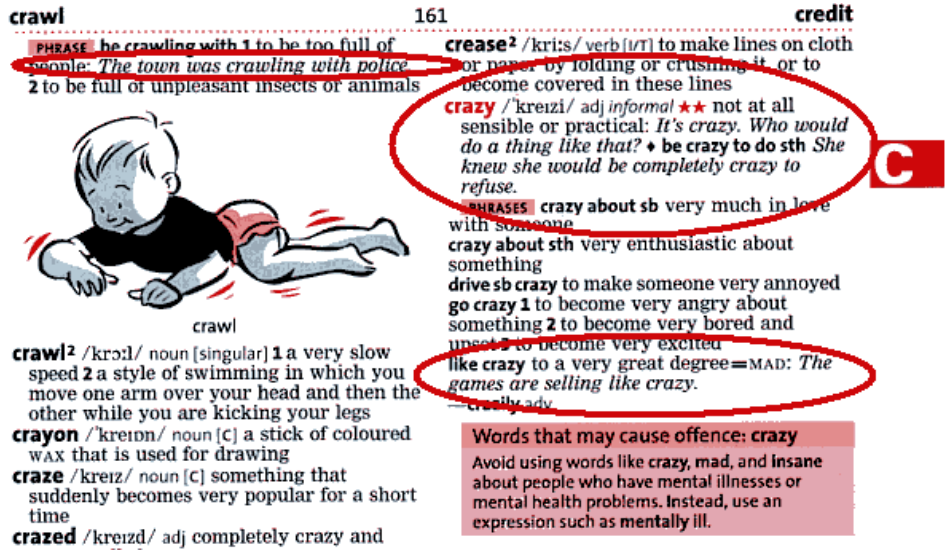
Problém: zdaleka ne všechna neznámá slova jsou česká, tedy začleněná do systému jazyka. Zejména asi nesklonné názvy, jména, případně i cizojazyčná slova či celé citáty. Hodně z nich by asi bylo možné odhadovat pomocí velkého písmena na začátku a absence diakritiky. Zatím nijak netestováno/neřešeno.

GDEX

Good Dictionary EXamples
Dobré slovníkové příklady

Miloš Husák
xhusak@mail.muni.cz

Slovníkový příklad



crawl 161 credit

PHRASE be crawling with 1 to be too full of people: *The town was crawling with police* 2 to be full of unpleasant insects or animals

crawl

crawl² /krɔːl/ noun [singular] **1** a very slow speed **2** a style of swimming in which you move one arm over your head and then the other while you are kicking your legs

crayon /ˈkreɪɒn/ noun [C] a stick of coloured WAX that is used for drawing

craze /kreɪz/ noun [C] something that suddenly becomes very popular for a short time

crazed /kreɪzd/ adj completely crazy and

crease² /kriːs/ verb [I/T] to make lines on cloth or paper by folding or crushing it, or to become covered in these lines

crazy /ˈkreɪzi/ adj *informal* ★★ not at all sensible or practical: *It's crazy. Who would do a thing like that?* ♦ **be crazy to do sth** *She knew she would be completely crazy to refuse.*

PHRASES **crazy about sb** very much in love with someone

crazy about sth very enthusiastic about something

drive sb crazy to make someone very annoyed

go crazy 1 to become very angry about something **2** to become very bored and upset **3** to become very excited

like crazy to a very great degree=MAD: *The games are selling like crazy.*

—crazily adv

C

Words that may cause offence: crazy

Avoid using words like **crazy**, **mad**, and **insane** about people who have mental illnesses or mental health problems. Instead, use an expression such as **mentally ill**.

Jak jsem ke slovníkům přišel/přešel

- brigáda "na léto"
- řecký korpus
- ohodnocování slovníkových příkladů
- Python, Manatee, SkE, WebBootCat ....
- bakalářská práce

Cíl práce

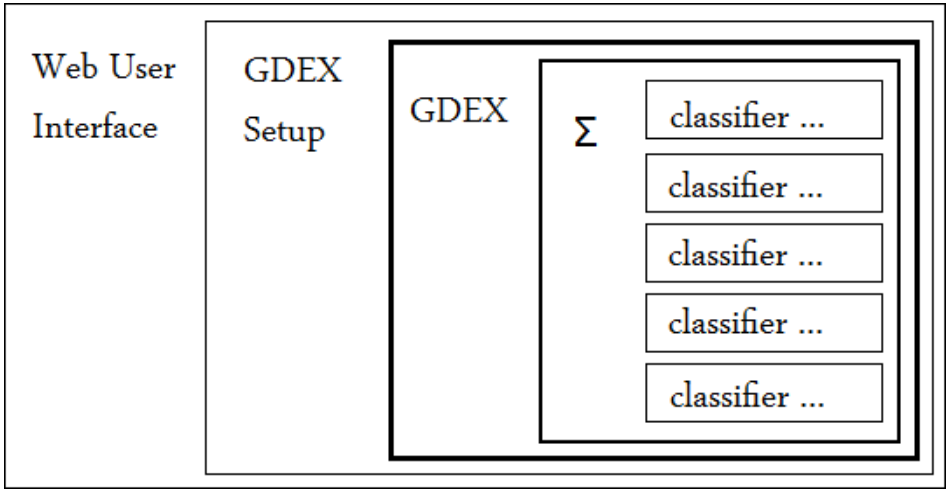
Zautomatizovat (alespoň částečně) vyhledávání slovníkových příkladů v korpusu.

- zjistit, jak se dobré příklady dají rozpoznat
- zkombinovat různé přístupy
- vytvořit snadno použitelné rozhraní

Dobrý příklad je ...

- informativní, samopopisný
  - pozice slova ve větě
  - synonyma
  - redundance
- snadno čitelný
  - jednoduchá větná struktura
  - krátká slova
  - běžná jména (politické problémy)
  - častá slovní spojení

GDEX



GDEX

GDEX je srdcem a tělem ohodnocování vět.

- hostí hlavní klasifikátor
- poskytuje jednotné rozhraní všem klasifikátorům
- spravuje dostupné prostředky

Klasifikátory

Klasifikátory jsou mozkem ohodnocování vět.

- každou větu ohodnotí číslem z $< 0, 1 >$
- silné a slabé klasifikátory
- délka vět, délka slov, četnost slov, počty významů, ...
- vážený průměr, progresivní skóre, přibližné progresivní skóre

Měření efektivity

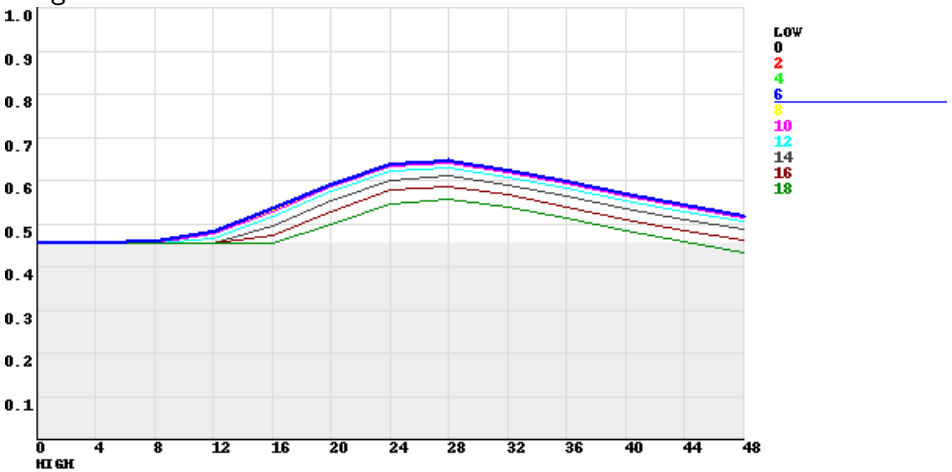
$$benchmark = 1 - \frac{\sum_{i \in concordances} \frac{\sum_{j \in goodexamples_i} n_{i,j}}{n_{last\_good\_example_i}+1}}{\sum_{i \in concordances} |goodexamples_i|}$$

Měření efektivity

- průměrná pozice dobrých slovníkových příkladů
- ručně značkované konkordance
- optimalizace parametrů klasifikátorů
- vyvažování důležitostí klasifikátorů

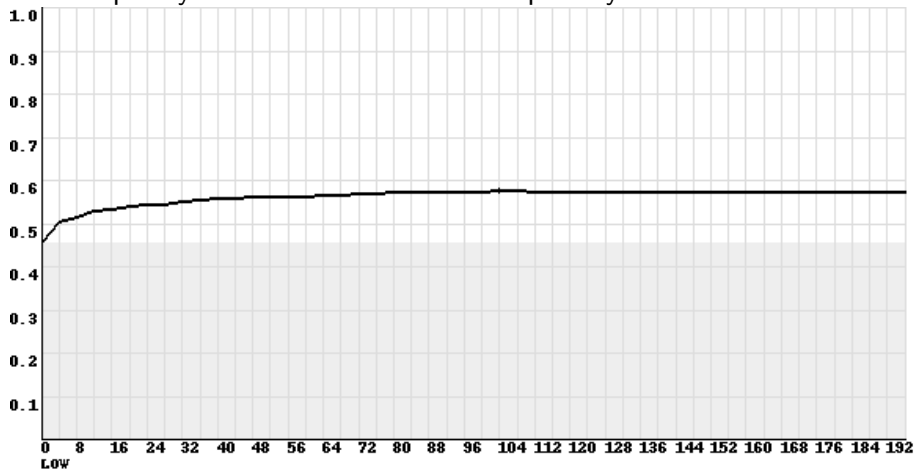
Dobrý klasifikátor I.

Length score - délka vět



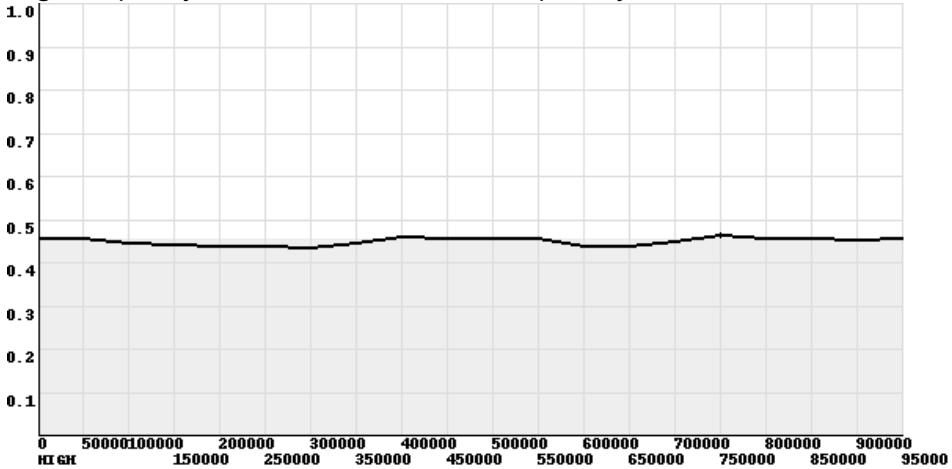
Dobrý klasifikátor II.

Low frequency score - minimální četnost použitých slov



Špatný klasifikátor

High frequency score - maximální četnost použitých slov



Ukázky

- GDEX Setup
- Sketch Engine
- GDEX Demo Dictionary

GDEX Setup

Config file | Constants | Classifiers | Preview | Learn | Archive

Name		Weight	Parameters				
weighted score	0 (weighted)	0					
length score	1 (length)	15	LOW :	10	HIGH :	25	
low frequency score	1 (lowfreq)	15	LOW :	44			
verbs score	2 (verbs)	5	VERBNUM :	3			
position score	1 (position)	1	IDEAL :	0.2	TOLERANCE :	0.6	
whole sentence	3 (sentence)	4					
long words score	1 (wordlen)	10	MAX :	7	PENALTY :	0.2	
wordlist min	3 (wordlist_min)	5	WORDLIST :	['the', 'be', 'to'	MIN :	0.25	
all colloc. avg	2 (all_colloc_avg)	10	FUNC :	DICE			
	Create new classifie		Strong classifier	Create/Update			

SketchEngine

http://beta.sketchengine.co.uk/

Home | Concordance | Word List | Word Sketch | Thesaurus | Sketch-Diff | View

Corpus: **British National Corpus**
conc description

View options

Attributes
- [x] word
- [ ] ambtag
- [ ] lempos
- [ ] tag
- [ ] lemma
- [ ] lc

Display attributes
- ( ) For each token
- (•) KWIC tokens only

Structures: <bncdoc> <stext> <text> <s> <p> <align> <caption> <hi> <lb> <bibl> <body> <div1>

References: Token number; bncdoc.id; bncdoc.date; bncdoc.author; bncdoc.title; bncdoc.info; Text availability; Text type; Publication date; David Lee's classification; Domain for written corpus texts; Domain for context-governed spoken material; Medium for written corpus texts

Page size (number of lines): 20
KWIC Context size (number of characters): 40

- [x] Icon for one-click sentence copying
- [ ] Sort concordance lines best-first. Number of lines to be sorted: 100

Change View Options

GDEX Demo Dictionary

http://forbetterenglish.sketchengine.co.uk/

The GDEX Demo Dictionary

This is an experimental automatic collocations dictionary, based on the Sketch Engine technology. Methods are described in Kilgarriff et al 2008: GDEX: Automatically finding good dictionary examples in a corpus. Proc EURALEX, Barcelona, Spain.

Enter an (English) word here to see its collocations, each with an example sentences.

Find

example (n)

pp_of	practice :	There are many examples of good practice being carried out by them on a daily basis .
	architecture :	West Hall , Silver Street The original building , opened in 1938 , was a striking example of library architecture of the nineteens-thirties .
	type :	The following are examples of the various types of groups .
	kind :	An example of this kind of mistaken identity , occurred in 1932 , when a crocodile-like creature was seen in the loch .
	genre :	However , this is only one , startling , example of the genre .
	latter :	An example of the latter is the specific knowledge attribute , used for holding subordinate and cross references .
object_of	cite :	We can cite numerous examples throughout history in connection with this topic .
	give :	Here she gives two examples of companies that are finding a way to do all three .
	follow :	If you have ever been tempted to believe what you are told then consider some of the following examples .
	shine :	A shining example for all of humanity she is super awesome .
	set :	Show Them How Jamie is a highly visible leader who sets a great example to his team .
	see :	To see the next example , use the ' page down ' button .
a_modifier	prime :	You just had to choose your favourite marque and there would be prime examples parked up across the circuit .
	typical :	A typical example of this condition of anxiety hysteria is seen in the case of a man who suffered from Awriters ' cramp ' .
	fine :	The Edna Wright LP is a fine example of mid seventies soul .
	classic :	Physics is a classic example of complexity in studies .
	above :	Aliasing Aliasing means that more than one reference is tied to the same object , as in the above example .
	excellent :	An excellent example is at Rey Cross near Bowes in County Durham .
modifies	sentence :	If the mouse is placed on an example sentence , the English translation will pop up on the screen .
n_modifier	case :	In all cases examples are given for both the humanities style and the scientific style .

Budoucí práce

- implementovat další klasifikátory
- naučit GDEX i na jiných datech/jazycích
- reimplementovat GDEX v C/C++