Závislostní a složkové modelování syntaxe jazyků*

Martin Plátek[1] a Tomáš Holan[2]

[1] KTIML MFF UK, Praha
platek@ksi.ms.mff.cuni.cz
[2] KSVI MFF UK, Praha
holan@ksvi.mff.cuni.cz

1 Úvod

V tomto textu se snažíme čtenáře seznámit se základními prvky modelování syntaxe přirozených jazyků. Předpokládáme, že se již setkal s teorií grafů a formálních jazyků.

Protože náš mateřský jazyk má bohaté tvarosloví a volný slovosled, klademe důraz na techniky, které si s těmito jevy dovedou poradit. Řadu zde zavedených pojmů čtenář v základních kursech formálních gramatik nenajde. Např. zde zavedené závislostní gramatiky dovolují redukovat nesouvislé podřetězy analyzovaného řetězu. Na tuto skutečnost navazují prezentované techniky algoritmů rozpoznávání a syntaktické analýzy. Nicméně se snažíme na pojmy a techniky z úvodních kurzů navázat. Nejprve zařazujeme jednoduché klasické typy formálních gramatik (kategoriální a bezkontextové) a jim odpovídající varianty algoritmů rozpoznávání a syntaktické analýzy (varianty CYK-algoritmu).

2 Základní syntaktické struktury

V této sekci zavádíme základní syntaktické struktury užívané v souvislosti s větnou skladbou. Jsou to závislostní stromy, složkové stromy a DR-stromy. DR-stromy integrují vlastnosti prvních dvou typů, mezi lingvisty jsou (pod tímto jménem) méně známé. Všechny tři typy syntaktických struktur zavádíme tak, že odpovídají pojmu stromu ve smyslu teorie grafů. Z těchto struktur lze ale vyčíst i řadu informací nesouvisejících se stromovou strukturou, například informace o pozici uzlů, která je pro naše úvahy velmi důležitá.

2.1 Závislostní stromy

Závislostní strom (dependency tree, D-strom) je v evropské lingvistice obvyklým formálním prostředkem pro znázorňování struktury věty. Tradiční definici lze nalézt například v [Kunze, 1972] nebo v [Nebeský, 1972]. My uvádíme modifikovanou definici, zdůrazňující geometrické a datové aspekty tohoto pojmu.

* Práce vznikla za podpory projektů GAČR s čísly 201/02/1456 a 201/04/2152.

V prvním přiblížení můžeme závislostní strom popsat pomocí vztahu označovaného jako *závislost*. Ten se v naší (i kontinentální) lingvistice používá k formalizaci jazykového jevu zvaného *podřízenost*. Podřízenosti jsou syntakticky rozlišitelné a jejich souhrn tvoří významový větný invariant. Jinými syntakticky rozlišitelnými jevy jsou koordinace (souřadnosti) a apozice. Ty nejsou vhodné pro modelování pomocí závislostí.

Ve folklóru (výpočetní) lingvistiky jsou slova závislost a podřízenost užívána jako synonyma. To vede v diskusích k častým nedorozuměním, jakým bychom se rádi vyhnuli. Slovo závislost zde užíváme jako pojem spjatý s dalšími formálními pojmy. Máme na mysli závislostní stromy, závislostní gramatiky, kategoriální gramatiky a další. Pomocí závislostí lze modelovat nejen podřízenost, ale i jiné jevy. Např. v tree-banku analytických stromů (PDT, viz [Hajič, 1998]) jsou pomocí závislostí zachyceny tzv. předložkové skupiny, vazby typu buď-nebo, vazby interpunkčních znamének se slovními tvary a podobně. V současném 'natural language processing' se datový typ 'závislostní strom' užívá s minimálními a obvykle zcela nespecifikovanými ohledy na syntaktický fenomén 'podřízenosti'. To je velmi nešťastné.

Přejděme k neformální charakteristice pojmu 'závislostní strom' nad danou větou. Každé slovo ve větě, až na jedno, je (přímo) závislé na některém jiném slově; tyto vztahy závislosti představují částečné uspořádání, které můžeme zobrazit jako strom. Uzly stromu odpovídají jednotlivým výskytům slovních tvarů, nesou v sobě informaci o slovním tvaru, o jeho pořadí ve větě a o uzlu, na němž tento uzel závisí. Uzel představující minimum zmíněného částečného uspořádání tvoří kořen stromu.

Závislostní stromy je zvykem znázorňovat buď jako hierarchizovaný řetězec, kde je závislost ilustrovaná pomocí šipek [Melčuk, 1988]:

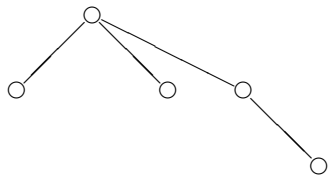
Obrázek 1. První způsob znázornění závislostního stromu

nebo jako stromový graf s uzly odrážející v horizontálním směru pořadí slov ve větě a ve vertikálním směru hierarchii závislostí:

Přistupme nyní k definici formalizující druhý typ vizualizace závislostního stromu.

Definice 1 (Závislostní strom, D-strom) Nechť A je konečná abeceda (slovník), nechť R je konečná množina funktorů, nechť $w = a_1 \dots a_n$ je věta, $w \in A^*$, nechť T je strom s kořenem. Řekneme, že T je *závislostní strom* (D-strom, „D-" podle „dependency tree") nad větou w, jestliže pro každé $i \in \{1 \dots n\}$ existuje právě jeden uzel stromu T a libovolný uzel stromu T je pětice tvaru $[a_i, i, v_i, d_i, R_i]$, kde

- $i \in \{1 \dots n\}$
- a_i je i-té slovo věty w



Obrázek 2. Druhý způsob znázornění závislostního stromu

- $R_i \in R \cup \{\lambda\}$; $v_i, d_i \in \{0 \dots n\}$;
  R_i slouží k označení hrany symbolem pro druh závislosti, i zachycuje horizontální pozici uzlu
- v_i je vzdálenost (měřená v počtu hran) uzlu od kořene stromu,
- d_i je horizontální pozice řídícího uzlu (existuje-li), jinak $d_i = 0$
- $v_i = 0$ právě když $d_i = 0$. V tom případě je $R_i = \lambda$ a uzel $[a_i, i, 0, 0, \lambda]$ je kořenem stromu T.
- je-li $u = [a_i, i, v_i, d_i, R_i]$ a $d_i \neq 0$ (T obsahuje $n-1$ takových uzlů), potom existuje vždy právě jeden uzel $v = [a_{d_i}, d_i, v_i - 1, d_{d_i}, R_{d_i}]$. Dvojice (u, v) tvoří orientovanou (od závislého uzlu k řídícímu uzlu) hranu stromu T označenou symbolem R_i.

Je-li $u = [a_i, i, v_i, d_i, R_i]$, budeme a_i označovat jako *symbol* uzlu u, i jako jeho *horizontální index*, v_i jako jeho *vertikální index*, d_i jako jeho *dominační index* a R_i je funktorem hrany vycházející z u, nebo funktorem kořene. Funktor hrany může chybět.Tuto skutečnost označujeme symbolem λ.

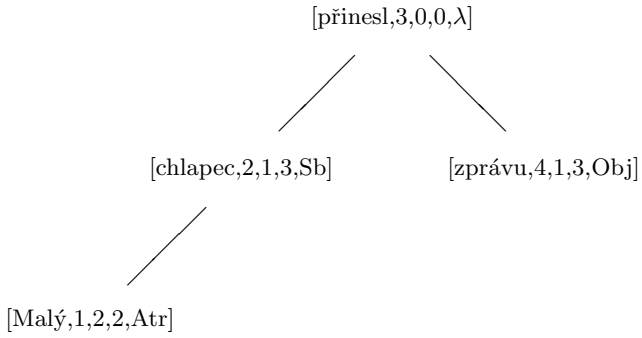
Zavedeme ještě standardním způsobem stupeň uzlu u D-stromu T. *Stupněm* u nazveme počet hran, které vedou do u (ze závislých uzlů). *Stupněm stromu* T nazveme maximum ze stupňů všech uzlů stromu T.

Poznámka 2 Neřekneme-li jinak, budeme v této i dalších definicích, kdykoli budeme hovořit o hranách a cestách, vždy rozumět orientované hrany a orientované cesty.

Poznámka 3 Můžeme si povšimnout, že vertikální indexy uzlů ve stromě lze odvodit z ostatních informací závislostního stromu. Reprezentují však geometrickou konvenci, která je s pojmem závislostního stromu tradičně spojována [Nebeský, 1972]. Nemusíme tak tuto konvenci zavádět zvlášť, jak bývalo zvykem.

Příklad 4 *Obrázek 3 znázorňuje závislostní strom T_1 nad větou „Malý chlapec přinesl zprávu".*

Uzly stromu jsou: u_1=[Malý,1,2,2,Atr], u_2=[chlapec,2,1,3,Sb], u_3=[přinesl,3,0,λ], u_4=[zprávu,4,1,3,obj]; hrany stromu jsou (u_1, u_2), (u_2, u_3) a (u_4, u_3).



Obrázek 3. Závislostní strom T_1 nad větou ,,Malý chlapec přinesl zprávu"

Nyní zavedeme pojem *pokrytí uzlu D-stromu.* Definovaný pojem určuje, jakou část věty pokrývá podstrom s kořenem v určitém uzlu. Pomocí pokrytí později zavedeme (ne)projektivitu jako případ, kdy pokrytí (ne)tvoří souvislý (pod)řetězec.

Definice 5 (Pokrytí uzlu D-stromu) Nechť T je D-strom a nechť u je uzel stromu T. $Cov(u, T)$ označíme množinu horizontálních indexů všech uzlů stromu T, ze kterých vede cesta do u. Uvažujeme i prázdnou cestu a tedy $Cov(u, T)$ vždy obsahuje také horizontální index uzlu u.

Řekneme, že $Cov(u, T)$ je *pokrytí uzlu u (podle stromu T).*

Příklad 6 *Je-li T_1 závislostní strom z minulého příkladu a u_1, u_2, u_3, u_4 jeho uzly, potom $Cov(u_1, T_1) = \{1\}$, $Cov(u_2, T_1) = \{1, 2\}$, $Cov(u_3, T_1) = \{1, 2, 3, 4\}$, $Cov(u_4, T_1) = \{4\}$.*

Zatím jsme se nezabývali tím, jak může vypadat pokrytí; víme jen tolik, že u stromu nad větou $a_1 \dots a_n$ je pokrytí kteréhokoliv uzlu vždy podmnožina množiny $\{1 \dots n\}$.

Nyní zavedeme pojem *díry.*

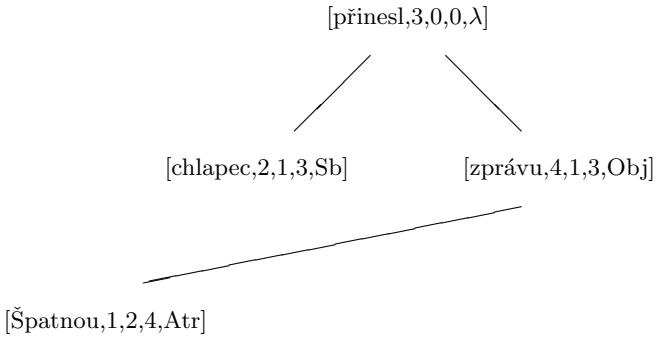
Definice 7 (Díra (v pokrytí))
Nechť T je závislostní strom nad větou $w = a_1 \dots a_n$, nechť u je uzel stromu T a nechť

$$Cov(u, T) = \{i_1, i_2, \dots, i_m\}, i_1 < i_2 < \dots < i_{m-1} < i_m.$$

Řekneme, že dvojice (i_j, i_{j+1}) tvoří *díru v $Cov(u, T)$*, jestliže
$1 \leq j < m$, a zároveň $i_{j+1} - i_j > 1$.

Příklad 8 *Pokrytí uzlů stromu T_1 (Obrázek 3) neobsahují žádné díry.*

Příklad 9 *Obrázek 4 znázorňuje závislostní strom T_2 nad větou ,,Špatnou chlapec přinesl zprávu".*



Obrázek 4. Závislostní strom T_2 nad větou „Špatnou chlapec přinesl zprávu"

Uzly stromu jsou: u_1*=[Špatnou,1,2,4,Atr],* u_2*=[chlapec,2,1,3,Sb],* u_3*=[přinesl,3,0,0,λ],* u_4*=[zprávu,4,1,3,Obj]; hrany stromu jsou* (u_1, u_4), (u_2, u_3) *a* (u_4, u_3).

Pokrytí uzlů stromu T_2 *jsou:* $Cov(u_1, T_2) = \{1\}$, $Cov(u_2, T_2) = \{2\}$, $Cov(u_3, T_2) = \{1, 2, 3, 4\}$, $Cov(u_4, T_2) = \{1, 4\}$.

Pokrytí čtvrtého uzlu $Cov(u_4, T_2) = \{1, 4\}$ *obsahuje díru.*

Pojmy *projektivita* a *neprojektivita* jsou v souvislosti se závislostními stromy běžně užívány k vyjadřování informací o slovosledu věty. Definice těchto pojmů lze nalézt například v [Nebeský, 1972].

Budeme-li v takové souvislosti hovořit o větě, budeme mít vždy na mysli větu spolu s některým závislostním stromem nad touto větou.

Neprojektivní závislost (neprojektivitu) lze neformálně popsat jako vztah závislosti mezi dvěma slovy oddělenými ve větě třetím slovem, které nezávisí (ani zprostředkovaně) na žádném z nich.

D-strom nad danou větou *je neprojektivní*, obsahuje-li nějakou neprojektivní závislost. V opačném případě je D-strom *projektivní*.

Například závislostní strom T_1 nad větou „Malý chlapec přinesl zprávu" , uvedený v Příkladu 1, je projektivní. Naopak D-strom T_2 nad větou „Špatnou chlapec přinesl zprávu", ve kterém se přívlastek „Špatnou" vztahuje ke slovu „zprávu", je neprojektivní.

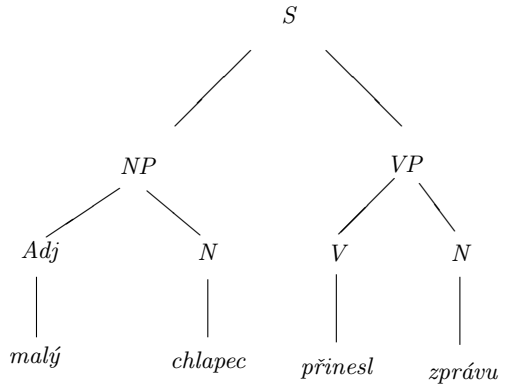
Projektivita či neprojektivita D-stromu T nad větou v je projevem vztahu slovosledu věty v ke struktuře stromu T. V článku [Holan at al., 1998] bylo ukázáno, že míra neprojektivity D-stromů nad větou, resp. horní hranice této míry ovlivňuje složitost syntaktické analýzy.

Nyní uvedeme definici projektivity a neprojektivity.

Definice 10 (Projektivita, neprojektivita) Nechť T je závislostní strom nad větou w. Je-li pokrytí každého uzlu u stromu T bez díry, budeme říkat, že T je projektivní (D-projektivní). Obsahuje-li T alespoň jeden uzel, jehož pokrytí má díru, říkáme, že T je neprojektivní (D-neprojektivní).

2.2 Složkové stromy

V této sekci zavádíme složkové stromy a systémy bezprostředních složek. Snažíme se, aby složkové stromy průhledně reprezentovaly systémy bezprostředních složek (immediate constituens) nad větou a to i systémy složek nespojitých. Složkové stromy (C-stromy) dělí větu postupně na části (složky), přičemž nepřihlížejí k rozlišení na podřadné a řídící složky, ani si nekladou za cíl odlišit vzájemně souřadné složky od ostatních. Diagram jednoduchého složkového stromu vidíme na Obr.5.



Obrázek 5. Diagram složkového stromu T_s nad větou „Malý chlapec přinesl zprávu"

Definice 11 (C-strom)
Nechť A je konečná množina symbolů (terminálních symbolů), $w = a_1 \dots a_n$ je věta, kde $a_1, \dots a_n \in A$. Nechť K_t je konečná množina symbolů (složkových kategorií), $K_t \cap A = \emptyset$. Řekneme, že strom T je *složkový strom (C-strom) nad větou* w, jestliže se množina jeho uzlů dělí na listy (terminální uzly) a složkové uzly (mezi složkové uzly patří i kořen stromu T) a T má přitom následující vlastnosti:

a) Pro každé $1 \leq i \leq n$ existuje právě jeden list stromu T tvaru $u_i = [a_i, \{i\}]$.

b) T má kořen tvaru $[A_r, \{1, ..., n\}]$, kde $A_r \in K_t$.

c) Libovolný složkový uzel u má tvar $[A_s, C_s]$, kde $A_s \in K_t$ a $C_s \subseteq \{1, ..., n\}$.

d) Nechť $u = [A_u, C_u]$, $v = [A_v, C_v]$ jsou různé uzly stromu T. Platí, že buď C_v je vlastní podmnožinou C_u, nebo C_u je vlastní podmnožinou C_v, nebo C_u a C_v jsou disjunktní, nebo $C_v = C_u$. V posledním případě je jeden z uzlů u, v listem a C_v, C_u jsou jednoprvkové množiny. Pokud C_u je vlastní podmnožinou C_v, tak z v vede cesta do u směrem od kořene k listu. Pokud v je listem, tak do něho vede cesta ze všech uzlů $y = [A_y, C_y]$ takových, že $C_v \subseteq C_y$.

e) Nechť $u = [A, C]$, $v_1 = [A_1, C_1]$, $v_2 = [A_2, C_2]$,..., $v_k = [A_k, C_k]$ jsou uzly T, kde u je matkou pro $v_1, ..., v_k$ a žádné další dcery nemá. Pak $C = C_1 \cup C_2 \cup ... C_k$.

Nechť $u = [A_s, C_s]$ je uzel stromu T. A_s nazýváme symbolem uzlu u, C_s (bezprostřední) složkou uzlu u a značíme ji $Cov(u, T)$ (připomíná pokrytí z předchozí sekce).

Nechť $w = a_1 \dots a_n$ je věta, T je C-stromem nad w a $U = \{u_1, ..., u_m\}$ je množina jeho složkových uzlů. Pak říkáme, že množina složek $\{Cov(u_i, T); 1 \leq i \leq m\}$ tvoří systém (bezprostředních) složek nad w. Zcela stejným způsobem jako pro D-stromy lze pro C-stromy zavést pojmy stupně uzlu a stupně C-stromu.

Mezi uzly C-stromu s disjuktními složkami se obvykle uvažuje uspořádání, budeme mu říkat *přirozené uspořádání*, dané následujícím uspořádáním jejich složek: řekneme, že *složka P_1 předchází složce P_2* pokud maximum složky P_1 je menší než maximum složky P_2.

Příklad 12 *Příklad složkového stromu T_s podle obr.5 nad větou „Malý chlapec přinesl zprávu". Složkový strom je (podle definice) charakterizován svými uzly, takže abychom charakterizovali T_s, uvedeme úplnou množinu jeho uzlů.*

Listy T_s mají následující tvar:
[Malý,{1}], [chlapec,{2}], [přinesl,{3}], [zprávu,{4}].
Kořen T_s:
[S,{1,2,3,4}].
Vnitřní uzly T_s:
[Adj,{1}], [N,{2}], [V,{3}], [N,{4}], [NP,{1,2}], [VP,{3,4}].

Je užitečné si uvědomit, že v lingvistice hraje prvotní úlohu spíše systém složek, než složkový strom. Složkový strom je datový typ, který nás upozorňuje na fakt, že systémy složek mají (definitoricky) hierarchický charakter, a že složky mívají svá označení, která charakterizují jejich lexikální obsazení.

V následující definici zavádíme pojem díry podobně jako v předchozí sekci a pomocí něj zavádíme pojem (ne)souvislosti C-stromu, který je analogií (ne)projektivity u D-stromů.

Definice 13 (Díra ve složce)
Nechť T je C-strom (systém složek) nad větou $w = a_1 \dots a_n$, nechť $Cov(u, T)$ je jeho složkou. Nechť

$$Cov(u, T) = \{i_1, i_2, \dots, i_m\}, i_1 < i_2 < \dots < i_{m-1} < i_m.$$

Řekneme, že dvojice (i_j, i_{j+1}) tvoří *díru ve složce $Cov(u, T)$*, jestliže $1 \leq j < m$, a zároveň $i_{j+1} - i_j > 1$.

Budeme říkat, že T je souvislý, pokud žádná jeho složka neobsahuje díru. V opačném případě říkáme, že T není souvislý.

Poznámka 14 Lingvisté, pokud hovoří o složkových stromech, mají nejčastěji na mysli souvislé C-stromy, či ještě častěji derivační stromy bezkontextových gramatik. Abychom zvýraznili fakt, že i v jazycích s pevným slovosledem jako je angličtina, je třeba brát v úvahu nesouvislé složky, uvažujeme složkové stromy s nesouvislými složkami. Ostatně v lingvistickém folklóru se o nesouvislých složkách běžně hovoří. Upozorňujeme tak na skutečnost, že ani ve složkové syntaxi přirozených jazyků nevystačíme s derivačními (frázovými) stromy (bez)kontextových gramatik jako s univerzálním datovým typem.

Pozornému čtenáři jistě neujde podobnost mezi systémem bezprostředních složek C-stromu a množinou pokrytí uzlů D-stromu. Skutečně, pokud opomeneme podmínku e), splňuje množina pokrytí uzlů D-stromů všechny ostatní požadavky pro systém složek nad příslušnou větou.

Ještě zavedeme (pro potřeby následujících kapitol) pojem C-podstromu nad větou w. Tento pojem umožňuje zachytit strukturu jednotlivých složek a je užitečný pro formulaci invariantů algoritmů syntaktické analýzy.

Definice 15 (C-podstrom)
Nechť $w = a_1 \dots a_n$ je věta, kde $a_1, \dots a_n \in A^+$. Nechť existuje C-strom T_1 nad větou w a T_2 je jeho podstromem s následujícími vlastnostmi:

Každý uzel T_2 je uzlem T_1 a T_2 přebírá ještě tyto vlastnosti T_1:

a) Libovolný list stromu T_2 je dvojice tvaru $u = [a_{i_u}, \{i_u\}]$, kde $1 \leq i_u \leq n$.

d) Nechť $u = [A_u, C_u]$, $v = [A_v, C_v]$ jsou libovolné uzly stromu T_2. Platí, že buď C_v je vlastní podmnožinou C_u, nebo C_u je vlastní podmnožinou C_v, nebo C_u a C_v jsou disjunktní, nebo $C_v = C_u$. V posledním případě je jeden z uzlů u, v listem. Pokud je C_u vlastní podmnožinou C_v, vede z v cesta do u směrem od kořene k listu. Pokud je v listem, vede do něj cesta ze všech uzlů $y = [A_y, C_y]$ takových, že $C_v \subseteq C_y$.

e) Nechť $u = [A, C]$, $v_1 = [A_1, C_1]$, $v_2 = [A_2, C_2]$,..., $v_k = [A_k, C_k]$ jsou uzly T, kde u je matkou pro $v_1, ..., v_k$ a žádné další dcery nemá. Pak $C = C_1 \cup C_2 \cup ... C_k$.

O T_2 budeme říkat, že je C-podstromem nad w. Pokud T obsahuje pouze uzly, které mají souvislá pokrytí (bez díry), říkáme, že T_2 je souvislý.

2.3 DR-stromy

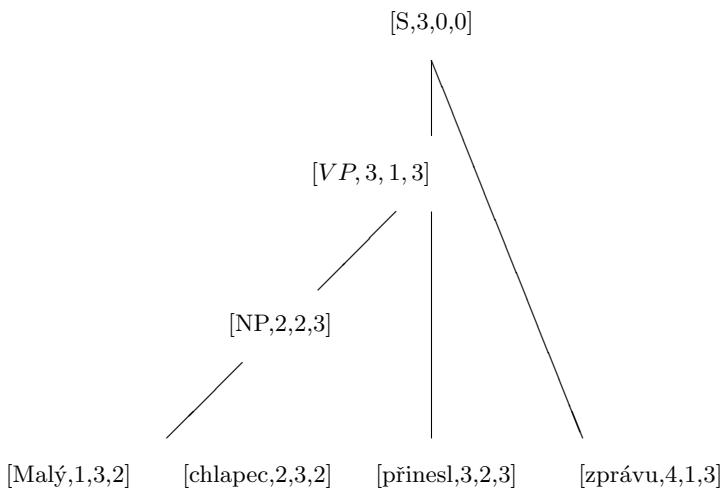
V této sekci zavedeme pojem DR-stromu. Je to datový typ, který je blízký modelování pomocí HPSG ([Pollard, Sag, 1994]). DR-strom zavádíme tak, aby zachycoval nejen závislosti analyzované věty, ale i pořadí (historii) u těch závislostí, které vedou v závislostním stromě do jednoho uzlu. Tedy přidává částečné uspořádání hranám a zprostředkovaně i uzlům se společnou matkou. Druhým cílem je zavést DR-stromy tak, aby plně zachycovaly informaci, kterou nesou složkové stromy.

Sám název DR-stromu vychází ze slov *delete – vypustit* a *rewrite – přepsat* a odráží představu postupné konstrukce závislostního stromu nad danou větou pomocí automatu se dvěma hlavami.

Představujeme si, že věta je uložena na pásce (či spíše lineárním seznamu) a automat má dvě hlavy, které se mohou po této pásce pohybovat. První hlava je *vypouštěcí* — ta dokáže z pásky/věty úplně vypustit políčko/slovo, na kterém právě stojí. Druhá hlava je přepisovací — tato hlava může změnit obsah navštíveného políčka. Automat pracuje v krocích. V každém kroku rozmístí hlavy na pásce tak, aby stály na slovech, mezi nimiž má být vyznačena závislost — vypouštěcí hlava na závislém slově a přepisovací hlava na příslušném slově řídícím. Krok práce automatu se provede tím, že automat vypustí slovo pod vypouštěcí hlavou, a případně přepíše slovo pod přepisovací hlavou.

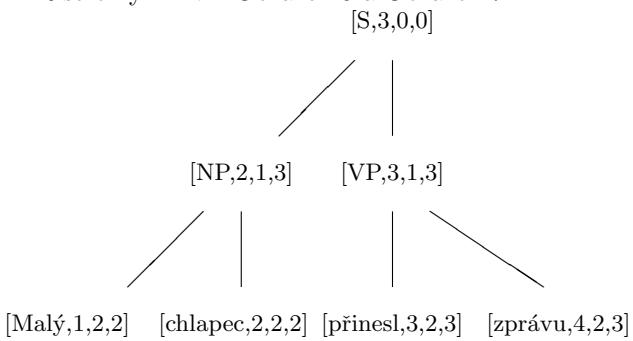
V DR-stromu přepsání znázorňuje svislá hrana (přepsané slovo zůstává na svém místě) a vypuštění odpovídá šikmá hrana, shodná s šikmou hranou v odpovídajícím závislostním stromě.

Připustíme i druhou variantu kroku spočívající pouze v přepsání symbolu pod přepisovací hranou, aniž by nějaký symbol byl vypuštěn. Takovým případům odpovídají v DR-stromu uzly s jedinou dcerou. Tato dcera je k uzlu připojena svislou, přepisovací hranou.



Obrázek 6. DR-strom strom T_{11} nad větou ,,Malý chlapec přinesl zprávu"

Různým pořadím vypouštění ovšem můžeme pro stejné závislosti získat různé DR-stromy. — viz Obrázek 6 a Obrázek 7.



Obrázek 7. DR-strom strom T_{21} nad větou ,,Malý chlapec přinesl zprávu"

Nyní přistupme k formální definici DR-stromu.

Definice 16 (DR-strom)
Nechť $w = a_1 \dots a_n$ je věta a A_t je konečná množina symbolů (slov), kde $a_1, \dots a_n \in A_t$. Nechť K_t je také konečná množina symbolů. Řekneme, že

strom T je *DR-strom nad větou w (s terminály z A_t a kategoriemi z K_t)*, jestliže libovolný uzel stromu T je čtveřice tvaru $u = [A_u, i_u, v_u, d_u]$, kde

- $A_u \in (A_t \cup K_t)$; A_u nazýváme *symbol* uzlu.
- $i_u \in \{1 \dots n\}$; i_u nazýváme *horizontální index* uzlu, udává horizontální pozici uzlu ve větě.
- pro každé $i \in \{1 \dots n\}$ existuje právě jeden list $u = [A_u, i_u, v_u, d_u]$ stromu T takový, že $i_u = i$ a $A_u = a_i$;
- $d_u \in \{0 \dots n\}$;
  d_u nazýváme *dominační index* uzlu u, udává horizontální index bezprostředně nadřízeného uzlu nebo v případě $d_u = 0$ absenci nadřízeného uzlu.
- $v_u \in \{0 \dots n\}$;
  $v_u = 0$ právě když $d_u = 0$, v tom případě je uzel $[A_u, i_u, 0, 0]$ kořenem stromu T. v_u nazýváme *vertikální index* uzlu, udává vzdálenost uzlu od kořene stromu měřenou v počtu hran.
- Je-li $u = [A_u, i_u, v_u, d_u]$ a $d_u \neq 0$, potom existuje právě jeden uzel $v = [A_v, d_u, v_u - 1, d_v]$; dvojice (u, v) tvoří hranu stromu T, orientovanou od listu ke kořenu.
  Je-li $d_u = i_u$, hovoříme o hraně *svislé (přepisovací)*.
  Je-li $d_u \neq i_u$, hovoříme o hraně *šikmé (vypouštěcí)*. V tomto případě existuje také (pro nějaký symbol A_{uu}) uzel tvaru $uu = [A_{uu}, d_u, v_u, d_u]$ stromu T (vede-li do uzlu v šikmá hrana, vede do něj právě jedna svislá hrana).
  Je-li $d_u > i_u$, hovoříme o R-hraně, je-li $d_u < i_u$, hovoříme o L-hraně.
- je-li $u = [A_u, i_u, v_u, d_u]$ uzel, potom existuje nejvýše jeden uzel $v = [A_v, i_v, v_u + 1, i_u]$, takový, že $i_v \neq i_u$ (do každého uzlu vede nejvýše jedna šikmá hrana).
- jsou-li $u = [A_u, i, v_u, d_u]$ a $v = [A_v, i, v_v, d_v]$ uzly stromu T (se stejným horizontálním indexem), potom platí
  - $v_u = v_v \Rightarrow u = v$
    hodnota horizontálního indexu spolu s hodnotou vertikálního indexu jednoznačně určují uzel stromu T.
  - $d_u \neq i \Rightarrow v_v \geq v_u$, $d_v \neq i \Rightarrow v_v \leq v_u$
    pro každé i existuje nejvýše jeden uzel s horizontálním indexem i, z něhož vede šikmá hrana. Je to uzel, který má ze všech uzlů s horizontálním indexem i nejmenší hodnotu vertikálního indexu (nejmenší vzdálenost od kořene).
  - $v_v > v_u \Rightarrow$ existuje uzel $v' = [A_{v'}, i, v_v - 1, i]$

Poznámka 17 V případech, kdy nám nezáleží na množinách A_t, K_t, budeme říkat pouze *DR-strom nad větou w*.

Můžeme si povšimnout, že hrany DR-stromů, na rozdíl od D-stromů, nenesou informaci o funktorech. Tato informace bývá (v počítačových ligvistických aplikacích) obsažena v symbolech uzlů určujících šikmou hranu. Toto pozorování formalizujeme v následující definici, která popisuje jeden z obvyklých způsobů, jak z DR-stromů získat D-stromy. Tento způsob spočívá v kontrakci libovolné svislé cesty do jediného uzlu a ve vybraném způsobu výpočtu funktorů.

Definice 18 (D-kontrakce DR-stromu) Nechť $w = a_1 \dots a_n$ je věta, T je D-strom nad větou w s množinou terminálů A_t a funktorů R , TR je DR-strom s množinou terminálů A_t a kategorií K_t nad větou $w \in {A_t}^*$, a f_c je funkcí z $(A_t \cup K_t)^2$ do $R \cup \{\lambda\}$; f_c nazýváme D-kontrakční funkcí.

Řekneme, že D-strom T *je D-kontrakcí DR-stromu* TR podle f_c, platí-li, že pro každý $u = [a_i, i, v_i, d_i, R_i]$ uzel stromu T existuje uzel uu stromu TR tvaru $uu = [A, i, v_{uu}, d_i]$ a list uv stromu TR tvaru $uv = [a_i, i, v_{uv}, d_v]$. Pokud u není kořenem, tak existuje šikmá hrana z uu do uzlu uu_1 (který má horizontální index d_i a symbol B) taková, že $f_c(A, B) = R_i$. Připomeňme si, že pokud je u kořenem, tak definitoricky $R_i = \lambda$.

Uzly uv a uu nemusí být v obecném případě různé.

Skutečnost, že T je D-kontrakcí TR podle f_c, budeme značit $T = Dkon(TR, f_c)$.

Skutečnost, že T je D-kontrakcí TR podle takové kontrakční funkce f_λ, která každému argumentu přiřazuje λ (prázdné slovo) budeme značit $T = Dkon(TR)$. Práce studující formální závislostní gramatiky často funktory zanedbávají. Této praxe se také v příštích sekcích přidržíme přesto, že určování a studium funktorů je prvotním úkolem závislostních syntaxí z pohledu lingvisty.

Z definice D-stromu plyne, že dominační index žádného uzlu se nerovná jeho horizontálnímu indexu, tedy že na libovolné horizontální pozici je nejvýše jeden uzel D-stromu. Povšimněme si následujícího tvrzení, které přímo vyplývá z definice kontrakce.

Tvrzení 19 *Je-li D-strom T kontrakcí DR-stromu TR, existuje pro každý $u = [a_i, i, v_i, d_i, R_i]$ uzel stromu T* právě jeden *uzel uu stromu TR shodující se s u v horizontálním a dominačním indexu.*

Nyní definici pojmů *pokrytí* a *díra v pokrytí* přeneseme z D-stromů na DR-stromy. Tyto pojmy se kromě typu stromů, na kterých jsou zavedeny, nebudou nijak lišit od pojmů definovaných na závislostních stromech.

Definice 20 (Pokrytí uzlu DR-stromu)
Nechť TR je DR-strom a nechť u je uzel stromu TR. $Cov(u, TR)$ označíme množinu horizontálních indexů všech uzlů stromu TR, ze kterých vede cesta do u. Uvažujeme i prázdnou cestu a tedy $Cov(u, TR)$ vždy obsahuje také horizontální index uzlu u.

Řekneme, že $Cov(u, TR)$ je *pokrytí uzlu u (podle stromu TR).*

Definice 21 (Díra (v pokrytí) DR-stromu)
Nechť T je DR- strom nad větou $w = a_1 \dots a_n$, nechť u je uzel stromu T a nechť

$$Cov(u, T) = \{i_1, i_2, \dots, i_m\}, i_1 < i_2 < \dots < i_{m-1} < i_m.$$

Řekneme, že dvojice (i_j, i_{j+1}) tvoří *díru v $Cov(u, T)$*, jestliže $1 \leq j < m$, a zároveň $i_{j+1} - i_j > 1$.

Budeme říkat, že T je DR-projektivní, pokud žádný z jeho uzlů nemá v pokrytí díru. V opačném případě říkáme, že T není DR-projektivní.

Předchozí část této sekce navazovala na sekci o D-stromech. Následujícím tvrzením navážeme na sekci o složkových stromech. Toto tvrzení říká, že DR-strom lze použít jako reprezentanta jistého systému složek.

Tvrzení 22 *Nechť $w = a_1...a_n$ a TR je DR-strom nad w s terminální abecedou A_t a kategoriální abecedou K_t, Nechť symbol kořene stromu TR a všech jeho vnitřních uzlů je z K_t a nechť uzly, do kterých vede jen jedna hrana, jsou uzly, do kterých vede hrana z listů. Nechť $u_1, ..., u_k$ je množina uzlů stromu TR a A_i je symbolem uzlu u_i pro $1 \leq i \leq k$. Pak množina dvojic tvaru $[A_i, Cov(u_i, TR)]$, kde $1 \leq i \leq k$, tvoří množinu uzlů C-stromu T_s nad w, který je grafově isomorfní se stromem TRr, který vznikne z TR obrácením orientace jeho hran.*

Důkaz:

Isomorfismus je dán (prostou) funkcí $f : f(u_i) = [A_i, Cov(u_i, TR)]$ pro $1 \leq i \leq k$.

Není těžké ověřit, že uzly sestrojené podle tvrzení, a hrany sestrojené podle definice složkového stromu dávají (C-)strom T_s f-isomorfní se stromem TRr. Q.E.D.

Poznámka 23 Předchozí tvrzení ilustrují fakt, že DR-strom nese informaci o jistém D-stromě a fakticky také o jistém systému složek.

Není obtížné si představit typ kontrakce, která by pouze nějakým způsobem zkracovala svislé cesty v DR-stromu. Získali bychom tím analogie DR-stromů s uzly neomezeného stupně. Účelem tohoto kroku je formulovat prostředek pro vyjadřování systémů složek s neomezeným počtem rovnocenných složek.

3 Gramatiky a syntaktická analýza

V této kapitole se budeme zabývat formálními gramatikami a syntaktickými analýzami podle jednotlivých typů formálních gramatik. Gramatiky jsou pro nás v první řadě prostředkem, který dovoluje zavádět syntaktické analýzy, většinou nekonečné množiny syntaktických struktur. Náš výklad rozšiřuje výklad ze základních učebnic formálních gramatik a jazyků. Tam bývá formální gramatika pojímána hlavně jako prostředek pro modelování formálních jazyků, tj. množin vět a strukturální povaha stavby vět bývá vykládána zjednodušeně.

Postupně se budeme věnovat třem typům gramatik: kategoriálním gramatikám, bezkontextovým gramatikám, a zobecněným závislostním gramatikám. Dodržíme tak chronologii vzniku těchto pojmů. Kategoriální gramatiky budeme prezentovat jako (historicky první moderní) formální prostředek modelující syntax. Bezkontextové a závislostní gramatiky budeme interpretovat jako gramatiky analytické (gramatiky spojené s představou postupného rozboru daných vět), stejně jako kategoriální gramatiky, pro které je tento způsob zavedení typický. Snadněji tak uvedeme studované typy gramatik do souvislosti s odpovídajícími typy syntaktických analýz. Hlouběji se problematikou závislostních analýz zabývají práce [Holan, 2001], [Plátek at al., 2002]. V závěru této kapitoly ukážeme, že kategoriální gramatiky lze velmi přirozeně používat pro zadávání

DR- a D-analýz. V souladu s klasickou literaturou [Gladkij, 1973] zavádíme pomocí kategoriálních gramatik pouze projektivní analýzy.

3.1 Kategoriální gramatiky

Tato kapitola vychází z kapitol o kategoriálních gramatikách knih [Partee at al., 1993] a [Gladkij, 1973]. První kniha preferuje aplikaci kategoriálních gramatik pro složkovou syntaxi. My, inspirováni druhou knihou, ukazujeme, jak souvisí kategoriální gramatiky se závislostní syntaxí.

Kategoriální gramatiky (KG) byly zavedeny v práci [Ajdukiewicz, 1935] o sémantických kategoriích logických jazyků. Bar-Hillel v knize [Bar-Hillel, 1964] představil kategoriální gramatiky jako prostředek pro popis syntaxe přirozených jazyků. My představujeme kategoriální gramatiky podobně, tedy jako formální prostředek, který preferuje metodu tzv. lexikalizace, tj. přiřazování informací o větné skladbě k jednotlivým slovním výskytům.

Kategoriální gramatiky mají ve své klasické podobě možnost produkovat D-stromy, avšak pouze se stupněm omezeným konstantou. Nedovolují tedy úplně přímočaře modelovat podřízenost v přirozených jazycích, neboť např. počet podřízených volných doplnění sloves i jmen není omezen pevně danou hranicí ani v češtině ani v dalších jazycích.

Základní idea kategoriálních gramatik spočívá v tom, že lexikálním položkám jsou přiřazovány nejen základní lexikální kategorie, ale i složené kategorie, které zachycují také omezený počet (podle stupně složitosti kategorií) syntaktických valencí. Řetězy kategorií mohou být redukovány pomocí redukčních pravidel ve formě $B(B\backslash A) \Rightarrow A$, $(A/B)B \Rightarrow A$. Vezměme si příklad české věty a zjednodušenou kategoriální gramatiku, která popisuje její strukturu. Věta má následující tvar:

Jana pracuje velmi rychle.

Jménu *Jana* přiřadíme jednoduchou kategorii Sb (Subjekt). Kategorie slovesa *pracuje* ($(Sb\backslash Vb)/Mv$) odpovídá zjednodušenému valenčnímu rámci tohoto slovesa, t.j. sloveso je rozvíjeno subjektem a jedním příslovečným určením. Použijeme-li obvyklou míru složitosti kategorií, totiž počet levých i pravých lomítek obsažených v kategorii, vidíme, že tato kategorie má složitost 2. Kategorie slova *velmi* je jednoduchá: Mad (modifikuje, rozvíjí příslovečné určení). Kategorie slova *rychle* má stupeň 1: ($Mad\backslash Mv$) (modifikuje sloveso, je rozvíjeno kvantifikujícím příslovcem).

Podívejme se, jak jsou výše popsaná věta a příslušný řetěz kategorií postupně redukovány:

$$\begin{pmatrix} Sb & ((Sb\backslash Vb)/Mv) & Mad & (Mad\backslash Mv) \\ Jana & pracuje & velmi & rychle \end{pmatrix}$$

$\Rightarrow$

$$\begin{pmatrix} Sb & ((Sb\backslash Vb)/Mv) & Mv \\ Jana & pracuje & rychle \end{pmatrix}$$

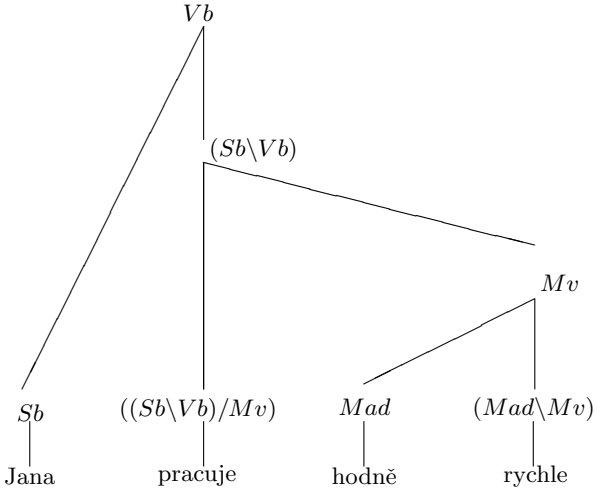
$\Rightarrow$

$$\begin{pmatrix} Sb & (Sb\backslash Vb) \\ Jana & pracuje \end{pmatrix}$$

$\Rightarrow$

$$\begin{pmatrix} Vb \\ pracuje \end{pmatrix}$$

Redukování podle kategoriální gramatiky odpovídá představě postupného vypouštění a přepisování symbolů zařízením se dvěma hlavami. Historii postupné redukce dané věty podle dané KG lze zachycovat DR-stromem. Výše uvedené redukci odpovídá obrázek 8, který tuto skutečnost ilustruje.



Obrázek 8. DR-strom k redukci podle KG

Podáme nyní standardní formální definici kategoriálních gramatik.

Definice 24 Nechť V_A je konečná množina symbolů. Říkáme jim kategoriální symboly. Množina C, tj. množina složených kategoriálních symbolů (kategorií) je dána následovně:

(1) Pro všechna $x \in V_A, x \in C$.
(2) Pokud x a y jsou z C, pak i (x/y) a $(x\backslash y)$ jsou z C.
(3) Nic dalšího do C nepatří.

Kategoriální gramatika je čtveřice $G = (V_T, V_A, S, F)$, kde
V_T je konečná množina terminálních symbolů,
V_A je konečná množina kategoriálních symbolů,
$S \subset V_A$,
F je funkce z V_T do $P_{fin}(C)$: tj. je to funkce přiřazující každému terminálnímu symbolu konečnou množinu kategorií z C.

Dále zavedeme binární relaci $\Rightarrow$, tj. elementární redukci na párech řetězů nad abecedou C:

Pro libovolné kategorie $X, Y \in C$ a libovolné konečné řetězy kategorií α, β lze provádět pouze následující dva typy redukcí:

$\alpha\ (X/Y)\ Y\ \beta \Rightarrow \alpha\ X\ \beta$,

$\alpha\ Y\ (Y\backslash X)\ \beta \Rightarrow \alpha\ X\ \beta$.

Relace $\Rightarrow^*$ (redukce žádným nebo více kroky) je reflexivní a transitivní uzávěr relace $\Rightarrow$.

Budeme říkat, že řetěz $w \in {V_T}^*$ je rozpoznáván pomocí G právě, když pro $w = w_1...w_n$, $w_1, ..., w_n \in V_T$ existuje řetěz kategorií $C_1...C_n$ takový, že $C_i \in F(w_i)$ pro všechna $1 \leq i \leq n$, a $C_1...C_n \Rightarrow^* Cs$, kde $Cs \in S$.

Jinými slovy, $w = w_1...w_n$ je rozpoznáván pomocí G, právě když existuje pomocí F nějaký jednoznačný výběr kategorií k symbolům řetězu w takový, že takto vybraný řetěz kategorií lze postupně redukovat na nějaký symbol z S. Budeme psát $L(G) = \{w|w$ je rozpoznáván pomocí $G\}$.

Poznámka 25 V této kapitole nebudeme formálně přiřazovat DR- a D-analýzy kategoriálním gramatikám. S tímto krokem posečkáme, abychom mohli využít aparátu spojeného se závislostními gramatikami (D-gramatikami).

3.2 Bezkontextové gramatiky a rozpoznávání

V této sekci zavádíme bezkontextové gramatiky (CF-gramatiky) a uvádíme je do souvislosti s C-stromy a tím i se systémy bezprostředních složek. Snažíme se pojem bezkontextové gramatiky a pojem (složkové) syntaktické analýzy uvést do vzájemného vztahu zavedením složkových analýz bezkontextových gramatik. Omezíme se na bezkontextové gramatiky ve formě, která odpovídá pojmu C-strom.

Definice 26 (CF-gramatika)
CF-gramatika (složková bezkontextová gramatika) je čtveřice

$$G = (V_T, V_N, S, P)$$

kde V_T je konečná množina terminálů, V_N je konečná množina neterminálů, $S \in V_N$ je startovací (přijímací) symbol a P je množina přepisovacích pravidel dvou tvarů:

a) $A \rightarrow \alpha$, kde $\alpha \in V_N^+$, $|\alpha| > 1$,
b) $A \rightarrow a$, kde $a \in V_T$.

Pokud pro pravidla typu a) platí omezení, že $\alpha \leq 2$ říkáme, že G je v Chomského normální formě.

Pravidla tvaru $A \rightarrow \alpha$ mohou být použita k *redukci* libovolného řetězce w aplikací na jakýkoliv výskyt řetězce α ve w. Řetězec α se použitím pravidla $A \rightarrow \alpha$ přepíše na A, tedy z řetězce $w = w_1\alpha w_2$ redukcí podle pravidla $A \rightarrow \alpha$ vznikne řetězec $w_1 A\ w_2$.

Poznamenejme, že v učebnicích se častěji v souvislosti s CF-gramatikami zavádí relace odvození (derivace), která je relací obrácenou k redukci. Relace odvození však navozuje představu generování namísto syntatické analýzy. Zde hovoříme o redukci, abychom zdůraznili skutečnost, že se budeme zabývat syntaktickou analýzou.

Následující definice zavádí standardním způsobem rozpoznávání věty a jazyka CF-gramatikou. Později uvedeme (ekvivalentní) definice pomocí složkových stromů, které budou z našeho hlediska výstižnější.

Definice 27 (CF-gramatika rozpoznává větu)
Řekneme, že CF-gramatika

$$G = (V_T, V_N, S, P)$$

rozpoznává větu $w \in V_T{}^*$, lze-li větu w opakovanou redukcí podle pravidel gramatiky G přepsat na řetěz sestávající z jediného (přijímacího) symbolu S.

Definice 28 (CF-gramatika rozpoznává jazyk)
Řekneme, že CF-gramatika G rozpoznává jazyk L, obsahuje-li L právě všechny věty rozpoznávané gramatikou G. Jazyk rozpoznávaný gramatikou G budeme značit $L(G)$.

3.3 CYK-algoritmus pro problém rozpoznávání

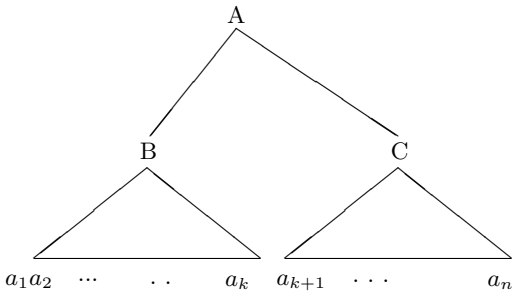
Uvažujme následující úlohu: Je dána CF-gramatika $G = (V_T, V_N, S, P)$, věta $x \in V_T^+$ a máme algoritmicky rozhodnout, zda $x \in L(G)$ (problém rozpoznávání pro bezkontextové jazyky).

Představíme efektivní algoritmus pro rozpoznávání bezkontextových jazyků, označovaný jako CYK. Omezujícím předpokladem tohoto algoritmu je, že jazyk $L(G)$ je zadán pomocí gramatiky G v Chomského normální formě. Pojmenování tohoto algoritmu je odvozeno z počátečních písmen jeho autorů: Cockeho, Youngera a Kasamiho.

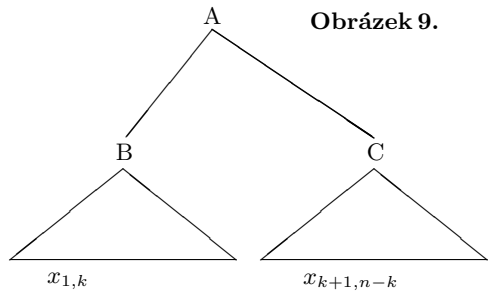
Nejprve přiblížíme základní myšlenku CYK-algoritmu. Pokud (pod)řetěz x, kde $x = a$ je délky 1, lze redukovat, pak pouze přímo pravidlem tvaru $A \rightarrow a$. Platí-li ale, že lze postupně redukovat $x = a_1 a_2 ... a_n$, kde $n \geq 2$ na symbol A, pak poslední pravidlo užité k redukci musí mít tvar $A \rightarrow BC$. Nahlédneme, že do B lze redukovat jistý prefix x a do C jistý sufix x. Musí tedy existovat k takové, že $1 \leq k < n$ a prefix $a_1 ... a_k$ lze zredukovat do B. Podobně sufix $a_{k+1} ... a_n$ lze zredukovat do C (viz obr. 9).

Vidíme, že problém redukce pro x délky n lze rozložit na dvojice podobných problémů pro věty délky k a $n - k$ pro vhodná k. Máme-li najít všechny možné redukce řetězu x, budeme potřebovat rozklady na dvě části pro všechna možná k mezi 1 a n. Informace o redukovatelnosti budeme ukládat do tabulky tak, abychom je mohli jednoduše skládat.

Budeme využívat následující notace: Pro větu x označíme jako $x_{i,j}$ podřetězec věty x začínající na pozici i, o délce j (tedy končící na pozici $i + j - 1$). Pomocí této notace si upravíme obrázek 9 na obr. 10.



Obrázek 9.



Obrázek 10.

Následující algoritmus pracuje s tabulkou (maticí) `TL[1..n,1..n]`, přičemž využívá pouze horní trojúhelníkovou matici (tj. vedlejší diagonálu a to co je nad ní). Pro $j = 1, ..., n$ a pro $i = 1, ..., n+1-j$ bude (postupně) do `TL[i,j]` ukládat neterminály, na které je možno redukovat podřetěz $x_{i,j}$.

Vstupní slovo $x = a_1...a_n$ bude z $L(G)$, pokud následující algoritmus vytvoří tabulku, kde $S \in$ `TL[1,n]`.

CYK - algoritmus

Vstup : $x = a_1...a_n$, CF-gramatika $G = (V_T, V_N, S, P)$, kde $x \in V_T^+$.

```
FOR i := 1 TO n DO (* j = 1 *)
   TL[i,1] := {A |A → a_i ∈ P} END;
FOR j := 2 TO n DO (* j > 1 *)
   FOR i := 1 TO n+1-j DO
      TL[i,j] := ∅;
      FOR k := 1 TO j-1 DO
         TL[i,j] := TL[i,j] ∪ {A |A → BC ∈ P,
         B ∈ TL[i,k],C ∈ TL[i+k,j-k]}
      END;
   END;
END;
IF S ∈ TL[1,n] THEN
   WriteString('x patří do L(G)')
```

```
ELSE
   WriteString('x nepatří do L(G)')
END
```

Není těžké nahlédnout, že tento algoritmus pro pevně danou gramatiku vykonává řádově n^3 kroků a používá řádově n^2 paměťových míst.

Příklad 29 Jazyk $L = \{a^n b^n c^m | n, m \geq 1\}$ je bezkontextový. Je rozpoznáván gramatikou G_1 s pravidly

$$\begin{aligned} S &\rightarrow AB \\ A &\rightarrow CD|CAD \\ C &\rightarrow a \\ D &\rightarrow b \\ B &\rightarrow c|EB \\ E &\rightarrow c \end{aligned}$$

G_1 lze převést do Chomského normální formy na G_2 následovně:

$$\begin{aligned} S &\rightarrow AB \\ A &\rightarrow CD|CF \\ B &\rightarrow c|EB \\ C &\rightarrow a \\ D &\rightarrow b \\ E &\rightarrow c \\ F &\rightarrow AD \end{aligned}$$

Vezměme $x = aaabbbcc$. CYK-algoritmus vytvoří tabulku na obr. 11. Protože nejspodnější políčko tabulky obsahuje S, x patří do rozpoznávaného jazyka.

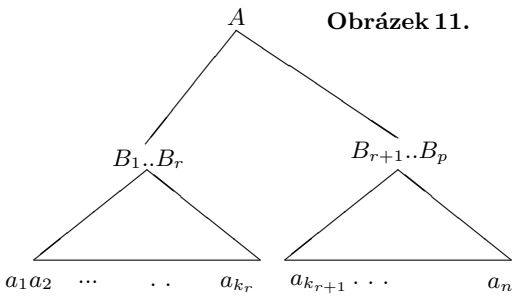
3.4 Zobecněný CYK algoritmus pro rozpoznávání

Zobecněný CYK (ZCYK) je schopen pracovat i s CF-gramatikami, které nejsou v Chomského normální formě. Je založen na podobných pozorováních jako CYK.

Pokud (pod)řetěz x, který je délky 1, lze redukovat, pak pouze přímo pravidlem tvaru $A \rightarrow a$, kde $x = a$. Platí-li ale, že lze postupně redukovat $x = a_1 a_2 ... a_n$, kde $n \geq 2$ na symbol A, pak poslední užité pravidlo k redukci musí mít tvar $A \rightarrow B_1 ... B_p$, kde $p > 1$. Nahlédneme, že pro každé $1 < r \leq p$ do řetězu neterminálů $B_1 ... B_r$ lze redukovat jistý prefix x a do řetězu neterminálů $B_{r+1} ... B_k$ zbylý sufix x. Musí tedy pro každé $1 < r \leq p$ existovat k_r takové, že $1 \leq k_r < n$, prefix $a_1 ... a_{k_r}$ lze zredukovat do $B_1 ... B_r$ a sufix $a_{k_{r+1}} ... a_n$ lze zredukovat do $B_{r+1} ... B_k$ (viz obr. 12). Takové rozdělení pravé strany pravidla

a	a	a	b	b	b	c	c
C	C	C	D	D	D	E,B	E,B
		A				B	
		F					
	A						
	F						
A							
S							
S							

Obrázek 11.



Obrázek 12.

na dvě části budeme označovat tečkou. Tato notace je používaná u řady dalších algoritmů.

Vidíme, že úlohu redukce (pod)řetězce x délky n do řetězu $B_1..B_p$ lze rozdělit na dvojice úloh:

(a) najít prefix x o vhodné délce k_{p-1}, který lze zredukovat do řetězu $B_1..B_{p-1}$.
(b) potvrdit, že suffix x o délce $|x| - k_{p-1}$ lze zredukovat do symbolu B_p.

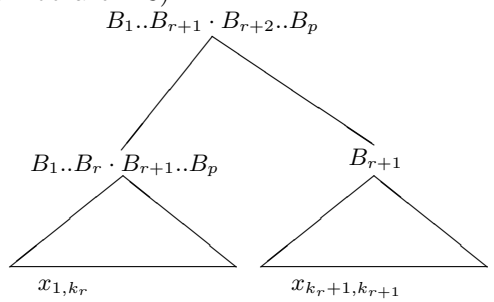
Dále vidíme, že problém (a) lze opět rozdělit na dvojice problémů:

(aa) najít prefix $x_{1,k_{p-1}}$ o vhodné délce k_{p-2}, který lze zredukovat do řetězu $B_1..B_{p-2}$.
(bb) potvrdit, že suffix $x_{1,k_{p-1}}$ o délce $|x_{1,k_{p-1}}| - k_{p-2}$ lze zredukovat do symbolu B_p.

Abychom ověřili, zda lze x zredukovat na řetěz $B_1..B_p$, lze postupně hledat prefixy x, které lze redukovat do řetězů $B_1..B_r$ pro r, které roste od 1 do p.

Pokud se potvrdí, že lze x zredukovat na řetěz $B_1..B_p$, který tvoří pravou stranu pravidla $A \to B_1..B_p$, tak se také potvrdí, že x lze zredukovat do A.

Budeme využívat následující notace: Nechť $B_1..B_p$ je pravá strana nějakého pravidla, $r \geq 1$. Pomocí (tečkové) notace $B_1..B_r \cdot B_{r+1}..B_p$ budeme označovat skutečnost, že do prefixu $B_1..B_r$ lze zredukovat nějaký souvislý úsek vstupního slova (viz obrázek 13).



Obrázek 13.

ZCYK pracuje se dvěma maticemi `TL[1..n,1..n]` a `TP[1..n,1..n]`, přičemž také využívá pouze horní trojúhelníkovou část matice. Stejně jako algoritmus CYK bude pro $j = 1, ..., n$ a pro $i = 1, ..., n + 1 - j$ do `TL[i,j]` ukládat neterminály, na ktere je možno redukovat podřetěz $x_{i,j}$.

Pro $j = 1, ..., n$ a pro $i = 1, ..., n + 1 - j$ bude ZCYK do `TP[i,j]` ukládat řetězy tvaru $B_1..B_r \cdot B_{r+1}..B_p$, kde $B_1..B_rB_{r+1}..B_p$ je pravou stranou nějakého pravidla gramatiky a na řetěz $B_1..B_r$ je možno zredukovat řetěz $x_{i,j}$.

Vstupní slovo $x = a_1...a_n$ je z $L(G)$, pokud po skončení výpočtu algoritmem ZCYK bude $S \in$ `TL[1,n]`.

ZCYK budeme prezentovat poněkud realističtěji než CYK. Gramatiku budeme považovat také za vstupní parametr algoritmu. Za výstupní parametr budeme považovat obě počítané tabulky. Později ukážeme, že výsledné tabulky tvoří základní informaci o syntaktické analýze pro danou vstupní větu a danou gramatiku.

ZCYK - algoritmus

Vstup : $x = a_1...a_n$, CF-gramatika $G = (V_T, V_N, S, P)$, kde $x \in V_T^+$.
Výstup: Tabulky `TL[1..n,1..n]`, `TP[1..n,1..n]`.

```
FOR i := 1 TO n DO (* j = 1 *)
   TL[i,1] := {A |A  → a_i ∈ P}
   TP[i,1] := {A · α|B → Aα ∈ P, A  ∈ TL[i,1]}
END;
FOR j := 2 TO n DO (* j > 1 *)
   FOR i := 1 TO n+1-j DO
      TP[i,j] := ∅;
      FOR k := 1 TO j-1 DO
```

```
      TP[i,j] := TP[i,j] ∪ {B_1..B_{r+1} · B_{r+2}..B_p|
      B_1..B_r · B_{r+1}..B_p ∈ TP[i,k], B_{r+1} ∈ TL[i+k,j-k]}
    END;
    TL[i,j] := {A |A  → B_1..B_p ∈ P, B_1..B_p· ∈ TP[i,j]}
    TP[i,j] := TP[i,j] ∪ {A · α|C → A · α ∈ P, A  ∈ TL[i,j]}
  END;
END;
IF S ∈ TL[1,n] THEN
  WriteString('x patří do L(G)')
ELSE
  WriteString('x nepatří do L(G)')
END
```

Podobně jako CYK také ZCYK vykonává řádově n^3 kroků a používá řádově n^2 paměťových míst, uvažujeme-li pevně zvolenou gramatiku. Jelikož však gramatika je považována za vstupní parametr, je pro korektní odhad složitosti výpočtu nutné uvažovat také faktor odpovídající velikosti gramatiky. Je třeba si také uvědomit, že množinové operace užívané v ZCYKu nelze považovat za elementární. Jejich složitostní ocenění necháme na čtenáři a jeho představách o postupech realizujících tyto operace. V kapitole o D-gramatikách je ukázána cesta, jak je možno v podobných případech postupovat.

Příklad 30 Vezmeme nejprve (druhou) gramatiku G_2 z příkladu 29. Budeme tak mít možnost porovnat práci CYK´u a ZCYK´u na stejné gramatice a stejném řetězu a budeme ilustrovat skutečnost, že ZCYK je zjemněním CYK´u.

$$\begin{gathered} S \rightarrow AB \\ A \rightarrow CD|CF \\ B \rightarrow c|EB \\ C \rightarrow a \\ D \rightarrow b \\ E \rightarrow c \\ F \rightarrow AD \end{gathered}$$

Vezměme $x = aaabbbcc$. ZCYK-algoritmus vytvoří tabulky TL a TP. Tabulka TL je stejná tabulka jako na obrázku 11. TP najdeme na obrázku 14. Protože nejspodnějí políčko tabulky TL obsahuje symbol S, řetězec x patří do rozpoznávaného jazyka.

Příklad 31 V tomto příkladě pracujeme s gramatikou G_1 z příkladu 29. Ilustrujeme tak práci ZCYK´u s gramatikou, která není v Chomského normální formě, tedy skutečnost, že ZCYK je zobecněním CYK´u.

a	a	a	b	b	b	c	c
C.D C.F	C.D C.F	C.D C.F				E.B	E.B
		CD. A.B A.D				EB.	
		AD.					
	CF. A.B A.D						
	AD.						
CF. A.B A.D							
AB.							
AB.							

Obrázek 14. Tabulka TP

$$
\begin{aligned}
S &\rightarrow AB \\
A &\rightarrow CD|CAD \\
C &\rightarrow a \\
D &\rightarrow b \\
B &\rightarrow c|EB \\
E &\rightarrow c
\end{aligned}
$$

Vezměme $x = aaabbbcc$. ZCYK-algoritmus vytvoří tabulky TL a TP. Tabulka TL je na obrázku 15. TP najdeme na obrázku 16. Jelikož nejspodnějí políčko tabulky TL obsahuje S, tak x patří do rozpoznávaného jazyka.

3.5 CF-gramatiky a C-analýzy

V této sekci se budeme zabývat syntaktickými analýzami podle CF-gramatik (C-analýzami). Nejprve zavedeme pojem *složkový (pod)strom (C-(pod)strom) podle CF-gramatiky*. C-analýzu podle gramatiky pak zavedeme jako množinu všech C-stromů podle dané CF-gramatiky.

Definice 32 (C-(pod)strom podle CF-gramatiky) Nechť $w = a_1, ..., a_n \in {V_T}^+$ je věta, nechť T je souvislý C-podstrom nad w, nechť $G = (V_T, V_N, S, P)$ je CF-gramatika.

Řekneme, že T je *C-podstromem podle CF-gramatiky G nad větou w*, jestliže splňuje podmínku:

Nechť uzel u má tvar $u = [A, Cov(u, T)]$, nechť u má k dcer, kde $k > 0$, a dcery mají po řadě tvar $v_1 = [B_1, Cov(v_1, T)]$,...,$v_k = [B_k, Cov(v_k, T)]$. Nechť

a	a	a	b	b	b	c	c
C	C	C	D	D	D	E,B	E,B
		A				B	
	A						
A							
S							
S							

Obrázek 15.

posloupnost dcer $v_1, ..., v_k$ je uspořádaná podle přirozeného uspořádání (viz 11). Pak existuje pravidlo gramatiky G tvaru

$$A \to B_1, ..., B_k.$$

Řekneme, že T je *C-stromem podle CF-gramatiky G nad větou w*, jestliže je C-podstromem podle G nad větou w a kořen stromu T je tvaru $[S, \{1, ..., n\}]$.

Připomeňme si, že z definice souvislého C-(pod)stromu vyplývá, že žádné z $k+1$ pokrytí $Cov(u, T)$, $Cov(v_1, T)$,...,$Cov(v_k, T)$ neobsahuje díru, a tedy že pokrytí $Cov(v_1, T)$,...,$Cov(v_k, T)$ na sebe podle přirozeného uspořádání souvisle navazují tak, že jejich sjednocení je rovno $Cov(u, T)$.

Skutečnost, že T je C-stromem nad w podle G, znamená, že T představuje návod, jak pomocí pravidel gramatiky redukovat danou větu až k přijímacímu symbolu S. To ale znamená, že gramatika G rozpoznává danou větu w ve smyslu dříve uvedené definice.

Naopak, pokud daná gramatika rozpoznává určitou větu, lze sestrojit C-strom odpovídající (nějakému existujícímu) postupu redukce věty do přijímacího symbolu.

Pojem C-podstromu je technický pojem, který nám umožní zavést pojem parsingu.

C-podstrom nad w podle G můžeme považovat za návod, jak pomocí pravidel gramatiky redukovat podřetěz dané věty až k jedinému (neterminálnímu) symbolu, který nemusí být přijímací.

Nyní zavedeme pojem C-analýzy podle podle dané CF-gramatiky a další příbuzné pojmy.

a	a	a	b	b	b	c	c
C.D C.AD	C.D C.AD	C.D C.AD				E.B	E.B
		CD. A.B				EB.	
	CA.D						
	CAD. A.B						
CA.D							
CAD. A.B							
AB.							
AB.							

Obrázek 16. Tabulka TP2

Definice 33 (C-analýza podle CF-gramatiky)
Nechť $G = (V_T, V_N, S, P)$ je C-gramatika. C je C-analýza nad abecedou V_T.

Řekneme, že $C(G)$ je *C-analýzou podle CF-gramatiky* G, jestliže platí

$$C(G) = \{T | T \textit{ je } \text{C-strom } \textit{podle } G\}$$

Řekneme, že RIC je rozšířenou *C-analýzou podle CF-gramatiky* G, jestliže platí

$$RIC(G) = \{T | T \textit{ je } \text{C-podstrom } \textit{podle } G.\}$$

Nechť $w \in {V_T}^+$. Řekneme, že $C(w, G)$ je *C-analýzou podle CF-gramatiky* G *nad* w, jestliže platí

$$C(w, G) = \{T | T \textit{ je } \text{C-strom } \textit{nad } w \textit{ podle } G.\}$$

Řekneme, že $RIC(w, G)$ je rozšířenou *C-analýzou podle CF-gramatiky* G *nad* w, jestliže platí

$$RIC(w, G) = \{T | T \textit{ je } \text{C-podstrom } \textit{nad } w \textit{ podle } G.\}$$

V pojmu C-analýzy jsme získali třídu syntaktických rozborů složkového typu, vyjadřovací silou odpovídající bezkontextovým gramatikám. Určením CF-gramatiky tedy získáme i to podstatné z hlediska syntaxe modelovaného jazyka: její C-analýzu.

3.6 Rozšířený C-parsing a ZCYK

V této sekci se budeme věnovat vztahu C-analýzy (jednoduché i rozšířené) podle dané CF-gramatiky a výstupů algoritmu ZCYK. Za tímto účelem zavádíme pojem *C-parsingu* a rozšířeného C-parsingu. Ukážeme, že výpočet C-parsingu je významným krokem při výpočtu C-analýzy. C-parsing bude zaveden jako množina položek jistého tvaru. Tímto způsobem představíme čtenáři obecně rozšířenou techniku pro formulaci algoritmů rozpoznávání a syntaktické analýzy bezkontextových jazyků. Více o těchto technikách viz [Sikkel, 1993].

Definice 34 (Položka C-(pod)stromu)
Zápisem $\langle i, j\rangle$, kde $i \leq j$, budeme označovat množinu $\{i, i+1, \ldots, j\}$.

Nechť Tr je souvislý C-(pod)strom nad nějakou větou. Řekneme, že trojice $P = [A, l, p]$ je *L-položkou C-stromu* Tr, právě když C-(pod)strom Tr obsahuje uzel $u = [A, Cov(u, Tr)]$ takový, že

$\langle l, p\rangle = Cov(u, Tr)$.

Řekneme, že trojice $P = [A_1 ... A_j \cdot A_{j+1} ... A_k, l, p]$ je *P-položkou C-stromu* Tr, právě když C-strom Tr obsahuje pro nějaké A uzel tvaru $u_m = [A, \langle l, p_m\rangle]$ s dcerami tvaru $u_i = [A_i, \langle l_i, p_i\rangle]$, kde pro $1 \leq i \leq j$ dcery u_i tvoří posloupnost podle přirozeného uspořádání takovou, že $l = l_1$ a $p_j = p \leq p_m$ a $p_i + 1 = l_{i+1}$ pro $1 < j$. Pokud $j = k$, pak $p_j = p = p_m$.

Označení L- a P- u položek vychází z představy, že tyto položky jsou tvořeny pomocí levých a pravých stran pravidel CF-gramatiky.

Uvádíme následující zřejmé tvrzení, které přímo vyplývá ze vzájemně jednoznačného přiřazení L-položek a uzlů C-stromu a ze skutečnosti, že C-strom je určen množinou svých uzlů.

Tvrzení 35
Souvislý C-strom je určen množinou svých L-položek.

Definice 36 (Položky podle gramatiky nad větou)
Nechť $G = (V_T, V_N, Pr, S)$ je C-gramatika, $w = a_1 \ldots a_n \in {V_T}^+$, $P = [\alpha, i_1, i_2]$ je L-položkou (P-položkou) nějakého C-(pod)stromu podle gramatiky G nad větou w.

Potom říkáme, že $P = [\alpha, i_1, i_2]$ je *L-položkou (P-položkou) podle C-gramatiky G nad větou w*,

Definice 37 (C-parsing nad větou a jeho položky)
Nechť $C(w, G)$ je C-analýza podle CF-gramatiky $G = (V_T, V_N, S, Pr)$ nad větou $w = a_1 \ldots a_n$.

Řekneme, že L-položka je *položkou C-parsingu* $CP(w, G)$, je-li L-položkou nějakého C-stromu $Tr \in C(w, G)$. Tedy $CP(w, G) = \{P |\, existuje\, Tr \in C(w), P$ je L-položkou $Tr\}$.

Definice 38 (Rozšířený C-parsing nad větou)
Nechť $RIC(w, G)$ je rozšířená C-analýza podle CF-gramatiky $G = (V_T, V_N, S, Pr)$ nad větou $w = a_1 \ldots a_n$.

Řekneme, že L-položka (P-položka) je *L-položkou (P-položkou) rozšířeného parsingu* $RICP(w, G)$, je-li L-položkou (P-položkou) nějakého C-podstromu $Tr \in RIC(w, G)$. Tedy $RICP(w) = \{P | P$ existuje Tr, $Tr \in RIC(w, G)$ takový, že P je L- nebo P-položkou $Tr\}$.

Nechť $1 \leq i \leq j \leq n$. Jako $LP(i, j, w, G)$ označíme množinu levých položek tvaru $[A, i, j]$ patřících do $RICP(w, G)$.

Jako $PP(i, j, w, G)$ označíme množinu pravých položek tvaru $[\alpha, i, j]$ patřících do $RICP(w, G)$.

Předchozí označení jsme zavedli proto, abychom mohli ukázat vztah mezi oběma typy parsingu a výstupy algoritmu ZCYK. Tento vztah je zformulován v následujícím tvrzení. Tvrzení je přímočarým důsledkem definic použitých pojmů.

Tvrzení 39 *Nechť je dána věta* $w \in {V_T}^+, |w| = n$, *CF-gramatika* $G = (V_T, V_N, S, Pr)$ *a nechť tabulky* `TL[1..n,1..n]` *a* `TP[1..n,1..n]` *tvoří výstupy algoritmu ZCYK pro vstupy* w *a* G.

Pro $1 \leq j \leq n$ *a* $1 \leq i \leq n + 1 - j$ *platí, že* $A \in$ `TL[i,i+1-j]`, *právě když* $[A, i, j] \in LP(i, j, w, G)$ *a* $\alpha \in$ `TP[i,i+1-j]`, *právě když* $[\alpha, i, j] \in PP(i, j, w, G)$.

Poznámka 40 Vidíme, že ZCYK prostřednictvím tabulek počítá parsing a rozšířený parsing k dané větě a k dané gramatice. Vidíme, že parsingu odpovídá tabulka `TL`. Parsing tvoří zhuštěnou informaci o syntaktické analýze pro danou větu a danou gramatiku. Způsob, jak lze z parsingu počítat vlastní syntaktickou analýzu, ukážeme v příští kapitole o D-gramatikách.

3.7 D-gramatiky

V této sekci zavádíme D-gramatiky vhodné i pro jazyky s volným slovosledem. Inspirujeme se pojmem dominačních (zobecněných závislostních) gramatik [Gladkij, 1973], ale uvolňujeme podmínky pro použití pravidel těchto gramatik a poněkud měníme jejich tvar. Tvar měníme proto, abychom se co nejvíce přiblížili tradici české matematické lingvistiky. Účelem uvolnění podmínek pro aplikaci pravidel je modelování volných vztahů mezi závislostmi a slovosledem. Výklad si zjednodušíme tím, že budeme uvažovat pouze D-stromy bez funktorů a tedy triviální kontrakční funkce, které každému argumentu přiřazují prázdné slovo. Využíváme toho, že vše podstatné o kontrakčních funkcích bylo řečeno v předchozích kapitolách a diskusi o funktorech zde prostě vynecháme. Tím se přiblížíme výše citovanému prameni, kde se také pracuje s D-stromy bez funktorů.

Definice 41 (D-gramatika)
D-gramatika (Dependency grammar) je čtveřice

$$G = (V_T, V_N, S_t, P)$$

kde V_T je konečná množina terminálů, V_N je konečná množina neterminálů,

$S_t \subseteq (V_T \cup V_N)$ je množina počátečních symbolů a P je množina přepisovacích pravidel tří typů:

$A \rightarrow_L BC$, kde $A, B, C \in (V_T \cup V_N)$,
$A \rightarrow_R BC$, kde $A, B, C \in (V_T \cup V_N)$,
$A \rightarrow B$, kde $A, B \in (V_T \cup V_N)$.

Písmena L resp. R v indexu pravidla znamenají, že první resp. druhý symbol pravé strany pravidla je *řídící*; druhý resp. první symbol pravé strany pravidla je potom *závislý*. Má-li pravidlo na pravé straně pouze jeden symbol, považujeme tento symbol za *řídící*.

Pravidlo má následující význam (pro redukci): závislý symbol bude vymazán (obsahuje-li pravá strana pravidla nějaký závislý symbol) a řídící symbol bude přepsán (nahrazen) symbolem z levé strany pravidla.

Vezměme pravidla tvaru

(1a) $A \rightarrow_L BC$,

(1b) $A \rightarrow_R BC$.

Redukci řetězu w podle pravidla (1a) lze uplatnit pro libovolný současný výskyt symbolů B, C v řetězci w, kde symbol B předchází (ne nutně bezprostředně!) symbolu C. Redukce podle (1a) znamená, že symbol B je přepsán symbolem A a symbol C je z řetězu vypuštěn.

Za stejných předpokladů lze redukovat w podle pravidla (1b). Redukce podle (1b) znamená přepsat na A symbol C a vypustit z w symbol B.

Poznámka 42 V definici D-gramatik připouštíme výskyt terminálů i na levé straně pravidel. Je to přirozené, uvědomíme-li si, že obvykle jako terminály užíváme výsledky lexikální analýzy (či jim odpovídající symboly) a že v závislostní syntaxi nemáme povinnost pojmenovávat složky. Neterminály se používají z různých technickýck důvodů. Jeden z podstatných důvodů použití neterminálů v závislostních gramatikách, je potřeba záznamu redukcí aktantů sloves.

Příklad 43 Nechť gramatika G obsahuje pravidlo $A \rightarrow_L BC$, potom řetězec $BBBC$ může být jedním použitím tohoto pravidla zredukován na libovolný z řetězců BBA, BAB a ABB.

Následující definice zavádí standardním způsobem rozpoznávání věty a jazyka D-gramatikou. Později uvedeme (ekvivalentní) definice pomocí DR-stromů, které budou dávat do souvislosti syntaktickou analýzu a rozpoznávání.

Definice 44 (Gramatika rozpoznává větu)
Řekneme, že D-gramatika $G = (V_T, V_N, S_t, P)$ rozpoznává větu $w \in V_T^+$, lze-li větu w opakovanou redukcí pravidly gramatiky G přepsat na větu sestávající z jediného symbolu patřícího do množiny S_t počátečních symbolů gramatiky G.

Definice 45 (Gramatika rozpoznává jazyk)
Řekneme, že D-gramatika G rozpoznává jazyk L, obsahuje-li L právě všechny věty rozpoznávané gramatikou G. Jazyk rozpoznávaný gramatikou G budeme značit $L(G)$.

3.8 DR-analýzy a D-analýzy

V této sekci D-gramatikám přiřadíme DR-analýzy a D-analýzy. Nejprve zavedeme pojem *DR-strom podle D-gramatiky.*

Definice 46 (DR-strom podle D-gramatiky) Nechť w je věta a T je DR-strom nad větou w. Dále nechť G je D-gramatika.

Řekneme, že T je *DR-strom podle D-gramatiky G nad větou w*, jestliže pro každý uzel u tvaru $u = [A, i, v_u, d_u]$ stromu T platí:

1. má-li u jedinou dceru $v = [B, j, v_v, d_v]$
   (z definice DR-stromu plyne, že potom musí platit $d_v = i = j$, $v_v = v_u + 1$),
   potom gramatika G obsahuje pravidlo
   $$A \to B$$
2. má-li u dvě dcery $v = [B, j, v_v, d_v]$ a $w = [C, k, v_w, d_w]$ takové, že $j < k$
   (z definice DR-stromu potom plyne $v_v = v_w = v_u + 1$, $d_v = d_w = i$ a buď $i = j$, nebo $i = k$),
   potom gramatika G obsahuje pravidlo
   $$A \to_x BC,$$
   kde $x = L$, pokud $i = j$, a $x = R$, pokud $i = k$.
3. je-li u kořen stromu T
   (z definice DR-stromu potom plyne $v_u = 0$, $d_u = 0$),
   potom symbol A je prvkem množiny počátečních symbolů gramatiky G.

Poznámka 47 To, že DR-strom nad nějakou větou je DR-stromem podle určité gramatiky, znamená, že tento DR-strom představuje návod, jak pomocí pravidel gramatiky redukovat danou větu až k jedinému symbolu — některému z počátečních symbolů gramatiky. Stejně tak si můžeme představit, že DR-strom je záznamem historie postupného redukování věty podle dané gramatiky.

Existence DR-stromu nad danou větou podle gramatiky znamená, že gramatika *rozpoznává danou větu* ve smyslu dříve uvedené definice.

Naopak, pokud daná gramatika rozpoznává určitou větu, lze sestrojit DR-strom odpovídající (nějakému existujícímu) postupu redukce věty do počátečního symbolu.

D-gramatika G tedy rozpoznává větu w právě tehdy, když existuje DR-strom podle D-gramatiky G nad větou w. Této ekvivalence budeme dále využívat.

Nyní uvedeme další definici, dávající do souvislosti s D-gramatikou ne už jen jeden DR-strom nad jednou větou, ale analýzu — funkci přiřazující větám množiny stromů.

Definice 48 (DR-analýza podle D-gramatiky)
Nechť $G = (V_T, V_N, S_t, P)$ je D-gramatika.

Řekneme, že $DR(G)$ je *DR-analýzou podle D-gramatiky G* , jestliže platí

$DR(G) = \{T|\ ex.\ w \in V_T{}^+, T\ je\ DR-strom\ nad\ w\ podle\ G\}$.

Pro každé $w \in V_T{}^+$ vezmeme

$DR(w, G) = \{T|\ T\ je\ DR-strom\ nad\ w\ podle\ G\}$. Budeme říkat, že $DR(w)$ je *DR-analýzou w podle G*.

Zavedeme ještě analogický pojem pro závislostní (D-)analýzu.

Definice 49 (Kontrakce analýzy) Nechť $DR(G)$ je DR-analýza podle G, kde $G = (V_T, V_N, S_t, P)$ je D-gramatika, nechť $DR(G, w)$ je DR-analýza w podle G.

Řekneme, že $D(G)$ je *D-analýza podle G*, pokud platí:

$$[((\forall T \in\ D(G))(\exists TR \in\ DR(G))\ T = Dkon(TR))$$

$$\wedge((\forall TR \in\ DR(G))\ Dkon(TR) \in\ D(G))]$$

— tj. $D(G)$ je právě množina všech D-stromů vzniklých D-kontrakcí ze všech DR-stromů z $DR(G)$.

Řekneme, že $D(w, G)$ *je D-analýza w* podle G pokud platí:

$$[((\forall T \in\ D(w, G))(\exists TR \in\ DR(w, G))\ T = Dkon(TR))$$

$$\wedge((\forall TR \in\ DR(w, G))\ Dkon(TR) \in\ D(w, G))]$$

— tj. $D(w, G)$ je právě množina všech D-stromů vzniklých D-kontrakcí ze všech DR-stromů z $DR(w, G)$.

3.9 DR-parsing

V této sekci se budeme věnovat výpočtu DR-analýzy podle dané D-gramatiky a podle jistých přirozených omezení. Zavedeme pojem *parsing* a budeme se zabývat složitostí jeho výpočtu. Dále ukážeme postup, jak pomocí parsingu postupně nacházet DR-stromy patřící do hledané DR-analýzy.

V minulých sekcích jsme zavedli pojmy DR- resp. D- analýzy podle gramatiky jako funkce, které každé konkrétní větě přiřadí množinu DR- resp. D-stromů.

V této sekci se omezíme se pouze na D-gramatiky *neobsahující unární pravidla*. Uvedené definice, tvrzení a postupy by bylo možno s určitou modifikací vyslovit i pro gramatiky s unárními pravidly, k našemu cíli, kterým je výpočet analýz vět přirozeného jazyka, je však nutně nepotřebujeme a takové definice by byly méně přehledné. V této souvislosti budeme v následujícím textu hovořit o DR-stromech resp. DR-analýzách bez unárních uzlů a o D-gramatikách bez unárních pravidel.

Označení 50
Připomeňme si, že zápisem $\langle i, j\rangle$, kde $i \leq j$, budeme označovat množinu $\{i, i+1, \ldots, j\}$.

Definice 51 (Položka)
Položkou nazveme $(2k+2)$-tici $P = [A, h, i_1, \ldots, i_{2k}]$, kde $i_1 \leq i_2 < i_3 \leq i_4 \ldots i_{2k-1} \leq i_{2k}$ a $h \in \langle i_1, i_2 \rangle \cup \langle i_3, i_4 \rangle \cup \ldots \cup \langle i_{2k-1}, i_{2k} \rangle$.

A budeme nazývat *symbol položky*, h *horizontální index položky*, množinu $\langle i_1, i_2 \rangle \cup \langle i_3, i_4 \rangle \cup \ldots \cup \langle i_{2k-1}, i_{2k} \rangle$ *pokrytí položky*. Pokrytí položky P budeme zkráceně zapisovat jako $Cov(P)$.
O hodnotě $2k$ budeme hovořit jako o *velikosti položky*.

Definice 52 (Položka DR-stromu)
Nechť Tr je DR-strom bez unárních uzlů.

Řekneme, že položka $P = [A_P, h_P, i_1, \ldots, i_{2k}]$ je *položkou DR-stromu* Tr, pokud DR-strom Tr obsahuje uzel $u = [A, h, v, d]$ takový, že $A_P = A$, $h_P = h$ a $Cov(P) = Cov(u, Tr)$.

Tvrzení 53
DR strom bez unárních uzlů je určen množinou svých položek.

Důkaz:

Nechť je dána množina položek nějakého DR-stromu Tr nad větou $w = a_1 \ldots a_n$. Ukážeme postup, jímž lze z takové množiny určit všechny uzly a hrany DR-stromu Tr.

Začneme položkou s největším pokrytím. Existuje jediná taková položka a tato položka bude mít tvar $P = [A_P, h_P, 1, n]$, kde symbol A_P musí patřit mezi počáteční symboly gramatiky a číslo n odpovídá délce věty w (toto číslo bychom v případě potřeby mohli určit z celkového počtu položek $2n-1$ pomocí známého vztahu mezi počtem uzlů a počtem listů binárního stromu).

Této položce bude odpovídat uzel tvaru $[A_P, h_P, 0, 0]$ — kořen stromu Tr.

Nyní popíšeme odvozovací krok:

Předpokládáme, že k položce P jsme již určili odpovídající uzel stromu u_P.

Je-li $|Cov(P)| > 1$, potom najdeme dvě položky $Q = [A_Q, h_Q, \ldots]$ a $R = [A_R, h_R, \ldots]$ různé od P takové, že platí $Cov(Q) \cup Cov(R) = Cov(P)$. V množině položek stromu Tr musí existovat právě jedna taková dvojice položek a navíc pro ni musí platit $Cov(Q) \cap Cov(R) = \emptyset$.

Každá z položek Q a R bude odpovídat jednomu uzlu stromu,
$u_Q = [A_Q, h_Q, v_Q, d_Q]$ a $u_R = [A_R, h_R, v_R, d_R]$.

Symboly a horizontální indexy těchto uzlů budou převzaty z příslušných položek. Dominační indexy obou uzlů u_Q a u_R budou rovny h_P, vertikální indexy budou o 1 větší než vertikální index uzlu u_P, tedy $v_P + 1$.

Musí platit $h_Q = h_P$ nebo $h_R = h_P$. V prvním případě povede z u_Q do u_P vertikální hrana a z u_R do u_P šikmá hrana, v druhém případě naopak.

Na obě položky (Q a R) a oba uzly (u_Q a u_R) opět aplikujeme právě popsaný odvozovací krok.

Popsaný odvozovací krok pro každou položku a jí odpovídající uzel DR-stromu Tr určí celý podstrom tohoto uzlu. Provedeme-li tento postup pro položku odpovídající kořeni stromu, získáme celý strom Tr. Q.E.D.

Poznámka 54 Opačné tvrzení — že DR strom bez unárních uzlů určuje svou množinu položek — je triviálním důsledkem definice položky DR-stromu. Vidíme vzájemně jednoznačný vztah mezi DR-stromem bez unárních uzlů a množinou jeho položek.

Poznámka 55 Pokud by strom Tr obsahoval unární uzly, existovalo by v množině položek stromu Tr více položek se stejným pokrytím i horizontálními indexy a popsaný postup by nebyl použitelný.

Definice 56 (Položka podle gramatiky nad větou)
Nechť G je D-gramatika bez unárních pravidel a nechť $w = a_1 \dots a_n$ je věta.

Řekneme, že položka $P = [A_P, h_P, i_1, \dots, i_{2k}]$ je *položkou podle gramatiky G nad větou w*, je-li P položkou nějakého DR-stromu podle gramatiky G nad větou w.

Poznámka 57 DR-strom podle gramatiky bez unárních pravidel je DR-stromem bez unárních uzlů.

Definice 58 (Položka analýzy nad větou)
Nechť $DR(G)$ je DR-analýza podle závislostní gramatiky bez unárních pravidel G a nechť $w = a_1 \dots a_n$ je věta.

Řekneme, že položka $P = [A_P, h_P, i_1, \dots, i_{2k}]$ je *položkou analýzy DR nad větou w*, je-li *položkou nějakého DR-stromu $Tr \in DR(w, G)$*.

Nyní zavedeme pojem parsing odpovídající výsledné množině syntaktické analýzy zdola.

Definice 59 (DR-parsing) Nechť G je D-gramatika bez unárních pravidel a $w = a_1 \dots a_n$ je věta.

DR-parsing podle gramatiky G nad větou w nazveme nejmenší množinu P_{RS} položek takovou, že

- $[a_i, i, i, i] \in P_{RS}$
  — tj. pro každé slovo věty w obsahuje P_{RS} položku DR-stromu nad touto větou odpovídající jeho listu
- jsou-li $P, Q \in P_{RS}$, $P = [B, i_P, i_1, \dots, i_{2k_P}]$, $Q = [C, i_Q, j_1, \dots, j_{2k_Q}]$ dvě položky takové, že $Cov(P) \cap Cov(Q) = \emptyset$, platí-li $i_P < i_Q$ a gramatika G obsahuje pravidlo $A \rightarrow_L BC$ resp. $A \rightarrow_R BC$,
  potom P_{RS} obsahuje i položku $R = [A, i_R, l_1, \dots, l_{2k_R}]$, kde $Cov(R) = Cov(P) \cup Cov(Q)$ a $i_R = i_P$ resp. $i_R = i_Q$.
  Říkáme, že položky P a Q tvoří *rozklad položky R* v parsingu P_{RS}.

Prvky P_{RS} budeme nazývat *položky parsingu P_{RS}*.

Na parsing můžeme uplatnit následující omezení (neprojektivity).

Definice 60 Nechť G je D-gramatika bez unárních pravidel, $w = a_1 \dots a_n$ je věta a P_{RS} je parsing podle gramatiky G nad větou w. Dále nechť $d \geq 0$ je celé číslo. Množinu

$$P = \{[A_P, i_P, i_1, \dots, i_{2k_P}] | P \in \ P_{RS}, k_P \leq d + 1\}$$

nazveme *parsing omezený počtem děr d.*

Tvrzení 61 *Je-li $G = (V_N, V_T, St, Pr)$ D-gramatika bez unárních pravidel, $w = a_1 \dots a_n$ věta a $d \geq 0$ celé číslo, potom P_{RS} parsing podle gramatiky G nad větou w omezený počtem děr d má polynomiální velikost a lze jej určit s polynomiální složitostí, časovou i prostorovou, vzhledem k délce věty n.*

Důkaz:

Z předpokladů plyne, že velikosti položek (čísla 2k) nejsou větší než $2d + 2$.

Položek s touto vlastností jistě nemůže existovat více než $|V_N \cup V_T| \times n \times n^{2d+2}$, tedy polynomiální počet vzhledem k délce věty.

Složitost výpočtu je shora omezena například složitostí jednoduchého algoritmu, který vychází od n položek $Pi = [a_i, i, i, i]$ odpovídajících listům DR-stromů a postupně pro všechny dvojice položek zkouší, lze-li jejich spojením na základě jejich symbolů, horizontálních indexů, pravidel gramatiky, pokrytí a předepsané restrikce počtu děr v pokrytí odvodit jinou položku. Pokud ano, zkontroluje ještě, zda se odvozená položka neshoduje s některou položkou odvozenou již dříve.

Je-li cílový počet položek omezen konstantou $K = |V_N \cup V_T| \times n^{2d+3}$, nebude tento proces trvat déle než K^3 kroků, kde za elementární kroky považujeme zjištění, zda lze ze dvou daných položek odvodit další položku a test shodnosti dvou položek — a má tedy také polynomiální složitost vzhledem k délce věty n. Q.E.D.

DR-parsingu využijeme k získání informací o DR-analýze.

Tvrzení 62 *Nechť G je D-gramatika bez unárních pravidel, nechť $w = a_1 \dots a_n$ je věta a P_{RS} je DR-parsing podle gramatiky G nad větou w. Pak P_{RS} obsahuje všechny položky DR-analýzy podle gramatiky G nad větou w.*

Důkaz:

Nechť $DR(G)$ je DR-analýza podle G. Zvolme libovolný $Tr \in \ DR(w, G)$, ukážeme, že P_{RS} obsahuje všechny položky stromu Tr.

Postupujme indukcí podle velikosti pokrytí uzlů stromu Tr.

Položky s jednoprvkovým pokrytím odpovídají listům stromu a musí mít tvar $P_i = [a_i, i, i, i]$. Takové položky parsing podle gramatiky G nad větou w obsahuje z definice.

Nyní předpokládejme, že již víme, že P_{RS} obsahuje všechny položky stromu Tr s pokrytím velikosti nejvýše $k - 1$, kde $k > 1$.

Uzly s pokrytím velikosti k nejsou listy (protože $k > 1$). Takový uzel u musí tedy mít dvě dcery (protože G neobsahuje unární pravidla), pokrytí těchto dcer jsou neprázdná a disjunktní, tedy menší než k, tedy podle indukčního

předpokladu již víme, že parsing P_{RS} obsahuje položky odpovídající těmto dcerám. Podle definice parsingu potom parsing musí obsahovat i položku odpovídající uzlu u. Q.E.D.

Poznámka 63 Viděli jsme, že množina položek DR-stromu jednoznačně určuje DR-strom. Na druhé straně daný parsing sice obsahuje sjednocení množin položek všech stromů určité DR-analýzy nad určitou větou, ale toto sjednocení, bez znalosti gramatiky, nemusí postačovat k rekonstrukci původních množin položek jednotlivých stromů.

Označení 64 Nechť G je D-gramatika bez unárních pravidel, w je věta a $P_{RS} = \{P_1, \ldots, P_n\}$ je DR-parsing podle gramatiky G nad w. Předpokládejme zároveň, že jednotlivé položky parsingu P_{RS} jsou pevně očíslovány.

Řekneme, že položka P_i je *odvozená*, pokud platí $|Cov(P_i)| > 1$.

S jakoukoliv odvozenou položkou P_i musí P_{RS} obsahovat také jednu nebo více dvojic položek (P_j, P_k) tvořících *rozklad položky P_i v parsingu P_{RS}*.

Rozklady téže položky lze uspořádat lexikograficky podle velikostí indexů, první z nich označíme za *první rozklad položky P_i* a ke každému rozkladu kromě *posledního* dokážeme nalézt *následující rozklad*.

Označení 65 Nechť G je D-gramatika bez unárních pravidel, w je věta $P_{RS} = \{P_1, \ldots, P_n\}$ je parsing podle gramatiky G nad w a $P \in P_{RS}$ je položka.

Definujme množinu položek $M(P)$ jako nejmenší množinu takovou, která obsahuje položku P a která zároveň s každou odvozenou položkou obsahuje i položky tvořící její první rozklad v parsingu P_{RS}.

DR-strom, který k této množině položek sestrojíme postupem popsaným v důkazu Tvrzení 53 nazveme *první podstrom položky P v parsingu P_{RS}*.

Tvrzení 66 *Nechť $w = a_1 \ldots a_n$ je věta, G je D-gramatika bez unárních pravidel, $P_{RS} = \{P_1, P_2, \ldots, P_m\}$ je parsing podle gramatiky G nad větou w a $DR(G)$ je DR-analýza podle G. Potom lze polynomiálním počtem kroků vzhledem k m a n určit, že množina $DR(w, G)$ je prázdná nebo nalézt první DR-strom $Tr \in DR(w, G)$.*

Důkaz:

Určení, zda množina $DR(w, G)$ je prázdná, je ekvivalentní zjištění, zda parsing P_{RS} obsahuje alespoň jednu položku P_{root} takovou, že $Cov(P_{root}) = \{1, \ldots, n\}$ a symbol v položce je startovací symbol. Pokud ano, hledaný DR-strom bude *první podstrom položky P_{root}*.

Q.E.D.

Tvrzení 67 *Nechť $w = a_1 \ldots a_n$ věta, G je D-gramatika bez unárních pravidel, $P_{RS} = \{P_1, P_2, \ldots, P_m\}$ je parsing podle gramatiky G nad větou w, $DR(G)$ je DR-analýza podle G, a $Tr_s \in DR(w, G)$ je DR-strom nad w podle analýzy DR. Potom existuje uspořádání U na množině $DR(w, G)$ a algoritmus, který po polynomiálně omezeném počtu kroků vzhledem k hodnotám m a n vydá strom $Tr_{s+1} \in DR(w, G)$, který podle uspořádání U následuje Tr_s, nebo zjistí, že takový strom již neexistuje.*

Důkaz:

Ukážeme způsob, jak nalézt další strom; uspořádání U bude dáno pořadím, v němž algoritmus nachází jednotlivé stromy.

Další strom nalezneme následujícím algoritmem:

1. Uzly stromu Tr_s očíslujeme podle jejich pozice ve stromě průchodem preorder, tj. vždy nejprve matky, potom celý levý podstrom a potom celý pravý podstrom. Jako P_{root} označíme položku odpovídající kořeni stromu Tr_s.
2. Procházíme uzly, které nejsou listy, v klesajícím pořadí podle očíslování, pro každý uzel určíme odpovídající položku $P_i \in P_{RS}$, podle dcer tohoto uzlu a jim odpovídajících položek určíme (P_j, P_k) rozklad položky P_i a zkoumáme, zda pro tento uzel a tento rozklad existuje následující rozklad podle parsingu P_{RS}.
3. Nalezneme-li uzel u, kde k odpovídající položce P_i a jejímu rozkladu (P_j, P_k) existuje následující rozklad $(P_{j'}, P_{k'})$, potom
   (a) změníme podstrom uzlu u tak, že jeho dcery budou uzly odpovídající položkám $P_{j'}$ a $P_{k'}$ a jejich podstromy budou první podstromy položek $P_{j'}$ a $P_{k'}$.
   (b) Dále pro každý uzel v v upraveném stromě, který ve svém levém podstromu obsahuje uzel u, určíme použitý rozklad (P_{j_u}, P_{k_u}) a jeho pravý podstrom nahradíme prvním podstromem položky P_{k_u}.
4. Nenalezneme-li žádný uzel u, pro který by existoval další rozklad, vyčerpali jsme všechny stromy s kořenem odpovídajícím položce P_{root}.
   Hledáme tedy další položku pokrývající celou větu. Najde-li se taková položka, tento algoritmus vyhledá první strom, který ji odpovídá, výše popsaným způsobem.
5. Neexistuje-li již žádná další položka pokrývající celou větu, neexistuje další DR-strom $Tr_{s+1} \in DR(w, G)$.

Je zřejmé, že popsaný postup prochází všechny možnosti sestrojení dalšího stromu.

Považujme zjištění, zda lze ze dvou daných položek odvodit třetí danou položku, za elementární krok. Algoritmus prochází nejvýše $(2n-1)$ uzlů, pro každý z nich prohlíží nejvýše $m(m-1)/2$ dvojic položek, aby zjistil, zda existuje následující rozklad příslušného uzlu, tedy $(2n-1)m(m-1)/2$ kroků. Nejvýše stejně dlouho trvá nalezení prvního rozkladu všech položek v novém podstromu a vytvoření prvních podstromů všech pravých dcer matky uzlu u (protože pro každý uzel bude algoritmus první podstrom vytvářet nejvýše jednou).

Složitost algoritmu tedy není větší než nm^2. Q.E.D.

Předchozí dvě tvrzení říkají, že ačkoliv složitost syntaktické analýzy i v případě omezené DR-neprojektivity může být (více než) exponenciální vzhledem k délce věty, lze tuto analýzu provádět tak, že v polynomiálně omezeném čase rozhodneme, zda existuje nějaký strom nad větou podle dané analýzy a potom postupně, opět v polynomiálně omezených časech, hledáme po jednom další stromy až do celkového počtu. Tato tvrzení představují návod, jak můžeme postupovat při výpočtu analýz v konkrétních programech.

Poznámka 68 V této sekci hovoříme o DR-analýzách, přestože žádaným výsledkem z pohledu závislostní syntaxe je D-analýza. D-analýza je však definována jako kontrakce jednotlivých stromů DR-analýzy a výpočet D-analýzy tedy sestává z výpočtu DR-analýzy a potom určení kontrakcí stromů a případně vyloučení shodných D-stromů. Přitom výpočet kontrakce DR-stromu nad větou délky n i testy shodnosti nalezených D-stromů jsou z hlediska složitosti výpočtu nevýznamné, protože leží pod odhady složitosti prezentovaných algoritmů.

Důvodem, proč D-analýzy zavádíme pouze jako odvozené od DR-analýz, je naše přednostní orientace na tvorbu a ladění gramatik, kde DR-stromy poskytují více informací než D-stromy. DR-analýza může reprezentovat syntaktickou analýzu ve smyslu HPSG ([Pollard, Sag, 1994]). Ostatně s analogií DR-stromů pracoval již [Gaifman, 1965] při porovnávání síly složkových a závislostních gramatik. Za významná považujeme omezení slovosledu, která lze zavést pomocí DR-stromů (viz [Holan, 2001], [Plátek at al., 2002]) a která vedou k polynomiální složitosti problému rozpoznávání..

3.10 Kategoriální gramatiky a DR-analýzy

V této kapitole přiřadíme kategoriálním gramatikám odpovídající DR-analýzy a D-analýzy. Technicky to provedeme tak, že každé kategoriální gramatice přiřadíme D-gramatiku, která přesně simuluje redukce podle kategoriální gramatik. To ovšem znamená, že budeme uvažovat pouze projektivní DR-stromy.

Definice 69 Nechť $Kg = (V_T, V_A, S, F)$ je kategoriální gramatika, Připomeňme, že F je funkce z V_T do $P_{fin}(C)$: tj. je to funkce přiřazující každému terminálnímu symbolu konečnou množinu kategorií.

Kategoriální gramatice Kg přiřadíme D-gramatiku $Dg(Kg) = (V_T, V_N, S, P)$, kde P a V_N budou popsány níže.

P bude množina přepisovacích pravidel tří typů:

(l) $A \rightarrow_L BC$, kde $A, B, C \in V_N$,
(r) $A \rightarrow_R BC$, kde $A, B, C \in V_N$,
(v) $A \rightarrow a$, kde $A \in V_N, a \in V_T$.

Označme jako $P_l = \{p \in P |\ p$ je pravidlo typu (l)$\}$,
$P_r = \{p \in P |\ p$ je pravidlo typu (r)$\}$,
$P_v = \{p \in P |\ p$ je pravidlo typu (v)$\}$.

Nejjednodušší je definovat množinu P_v: $P_v = \{X \rightarrow a | a \in V_T; X \in F(a)\}$.
Nyní postupně zavedeme množiny V_N, P_l, P_r.
$N_f = \{k |$ existuje $a \in V_T; k \in F(a)\}$.

(i) Každé $A \in N_f$ je prvkem V_N.
(ii) Nechť $(\alpha/\beta) \in V_N$. Pak $\alpha, \beta \in V_N$ a $\alpha \rightarrow_L (\alpha/\beta)\beta \in P_l$
(iii) Nechť $(\alpha\backslash\beta) \in V_N$. Pak $\alpha, \beta \in V_N$ a $\beta \rightarrow_R \alpha(\alpha\backslash\beta) \in P_r$
(iv) V množinách V_N, P_l a P_r jsou jen prvky, které vznikly podle pravidel (i), (ii), (iii).

Definice 70 (DR-analýza podle kategoriální gramatiky)
Nechť $Kg = (V_T, V_A, S, F)$ je kategoriální gramatika a $Dg(Kg) = (V_T, V_N, S, P)$ je D-gramatika přiřazená ke Kg.

Řekneme, že $DR(Kg)$ je *DR-analýzou podle kategoriální gramatiky* Kg, jestliže platí

$DR(Kg) = DR(Dg(Kg))$.

Pro každé $w \in {V_T}^+$ vezmeme

$DR(w, Kg) = DR(w, Dg(Kg))$. Budeme říkat, že $DR(w, Kg)$ je DR-analýzou nad w podle Kg.

Definice 71 (D-analýza podle kategoriální gramatiky)

Nechť $Kg = (V_T, V_A, S, F)$ je kategoriální gramatika a $Dg(Kg) = (V_T, V_N, S, P)$ je D-gramatika přiřazená ke Kg.

Platí-li, že

$$D(Kg) = D(Dg(Kg))$$

řekneme, že $D(Kg)$ je *D-analýzou podle podle kategoriální gramatiky* Kg. Prvky $D(Kg)$ budeme nazývat *D-stromy podle kategoriální gramatiky* Kg.

Pro každé $w \in {V_T}^+$ vezmeme $D(w, Kg) = D(w, Dg(Kg))$. Budeme říkat, že $D(w, Kg)$ je D-analýzou w podle Kg.

Poznámka 72 V předchozích definicích jsme brali v úvahu klasické kategoriální gramatiky, jejichž analýzy reprezentují množiny projektivních DR-stromů a popřípadě množiny projektivních D-stromů. Úvahy o parsingu a syntaktické analýze kategoriálních gramatik lze podle předchozích definic převést na úvahy o (ne)projektivním parsingu a syntaktické analýze závislostních gramatik. To je možné proto, že Dg(Kg) lze snadno rozdělit na dvě části. Unární pravidla Dg(Kg) slouží pouze pro lexikální analýzu, která se pro potřeby výpočetní lingvistiky odděluje od vlastní syntaktické analýzy. Pro formulaci syntaktické analýzy slouží u Dg(Kg) binární pravidla.

4 Shrnutí

Provádět syntaktickou analýzu češtiny je jiná úloha, než syntaktická analýza angličtiny. Čeština je jazyk s volným slovosledem, tedy jazyk, jehož závislosti (valence) a slovosled se vzájemně ovlivňují relativně málo a tyto souvislosti je teprve třeba pracně hledat. Naší snahou bylo připravit aparát, kde zmíněné souvislosti lze formulovat pomocí různých typů omezení (constraints). V pracech [Holan, 2001], [Holan at al., 2000], [Plátek at al., 2002] se ukazuje, že prezentované pojetí D-gramatik dává pro formulaci slovosledných omezení velký prostor. Například jsme schopni sestavit škálu omezení takovou, že umožní nastavit stupeň volnosti slovosledu. Je též ukázáno, že tímto stupněm volnosti slovosledu lze měřit vzdálenost od stupně volnosti slovosledu, který povolují bezkontextové gramatiky. To je důležité pro porovnání slovosledu u různých jazyků.

V češtině byly nalezeny jevy, které v principu dovolují konstrukce, pomocí nichž lze překročit libovolný pevně zvolený stupeň této volnosti (viz [Holan at al., 2000]). Tím se čeština výrazně liší od angličtiny, pro kterou nebyly nalezeny konstrukce, převyšující stupeň dva. Naopak čeština je v tomto ohledu velmi blízká latině, nemluvě o dalších slovanských jazycích.

Účelem tohoto textu bylo seznámit čtenáře se základními pojmy a technikami vhodnými pro pochopení problematiky syntaktické analýzy přirozených jazyků, zejména češtiny.

Reference

[Ajdukiewicz, 1935]
Ajdukiewicz, K.: *Die syntaktische Konnexität,* Studia Philosophica, 1935, No 1, pp. 1–27

[Bar-Hillel, 1964]
Bar-Hillel Y.: *Language and Information,,* Reading (Mass.), 1964

[Gaifman, 1965]
Gaifman H.: *Dependency systems and phrase structure systems,* Information and Control, 8, No 3, 1965, pp. 304–337

[Gladkij, 1973]
Alexej V. Gladkij: *Formalnyje gramatiki i jaziki,* Nauka, Moskva, 1973

[Hajič, 1998]
Jan Hajič: *Building a Syntactically Annotated Corpus: The Prague Dependency Treebank,* In: Issues of Valency and Meaning, Karolinum, Praha 1998, pp. 106–132

[Holan, 2001]
Tomáš Holan: *Nástroje pro vývoj závislostních analyzátorů přirozených jazyků s volným slovosledem,* MFF UK Praha, 2001

[Holan at al., 1995]
Tomáš Holan, Vladislav Kuboň, Martin Plátek: *An Implementation of Syntactic Analysis of Czech,* In: Fourth International Workshop on Parsing Technologies, ed. Bunt H., Praha, Institute of Formal and Applied Linguistics, Charles University, Prague, 1995, pp. 126–135

[Holan at al., 1997]
Tomáš Holan, Vladislav Kuboň, Martin Plátek: *A Prototype of a Grammar Checker for Czech,* In: Proceedings of the Fifth Conference on Applied Natural Language Processing, pp. 147–154, Association for Computational Linguistics, Washington, DC, USA, 1997

[Holan at al., 1998]
Tomáš Holan, Vladislav Kuboň, Karel Oliva, Martin Plátek: *Two Useful Measures of Word Order Complexity,* In: Processing of Dependency-based Grammars: Proceedings of the Workshop, pp. 21–28, ACL, Montreal, 1998

[Holan at al., 2000]
Tomáš Holan, Vladislav Kuboň, Karel Oliva, Martin Plátek: *On Complexity of Word Order,* In: Les grammaires de dépendance - Traitement automatique des langues Vol 41, No 1 (2000) (special issue on dependency grammars of the journal Traitement automatique des langues, guest editor Sylvain Kahane), pp. 273–300

[Kuboň, 2001]
Vladislav Kuboň: *Problems of Robust Parsing for Czech,* Disertační práce, MFF UK, Praha, 2001

[Kunze, 1972]
Jürgen Kunze: *Die Auslassbarkeit von Satzteilen bei koordinativen Verbindungen in Deutschen,* Akademie-Verlag-Berlin, 1972

[Kunze, 1975]
Jürgen Kunze: *Abhängigkeitsgrammatik,* Akademie-Verlag, Berlin 1975

[Melčuk, 1988]
Igor A. Melčuk: *Dependency Syntax: Theory and Practice,* State University of New York Press, 1988

[Nebeský, 1972]
Ladislav Nebeský: *A Projectivity Theorem,* In: Prague Studies in Mathematical Linguistics (3), Academia, Praha, 1972, pp. 165–169

[Panevová, 1980]
Panevová, J.: *Formy a funkce ve stavbě české věty.,* Academia, Praha, 1980.

[Partee at al., 1993]
Partee, Barbara H., Alice ter Meulen and Robert Wall: *Mathematical Methods in Linguistics,* Dordrecht: Kluwer, 1990. Second printing, corrected first edition, 1993

[Plátek at al., 2002]
Plátek M., Holan T., Kuboň V., Oliva K.: *Word-Order Relaxations and Restrictions within a Dependency Grammar,* in: G. Andrejková, S. Krajči (Eds.): ITAT'2001, Workshop on Theory and Practice of Information Technologies, Zuberec, Slovakia, September, pp. 5–26, 2001

[Pollard, Sag, 1994]
Pollard, C. and I. Sag : *Head-Driven Phrase Structure Grammar,* Chicago: University of Chicago Press, and Stanford: CSLI Publications, 1984

[Schöning,1992]
Uwe Schöning: *Theoretische Informatik kurz gefaßt,* BI-Wiss.-Verl., Manheim, 1992

[Sgall, 1967]
Sgall, P.: *Generativní popis jazyka a česká deklinace.,* Academia, Praha, 1967

[Sgall at all, 1986]
Sgall, P., Hajičová, E., Panevová, J.: *The Meaning of the Sentence in Its Semantic and Pragmatic Aspects,* (ed. by J. Mey), Dordrecht:Reidel and Prague:Academia, 1986

[Sikkel, 1993]
K. Sikkel.: *Parsing Schemata - a framework for specification and analysis of parsing algorithms.,* Texts in Theoretical Computer Science - An EATCS Series, Springer-Verlag, Berlin/Heidelberg/New York

[Šmilauer, 1958]
Šmilauer, V.: *Učebnice větného rozboru.,* Skripta FF UK, SPN, Praha, 1958