

Funkční generativní popis a restartovací automaty¹

Martin Plátek¹ Markéta Lopatková²

¹KTIML, MFF UK, Praha

²ÚFAL, MFF UK, Praha

Mísečky 2009

Autoři jsou podporováni programem 'Informační společnost',
projekt č. 1ET100300517.

Obsah

- 1 O Funkčním Generativním Popisu
- 2 Redukční analýza pro FGP
- 3 Restartovací automaty a FGP
- 4 Teoretické výsledky a FGP
- 5 Výhledy

O Funkčním Generativním Popisu

Navazujeme na práce o **F**unkčním **G**enerativním **P**opisu (**FGP**)
FGP – popis typu meaning - text, rozčleněný (podle typů kategorií a jednotek) do (3-5) rovin.

Vyvíjený od 60.let **L**aboratoří **A**lgebraické **L**ingvistiky pod vedením P.Sgalla.

Dnes – Ústav formální a aplikované lingvistiky (**UFAL**).

Podobné popisy :

- **I.Melčuk** (**ruština, asi i francouština**),
- **J.Kunze – němčina** .

Sgallova metoda je postatně odlišná od Chomského přístupu.

O Funkčním Generativním Popisu

Snažíme se vystihnout podstatu Sgallovy metody a výstižně ji reprezentovat restartovacími automaty.

O FGP

Zdroje k přednášce

PS78 Plátek, M., and P. Sgall: A Scale of Context-Sensitive Languages: Applications to Natural Language. *Information and Control* Vol. **38**(1) (1978) 1–20

LPS07 Lopatková, M., M. Plátek, and P. Sgall: Towards a formal model for functional generative description, analysis by reduction and restarting automata. *The Prague Bulletin of Mathematical Linguistics* Vol. **87** (2007) 7–26

Formální popisy přirozených jazyků (FPPJ)

Základní úkoly

Formální popisy přirozeného jazyka (typu **meaning-text**) mají za úkol vymezit následujícími požadavky (t.zv. **explikativní sílu**):

- množinu správně utvořených vět daného jazyka **LL** (t.j. **generativní sílu**)
- množinu **LM** významových (tektogramatických) reprezentací vět
- **SH**-relaci (**rozlišovací relací**) mezi LL a LM popisující synonymii a homonymii; (se zvláštním ohledem na valenční synonymii, aktuální členění a **slovosled**)
- množinu **zjednoznačňujících kategorií, umožňujících bezchybnou (závislostní) redukční analýzu**

FPPJ

Další úkoly

FGP tradičně jako rozlišovací relaci používá formální překlad (binární relaci nad dvěma formálními jazyky). Rozlišovací relaci budeme prezentovat jako formální překlad i zde.

Další úkoly:

- zachytit stavbu vět z LL (povrchovou redukční analýzou)
- zachytit stavbu vět z LM (tektogramatickou redukční analýzou)
- umožnit porovnávání stavby vět v LL i LM kvůli slovosledu a aktuálnímu členění

Čtyři roviny FGP

- rovina slovních tvarů a interpunkce (*w*-rovina).
Odpovídá jazyku LC;
- morfemická rovina (*m*-rovina), která pro každý slovní tvar či interpunkční znaménko udává jeho lemma a morfologické kategorie;
- rovina povrchové syntaxe FGP (*a*-rovina) zachycující vztahy jako ‘býti_subjektem’, ‘býti_objektem’, ‘býti_přívlastkem’ ... ;
- rovina jazykového významu, označovaná jako **tektogramatická rovina** (*t*-rovina), která specifikuje zejména tzv. hloubkové role, gramatémy a aktuální členění.
Odpovídá formálnímu jazyku LM.

Příklad redukční analýzy

Vstupní věta a 1. krok

<i>Našeho</i>	<i>muj.PSMS4</i>	Atr	
<i>Karla</i>	<i>Karel.NNMS4</i>	Obj	
<i>plánujeme</i>	<i>plánovat.VB-P-</i>	Pred	[my].ACT_t <i>plánovat.PRED.Fr1_f</i> <i>Karel.PAT_c</i> <i>my.APP_f</i>
<i>poslat</i>	<i>poslat.Vf- - -</i>	Obj	[my].ACT_t <i>poslat.PAT.Fr2_f</i>
<i>na</i>	<i>na.RR- - 4</i>	AuxP	
<i>příští</i>	<i>příští.AA4IS</i>	Atr	
<i>rok</i>	<i>rok.NNIS4</i>	Adv	<i>rok.THL_f</i> <i>příští.RSTR_f</i>
<i>do</i>	<i>do.RR- - 2</i>	AuxP	
<i>Anglie</i>	<i>Anglie.NNFS2</i>	Adv	<i>Anglie.DIR3.basic_f</i>
<i>.</i>	<i>..Z: - - -</i>	AuxK	

Příklad redukční analýzy

2. krok

<i>Karla</i>	<i>Karel.NNMS4</i>	Obj	[my].ACT_t
<i>plánujeme</i>	<i>plánovat.VB-P-</i>	Pred	<i>plánovat.PRED.Fr1_f</i> <i>Karel.PAT_c</i>
<i>poslat</i>	<i>poslat.Vf- - -</i>	Obj	[my].ACT_t
<i>na</i>	<i>na.RR- - 4</i>	AuxP	<i>poslat.PAT.Fr2_f</i>
<i>příští</i>	<i>příští.AA4IS</i>	Atr	
<i>rok</i>	<i>rok.NNIS4</i>	Adv	<i>rok.THL_f</i> <i>příští.RSTR_f</i>
<i>do</i>	<i>do.RR- - 2</i>	AuxP	
<i>Anglie</i>	<i>Anglie.NNFS2</i>	Adv	<i>Anglie.DIR3.basic_f</i>
.	<i>..Z: - - -</i>	AuxK	

Příklad redukční analýzy

3. krok

<i>Karla</i>	<i>Karel.NNMS4</i>	Obj	[my].ACT_t
<i>plánujeme</i>	<i>plánovat.VB-P-</i>	Pred	<i>plánovat.PRED.Fr1_f</i> <i>Karel.PAT_c</i>
<i>poslat</i>	<i>poslat.Vf- - -</i>	Obj	[my].ACT_t <i>poslat.PAT.Fr2_f</i>
<i>na</i>	<i>na.RR- - 4</i>	AuxP	
<i>rok</i>	<i>rok.NNIS4</i>	Adv	<i>rok.THL_f</i>
<i>do</i>	<i>do.RR- - 2</i>	AuxP	
<i>Anglie</i>	<i>Anglie.NNFS2</i>	Adv	<i>Anglie.DIR3.basic_f</i>
.	<i>..Z: - - -</i>	AuxK	

Příklad redukční analýzy

4.krok

<i>Karla</i>	<i>Karel.NNMS4</i>	Obj	[my].ACT_t
<i>plánujeme</i>	<i>plánovat.VB-P-</i>	Pred	<i>plánovat.PRED.Fr1_f</i>
			<i>Karel.PAT_c</i>
			[my].ACT_t
<i>poslat</i>	<i>poslat.Vf- - -</i>	Obj	<i>poslat.PAT.Fr2_f</i>
<i>do</i>	<i>do.RR- - 2</i>	AuxP	
<i>Anglie</i>	<i>Anglie.NNFS2</i>	Adv	<i>Anglie.DIR3.basic_f</i>
.	<i>..Z: - - -</i>	AuxK	

Příklad redukční analýzy

neredukovatelná věta

→

<i>plánujeme</i>	<i>plánovat.VB-P-</i>	Pred	[my].ACT_t <i>plánovat.PRED.Fr1_f</i> .PAT_f
.	..Z: - - -	AuxK	

Vlastnosti závislostní redukční analýzy

Lexikalizované roviny

- Závislostní redukční analýza je po jednotlivých rovinách lexikalizovaná, t.j. (základní) kategorie jsou vázány k lexikálním jednotkám, ke každé lexikální jednotce se jich váže omezený počet.
- Lexikální jednotky na povrchových rovinách bývají neřádka rozmístěny jinak než na tektogramatické rovině.
- Chomského základní kategorie bývají přiřazeny souvislým úsekům vět, jejichž délku v obecném případě nelze omezit

Restartovací automaty

Zdroje

KP94 V. Kuboň and M. Plátek. A grammar based approach to a grammar checking of free word order languages. In: *COLING'94, Proc., Vol. II*, Kyoto, Japan, 1994, 906–910.

JMPV99 P. Jančar, F. Mráz, M. Plátek, and J. Vogel. On monotonic automata with a restart operation. *J. Autom. Lang. Comb.* 4 (1999) 283-292.

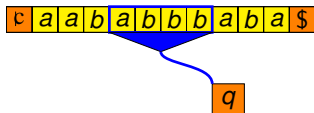
Např. charakterizace DCFL deterministickými monotonními restartovacími automaty, které nepoužívají pomocné symboly a pouze vypouštějí. Ukázáno, že charakterizace CFL potřebuje pomocné symboly.

Restartovací automaty

Zdroje

- MMOP** Messerschmidt H., F. Mráz, F. Otto, and M. Plátek:
Correctness Preservation and Complexity of Simple
RL-Automata. In *Lecture Notes in Computer Science*, Vol.
4094 (2006) 162–172
- OP** Friedrich Otto, Martin Plátek: A Two-Dimensional
Taxonomy of Proper Languages of Lexicalized
FRR-Automata. LATA 2008: 409-420

Obecný model restartovacího automatu



- Podstatné prvky:
 - ohraničená pružná páska
 - konečná řídicí jednotka s (re)startovacím stavem q_0
 - pracovní okno pevné velikosti
 - operace: posun doprava, posun doleva, přepis obsahu okna, restart, accept, reject

Výpočet restartovacího automatu

Cykly, koncovky

ϵ a $\$$ nelze během výpočtu přepsat nebo vypustit

Cyklus : část výpočtu mezi dvěma (re)starty

Koncovka : část výpočtu mezi posledním restartem a zastavením.

V koncovce se nevypouští!

$$\begin{aligned} w_1 &= \text{[yellow bar with 2 red segments]} \\ w_2 &= \text{[yellow bar with 1 red segment]} \\ w_2 &= \text{[yellow bar with 3 red segments]} \\ &\dots \\ w_{n-1} &= \text{[yellow bar with 2 red segments]} \\ w_n &= \text{[yellow bar]} \end{aligned} \quad w_1 \vdash M^c w_2 \vdash M^c \dots \vdash M^c w_{n-1} \vdash M^c w_n$$

Poznámka: pokud M nepřijímá w_n pak M nepřijímá žádné z

w_1, \dots, w_n

Typy restartovacích automatů z aplikačního hlediska

Chomského restartovací analyzátoři, redukční analyzátoři

Redukční analyzátoři

- (re)startuje na levém konci pásky a z restartovacího stavu bez omezení na přítomnost kategorií na pásce
- pro redukční analýzu požadujeme, aby přepisy nic nepřidávaly a alespoň jeden přepis v cyklu zkracoval. Startuje se v restartovací konfiguraci, t.j. s kategoriemi na pásce
- pokud M přijímá w_1 pak M přijímá všechna w_1, \dots, w_n (correctness preserving property),

Chomského restartovací analyzátoři

- smějí (omezeně) připisovat kategorie. Startují z počáteční konfigurace, t.j. zcela bez kategorií
- pokud M nepřijímá w_1 pak M nepřijímá všechna w_1, \dots, w_n (error preserving property)

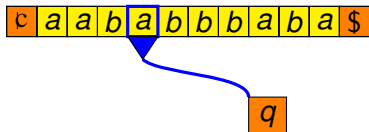
t-restartovací automat (t-sRL-automat)

Závislostní redukční analyzátor

Vlastnosti:

- pracovní okno velikosti 1
- hlavně vypouští políčka (symboly), okrajově přesunuje políčka (např. kvůli se), či případně mění jejich obsah (zatím nepoužíváme)
- během cyklu vypouští nejméně jedno políčko a nejvíce t políček
- vypouštění i v jednom cyklu nemusí být lokální

t-sRL-automat



$M = (Q, \Sigma, \delta, q_0, c, \$)$:

- Q konečná množina stavů
- Σ konečná množina symbolů (slovník)
- $c, \$$ jsou zarážky, $\{c, \$\} \cap \Sigma = \emptyset$
- $q_0 \in Q$ je startovací stav
- δ konečná množina instrukcí

Jazyky rozpoznávané t-sRL-automatem

- M přijímá řetěz w – existuje přijímací výpočet z (re)startovací konfigurace $q_0cw\$$
- $L_C(M)$ – jazyk řetězů, které M přijímá – **charakteristický jazyk** automatu M .
- $S_C(M)$ – **jednoduchý jazyk** automatu M – jazyk řetězů přijímaných bez restartu.

Základní vlastnosti t-sRL-automatů

Zachovávání chyby a správnosti

Fakt

- 1 Každý t-sRL-automat *zachovává chybu*.
[$u \notin L_C(M)$ a $u \vdash^{c^*}_M v$ implikuje, že $v \notin L_C(M)$.]
- 2 Každý deterministický t-sRL-automat zachovává správnost.
[$u \in L_C(M)$ a $u \vdash^{c^*}_M v$ implikuje, že $v \in L_C(M)$.]
- 3 Existují nedeterministické t-sRL-automaty, které nezachovávají správnost.

Základní vlastnosti t-sRL-automatů

Zachovávání chyby a správnosti

Fakt

- 1 Každý t-sRL-automat zachovává chybu.
[$u \notin L_C(M)$ a $u \vdash^{c^*}_M v$ implikuje, že $v \notin L_C(M)$.]
- 2 Každý deterministický t-sRL-automat zachovává správnost.
[$u \in L_C(M)$ a $u \vdash^{c^*}_M v$ implikuje, že $v \in L_C(M)$.]
- 3 Existují nedeterministické t-sRL-automaty, které nezachovávají správnost.

Meta-instrukce

- vypouštěcí meta-instrukce

$$(c \cdot E_0, a_1, E_1, a_2, E_2, \dots, E_{s-1}, a_s, E_s \cdot \$),$$

E_0, \dots, E_s jsou regulární jazyky, kterým říkáme **regulární omezení** a $a_1, \dots, a_s \in \Sigma$ jsou výskyty symbolů vypouštěné touto meta-instrukcí. Pokud $v_0 \in E_0, \dots, v_s \in E_s$, a

$$u = \boxed{v_0 \ a_1 \ v_1 \ a_2 \ \dots \ v_{s-1} \ a_s \ v_s}$$

$$v = \boxed{v_0 \ v_1 \ \dots \ v_{s-1} \ v_s}$$

Pak píšeme $u \vdash_M^c v$.

Meta-instrukce

- přijímací meta-instrukce

$(\epsilon \cdot E \cdot \$, \text{Accept}),$

kde E je regulární jazyk.

- Tato instrukce přijme každé slovo z E .

Formální příklad

- $LR_t := \{ c_0 w c_1 w c_2 \dots c_{t-1} w \mid w \in \{a, b\}^* \}$,
- t -sRL-automat M_t pro jazyk LR_t (Σ_0 označuje $\{a, b\}$):
 - (1) $(c c_0, a, \Sigma_0^* \cdot c_1, a, \Sigma_0^* \cdot c_2, \dots, \Sigma_0^* \cdot c_{t-1}, a, \Sigma_0^* \cdot \$)$,
 - (2) $(c c_0, b, \Sigma_0^* \cdot c_1, b, \Sigma_0^* \cdot c_2, \dots, \Sigma_0^* \cdot c_{t-1}, b, \Sigma_0^* \cdot \$)$,
 - (3) $(c c_0 c_1 \dots c_{t-1} \$, \text{Accept})$.
- Ukázkový výpočet automatu M_3 :
 - $c_0 a b b c_1 a b b c_2 a b b$
 - $c_0 b b c_1 b b c_2 b b$
 - $c_0 b c_1 b c_2 b$
 - $c_0 c_1 c_2$
 - Accept

Formální příklad

- $LR_t := \{ c_0 w c_1 w c_2 \dots c_{t-1} w \mid w \in \{a, b\}^* \}$,
- t -sRL-automat M_t pro jazyk LR_t (Σ_0 označuje $\{a, b\}$):
 - (1) $(c c_0, a, \Sigma_0^* \cdot c_1, a, \Sigma_0^* \cdot c_2, \dots, \Sigma_0^* \cdot c_{t-1}, a, \Sigma_0^* \cdot \$)$,
 - (2) $(c c_0, b, \Sigma_0^* \cdot c_1, b, \Sigma_0^* \cdot c_2, \dots, \Sigma_0^* \cdot c_{t-1}, b, \Sigma_0^* \cdot \$)$,
 - (3) $(c c_0 c_1 \dots c_{t-1} \$, \text{Accept})$.
- Ukázkový výpočet automatu M_3 :
 - $c_0 a b b c_1 a b b c_2 a b b$
 - $c_0 b b c_1 b b c_2 b b$
 - $c_0 b c_1 b c_2 b$
 - $c_0 c_1 c_2$
 - Accept

Formální příklad

- $LR_t := \{ c_0 w c_1 w c_2 \dots c_{t-1} w \mid w \in \{a, b\}^* \}$,
- t -sRL-automat M_t pro jazyk LR_t (Σ_0 označuje $\{a, b\}$):
 - (1) $(c c_0, a, \Sigma_0^* \cdot c_1, a, \Sigma_0^* \cdot c_2, \dots, \Sigma_0^* \cdot c_{t-1}, a, \Sigma_0^* \cdot \$)$,
 - (2) $(c c_0, b, \Sigma_0^* \cdot c_1, b, \Sigma_0^* \cdot c_2, \dots, \Sigma_0^* \cdot c_{t-1}, b, \Sigma_0^* \cdot \$)$,
 - (3) $(c c_0 c_1 \dots c_{t-1} \$, \text{Accept})$.
- Ukázkový výpočet automatu M_3 :
 - $c_0 a b b c_1 a b b c_2 a b b$
 - $c_0 b b c_1 b b c_2 b b$
 - $c_0 b c_1 b c_2 b$
 - $c_0 c_1 c_2$
 - Accept

Formální příklad

- $LR_t := \{ c_0 w c_1 w c_2 \dots c_{t-1} w \mid w \in \{a, b\}^* \}$,
- t -sRL-automat M_t pro jazyk LR_t (Σ_0 označuje $\{a, b\}$):
 - (1) $(c c_0, a, \Sigma_0^* \cdot c_1, a, \Sigma_0^* \cdot c_2, \dots, \Sigma_0^* \cdot c_{t-1}, a, \Sigma_0^* \cdot \$)$,
 - (2) $(c c_0, b, \Sigma_0^* \cdot c_1, b, \Sigma_0^* \cdot c_2, \dots, \Sigma_0^* \cdot c_{t-1}, b, \Sigma_0^* \cdot \$)$,
 - (3) $(c c_0 c_1 \dots c_{t-1} \$, \text{Accept})$.
- Ukázkový výpočet automatu M_3 :
 - $c_0 a b b c_1 a b b c_2 a b b$
 - $c_0 b b c_1 b b c_2 b b$
 - $c_0 b c_1 b c_2 b$
 - $c_0 c_1 c_2$
 - Accept

Formální příklad

- $LR_t := \{ c_0 w c_1 w c_2 \dots c_{t-1} w \mid w \in \{a, b\}^* \}$,
- t -sRL-automat M_t pro jazyk LR_t (Σ_0 označuje $\{a, b\}$):
 - (1) $(c c_0, a, \Sigma_0^* \cdot c_1, a, \Sigma_0^* \cdot c_2, \dots, \Sigma_0^* \cdot c_{t-1}, a, \Sigma_0^* \cdot \$)$,
 - (2) $(c c_0, b, \Sigma_0^* \cdot c_1, b, \Sigma_0^* \cdot c_2, \dots, \Sigma_0^* \cdot c_{t-1}, b, \Sigma_0^* \cdot \$)$,
 - (3) $(c c_0 c_1 \dots c_{t-1} \$, \text{Accept})$.
- Ukázkový výpočet automatu M_3 :
 - $c_0 a b b c_1 a b b c_2 a b b$
 - $c_0 b b c_1 b b c_2 b b$
 - $c_0 b c_1 b c_2 b$
 - $c_0 c_1 c_2$
 - Accept

Formální příklad

- $LR_t := \{ c_0 w c_1 w c_2 \dots c_{t-1} w \mid w \in \{a, b\}^* \}$,
- t -sRL-automat M_t pro jazyk LR_t (Σ_0 označuje $\{a, b\}$):
 - (1) $(c c_0, a, \Sigma_0^* \cdot c_1, a, \Sigma_0^* \cdot c_2, \dots, \Sigma_0^* \cdot c_{t-1}, a, \Sigma_0^* \cdot \$)$,
 - (2) $(c c_0, b, \Sigma_0^* \cdot c_1, b, \Sigma_0^* \cdot c_2, \dots, \Sigma_0^* \cdot c_{t-1}, b, \Sigma_0^* \cdot \$)$,
 - (3) $(c c_0 c_1 \dots c_{t-1} \$, \text{Accept})$.
- Ukázkový výpočet automatu M_3 :
 - $c_0 a b b c_1 a b b c_2 a b b$
 - $c_0 b b c_1 b b c_2 b b$
 - $c_0 b c_1 b c_2 b$
 - $c_0 c_1 c_2$
 - Accept

Formální příklad

- $LR_t := \{ c_0 w c_1 w c_2 \dots c_{t-1} w \mid w \in \{a, b\}^* \}$,
- t -sRL-automat M_t pro jazyk LR_t (Σ_0 označuje $\{a, b\}$):
 - (1) $(c c_0, a, \Sigma_0^* \cdot c_1, a, \Sigma_0^* \cdot c_2, \dots, \Sigma_0^* \cdot c_{t-1}, a, \Sigma_0^* \cdot \$)$,
 - (2) $(c c_0, b, \Sigma_0^* \cdot c_1, b, \Sigma_0^* \cdot c_2, \dots, \Sigma_0^* \cdot c_{t-1}, b, \Sigma_0^* \cdot \$)$,
 - (3) $(c c_0 c_1 \dots c_{t-1} \$, \text{Accept})$.
- Ukázkový výpočet automatu M_3 :
 - $c_0 a b b c_1 a b b c_2 a b b$
 - $c_0 b b c_1 b b c_2 b b$
 - $c_0 b c_1 b c_2 b$
 - $c_0 c_1 c_2$
 - Accept

Redukční analyzátor pro FGD :

t-sRL-automat M_{07} , jehož charakteristický slovník Σ je sjednocením čtyřech podslovníků $\Sigma_0, \dots, \Sigma_3$.

$L_C(M_{07})$ – *charakteristický jazyk analyzátoru M_{07} .*

$S_C(M_{07})$ – *jednoduchý jazyk analyzátoru M_{07}*

Požadované vlastnosti M_{07} a jeho kategorií, vyzorované ze závislostní redukční analýzy (manuální):

Řetězy z $S_C(M_{07})$ neobsahují více než předem odhadnutý počet (t) symbolů.

M_{07} zachovává korektnost.

M_{07} vypouští v každém cyklu alespoň jeden symbol z každého podslovníku.

Realizovatelné redukce v jednom řetězu w z $L_C(M_{07})$ jsou vzájemně disjunktní.

Nechť w, y jsou z $L_C(M_{07})$ a w je podřetězem y (i nesouvislým).

Pak M_{07} má umět (postupně) zredukovat y na w .

Opakování

Základní úkoly

- množinu **LL** správně utvořených vět daného přirozeného jazyka
- množinu **LM** významových (tektogramatických) reprezentací vět
- rozlišovací relaci **SH** mezi LL a LM popisující vztahy synonymie a homonymie

Projekce charakteristického jazyka do rovin

Projekce do vstupní (povrchové) roviny - Základní rys závislostních redukčních automatů

Vstupní rovina rozpoznávaná automatem M_{07} (LL) :

$L_0(M_{07})$ – získáme z $L_C(M_{07})$ odstraněním symbolů nepatřících do Σ_0 .

Tektogramatická (významová) rovina rozpoznávaná automatem M_{07} (LM) :

$L_3(M_{07})$ – získáme z $L_C(M_{07})$ odstraněním symbolů nepatřících do Σ_3 .

SH-relace

Rozlišovací (charakteristická) relace automatu M_{07} :

$$\text{SH}(M_{07}) :=$$

$\{(u, y) \mid u \in L_0(M_{07}) \ \& \ y \in L_3(M_{07}) \ \& \ \exists w \in L_C(M_{07}), \text{ kde } u \text{ vzniklo z } w \text{ vypuštěním symbolů nepatřících do } \Sigma_0 \text{ a } y \text{ vzniklo z } w \text{ vypuštěním symbolů nepatřících do } \Sigma_3\}.$

Sématicko-syntaktická syntéza

Nechť $y \in L_3(M_{07})$. **SH-syntéza** pro y :

$$\text{synt-SH}(M_{07}, y) = \{(u, y) \mid (u, y) \in \text{SH}(M_{07})\}$$

Sématicko-syntaktická analýza

Nechť $u \in L_0(M_{07})$. *SH-analýza* pro u :

$$\text{anal-SH}(M_{07}, u) = \{(u, y) \mid (u, y) \in \text{SH}(M)\}$$

Vlastnosti FGP (momentální stav)

$L_3(M_{07})$ – deterministický bezkontextový jazyk,
ex. redukční analyzátor, který zachovává korektnost a
neužívá pomocné symboly a pouze vypouští

$L_C(M_{07})$ – v deterministických kontextových jazycích,
relativně blízko deterministickým bezkontextovým
ex. restartovací automat, který zachovává korektnost a
nerozlišuje typy symbolů

$L_0(M_{07})$ – v kontextových jazycích, relativně blízko
bezkontextovým
ex. Chomského restartovací analyzátor, který
nezachovává korektnost a používá pomocné symboly

Typy teoretických výsledků poslední doby :

- Vliv omezujících vlastností redukci na charakteristické jazyky i ostatní typy jazyků.
- O vzdálenosti LL a LM od charakteristického jazyka.
- Učení redukčních systémů po meta-pravidlech a pomocí příkladů redukční analýzy (František Mráz, ..).
Připomíná postupnou stavbu FGD.
- Hledání lingvisticky interpretovatelných omezení pro restartovací automaty. Týká se hlavně vlastností slovosledu a valence.

Výhledy

- Studovat vlastnosti formálního překladu pomocí restartovacích automatů
- Integrovat závislostní gramatiky a restartovací automaty do jediného paradigmatu. Použít pro studium rozlišovacích relací.
- Naučit se pracovat současně s pozitivní i negativní informací.

Děkujeme