O roztržitosti a volnosti slovosledu pomocí restartovacích automatů*

Martin Plátek[1], František Mráz[1,2], Friedrich Otto[2], and Markéta Lopatková[1]

[1] MFF UK Praha, Malostranské nám. 25, 118 00 Praha 1, Česká Republika
[2] Fachbereich Mathematik/Informatik, Universität Kassel, 341 09 Kassel, Německo

Abstrakt. Zavádíme nový typ restartovacích automatů, abychom získali prostředek umožňující klasifikaci jevů souvisejících s valencí a slovosledem u přirozených jazyků. Studujeme dva základní typy omezení výpočtů: *j-roztržitost* interpretujeme jako míru nesouvislosti valencí (redukcí) a *j-volnost* jako míru volnosti slovosledu. Ukazujeme, že roztržitost lze měřit stejně u jazyků s volným slovosledem (vysokým stupněm volnosti) jako u jazyků s pevným slovosledem (nízkým stupněm volnosti). Přínos vidíme i v exaktním rozlišení mezi roztržitostí a volností slovosledu.

1 Úvod

Z teoretického pohledu jsou restartovací automaty flexibilním zobecněním analytických gramatik (např. frázových, kategoriálních, čistých a závislostních gramatik). Základní motivací pro studium restartovacích automatů je modelování redukční analýzy vět přirozených jazyků. Každý restartovací automat definuje *syntaktický systém*, který je formalizací redukční analýzy [5]. Syntaktické systémy nesou přímo informaci o slovníku (abecedě) vstupních slov (symbolů), seznamu (slovníku) kategorií, jazyce větných forem (charakteristickém jazyce), systému redukcí a zprostředkovaně i o vstupním (povrchovém, terminálním) jazyce. Ve schopnosti rozlišovat a popisovat syntaktické systémy přirozených jazyků se projevuje lingvistická kompetence mluvčího. V článku navrhujeme nový typ restartovacích automatů umožňující adekvátní klasifikaci jevů souvisejících s valencí a slovosledem v přirozených jazycích (více viz 1.1).

V technické části ukazujeme nekonečnou škálu tříd syntaktických systémů a tříd jazyků definovanou pomocí restartovacích automatů a vhodných omezení jejich výpočtů. Naším hlavním cílem je formalizovat stupně volnosti slovosledu. Podobnou problematiku jsme řešili již dříve [3,4]. Využívali jsme závislostní gramatiky a od nich odvozené pojmy. Restartovací automaty umožňují univerzálnější postup. Tento postup může být uplatněn (nepřímo) na různé typy gramatik, např. na frázové gramatiky Chomského typu, kategoriální, Marcusovy, čisté i závislostní gramatiky. Jim odpovídající analyzátory lze přímočaře konstruovat jako restartovací automaty. Syntaktické systémy takto získané dovolují sílu

* První, druhý autor a čtvrtá autorka byli částečně podporovaní GA ČR, grant č. 201/04/2102 a programem 'Informační společnost', projekt č. 1ET100300517. Druhý a třetí autor byli podporování grantem Deutsche Forschungsgemeinschaft.

těchto prostředků porovnávat (z lingvistického pohledu) výrazně adekvátněji než jen pomocí prosté generativní síly.

Omezujeme se na automaty, jejichž výpočetní síla není větší než lineárně omezené automaty. *Volně restartující automat*, krátce FRR-automat, je zařízení s jednou (pružnou) páskou, konečnou řídící jednotkou a pracovním oknem pevné velikosti. Každý výpočet je složen z posloupnosti *cyklů* a je uzavřen *koncovkou.* Během cyklu dochází ke zjednodušení obsahu pásky přepisováním. Počet přepsání v cyklu není v základním modelu omezen.

V práci jsou zavedeny a studovány dva základní typy omezení výpočtů. Tzv. j*-roztržitost* interpretujeme jako míru složitosti slovosledu, j*-volnost* interpretujeme jako míru volnosti slovosledu. Ukážeme, že roztržitost lze měřit stejně u jazyků s volným slovosedem (vysokým stupněm volnosti) jako u jazyků s pevným slovosledem (nízkým stupněm volnosti). Důležitým terminologickým přínosem práce je exaktní rozlišení mezi složitostí (roztržitostí) a volností slovosledu v jazykových jevech.

1.1 Lingvistická motivace

V této podsekci přímo navazujeme na text o redukční analýze [5]. Ilustrujme složitost slovosledu na dobře známé holandské konstrukci (příklad 1) s valenčním slovesným doplněním v infinitivu (tuto konstrukci nelze adekvátně popsat kontextovými gramatikami [7]). Holandština je považována za jazyk s pevným slovosledem, kde pozice jednotlivých větných členů je pevně dána, tj. syntaktické vztahy jsou určeny slovosledem.

Příklad 1 *Ik denk dat ik Cecilia de nijlpaarden zag voeren.*
(já-myslím-že-já-Cecilia-(člen)-hroši-viděl-krmit)
(Myslím, že jsem viděl Cecilii krmit hrochy.)

Obrázek 1.	*Ik denk dat ik Cecilia* `de nijlpaarden` *zag* `voeren`.
$\longrightarrow$	*Ik denk* `dat ik Cecilia zag` $\mathtt{VG}_{inf}$.
$\longrightarrow$	*Ik denk* $\mathtt{VG}_{dat}$.

Pokud chceme tuto větu redukovat, postupujeme způsobem zachyceným na obr. 1 (redukovaná slova jsou zvýrazněna). V prvním kroku jsou zpracována slova *de* (člen), *nijlpaarden* (hroši) a *voeren* (krmit) (jde o sloveso a jeho valenční doplnění; povšimněme si, že přepisovaná slova netvoří 'spojitou komponentu'). Tato slova jsou přepsána slovem VG_{inf}, které patří do seznamu kategorií (zachycuje skutečnost, že byla vypuštěna infinitivní konstrukce). Ve druhém kroku redukce pokračuje přepsáním slov *dat* (že), *ik* (já), *Cecilia*, *zag* (viděl) a VG_{inf} kategoriálním slovem VG_{dat} (označuje vypuštěnou závislou klauzi uvozenou spojkou *dat* (že); tato závislá klauze je opět tvořena slovesem a jeho valenčními doplněními).

Podobné konstrukce, kde je adekvátní redukovat v jednom kroku slova, která se nacházejí na 'různých místech' věty, jsou poměrně běžné například i v češtině (příklad 2), která nám slouží jako zástupce jazyků s volným slovosledem – syntaktické vztahy jsou zde dány převážně jinými prostředky než pozicí ve větě (v češtině zejména bohatou flexí).

Příklad 2 *Petr na otázku nenašel odpověď.*

Obrázek 2. *Petr* **na otázku** *nenašel* **odpověď.**
$\longrightarrow$ *Petr nenašel* NG_{acc}.

Redukce této věty je znázorněna na obr. 2. V redukčním kroku jsou přepsána slova *na, otázku* a *odpověď* kategoriálním slovem NG_{acc} (zachycuje skutečnost, že byla vypuštěna jmenná skupina realizující valenční doplnění slovesa, jejíž 'hlavu' tvoří podstatné jméno v akuzativu; tato jmenná skupina je tvořena substantivem a jeho valenčním doplněním). Stejně bychom mohli postupovat pro skoro všechny permutace této věty.

Poznamenejme zde, že v [4] jsou ukázány české konstrukce, kde je vhodné redukovat větu na více místech. Počet přepisovacích pozic je v čekých větách v principu omezen počtem povinných valenčních pozic sloves a substantiv.

V sekci 3 se budeme věnovat dvěma typům formálních jazyků, které reprezentují jednak jazyky s valencemi modelovanými pomocí redukcí na pevně určených pozicích, jednak jazyky s valencemi modelovanými pomocí redukcí symbolů na proměnlivých pozicích, tedy nezávisle na slovosledu.

2 Definice

Pojmy budeme zavádět nepříliš formálně. Spoléháme přitom u čtenáře na jisté zkušenosti z teorie formálních jazyků.[3]

Volně restartující automat, krátce FRR-automat, je (nedeterministický) automat M s konečnou řídící jednotkou reprezentovanou konečnou množinou stavů Q a konečnou přechododovou relací (tabulkou) T, konečným vstupní slovníkem I, konečným kategoriálním slovníkem (abecedou) C a pracovním oknem velikosti k, kde $k \geq 1$. M pracuje na (pružné) pásce s levou zarážkou ¢ a pravou zarážkou \$. Pro vstupní větu $w \in I^*$ je vstupním zápisem na pásce ¢w\$. M začíná výpočet ve svém startovním stavu q_0 s oknem na začátku pásky, ve výhledu s levou zarážkou ¢ a prvními $k-1$ symboly w. Řízen svojí přechodovou tabulkou, M na základě aktuáního stavu q a obsahu okna u může provádět následující typy kroků:

(a) *posun doprava* – posune okno o jednu pozici doprava a změní stav,
(b) *přepisovací krok* – nahradí obsah okna u řetězem v, změní stav a posune okno na pravého souseda řetězu v. Navíc požadujeme, aby v bylo *zjednodušením* u, tedy aby obsahovalo méně vstupních symbolů než u, nebo aby obsahovalo stejný počet vstupních symbolů jako u, ale bylo kratší než u.
(c) na pravém konci pásky se M buď zastaví a ohlásí *přijmutí*, nebo se zastaví a ohlásí *odmítnutí*, nebo *restartuje*, což znamená, že přemístí okno na levý konec pásky a přepne se znovu do startovního stavu.

[3] Abychom předcházeli nedorozuměním v lingvistické interpretaci navrhovaných pojmů, používáme často původní Chomského terminologii. Tedy např. *slovník* na místě abecedy, *věta* na místě slova.

Konfigurace automatu M je řetěz $\alpha q \beta$ kde $q \in Q$, a buď $\alpha = \lambda$ a $\beta \in \{¢\} \cdot (I \cup C)^* \cdot \{\$\}$, nebo $\alpha \in \{¢\} \cdot (I \cup C)^*$ a $\beta \in (I \cup C)^* \cdot \{\$\}$; q reprezentuje momentální stav, $\alpha\beta$ je momentální obsah pásky, a je konvencí, že okno obsahuje prvních k symbolů β, nebo celé β, pokud $|\beta| \leq k$. *Restartovací konfigurace* je tvaru $q_0 ¢ w \$$, kde $w \in (I \cup C)^*$.

Libovolný výpočet M sestává z fází. Fáze, zvaná *cyklus*, začíná restartovací konfigurací, hlava se pohybuje doprava prováděním posunů doprava a přepisovacími kroky do chvíle, kdy je proveden restart, a tudíž nastane nová restartovací konfigurace. V případě, že fáze nekončí restartem, nazýváme ji *koncovkou*, neboť nutně končí zastavením automatu. Požadujeme, aby každý cyklus automatu M provedl alespoň jedno přepsání – tedy každá nová fáze začíná na jednodušším řetězu než ta předchozí. Používáme notaci $u \vdash_M^c v$ pro označení cyklu M, který začíná v restartovací konfiguraci $q_0 ¢ u \$$ a končí restartovací konfigurací $q_0 ¢ v \$$; relace $\vdash_M^{c^*}$ je reflexivním a transitivním uzávěrem $\vdash_M^c$. Pár $\mathsf{RS}(M) := ((I \cup C)^*, \vdash_M^c)$ je nazýván *redukčním systémem* indukovaným M.

Větná forma $w \in (I \cup C)^*$ *je přijímána pomocí* M, pokud existuje přijímající výpočet startující z $q_0 ¢ w \$$. Jako $L_{IC}(M)$ označujeme jazyk větných forem přijímaných (rozpoznávaných) pomocí M. Říkáme, že $L_{IC}(M)$ je *charakteristický jazyk* rozpoznávaný M. Jazyk $L_I(M) := L_{IC}(M) \cap I^*$ všech vstupních vět přijímaných pomocí M je *vstupní jazyk rozpoznávaný* M. Čtveřici $BS(M) := (I, C, L_{IC}(M), \mathsf{RS}(M))$ nazýváme *syntaktickým systémem* automatu M. Syntaktické systémy reprezentují lingvistickou kompetenci, tedy znalost větné stavby, odpovědnost za adekvátně navržený systém (zjednoznačňujících) kategorií a umění interpretovat jejich souvislost s větnou stavbou. Větná stavba je v syntaktických systémech reprezentována (hlavně) redukčním systémem. Povšimněme si, že $BS(M)$ obsahuje úplnou informaci o $L_I(M), L_{IC}(M)$ a $\mathsf{RS}(M)$ a že $L_{IC}(M)$ přebírá významnější roli mezi jazyky reprezentovanými syntaktickým systémem. Poznamenejme ještě, že zcela analogicky lze zavést syntaktické systémy pro Chomského frázové grammatiky i pro nejrůznější typy kategoriálních, závislostních či jiných formálních gramatik. Tato skutečnost umožňuje porovnávat ('explikativní') sílu různých typů gramatik či analyzátorů pomocí syntaktických systémů.

Následující vlastnost FRR-automatů využíváme často implicitně v důkazech. Její zásluhou jsou FRR-automaty vhodným modelem redukční analýzy.

Fakt 1. (Zachovávání správnosti). *Nechť M je* FRR-*automat a nechť $u, v \in (I \cup C)^*$. Pokud $u \vdash_M^{c^*} v$ je částí přijímacího výpočtu M, potom $v \in L_{IC}(M)$.*

Dále se budeme věnovat omezením relevantním pro naše výpočty. Nejprve zavedeme pojem *monotonie.* Nechť $C := \alpha q \beta$ je *přepisovací konfigurace*, tedy konfigurace, v níž lze uplatnit přepisovací instrukci. Potom $|\beta|$ je *pravou distancí* C, kterou označíme $D_r(C)$. *Posloupnost přepisovacích konfigurací* $S = (C_1, C_2, \ldots, C_n)$ nazýváme *monotonní*, pokud $D_r(C_1) \geq D_r(C_2) \geq \ldots \geq D_r(C_n)$.

Nechť j je přirozené číslo. Budeme říkat, že posloupnost přepisovacích konfigurací $S = (C_1, C_2, \ldots, C_n)$ je *j-monotonní*, pokud existuje rozklad S do j

podposloupností

$$S_1 = (C_{1,1}, C_{1,2}, \cdots, C_{1,p_1}), \ldots, S_j = (C_{j,1}, C_{j,2}, \cdots, C_{j,p_j}),$$

takový, že každá S_i, $1 \leq i \leq j$ je monotonní. Povšimněme si, že posloupnost přepisovacích konfigurací $(C_1, C_2, \cdots, C_n)$ *není* j-monotonní právě tehdy, když existují indexy $1 \leq i_1 < i_2 < ... < i_{j+1} \leq n$ takové, že $D_r(C_{i_1}) < D_r(C_{i_2}) < \ldots < D_r(C_{i_{j+1}})$.

Výpočet FRR-automatu M je nazýván *j-monotonní*, pokud odpovídající posloupnost přepisovacích konfigurací je j-monotonní. *Výpočet* je *j-přepisující*, pokud žádný z jeho cyklů neobsahuje více než j přepisovacích kroků. Nakonec, *výpočet* je *j-roztržitý* pokud je jak j-přepisující, tak j-monotonní.

Notace. Pro libovolnou třídu A automatů a libovolný index $Y \in \{I, IC\}$ bude $\mathcal{L}_Y(\mathsf{A})$ označovat třídu Y-jazyků rozpoznatelných automaty z A a $\mathcal{BS}(\mathsf{A})$ bude označovat třídu syntaktických systémů určených A. Jako (D)CFL budeme označovat třídu (deterministických) bezkontextových jazyků, pomocí $|w|_b$ počet výskytů symbolu b v řetězu w a jako $\subset$ budeme označovat vlastní inkluzi. Občas budeme užívat regulární výrazy místo regulárních jazyků, λ značí prázdnou větu a $\mathbb{N}_+$ značí množinu kladných celých čísel.

3 Vlastnosti slovosledu

V této sekci studujeme omezení, která reprezentují složitost a volnost slovosledu formálních jazyků, nebo přesněji jejich syntaktických systémů. Volnost slovosledu syntaktického systému znamená odolnost proti permutacím jejich větných forem a proti redukcím. Zaměříme se hlavně na studium syntaktických systémů a charakteristických jazyků, abychom zdůraznili jejich význam pro modelování lingvistické kompetence.

Definice 2. *Pro* FRR-*automat* M, $i \in \mathbb{N}_+$ *a* $Y \in \{I, IC\}$ *zavádíme:*

$$L_Y(M,i) := \{\, w \in L_Y(M) \mid M \textit{ přijímá } w \textit{ pomocí } i\textit{-roztržitého výpočtu} \},$$
$$BS(M,i) := (I, C, L_{IC}(M,i), \mathsf{RS}(M)), \;\textit{pokud}\; BS(M) = (I, C, L_{IC}(M), \mathsf{RS}(M)).$$

Zřejmě platí, že pro každý FRR-automat M, každé $Y \in \{I, IC\}$ a každé $i \geq 1$ $L_Y(M,i) \subseteq L_Y(M,i+1)$ a $BS(M,i) \subseteq BS(M,i+1)$.

Definice 3. *Nechť* $i \in \mathbb{N}_+$, $Y \in \{I, IC\}$ *a* $\mathcal{A}$ *označuje (pod)typ* FRR-*automatu. Budeme značit:*

$$\mathcal{L}_Y(i, \mathcal{A}) := \{\, L_Y(M,i) \mid M \textit{ je } \mathcal{A}\textit{-automat} \},$$
$$\mathcal{BS}(i, \mathcal{A}) := \{\, BS(M,i) \mid M \textit{ je } \mathcal{A}\textit{-automat} \}.$$

Následující věta vznikla pouhým přeformulováním výsledků z [6].

Věta 1. $\mathsf{DCFL} \subset \mathcal{L}_{IC}(1, \mathsf{FRR}) \subset \mathsf{CFL} = \mathcal{L}_I(1, \mathsf{FRR})$.

Na tomto místě je užitečné si uvědomit, že $\mathcal{BS}(1, \mathsf{FRR})$ obsahuje jako vlastní podtřídy třídy syntaktických systémů bezkontextových grammatik (bez vypouštěcích pravidel) či (standardních) kategoriálních grammatik.

Nyní (z praktických důvodů) zavedeme omezený typ FRR-automatu, který budeme označovat jako MRR-*automat*. Děláme to proto, že přechodová relace (tabulka) MRR-automatu M se zadává pomocí konečné posloupnosti *meta-instrukcí* tvaru

$$(E_1, u_1 \to v_1, E_2, u_2 \to v_2, E_3, \ldots, E_i, u_i \to v_i, E_{i+1}),$$

kde $E_1, \ldots, E_{i+1}$ jsou regulární výrazy a pro každé $n = 1, \ldots, i$, $u_n, v_n \in (I \cup C)^*$ jsou věty takové, že v_n je zjednodušením u_n. Pravidla $u_1 \to v_1$, $u_2 \to v_2$, $\ldots$, $u_i \to v_i$ ztělesňují přepisovací kroky M. Při pokusu o provedení této meta-instrukce z konfigurace $q_0 ¢ w \$$ M neuspěje (a tedy skončí odmítnutím), pokud nebude w umožňovat faktorizaci tvaru $w = w_1 u_1 w_2 u_2 \ldots w_i u_i w_{i+1}$ takovou, že $¢ w_1 \in L(E_1)$, $w_2 \in L(E_2)$, $\ldots$, $w_{i+1}\$ \in L(E_{i+1})$, kde $L(E_n)$ označuje jazyk popsaný regulárním výrazem E_n. Na druhou stranu, pokud w má nějaké faktorizace tohoto tvaru, pak jedna taková faktorizace je vybrána nedeterministicky a $q_0 ¢ w \$$ je transformována do $q_0 ¢ w_1 v_1 w_2 v_2 w_3 \ldots w_i v_i w_{i+1} \$$. Abychom popsali koncovky přijímacích výpočtů M, používáme meta-instrukce tvaru $(¢ \cdot E \cdot \$, \mathsf{Accept})$, které přijímají věty z regulárního jazyka $L(E)$.

Pro každý MRR-automat M existuje číslo j_M takové, že žádný cyklus libovolného výpočtu M neobsahuje více než j_M přepisovacích kroků, to jest, MRR-automaty mají *omezené přepisování*. V tomto smyslu jsou MRR-automaty slabší než FRR-automaty.

Příklad 3. *Nechť M_1 je* MRR-*automat s $\{a, b\}$ jako vstupním i kategoriálním slovníkem (abecedou), který je zadán následujícími třemi meta-instrukcemi:*

(1) $(¢ \cdot a^+, abb \to b, b^* \cdot \$)$,
(2) $(¢ \cdot a^+, abb \to b, b^* \cdot a^+, abb \to b, b^* \cdot \$)$,
(3) $(¢ \cdot (ab + abab) \cdot \$,\ \mathsf{Accept})$.

Snadno nahlédneme, že $L_{IC}(M_1, 1) = \{\, a^n b^n \mid n > 0 \,\} \cup \{abab\}$, $L_{IC}(M_1, 2) = L_{IC}(M_1, 1) \cup \{\, a^n b^n a^n b^n \mid n > 0 \,\}$ a $L_{IC}(M_1) = L_{IC}(M_1, 2) \supset L_{IC}(M_1, 1)$.

Poznámka 1. $L_{IC}(M_1) \notin \mathsf{CFL}$ *a podle Věty 1 platí, že* $L_{IC}(M_1) \notin \mathcal{L}_{IC}(1, \mathsf{FRR})$.

Příklad 4. *Nechť M_2 je* MRR-*automat se stejnými slovníky jako M_1, který je zadán pomocí meta-instrukcí (1)–(3) z M_1 a následujících dvou meta-instrukcí:*

(4) $(¢ \cdot a^+, abb \to b, b^* \cdot ab \cdot \$)$,
(5) $(¢ \cdot ab \cdot a^+, abb \to b, b^* \cdot \$)$.

Platí, že $L_{IC}(M_2, 1) = \{\, a^n b^n, aba^n b^n, a^n b^n ab \mid n > 0 \,\}$, $L_{IC}(M_2, 2) = \{\, a^n b^n a^m b^m \mid n > 0,\ m \geq 0 \,\}$ a $L_{IC}(M_2) = L_{IC}(M_2, 2)$.

Oproti $L_{IC}(M_1)$ je jazyk $L_{IC}(M_2)$ bezkontextový a $L_{IC}(M_2)$ dokonce může být rozpoznáván pomocí MRR-automatu M' takového, že využívá pouze 1-roztržitých (tedy neroztržitých) výpočtů, tudíž $L_{IC}(M_2) \in \mathcal{L}_{IC}(1, \mathsf{FRR})$. M'

je zadán pomocí následujících meta-instrukcí

$$\begin{array}{ll}(1)\ (\text{¢} \cdot a^+, abb \to b, b^* \cdot \$), & (3)\ (\text{¢} \cdot a^+, abb \to b, b^* \cdot a^+ \cdot b^+ \cdot \$),\\ (2)\ (\text{¢} \cdot ab \to \lambda, a^+ \cdot b^+ \cdot \$), & (4)\ (\text{¢} \cdot ab \cdot \$,\ \mathsf{Accept}).\end{array}$$

Dále budeme využívat následující příklady jazyků, kde $j \in \mathbb{N}_+$:

$$LC_j := \{ c_0wc_1w \dots c_iw \dots c_{j-2}wc_{j-1}w \mid w \in \{a,b\}^* \},$$

$$LE_j := \{ w \in \{a_1, ..., a_j\}^* \mid |w|_{a_1} = |w|_{a_2} = ... = |w|_{a_j} \}.$$

Poznamenejme, že formální jazyky LC_j $(j \geq 1)$ jsou paralelou jevů v přirozených jazycích, kde jsou valence modelovány pomocí redukcí j-tic symbolů v pevně určených pozicích (s rostoucím počtem takových pozic), viz příklad 1. Formální jazyky LE_j $(j \geq 1)$ modelují případy, kdy jsou valence modelovány pomocí redukcí j-tic symbolů ve zcela volně umisťovaných pozicích, viz příklad 2.

Příklad 5. *Nechť $j \geq 1$ a MC_j je* MRR-*automat s $\Sigma_j := \{a, b, c_0, c_1, \dots, c_{j-1}\}$ jako vstupním i kategoriálním slovníkem a MC_j je zadán následující posloupností meta-instrukcí, kde $\Sigma_0 := \{a, b\}$:*

$$\begin{array}{l}(1)\ (\text{¢}c_0, a \to \lambda, \Sigma_0^* \cdot c_1, a \to \lambda, \Sigma_0^* \cdot c_2, \dots, \Sigma_0^* \cdot c_{j-1}, a \to \lambda, \Sigma_0^* \cdot \$);\\ (2)\ (\text{¢}c_0, b \to \lambda, \Sigma_0^* \cdot c_1, b \to \lambda, \Sigma_0^* \cdot c_2, \dots, \Sigma_0^* \cdot c_{j-1}, b \to \lambda, \Sigma_0^* \cdot \$);\\ (3)\ (\text{¢}c_0c_1 \dots c_{j-1}\$, \mathsf{Accept}).\end{array}$$

Je zřejmé, že platí $L_I(MC_j) = L_{IC}(MC_j) = LC_j$ a že MC_j je j-roztržité. Není těžké nahlédnout, že $LC_{j+1} \notin \mathcal{L}_{IC}(j, \mathsf{FRR})$, jelikož pro větu $x \in LC_{j+1}$ nestačí (v obecném případě) j přepisovacích operací v jednom cyklu na transformaci x na kratší větu patřící do LC_{j+1}. Tedy LC_{j+1} nemůže být charakteristickým jazykem pro žádný j-roztržitý FRR-automat.

Na druhou stranu pro každé $j \geq 1$ existuje FRR-automat $\overline{M}_j$ se vstupním slovníkem Σ_j a s odlišným kategoriálním slovníkem takový, že $L_I(\overline{M}_j, 2) = LC_j$. Uvažujme vstup tvaru $c_0w_1c_1w_2 \dots c_{j-2}w_{j-1}c_{j-1}w_j$ s $w_1, \dots, w_j \in \{a,b\}^*$. $\overline{M}_j$ postupně porovnává sousední faktory w_i a w_{i+1} $(1 \leq i \leq j-1)$ zleva doprava, k čemuž mu postačí dva průchody přepisů zleva doprava pro každý z těchto faktorů. Z toho vyplývá, že $\overline{M}_j$ je 2-roztržitý.

Tento příklad ilustruje fakt, že jazyk, pokud je posuzován jako vstupní jazyk nějakého FRR-automatu, může být klasifikován podstatně níže v hierarchii než v případě, kdy je posuzován jako charakteristický jazyk nějakého FRR-automatu. Důvodem tohoto jevu je skutečnost, že pro vstupní jazyky není zaručeno zachovávání správnosti (příslušnosti k jazyku) během přijímacích výpočtů.

Věta 2.

(a) $\mathcal{L}_{IC}(\mathsf{FRR}) \subset \mathcal{L}_I(\mathsf{FRR})$, *a pro všechna* $i \in \mathbb{N}_+$,
(b) $\mathcal{L}_{IC}(i, \mathsf{FRR}) \subset \mathcal{L}_I(i, \mathsf{FRR})$,
(c) $\mathcal{L}_{IC}(i, \mathsf{FRR}) \subset \mathcal{L}_{IC}(i+1, \mathsf{FRR}) \subset \mathcal{L}_{IC}(\mathsf{FRR})$,
(d) $\mathcal{BS}(i, \mathsf{FRR}) \subset \mathcal{BS}(i+1, \mathsf{FRR}) \subset \mathcal{BS}(\mathsf{FRR})$.

Zavádíme pojmy, které umožní formalizovat rozpětí volnosti slovosledu u syntaktických sytémů a charakteristických jazyků FRR-automatů. Jazyky LE_j reprezentují jazyky s rostoucí volností slovosledu.

Definice 4.
(a) *Dvě věty* $u, v \in \{a_1, ..., a_j\}^*$ *nazýváme* Parikhovsky equivalentní *a značíme* $u \equiv v$, *pokud* $|u|_{a_i} = |v|_{a_i}$ *platí pro všechna* $1 \leq i \leq j$.
(b) *Nechť* M *je* FRR-*automat a nechť* $u, v \in L_{IC}(M)$. *Budeme říkat, že věta* u *je* M-transformovatelná *do* v, *což označujeme pomocí* $u \equiv>_M v$, *pokud* $u \equiv v$ *a pro každé* u_1 *splňující* $u \vdash^c_M u_1$, *kde* $|u| > |u_1|$, *existuje* v_1 *takové, že* $v \vdash^{c^*}_M v_1$ *a* $u_1 \equiv v_1$.
(c) FRR-*automat* M *je* redukčně úplný, *pokud pro každé* $i > 0$ *a každé* $u \in L_{IC}(M, i+1)$, *existuje věta* $v \in L_{IC}(M, i)$ *taková, že* $u \equiv>_M v$.

Příklad 6. MRR-*automat* M_1 *z příkladu 3 je redukčně úplný. Vezměme* $n \geq 2$ *a* $u := a^n b^n a^n b^n \in L_{IC}(M_1, 2) \setminus L_{IC}(M_1, 1)$. *Vyberme* $v := a^{2n} b^{2n} \in L_{IC}(M_1, 1)$, *potom platí* $u \equiv>_{M_1} v$. *Je totiž vidět, že ze vztahů* $u \equiv v$ *a* $u \vdash^c_{M_1} u_1$ *vyplývá* $u_1 = a^{n-1} b^{n-1} a^{n-1} b^{n-1}$. *Dále je vidět, že* $v_1 := a^{2n-2} b^{2n-2} \in L_{IC}(M_1, 1)$ *splňuje podmínky* $v \vdash^{c^*}_{M_1} v_1$ *a* $v_1 \equiv u_1$.

Definice 5. *Redukčně úplný* FRR-*automat* M *je nazýván* j-volným *pro nějaké* $j \in \mathbb{N}_+$, *pokud pro každé* $i = 1, \ldots, j-1$, $L_{IC}(M, i+1) \notin \mathcal{L}_{IC}(i, \mathsf{FRR})$. *Jazyk* L_{IC}, *resp. syntaktický systém* BS *je* j-volný *pokud existuje* j-*volný* FRR-*automat* M *takový, že* $L_{IC} = L_{IC}(M)$, *resp.* $BS = BS(M)$.

Pojem volnosti slouží pro porovnávání volnosti slovosledu.

Tvrzení 6. *Pokud* FRR-*automat* M *je* j-*volný pro nějaké celé číslo* $j > 1$, *pak* M *je také* $(j-1)$-*volný.*

Příklad 7. MRR-*automat* M_1 *z příkladu 3 je 2-volný, ale není 3-volný.* MRR-*automat* M_2 *z příkladu 4 je 1-volný, ale není 2-volný.*

Příklad 8. *Jazyk* LC_j *patří do třídy* $\mathcal{L}_{IC}(j, \mathsf{FRR})$, *ale není* j-*volný pro libovolné* $j \geq 2$. *Například, pokud* M *je* MRR-*automat takový, že* $LC_j = L_{IC}(M)$, *potom* $L_{IC}(M, i)$ *je konečný pro všechna* $1 \leq i < \lceil j/2 \rceil$. *Povšimněme si pro zajímavost, že podjazyk* $\{c_0 a^n c_1 a^n \ldots c_{j-1} a^n \mid n \geq 0\}$ *jazyka* LC_j *je obsažen ve třídě* $\mathcal{L}_{IC}(\lceil j/2 \rceil, \mathsf{FRR})$.

Tvrzení 7. *Pro* $j \geq 2$ *je jazyk* LE_j $(j-1)$-*volný, ale není* j-*volný.*

Důkaz. Nechť $j \geq 2$ a M_j je (deterministický) MRR-automat s oběma slovníky rovnými $\Sigma_j := \{a_1, \ldots, a_j\}$, který je definován pomocí meta-instrukcí následujícího tvaru, kde π může být libovolnou permutací množiny indexů $\{1, 2, \ldots, j\}$ a i může nabývat libovolné hodnoty z množiny indexů $3, \ldots, j-1$:

$$\begin{aligned}
&(1.\pi)\ (\text{¢} \cdot a^*_{\pi(1)}, a_{\pi(1)} a_{\pi(2)} \to \lambda, \{a_{\pi(1)}, a_{\pi(2)}\}^*, a_{\pi(3)} \to \lambda, \\
&\qquad\qquad \ldots, (\Sigma_j \setminus \{a_{\pi(j)}\})^*, a_{\pi(j)} \to \lambda, \Sigma^*_j \cdot \$); \\
&(2.i)\ (\text{¢} \cdot a^*_2, a_2 a_1 \to \lambda, a^*_1, a_3 \to \lambda, a^*_3 \cdot a^*_1, a_4 \to \lambda, a^*_4 \cdot a^*_1, \\
&\qquad\qquad \ldots, a^*_i \cdot a^*_1, a_{i+1} \ldots a_j \to \lambda, (a_{i+1} \ldots a_j)^* \cdot \$); \\
&(3)\quad (\text{¢} \cdot \$, \mathsf{Accept}).
\end{aligned}$$

Pomocné tvrzení 1. $L_{IC}(M_j) = LE_j$ a každý výpočet M_j je $(j-1)$-roztržitý.

Důkaz. Pokud obsah pásky má tvar

$$w(i) := a_2^n a_1^{m_1} a_3^n a_1^{m_2} a_4^n \dots a_i^n a_1^{m_{i-1}} (a_{i+1} \dots a_j)^n$$

pro nějaká celá čísla $n, m_1, \dots, m_{i-1} > 0$, potom lze uplatnit meta-instrukci (2.i). Opakováním takovéto situace může M_j dospět k přijetí $w(i)$, právě když $w(i) \in LE_j$. Odpovídající přijímací výpočet je zřejmě i-roztržitý. Pokud obsah pásky má jiný tvar, pak M_j může uplatnit meta-instrukce typu (1.π) a tedy v odpovídajícím cyklu M_j používá přesně $j-1$ vypouštěcích operací. Pokud sestavíme přepisovací (vypouštěcí) konfigurace výpočtu M_j do $j-1$ podposloupností takovým způsobem, že i-tá vypouštěcí konfigurace každého cyklu je uložena do i-té podposloupnosti pro všechna $i = 1, \dots, j-1$, pak lze nahlédnout, že všechny $(j-1)$ takto získané podposloupnosti jsou monotonní. Vidíme také, že M_j rozpoznává LE_j. □

Můžeme se snadno přesvědčit, že M_j je redukčně úplný. Následujícím výsledkem lze uzavřít důkaz tvrzení 7.

Pomocné tvrzení 2. Pro všechna $i \in \{2, \dots, j-1\}$, $L_{IC}(M_j, i) \notin \mathcal{L}_{IC}(i-1, \mathsf{FRR})$.

Důkaz. Uvažujme vstup tvaru $w(i)$ (viz výše) takový, že $n = m_1 + \dots + m_{i-1}$, a všechna čísla $m_1, \dots, m_{i-1}$ jsou dostatečně velká. Abychom převedli $w(i)$ na kratší větu také patřící do $L_{IC}(M_j, i)$, tak nejméně jeden výskyt každého symbolu $a_1, a_2, \dots, a_j$ musí být vypuštěn. Pokud čísla $m_2, \dots, m_{i-1}$ jsou větší než velikost okna uvažovaného automatu, musí automat provést minimálně i operací vypuštění, abychom vypustili alespoň po jednom výskytu každého symbolu. □

□

Následující věta je přímým důsledkem předchozího tvrzení. Jednotlivá tvrzení ukazují, že existují nekonečné škály syntaktických systémů a charakteristických jazyků s ohledem na stupně volnosti i stupně roztržitosti slovosledu. Ukazuje navíc, že roztržitost slovosledu má smysl měřit při libovolném stupni volnosti slovosledu.

Věta 3. *Pro všechna* $i, j \in \mathbb{N}_+, i < j$,
(a) $\mathcal{L}_{IC}((j+1)\text{-volný-FRR}) \subset \mathcal{L}_{IC}(j\text{-volný-FRR})$.
(b) $\mathcal{BS}((j+1)\text{-volný-FRR}) \subset \mathcal{BS}(j\text{-volný-FRR})$.
(c) $\mathcal{L}_{IC}(i, j\text{-volný-FRR}) \subset \mathcal{L}_{IC}(i+1, j\text{-volný-FRR})$.
(d) $\mathcal{BS}(i, j\text{-volný-FRR}) \subset \mathcal{BS}(i+1, j\text{-volný-FRR})$.

4 Závěrečné poznámky

V příspěvku jsme zavedli koncept *j-roztržitosti*, který slouží jako míra složitosti slovosledu (jednotlivých syntaktických jevů) přirozených jazyků, zatímco pojem *j-volnosti* vypovídá o stupni volnosti slovovsledu. Ukázali jsme, že míra složitosti slovosledu může sloužit jako vhodný parametr výpočtu pro restartovací automaty, což se podstatným způsobem promítá do konstrukcí syntaktických analyzátorů přirozených jazyků, a to zejména jazyků s volným slovosledem.

Viděli jsme, že složitost slovosledu může být způsobena jak jazykovými jevy založenými na pevné pozici v syntaktické konstrukci (ilustrováno konstrukcí z holandštiny – Příklad 1), tak jevy, u nichž je syntaktická struktura dána jinými prostředky než slovoslednými (jako je např. valence v češtině – Příklad 2). Holandština, jazyk s pevným slovosledem, tvoří 2-roztržité konstrukce (se dvěma fixními pozicemi, kde dochází k přepisování), které ale nejsou 2-volné. Naopak podobné konstrukce v češtině jsou j-volné (kde j je dáno počtem valenčních doplnění valenci nesoucích členů věty, zejména sloves a substantiv): slovosled takových konstrukcí je možné zjednodušit při zachování redukční struktury (viz příklad z úvodu). Podobná pozorování platí i pro latinu, většinu slovanských jazyků a pro některé další typy jazyků, což nás vede k úvahám, které jsme prezentovali v našem příspěvku.

V blízké budoucnosti plánujeme charakterizovat *sílu* různých typů formalizmů, např. [1,2], na základě pojmů navržených v tomto příspěvku. To se zdá být slibné zejména pro formalizmy založené na topologických omezeních, jako je [1], a to díky zde studovaným, ve své podstatě topologickým omezením.

Reference

1. R. Debusmann, D. Duchier, J. Niehren. The XDG Grammar Development Kit: Topological Dependency Trees: A Constraint-based Account of Linear Precedence. In: *Proceedings of the MOZ04 Conference, 39th Annual Meeting of ACL,* Charleroi/BEL, LNCS 3389, Springer, 2004, 190–201.
2. R. Gramatovici, C. Martin-Vide. Contextual Grammars and Dependency Trees. In: B. Daille, E. Morin (eds.), *Actes de TALN. 10e Conference sur le Traitement Automatique des Langues Naturelles, TALN 2003*, Batz-sur-Mer, France, 2003, 135–144.
3. T. Holan, V. Kuboň, K. Oliva, M. Plátek. Two Useful Measures of Word Order Complexity. In: A. Polguere, S. Kahane (eds.), *Proceedings of the Coling '98 Workshop "Processing of Dependency-Based Grammars"*, University of Montreal, 1998, 21–28.
4. T. Holan, V. Kuboň, K. Oliva, M. Plátek. 2000. On Complexity of Word Order. In: Les grammaires de dépendance - Traitement automatique des langues (TAL), Vol. 41, No. 1 (q.ed. S. Kahane), pp. 273-300.
5. M. Lopatková, M. Plátek, V. Kuboň. Závislostní redukční analýza přirozených jazyků, In: P. Vojtáš(ed.), *ITAT 2004, Information Technologies-Applictions and Theory*, Proceedings, Slovakia, 2005, 165–175
6. F. Otto. Restarting automata and their relations to the Chomsky hierarchy. In: Z. Ésik and Z. Fülöp (eds.), *Developments in Language Theory, Proc. DLT'2003, LNCS 2710*, Springer, Berlin, 2003, 55–74.
7. M. Steedman. Linguistic and Computational Theories of Grammar. Course at University of Edinburgh, 2005, http://www.iccs.informatics.ed.ac.uk/~steedman/tl/home.html