Indexed Grammars and Global Index Grammars

Faculty of Information Technology, BUT

Modern Theoretical Computer Science (TID)

Adam Husár
29 October 2007

Contents

- Introduction
- Indexed grammars
- Global index grammars
- Global index grammars with regular rules
- Conclusion

Introduction, motivation

- Regulated grammars/Grammars with controlled derivations

- Contextual vs. context-free rules

- Natural language features modelling
  - reduplication: $\{xx \mid x \in N\}$,
  - multiple agreements: $\{a^n b^n c^n \mid n \geqq 1\}$, $\{a^n b^n c^n d^n \mid n \geqq 1\}$, ...
  - crossed agreements: $\{a^n b^m c^n d^m \mid m, n \geqq 1\}$.

- Examples of reduplication
  - *чуть* (a little, few), *чуть-чуть* (very few)
  - 人 *rén* (person), 人人 *rénrén* (everybody)
  - *bon* (good), *bonbon* (bonbon)
  - *hocus-pokus*, *fifty-fifty*

Indexed Grammars

- Introduced by Alfred V. Aho in 1968.
- Extension of context-free grammars.

- Definition: $G=(N,T,I,P,S)$,

  – *N, T* and *S* are defined as in CFG,
  – *I* is a finite set of finite sets (indices) of productions of the form $A \rightarrow w$, where $A \in N$ and $w \in V_G^*$ and
  – *P* is a finite set of productions of the form $A \rightarrow \alpha$, where $A \in N$ and $\alpha \in (NI^* \cup T)^*$.

Indexed Grammars, cont.

- $G=(N,T,I,P,S)$,
  - I is a finite set of finite sets of productions of the form $A \text{->} w$, where $A \in N$ and $w \in V_G^*$ and
  - P is a finite set of productions of the form $A \text{->} \alpha$, where $A \in N$ and $\alpha \in (NI^* \cup T)^*$.

- Derivation relation $\Rightarrow$: $x \Rightarrow y$, where $x,y \in (NI^* \cup T)^*$,

  if either

  $x = x_1 A \beta x_2$, for some $x_1, x_2 \in (NI^* \cup T)^*$, $A \in N$, $\beta \in I^*$,
  $A \rightarrow X_1\beta_1 X_2\beta_2 \ldots X_k\beta_k \in P$,
  $y = x_1 X_1\gamma_1 X_2\gamma_2 \ldots X_k\gamma_k x_2$, with $\gamma_j = \beta_j\beta$, for $X_j \in N$, and $\gamma_j = \varepsilon$, for $X_j \in T$, $1 \leqq j \leqq k$

  or

  $x = x_1 A i \beta x_2$, for some $x_1, x_2 \in (NI^* \cup T)^*$, $A \in N$, $i \in I$, $\beta \in I^*$,
  $A \rightarrow X_1 X_2 \ldots X_k \in i$,
  $y = x_1 X_1\gamma_1 X_2\gamma_2 \ldots X_k\gamma_k x_2$, with $\gamma_j = \beta$, for $X_j \in N$, and $\gamma_j = \varepsilon$, for $X_j \in T$, $1 \leqq j \leqq k$.

Indexed Grammars, example

- Derivation relation $\Rightarrow$: $x \Rightarrow y$, where $x, y \in (NI^* \cup T)^*$, if either

  $x = x_1 A \beta x_2$, for some $x_1, x_2 \in (NI^* \cup T)^*$, $A \in N$, $\beta \in I^*$,
  $A \to X_1\beta_1 X_2\beta_2 \ldots X_k\beta_k \in P$,
  $y = x_1 X_1\gamma_1 X_2\gamma_2 \ldots X_k\gamma_k x_2$, with $\gamma_j = \beta_j\beta$, for $X_j \in N$, and $\gamma_j = \varepsilon$, for $X_j \in T$, $1 \leqq j \leqq k$

  or

  $x = x_1 A i \beta x_2$, for some $x_1, x_2 \in (NI^* \cup T)^*$, $A \in N$, $i \in I$, $\beta \in I^*$,
  $A \to X_1 X_2 \ldots X_k \in i$,
  $y = x_1 X_1\gamma_1 X_2\gamma_2 \ldots X_k\gamma_k x_2$, with $\gamma_j = \beta$, for $X_j \in N$, and $\gamma_j = \varepsilon$, for $X_j \in T$, $1 \leqq j \leqq k$.

- $G = (\{S, A, B\}, \{a, b, c\}, \{f, g\}, P, S)$, where
  - $f = \{B \to b\}$, $g = \{B \to bB\}$,
  - $P = \{S \to aAfc,\ A \to aAgc,\ A \to B\}$.

- $S \Rightarrow aAfc \Rightarrow aaAgfcc \Rightarrow \ldots \Rightarrow a^nAg^{n-1}fc^n \Rightarrow$
  $a^nBgg^{n-2}fc^n \Rightarrow a^nbBg^{n-2}fc^n \Rightarrow a^nbbBg^{n-3}fc^n \Rightarrow \ldots \Rightarrow a^nb^{n-1}Bfc^n \Rightarrow a^nb^nc^n$

- $L(G) = \{a^nb^nc^n \mid n \geqq 1\}$.

Indexed Grammars, properties

- $\mathcal{L}(I) = \mathcal{L}(\varepsilon I)$
- $\mathcal{L}(CF) \subsetneq \mathcal{L}(I) \subsetneq \mathcal{L}(CS)$

- The family of global index languages is an Full Abstract Family of Languages.
- Closed under union, concatenation, Kleene-closure and intersection with regular sets.
- Not closed under intersection nor complement.

- Membership problem: NP-complete.
- Emptiness problem: decidable.

- $\mathcal{L}(I)$ can be recognized by nested stack automata.

Global Index Grammars

- Introduced by José M. Castano in 2003.
- IG $\Rightarrow$ LIG $\Rightarrow$ GIG
- Uses stack of indices as a global control structure.

- Definition: $G = (N, T, I, S, \#, P)$, where
  - *N, T* and *S* are defined as in CFG,
  - *I* is a set of stack indices,
  - # is a start stack symbol and
  - *P* is a finite set of productions having following forms:

a.i	$A \rightarrow_{\varepsilon} \alpha$	*(epsilon),*
a.ii	$A \rightarrow_{[y]} \alpha$	*(epsilon with constraints),*
b.	$A \rightarrow_{x} a\beta$	*(push),*
c.	$A \rightarrow_{\neg x} \alpha a \beta$	*(pop),*

where $x \in I$, $y \in \{I \cup \{\#\}\}$, $A \in N$, $\alpha, \beta \in (N \cup T)^*$ and $a \in T$.

Global Index Grammars, cont.

- $G = (N, T, I, S, \#, P)$, where
  - P is a finite set of productions having following forms:

    a.i $A \rightarrow_{\varepsilon} \alpha$ *(epsilon),*

    a.ii $A \rightarrow_{[y]} \alpha$ *(epsilon with constraints),*

    b. $A \rightarrow_{x} a\beta$ *(push),*

    c. $A \rightarrow_{\neg x} \alpha a\beta$ *(pop),*

    where $x \in I$, $y \in \{I \cup \{\#\}\}$, $A \in N$, $\alpha, \beta \in (N \cup T)^*$ and $a \in T$.

- Derivation relation $\Rightarrow$:

  a. If $A \rightarrow_{\mu} X_1 \ldots X_n$ is a production of type (a.), then:

  i: $\delta\#\beta A\gamma \Rightarrow_{\varepsilon} \delta\#\beta X_1 \ldots X_n\gamma$ or

  ii: $z\delta\#\beta A\gamma \Rightarrow_{[z]} z\delta\#\beta X_1 \ldots X_n\gamma$.

  b. If $A \rightarrow_{\mu} X_1 \ldots X_n$ is a production of type (b.), then:

  $\delta\#wA\gamma \Rightarrow_{z} z\delta\#wX_1 \ldots X_n\gamma$. *(leftmost derivation)*

  c. If $A \rightarrow_{\mu} X_1 \ldots X_n$ is a production of type (c.), then:

  $z\delta\#wA\gamma \Rightarrow_{\neg z} \delta\#wX_1 \ldots X_n\gamma$. *(leftmost derivation)*

$\beta, \gamma \in (N \cup T)^*$, $\delta \in I^*$, $z \in I \cup \{\varepsilon\}$, $w \in T^*$ and $X_i \in (N \cup T)$.

Global Index Grammars, example

- $G = (\{S, R, A, B, C, L\}, \{a, b\}, \{i, k\}, S, \#, P)$, where
- $P = \{$

  $S \rightarrow AS \mid BS \mid C,$

  $C \rightarrow RC \mid L,$

  $R \rightarrow_{\neg i} RA,\ R \rightarrow_{\neg k} RB,\ R \rightarrow_{[\#]} \varepsilon,$

  $A \rightarrow_{i} a,\ B \rightarrow_{k} b,$

  $L \rightarrow_{\neg i} La \mid a,\ L \rightarrow_{\neg k} Lb \mid b$

  $\}$

- $\#S \Rightarrow \#AS \Rightarrow_i i\#aS \Rightarrow i\#aBS \Rightarrow_k ki\#abS \Rightarrow ki\#abC \Rightarrow ki\#abRC \Rightarrow_{\neg k} i\#abRBC \Rightarrow_{\neg i} \#abRABC \Rightarrow_{[\#]} \#abABC \Rightarrow_i i\#abaBC \Rightarrow_k ki\#ababC \Rightarrow ki\#ababL \Rightarrow_{\neg k} i\#ababLb \Rightarrow_{\neg i} \#ababab$

- $L(G) = \{ ww^{+} \mid w \in \{a, b\}\}$

Global Index Grammars, properties

- $\mathcal{L}(CF) \subsetneq \mathcal{L}(GI) \subsetneq \mathcal{L}(CS)$
- $\mathcal{L}(GI) \subsetneq \mathcal{L}(I)$???

- The family of global index languages is an Abstract Family of Languages.
- Closed under union, concatenation, Kleene-closure and intersection with regular sets.
- Not closed under intersection nor complement.

- Membership problem computational complexity: ???.
- Emptiness problem: decidable.

- $\mathcal{L}(GI)$ can be recognized using two stack pushdown automata.

Global Index Grammars with regular rules

- Let's have a CFG $G = (N, T, P, S)$ in Chomsky normal form.
- Then we can obtain an equivalent global index grammar

$$G = (N \cup \{X\}, T, I, S, \#, P'), X \notin N,$$

where production set P' and index set I is constructed from P in the following way:

for every rule from P of the form:

a) $A \rightarrow BC$, add $A \rightarrow_C B$ and $X \rightarrow_{\neg C} C$ to P' and C to I,

b) $A \rightarrow a$, add $A \rightarrow aX$ and $A \rightarrow a$ to P'.

Global Index Grammars with regular rules, example

- Lets have a CFG with following rules:

  1: $E \to FD$, 2: $D \to +FD$, 3: $D \to \varepsilon$, 4: $F \to (E)$, 5: $F \to i$.

- An equivalent GIG would have these rules:

  1: $E \to_{D} F$, $X \to_{\neg D} D$,

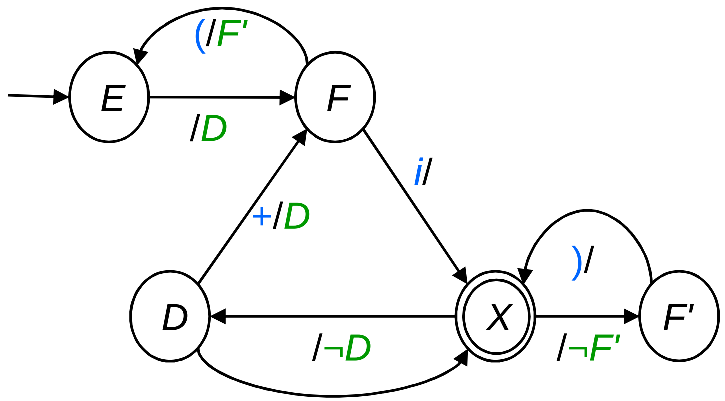
  2: $D \to_{D} +F$, $X \to_{\neg D} D$,

  3: $D \to \varepsilon$, $D \to X$,

  4: $F \to_{F'} (E$, $X \to_{\neg F'} F'$, $F' \to)$, $F' \to)X$,

  5: $F \to i$, $F \to iX$.

- Now we can represent this GIG using stack automata:

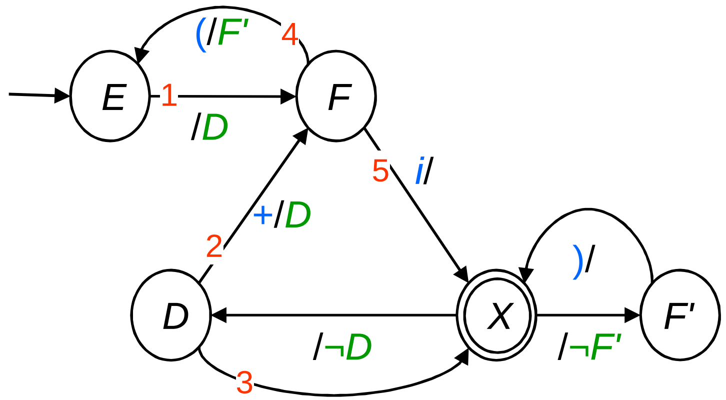


Global Index Grammars with regular rules, example, cont.

1) Original rules:

1: $E \rightarrow FD$,
2: $D \rightarrow +FD$,
3: $D \rightarrow \varepsilon$,
4: $F \rightarrow (E)$,
5: $F \rightarrow$ i.

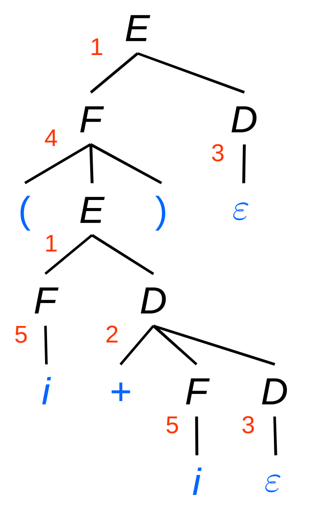
2) Equivalent stack automata:



3) For input string (i + i), this stack automata will have the following sequence of moves, where a configurations has this form: $((\Gamma \cup \{\#\})^* \times Q \times \Sigma^*)$:

$(\#, E, (i{+}i)) \mapsto_1 (D\#, F, (i{+}i)) \mapsto_4 (F'D\#, E, i{+}i)) \mapsto_1$
$(DF'D\#, F, i{+}i)) \mapsto_5 (DF'D\#, X, {+}i)) \mapsto$
$(F'D\#, D, {+}i)) \mapsto_2 (DF'D\#, F, i)) \mapsto_5$
$(DF'D\#, X,)) \mapsto (F'D\#, D,)) \mapsto_3$
$(F'D\#, X,)) \mapsto (D\#, F',)) \mapsto$
$(D\#, X,) \mapsto (\#, D,) \mapsto_3 (\#, X,)$.

4) Left parse 14152533



Conclusions

- Regulated grammars
- Indexed grammars
- Global index grammars
- Global index grammars with regular rules

References

[1] Aho, A. V.: Indexed grammars - an extension of context-free grammars. *Journal of the Association for Computing Machinery*, 15(4):647–671, 1968.

[2] Rozenberg, G., Salomaa, A. et al.: Handbook of Formal Languages, vol. 2. Springer Publishing, 2004.

[3] Castano, J. M.: Global Index Languages. PhD. Thesis, The Faculty of the Graduate School of Arts and Sciences, Brandeis University, 2004.