Bidirectional Contextual Grammars

Contextual Grammar without Choice

Contextual Grammar without Choice, $G = (T, P, S)$ (see [4])

T is a finite alphabet

P is a finite subset of $T^* \times T^*$, called contexts

S is a finite language over T

External Derivation Step

$x \Rightarrow y$ iff $y = uxv$, for a context $(u, v) \in P$

Generated Language

$L(G) = \{z \,:\, s \Rightarrow^* z, \text{ for any } s \in S\}$

Generative Power

$\mathcal{L}_{EC} = \mathcal{L}_{LIN_1}$ (languages described by linear grammars with 1 nonterminal)

Contextual Grammar without Choice – Example

Example

$G_{EC} = (\{a, b\}, \{(a, b)\}, \{\varepsilon\})$
$G_{LIN_1} = (\{S\}, \{a, b\}, \{S \rightarrow aSb, S \rightarrow \varepsilon\}, S)$

$\varepsilon \Rightarrow ab \Rightarrow aabb \Rightarrow aaabbb$ in G_{EC}
$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aaaSbbb$ in G_{LIN_1}

$L(G_{EC}) = L(G_{LIN_1}) = \{a^n b^n \ : \ n \geq 0\}$

Bidirectional Contextual Grammar

Bidirectional ([1]) Contextual Grammar, $G = (T \cup \{\$\}, P_d \cup P_r, S)$

T is a finite alphabet, \$ is a special symbol, $\$ \notin T$

P_d, P_r are finite subsets of $(T \cup \{\$\})^* \times (T \cup \{\$\})^*$

S is a finite language over $T \cup \{\$\}$

Computation Step

derivation $x \;{}_d\!\Rightarrow y$ iff $y = uxv$, for a context $(u, v) \in P_d$

reduction $y \;{}_r\!\Rightarrow x$ iff $y = uxv$, for a context $(u, v) \in P_r$

computation $x \Rightarrow y$ iff $x \;{}_d\!\Rightarrow y$ or $x \;{}_r\!\Rightarrow y$

\$-Bounded Language (see [5])

$${}_{\$}L(G) = \{z : s \Rightarrow^* \$z\$ \text{ in } G, z \in T^*, s \in S\}$$

Main Result

Successful Computation

Every computation of the form $s \Rightarrow^* \$z\$$, $s \in S, z \in T^*$ is said to be successful.

Turn

Every computation of the form

$$x \ {}_{d}\!\Rightarrow y \ {}_{r}\!\Rightarrow z \text{ or } x \ {}_{r}\!\Rightarrow y \ {}_{d}\!\Rightarrow z,$$

$x, y, z \in (T \cup \{\$\})^*$ is called turn. G is i-turn if every successful computation in G contains at most i turns.

Theorem

Let L be a recursively enumerable language. Then, there exists a one-turn bidirectional contextual grammar, G, such that $L = {}_{\$}L(G)$.

Queue Grammar (**QG**)

- we represent the recursively enumerable language by a queue grammar

Queue Grammar $G = (V, T, W, F, s, P)$

V is a finite alphabet of symbols

T is a set of terminals, $T \subset V$

W is a finite alphabet of states

F is a set of final states, $F \subset W$

s is a starting string, $s \in (V - T)(W - F)$

P is a finite set of productions of the form: (a, b, x, c)

a $\in V$
b $\in (W - F)$
x $\in V^*$
c $\in W$

Queue Grammar – Derivation Step

Derivation Step

If $u = arb$, $v = rxc$, $a \in V$, $r, x \in V^*$, $b, c \in W$, and $(a, b, x, c) \in P$, then $u \Rightarrow v\ [(a, b, x, c)]$.

Generated Language

$L(G) = \{w : s \Rightarrow^* wf, w \in T^*, f \in F\}$

Generative Power (see [2])

$\mathcal{L}_{QG} = \mathcal{L}_{RE}$

Lemma

For every **QG** *there exists an equivalent* **QG** *which generates every string so that it first uses only productions rewriting symbols over $(V - T)^*$, and then only symbols over T^* (proof, see [3]).*

Queue Grammar – Example

Example

$G_1 = (\{A, a, b\}, \{a, b\}, \{\bar{e}, \bar{f}\}, \{\bar{f}\}, A\bar{e}, P_1)$

$$\begin{aligned} P_1 = \{\ & 1 : (A, \bar{e}, bAa, \bar{e}), \\ & 2 : (A, \bar{e}, \varepsilon, \bar{f}), \\ & 3 : (a, \bar{e}, a, \bar{e}), \\ & 4 : (b, \bar{e}, b, \bar{e})\} \end{aligned}$$

$A\bar{e} \Rightarrow bAa\bar{e}\ [1] \Rightarrow Aab\bar{e}\ [4] \Rightarrow abbAa\bar{e}\ [1] \Rightarrow bbAaa\bar{e}\ [3] \Rightarrow bAaab\bar{e}\ [4]$
$\Rightarrow Aaabb\bar{e}\ [4] \Rightarrow aabb\bar{f}\ [2]$

$L(G_1) = \{a^n b^n \ : \ n \geq 0\}$

Proof Sketch

Basic Idea

1. represent the recursively enumerable language by a **QG**
2. simulate any derivation in the **QG** by a bidirectional contextual grammar using derivation steps (in the reverse order)
    1. simulate the last derivation step in the **QG** by a string from S in the contextual grammar
    2. simulate generation of words from T^+
    3. simulate generation of words from $(V - T)^*$
    4. simulate the starting string of the **QG**
3. verify the simulation by reduction steps

QG Simulation – Example

Example

$G_1 = (\{A, a, b\}, \{a, b\}, \{\bar{e}, \bar{f}\}, \{\bar{f}\}, A\bar{e}, P_1)$

$$P_1 = \{\ 1 : (A, \bar{e}, bAa, \bar{e}), \\ 2 : (A, \bar{e}, \varepsilon, \bar{f}), \\ 3 : (a, \bar{e}, a, \bar{e}), \\ 4 : (b, \bar{e}, b, \bar{e})\}$$

Queue	A
States	$\bar{e}$
Productions	

QG Simulation – Example

Example

$G_1 = (\{A, a, b\}, \{a, b\}, \{\bar{e}, \bar{f}\}, \{\bar{f}\}, A\bar{e}, P_1)$

$$
\begin{aligned}
P_1 = \{\; & 1 : (A, \bar{e}, bAa, \bar{e}),\\
& 2 : (A, \bar{e}, \varepsilon, \bar{f}),\\
& 3 : (a, \bar{e}, a, \bar{e}),\\
& 4 : (b, \bar{e}, b, \bar{e})\}
\end{aligned}
$$

Queue	A	b	A	a
States	$\bar{e}$	$\bar{e}$		
Productions	1			

QG Simulation – Example

Example

$G_1 = (\{A, a, b\}, \{a, b\}, \{\bar{e}, \bar{f}\}, \{\bar{f}\}, A\bar{e}, P_1)$

$$\begin{aligned} P_1 = \{\ & 1 : (A, \bar{e}, bAa, \bar{e}), \\ & 2 : (A, \bar{e}, \varepsilon, \bar{f}), \\ & 3 : (a, \bar{e}, a, \bar{e}), \\ & 4 : (b, \bar{e}, b, \bar{e})\} \end{aligned}$$

Queue	A	b	A	a	b
States	$\bar{e}$	$\bar{e}$	$\bar{e}$		
Productions	1	4			

QG Simulation – Example

Example

$G_1 = (\{A, a, b\}, \{a, b\}, \{\bar{e}, \bar{f}\}, \{\bar{f}\}, A\bar{e}, P_1)$

$$\begin{aligned} P_1 = \{\ & 1 : (A, \bar{e}, bAa, \bar{e}), \\ & 2 : (A, \bar{e}, \varepsilon, \bar{f}), \\ & 3 : (a, \bar{e}, a, \bar{e}), \\ & 4 : (b, \bar{e}, b, \bar{e})\} \end{aligned}$$

Queue	A	b	A	a	b	b	A	a
States	$\bar{e}$	$\bar{e}$	$\bar{e}$	$\bar{e}$				
Productions	1	4	1					

QG Simulation – Example

Example

$G_1 = (\{A, a, b\}, \{a, b\}, \{\bar{e}, \bar{f}\}, \{\bar{f}\}, A\bar{e}, P_1)$

$$
\begin{aligned}
P_1 = \{\; & 1 : (A, \bar{e}, bAa, \bar{e}),\\
& 2 : (A, \bar{e}, \varepsilon, \bar{f}),\\
& 3 : (a, \bar{e}, a, \bar{e}),\\
& 4 : (b, \bar{e}, b, \bar{e})\}
\end{aligned}
$$

Queue	A	b	A	a	b	b	A	a	a
States	$\bar{e}$	$\bar{e}$	$\bar{e}$	$\bar{e}$	$\bar{e}$				
Productions	1	4	1	3					

QG Simulation – Example

Example

$G_1 = (\{A, a, b\}, \{a, b\}, \{\bar{e}, \bar{f}\}, \{\bar{f}\}, A\bar{e}, P_1)$

$$
\begin{aligned}
P_1 = \{\ & 1 : (A, \bar{e}, bAa, \bar{e}),\\
& 2 : (A, \bar{e}, \varepsilon, \bar{f}),\\
& 3 : (a, \bar{e}, a, \bar{e}),\\
& 4 : (b, \bar{e}, b, \bar{e})\}
\end{aligned}
$$

Queue	A	b	A	a	b	b	A	a	a	b
States	$\bar{e}$	$\bar{e}$	$\bar{e}$	$\bar{e}$	$\bar{e}$	$\bar{e}$				
Productions	1	4	1	3	4					

QG Simulation – Example

Example

$G_1 = (\{A, a, b\}, \{a, b\}, \{\bar{e}, \bar{f}\}, \{\bar{f}\}, A\bar{e}, P_1)$

$$
\begin{aligned}
P_1 = \{\ & 1 : (A, \bar{e}, bAa, \bar{e}),\\
& 2 : (A, \bar{e}, \varepsilon, \bar{f}),\\
& 3 : (a, \bar{e}, a, \bar{e}),\\
& 4 : (b, \bar{e}, b, \bar{e})\}
\end{aligned}
$$

Queue	A	b	A	a	b	b	A	a	a	b	b
States	$\bar{e}$	$\bar{e}$	$\bar{e}$	$\bar{e}$	$\bar{e}$	$\bar{e}$	$\bar{e}$				
Productions	1	4	1	3	4	4					

QG Simulation – Example

Example

$G_1 = (\{A, a, b\}, \{a, b\}, \{\bar{e}, \bar{f}\}, \{\bar{f}\}, A\bar{e}, P_1)$

$P_1 = \{$ $1 : (A, \bar{e}, bAa, \bar{e}),$
$2 : (A, \bar{e}, \varepsilon, \bar{f}),$
$3 : (a, \bar{e}, a, \bar{e}),$
$4 : (b, \bar{e}, b, \bar{e})\}$

Queue	A	b	A	a	b	b	A	a	a	b	b
States	$\bar{e}$	$\bar{e}$	$\bar{e}$	$\bar{e}$	$\bar{e}$	$\bar{e}$	$\bar{e}$	$\bar{f}$			
Productions	1	4	1	3	4	4	2				

QG Simulation – Example

Example

$G_1 = (\{A, a, b\}, \{a, b\}, \{\bar{e}, \bar{f}\}, \{\bar{f}\}, A\bar{e}, P_1)$

$$\begin{aligned} P_1 = \{\ & 1 : (A, \bar{e}, bAa, \bar{e}), \\ & 2 : (A, \bar{e}, \varepsilon, \bar{f}), \\ & 3 : (a, \bar{e}, a, \bar{e}), \\ & 4 : (b, \bar{e}, b, \bar{e})\} \end{aligned}$$

Queue	A	b	A	a	b	b	A	a	a	b	b
States	$\bar{e}$	$\bar{e}$	$\bar{e}$	$\bar{e}$	$\bar{e}$	$\bar{e}$	$\bar{e}$	$\bar{f}$			
Productions	1	4	1	3	4	4	2				
Prod. (queue)	1,2	4	1,2	3	4	4	1,2				
Prod. (state)	1-4	1-4	1-4	1-3,4	1-4	1-4	1,2-4				
Simulated pr.	1	4	1	3	4	4	2				

QG Simulation – Example

Example

$$G_1 = (\{A, a, b\}, \{a, b\}, \{\bar{e}, \bar{f}\}, \{\bar{f}\}, A\bar{e}, P_1)$$

$$\begin{aligned} P_1 = \{\ & 1 : (A, \bar{e}, bAa, \bar{e}),\\ & 2 : (A, \bar{e}, \varepsilon, \bar{f}),\\ & 3 : (a, \bar{e}, a, \bar{e}),\\ & 4 : (b, \bar{e}, b, \bar{e})\} \end{aligned}$$

Queue	A	b	A	a	b	b	A	a	a	b	b
States	$\bar{e}$	$\bar{e}$	$\bar{e}$	$\bar{e}$	$\bar{e}$	$\bar{e}$	$\bar{e}$	$\bar{f}$			
Productions	1	4	1	3	4	4	2				
Prod. (queue)	1,2	4	1,2	3	4	4	1,2				
Prod. (state)	1-4	1-4	1-4	1-3,4	1-4	1-4	1,2-4				
Simulated pr.	1	4	1	3	4	4	2				

QG Simulation by Bidirectional Contextual Grammar

Simulation of **QG**

- expand the queue on the left-hand side of the sentential form
- check the states on the right-hand side of the sentential form

- terminals – a
- productions based on queue – b
- productions based on states – c
- simulated productions – d

Sentential Form before the Turn

$$\$b_n \ldots \$b_2\$b_1\$a_1a_2 \ldots a_n\$c_1\$\$d_1\$\$c_2\$\$d_2\$\$ \ldots c_n\$\$d_n\$\$$$

Construction I

- $Q = (V, T, W, F, s, R)$ such that $L(Q) = L$
- $o \in T$ – any symbol from T
- α – injective homomorphism from R to $\{o\}^+$
- $f(\varepsilon) = \varepsilon$ and $f(a) = \{\alpha((a, b, c_1 \dots c_n, d)) : (a, b, c_1 \dots c_n, d) \in R\}$ for all $a \in V$
- $g(b) = \{\alpha((a, b, c_1 \dots c_n, d)) : (a, b, c_1 \dots c_n, d) \in R\}$ for all $b \in W$

Constructed Bidirectional Contextual Grammar

$$G = (T \cup \{\$\}, P_d \cup P_r, S)$$

Simulation of the Last Step in **QG**

$$S = \{\, c_1 \dots c_n \$ \alpha((a, b, c_1 \dots c_n, d)) : (a, b, c_1 \dots c_n, d) \in R, \\ c_1, \dots, c_n \in T \text{ for some } n \geq 0, d \in F\}$$

Construction II

Simulation of **QG**'s Productions over T^+

For every $(a, b, c_1 \ldots c_n, d) \in R$, $c_1, \ldots, c_n \in T$, for some $n \geq 0$, $d \in (W - F)$, $\overline{d} \in g(d)$, add

$$(c_1 \ldots c_n, \$\$\overline{d}\$\$\alpha((a, b, c_1 \ldots c_n, d)))$$

to P_d.

Simulation of **QG**'s Productions over $(V - T)^+$

For every $(a, b, c_1 \ldots c_n, d) \in R$, $c_1, \ldots, c_n \in (V - T)$, for some $n \geq 0$, $d \in (W - F)$, $\overline{c}_1 \in f(c_1), \ldots, \overline{c}_n \in f(c_n)$, $\overline{d} \in g(d)$, add

$$(\overline{c}_1\$\overline{c}_2\$\ldots\overline{c}_n\$, \$\$\overline{d}\$\$\alpha((a, b, c_1 \ldots c_n, d)))$$

to P_d.

Construction III

Simulation of **QG**'s Starting String

For every $\overline{a}_0 \in f(a_0), \overline{q}_0 \in f(q_0)$ such that $s = a_0 q_0$, add

$$(\$\overline{a}_0\$, \$\$\overline{q}_0\$\$)$$

to P_d.

Verification of the Simulation

For every $r \in R$, add

$$(\$\alpha(r), \alpha(r)\$\$\alpha(r)\$\$)$$

to P_r.

Proof I

$$\begin{aligned} S &\approx c_1 \ldots c_n \$ \alpha((a, b, c_1 \ldots c_n, d)) \\ {}_1P_d &\approx (c_1 \ldots c_n, \$\$\overline{d}\$\$\alpha((a, b, c_1 \ldots c_n, d))) \\ {}_2P_d &\approx (\overline{c}_1\$\overline{c}_2\$ \ldots \overline{c}_n\$, \$\$\overline{d}\$\$\alpha((a, b, c_1 \ldots c_n, d))) \\ {}_3P_d &\approx (\$\overline{a}_0\$, \$\$\overline{q}_0\$\$) \\ P_r &\approx (\$\alpha(r), \alpha(r)\$\$\alpha(r)\$\$) \end{aligned}$$

- $s \notin {}_{\$}L(G)$, for any $s \in S$
- no production from ${}_1P_d \cup {}_2P_d \cup {}_3P_d$ is applied in the very last computation step

Therefore, the computation has the form

$$\begin{array}{cccc} s & \Rightarrow^+ & v & [\rho] \\ & {}_r\Rightarrow^+ & w & [\tau]. \end{array}$$

- ρ denotes productions from ${}_1P_d \cup {}_2P_d \cup {}_3P_d \cup P_r$
- τ denotes productions from P_r

Proof II

$P_r \approx (\$\alpha(r), \alpha(r)\$\$\alpha(r)\$\$)$

Incorrect Forms of v

- $\$p_{n-1}\ldots\$p_1\$w\$p_1\$\$p_1\$\$\ldots p_{n-1}\$\$p_{n-1}\$\$p_n\$\$p_n\$\$$ $_r{\Rightarrow^*}\ \$ \notin {}_{\$}L(G)$
- $\$p_{n-2}\ldots\$p_1\$w\$p_1\$\$p_1\$\$\ldots p_{n-1}\$\$p_{n-1}\$\$p_n\$\$p_n\$\$$ $_r{\Rightarrow^*}\ \$\$p_1\$\$ \notin {}_{\$}L(G)$
- $\$p_n\ldots\$p_1\$w\$p_1\$\$p_1\$\$\ldots p_{n-1}\$\$p_{n-1}\$\$$ $_r{\Rightarrow^*}\ \$p_1\$w\$ \notin {}_{\$}L(G)$

Correct Form of v

$\$p_n\$p_{n-1}\ldots\$p_1\$w\$p_1\$\$p_1\$\$\ldots p_{n-1}\$\$p_{n-1}\$\$p_n\$\$p_n\$\$$

Proof III

$$\begin{aligned}
S &\approx c_1 \ldots c_n \$ \alpha((a, b, c_1 \ldots c_n, d)) \\
{}_1P_d &\approx (c_1 \ldots c_n, \$\$\overline{d}\$\$\alpha((a, b, c_1 \ldots c_n, d))) \\
{}_2P_d &\approx (\overline{c}_1\$\overline{c}_2\$ \ldots \overline{c}_n\$, \$\$\overline{d}\$\$\alpha((a, b, c_1 \ldots c_n, d))) \\
{}_3P_d &\approx (\$\overline{a}_0\$, \$\$\overline{q}_0\$\$) \\
P_r &\approx (\$\alpha(r), \alpha(r)\$\$\alpha(r)\$\$)
\end{aligned}$$

- $p_r \in P_r$ cannot be used after ${}_1P_d \cup {}_2P_d$
- $p_r \in P_r$ can be used after $p_3 \in {}_3P_d$

Therefore, the computation has the form

$$\begin{array}{llll}
s & {}_d\Rightarrow^* & u_1 & [\xi_1] \\
 & {}_d\Rightarrow & u_2 & [p_3] \\
 & \Rightarrow^* & v & [\xi_2] \\
 & {}_r\Rightarrow^+ & w &
\end{array}$$

- ξ_1 is a sequence of productions from ${}_1P_d \cup {}_2P_d$
- ξ_2 is a sequence of productions from ${}_1P_d \cup {}_2P_d \cup {}_3P_d \cup P_r$

Proof IV

$$\begin{aligned} S &\approx c_1 \dots c_n \$ \alpha((a, b, c_1 \dots c_n, d)) \\ {}_1P_d &\approx (c_1 \dots c_n, \$\$\overline{d}\$\$\alpha((a, b, c_1 \dots c_n, d))) \\ {}_2P_d &\approx (\overline{c}_1\$\overline{c}_2\$ \dots \overline{c}_n\$, \$\$\overline{d}\$\$\alpha((a, b, c_1 \dots c_n, d))) \\ {}_3P_d &\approx (\$\overline{a}_0\$, \$\$\overline{q}_0\$\$) \\ P_r &\approx (\$\alpha(r), \alpha(r)\$\$\alpha(r)\$\$) \end{aligned}$$

The Form of u_2

$\$p_m\$p_{m-1} \dots \$p_1\$w\$p_1\$\$p_1\$\$ \dots p_{n-1}\$\$p_{n-1}\$\$p_n\$\$p_n\$\$$

- after any application of $p_r \in P_r$, \$ is at the end of the sentential form
- p_r followed by $p_d \in {}_1P_d \cup {}_2P_d \cup {}_3P_d$ leads to \$\$\$ which blocks the computation
- after $p_3 \in {}_3P_d$ only $p_r \in P_r$ can be used

Proof V

We obtain every successful computation in the form

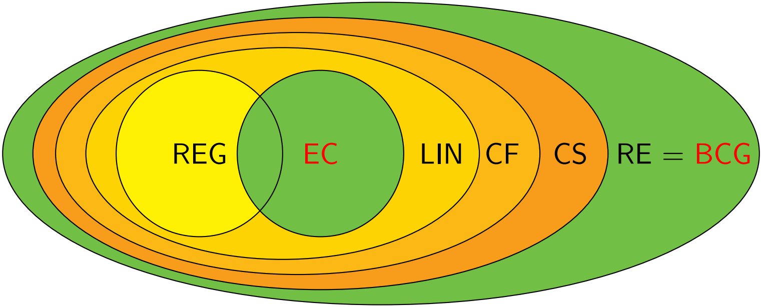
$$\begin{array}{llll} s & {}_d\!\Rightarrow^* & y & [\psi] \\ & {}_d\!\Rightarrow^* & u & [\zeta] \\ & {}_d\!\Rightarrow & v & [p_3] \\ & {}_r\!\Rightarrow^* & w & [\eta] \end{array}$$

- ψ is a sequence of productions from ${}_1P_d$
- ζ is a sequence of productions from ${}_2P_d$
- $p_3 \in {}_3P_d$
- η is a sequence of productions from P_r

Summary

- by the introduction of
    - reducing productions and
    - \$-bounded language,

  we greatly increase the power of contextual grammars



Further Investigation

- bidirectional contextual automata (machines)

Bibliography I

D. E. Appelt.
Bidirectional grammars and the design of natural language generation systems.
In *Proceedings of Third Conference on Theoretical Issues in Natural Language Processing (TINLAP-3)*, pages 185–191, Las Cruces, New Mexico, 1987.

H. C. M. Kleijn and G. Rozenberg.
On the generative power of regular pattern grammars.
Acta Informatica, 20:391–411, 1983.

A. Meduna.
Simultaneously one-turn two-pushdown automata.
International Journal of Computer Mathematics, 80:679–687, 2003.

Bibliography II

G. Paun.
Marcus contextual grammars.
Kluwer Academic Publishers, London, 1997.

A. Salomaa.
Formal Languages.
Academic Press, New York, 1973.