Practical problems with Chomsky-Schützenberger parsing for weighted multiple context-free grammars[1]

Tobias Denkinger

tobias.denkinger@tu-dresden.de

Institute of Theoretical Computer Science
Faculty of Computer Science
Technische Universität Dresden

2018-04-27

[1]based on T. Denkinger (2017). "Chomsky-Schützenberger parsing for weighted multiple context-free languages".

The problem: k-best parsing

parsing problem

Input:

- a grammar G
- a word w

Output:

- a derivation of w in G (not unique)

The problem: k-best parsing

k-best parsing problem [Huang and Chiang 2005]

Input:

- a weighted grammar G
- a suitable partial order $\trianglelefteq$ on the weights
- a number $k \in \mathbb{N}$
- a word w

Output:

- a sequence of k best derivations[2] of w in G

(not unique)

[2] w.r.t. weights and $\trianglelefteq$ (less is better)

Multiple context-free grammars

context-free grammars

$A \to \mathrm{a}A\mathrm{b}B$ composes strings

Multiple context-free grammars

context-free grammars

$$A \to \mathrm{a}A\mathrm{b}B$$

composes strings

$$A \to [\underbrace{(x, y) \mapsto \mathrm{a}x\mathrm{b}y}_{\Sigma^* \times \Sigma^* \to \Sigma^*}](A, B)$$

Multiple context-free grammars

context-free grammars

$$A \to \mathrm{a}A\mathrm{b}B$$ composes strings

$$A \to [\quad \mathrm{a}\,x\,\mathrm{b}\,y \quad](A, B)$$

Multiple context-free grammars

context-free grammars

$$A \to \mathrm{a}A\mathrm{b}B$$

composes strings

$$A \to [\quad \mathrm{a}\, x\, \mathrm{b}\, y \quad](A, B)$$

multiple context-free grammars [Seki, Matsumura, Fujii, and Kasami 1991]

$$A \to [\underbrace{((x_1, x_2), (y_1, y_2)) \mapsto (\mathrm{a}x_1y_2\mathrm{b}, y_1\mathrm{c}x_2)}_{(\Sigma^* \times \Sigma^*) \times (\Sigma^* \times \Sigma^*) \to (\Sigma^* \times \Sigma^*)}](A, B)$$

composes *tuples* of strings

Multiple context-free grammars

context-free grammars

$$A \to \mathrm{a}A\mathrm{b}B$$

composes strings

$$A \to [\quad \mathrm{a}\,x\,\mathrm{b}\,y \quad](A, B)$$

multiple context-free grammars [Seki, Matsumura, Fujii, and Kasami 1991]

$$A \to [\quad \mathrm{a}x_1y_2\mathrm{b}, y_1\mathrm{c}x_2 \quad](A, B)$$

composes *tuples* of strings

Multiple context-free grammars

context-free grammars

$$A \to \mathrm{a}A\mathrm{b}B$$

composes strings

$$A \to [\quad \mathrm{a}\, x\, \mathrm{b}\, y \quad](A, B)$$

multiple context-free grammars [Seki, Matsumura, Fujii, and Kasami 1991]

$$A \to [\quad \mathrm{a}x_1y_2\mathrm{b}, y_1\mathrm{c}x_2 \quad](A, B)$$

composes *tuples* of strings

$\Longrightarrow$ extra expressive power useful for natural language processing

The Chomsky-Schützenberger theorem

CS-theorem$_s$ [Chomsky and Schützenberger 1963]

Let L be a language. T.f.a.e.

1. $\exists$ CFG G s.t. $L = \mathrm{L}(G)$
2. $\exists$ regular language R,
   $\exists$ Dyck language D,
   $\exists$ homomorphism h
   s.t. $L = h(R \cap D)$

The Chomsky-Schützenberger theorem

CS-theorems [Chomsky and Schützenberger 1963] [Yoshinaka, Kaji, and Seki 2010]

Let L be a language. T.f.a.e.

1. $\exists$ MCFG G s.t. $L = \mathrm{L}(G)$
2. $\exists$ regular language R,
   $\exists$ multiple Dyck language D,
   $\exists$ homomorphism h
   s.t. $L = h(R \cap D)$

The Chomsky-Schützenberger theorem

CS-theorems [Chomsky and Schützenberger 1963] [Yoshinaka, Kaji, and Seki 2010]

Let L be a language. T.f.a.e.

1. $\exists$ MCFG G s.t. $L = \mathrm{L}(G)$
2. $\exists$ regular language R,
   $\exists$ multiple Dyck language D,
   $\exists$ homomorphism h
   s.t. $L = h(R \cap D)$

Idea [Hulden 2011, for CFGs]

Use the decomposition provided by (1. $\rightarrow$ 2.) for parsing.

From the CS-theorem to CS-parsing

$$w \in \mathrm{L}(G)$$

From the CS-theorem to CS-parsing

$$w \in \mathrm{L}(G) \iff w \in h(R \cap D) \qquad \text{(CS-theorem)}$$

From the CS-theorem to CS-parsing

$$w \in \mathrm{L}(G) \iff w \in h(R \cap D) \qquad \text{(CS-theorem)}$$
$$\iff \exists u \in R \cap D\colon h(u) = w$$

From the CS-theorem to CS-parsing

$$\begin{aligned} w \in \mathrm{L}(G) &\iff w \in h(R \cap D) && \text{(CS-theorem)} \\ &\iff \exists u \in R \cap D\colon h(u) = w \\ &\iff \exists u \in R \cap h^{-1}(w)\colon u \in D \end{aligned}$$

From the CS-theorem to CS-parsing

$$
\begin{aligned}
w \in \mathrm{L}(G) &\iff w \in h(R \cap D) && \text{(CS-theorem)} \\
&\iff \exists u \in R \cap D \colon h(u) = w \\
&\iff \exists u \in R \cap h^{-1}(w) \colon u \in D
\end{aligned}
$$

Observation

Each $u \in R \cap D$ encodes a derivation of G.

From the CS-theorem to CS-parsing

$$\begin{aligned} w \in \mathrm{L}(G) &\iff w \in h(R \cap D) && \text{(CS-theorem)} \\ &\iff \exists u \in R \cap D \colon h(u) = w \\ &\iff \exists u \in R \cap h^{-1}(w) \colon u \in D \end{aligned}$$

Observation

Each $u \in R \cap D$ encodes a derivation of G.

k-best CS-parsing

$$\mathrm{parse}_{G,\mathit{wt},k}(w)$$

From the CS-theorem to CS-parsing

$$\begin{aligned} w \in \mathrm{L}(G) &\iff w \in h(R \cap D) && \text{(CS-theorem)} \\ &\iff \exists u \in R \cap D \colon h(u) = w \\ &\iff \exists u \in R \cap h^{-1}(w) \colon u \in D \end{aligned}$$

Observation

Each $u \in R \cap D$ encodes a derivation of G.

k-best CS-parsing

$$\begin{aligned} &\mathrm{parse}_{G,\mathit{wt},k}(w) \\ &= (\mathrm{toDeriv} \circ \mathrm{take}_k \circ \mathrm{sort}^{\mathit{wt}}_{\trianglelefteq} \circ \mathrm{filter}_{\cap D})(R \cap h^{-1}(w)) \end{aligned}$$

From the CS-theorem to CS-parsing

$$\begin{aligned} w \in \mathrm{L}(G) &\iff w \in h(R \cap D) && \text{(CS-theorem)} \\ &\iff \exists u \in R \cap D\colon h(u) = w \\ &\iff \exists u \in R \cap h^{-1}(w)\colon u \in D \end{aligned}$$

Observation

Each $u \in R \cap D$ encodes a derivation of G.

k-best CS-parsing

$$\begin{aligned} &\mathrm{parse}_{G,\mathit{wt},k}(w) \\ &= (\mathrm{toDeriv} \circ \mathrm{take}_k \circ \mathrm{sort}^{\mathit{wt}}_{\trianglelefteq} \circ \mathrm{filter}_{\cap D})(R \cap h^{-1}(w)) \\ &= (\mathrm{toDeriv} \circ \mathrm{take}_k \circ \mathrm{filter}_{\cap D} \circ \mathrm{sort}^{\mathit{wt}}_{\trianglelefteq})(R \cap h^{-1}(w)) \end{aligned}$$

From the CS-theorem to CS-parsing

$$\begin{aligned} w \in \mathrm{L}(G) &\iff w \in h(R \cap D) && \text{(CS-theorem)} \\ &\iff \exists u \in R \cap D\colon h(u) = w \\ &\iff \exists u \in R \cap h^{-1}(w)\colon u \in D \end{aligned}$$

Observation

Each $u \in R \cap D$ encodes a derivation of G.

k-best CS-parsing

$$\begin{aligned} &\mathrm{parse}_{G,\mathit{wt},k}(w) \\ &= (\mathrm{toDeriv} \circ \mathrm{take}_k \circ \mathrm{sort}_{\trianglelefteq}^{\mathit{wt}} \circ \mathrm{filter}_{\cap D})(R \cap h^{-1}(w)) \\ &= (\mathrm{toDeriv} \circ \mathrm{take}_k \circ \mathrm{filter}_{\cap D} \circ \mathrm{sort}_{\trianglelefteq}^{\mathit{wt}})(R \cap h^{-1}(w)) \\ &= (\mathrm{toDeriv} \circ \mathrm{take}_k \circ \mathrm{filter}_{\cap D} \circ \mathrm{sort}_{\trianglelefteq})(R^{\mathit{wt}} \triangleright h^{-1}(w)) \end{aligned}$$

From the CS-theorem to CS-parsing

$$\begin{aligned} w \in \mathrm{L}(G) &\iff w \in h(R \cap D) && \text{(CS-theorem)} \\ &\iff \exists u \in R \cap D\colon h(u) = w \\ &\iff \exists u \in R \cap h^{-1}(w)\colon u \in D \end{aligned}$$

Observation

Each $u \in R \cap D$ encodes a derivation of G.

k-best CS-parsing

$$\begin{aligned} &\mathrm{parse}_{G,\mathit{wt},k}(w) \\ &= (\mathrm{toDeriv} \circ \mathrm{take}_k \circ \mathrm{sort}^{\mathit{wt}}_{\trianglelefteq} \circ \mathrm{filter}_{\cap D})(R \cap h^{-1}(w)) \\ &= (\mathrm{toDeriv} \circ \mathrm{take}_k \circ \mathrm{filter}_{\cap D} \circ \mathrm{sort}^{\mathit{wt}}_{\trianglelefteq})(R \cap h^{-1}(w)) \\ &= (\mathrm{toDeriv} \circ \mathrm{take}_k \circ \mathrm{filter}_{\cap D} \circ \underbrace{\mathrm{sort}_{\trianglelefteq})(R^{\mathit{wt}} \triangleright h^{-1}(w))}_{\text{enumerate from a weighted finite-state automaton}} \end{aligned}$$

Derivations and bracket words

$$\alpha: S \to [\ x_1 \quad x_2\ ](A)$$

$$\beta: A \to [\ \mathrm{a}\ \ x_1\ \ b\ \ ,\ \ c\ \ x_2\ ](A)$$

$$\beta: A \to [\ \mathrm{a}\ \ x_1\ \ b\ \ ,\ \ c\ \ x_2\ ](A)$$

$$\gamma: A \to [\ \varepsilon\ \ ,\ \ \varepsilon\ ]()$$

word $w =$

Derivations and bracket words

automaton for R:

$$\overset{S_1\bullet}{} \qquad \overset{\bullet\bar{S}_1}{}$$

$$\alpha: S \to [\ x_1 \quad x_2\ ](A)$$

$$A_1\bullet \qquad \bar{A}_1\bullet \ \bullet A_2 \qquad \bullet\bar{A}_2$$

$$\beta: A \to [\ \mathrm{a}\ \ x_1\ \ b\ \ ,\ \ c\ \ x_2\ ](A)$$

$$A_1\bullet \qquad \bar{A}_1\bullet \ \bullet A_2 \qquad \bullet\bar{A}_2$$

$$\beta: A \to [\ \mathrm{a}\ \ x_1\ \ b\ \ ,\ \ c\ \ x_2\ ](A)$$

$$A_1\bullet\bar{A}_1\bullet \ \bullet A_2\bullet\bar{A}_2$$

$$\gamma: A \to [\ \varepsilon\ \ ,\ \ \varepsilon\ ]()$$

word $w =$

Derivations and bracket words

automaton for R:

$$\alpha\colon S \to [\overset{S_1}{\bullet} x_1 \bullet\bullet x_2 \overset{\bar{S}_1}{\bullet}](A)$$

$$\beta\colon A \to [\overset{A_1}{\bullet} \mathrm{a} \bullet x_1 \bullet b \overset{\bar{A}_1}{\bullet}, \overset{A_2}{\bullet} c \bullet x_2 \overset{\bar{A}_2}{\bullet}](A)$$

$$\beta\colon A \to [\overset{A_1}{\bullet} \mathrm{a} \bullet x_1 \bullet b \overset{\bar{A}_1}{\bullet}, \overset{A_2}{\bullet} c \bullet x_2 \overset{\bar{A}_2}{\bullet}](A)$$

$$\gamma\colon A \to [\overset{A_1}{\bullet} \varepsilon \overset{\bar{A}_1}{\bullet}, \overset{A_2}{\bullet} \varepsilon \overset{\bar{A}_2}{\bullet}]()$$

word $w =$

Derivations and bracket words

$S_1 \bullet \qquad\qquad \bullet \bar{S_1}$

$\alpha\colon S \to [\bullet x_1 \bullet\bullet x_2 \bullet](A)$

$A_1 \bullet \qquad \bar{A_1}\bullet \;\; \bullet A_2 \qquad \bullet \bar{A_2}$

$\beta\colon A \to [\bullet \mathrm{a} \bullet x_1 \bullet b \bullet , \bullet c \bullet x_2 \bullet](A)$

$A_1 \bullet \qquad \bar{A_1}\bullet \;\; \bullet A_2 \qquad \bullet \bar{A_2}$

$\beta\colon A \to [\bullet \mathrm{a} \bullet x_1 \bullet b \bullet , \bullet c \bullet x_2 \bullet](A)$

$A_1 \bullet \bar{A_1} \bullet \;\; \bullet A_2 \bullet \bar{A_2}$

$\gamma\colon A \to [\bullet \varepsilon \bullet , \bullet \varepsilon \bullet]()$

word $w =$

automaton for R:

Derivations and bracket words

$$\alpha\colon S \to [\bullet x_1 \bullet\bullet x_2 \bullet](A)$$

S_1 ... $\bar{S}_1$

$$\beta\colon A \to [\bullet \mathrm{a} \bullet x_1 \bullet b \bullet, \bullet c \bullet x_2 \bullet](A)$$

A_1 ... $\bar{A}_1$ A_2 ... $\bar{A}_2$

$$\beta\colon A \to [\bullet \mathrm{a} \bullet x_1 \bullet b \bullet, \bullet c \bullet x_2 \bullet](A)$$

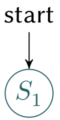
A_1 ... $\bar{A}_1$ A_2 ... $\bar{A}_2$

$$\gamma\colon A \to [\bullet \varepsilon \bullet, \bullet \varepsilon \bullet]()$$

$A_1 \bullet \bar{A}_1 \bullet \quad \bullet A_2 \bullet \bar{A}_2$

word $w =$

automaton for R:



Derivations and bracket words

$S_1 \bullet \qquad \bullet \bar{S}_1$

$$\alpha\colon S \to [\bullet x_1 \bullet\bullet x_2 \bullet](A)$$

$A_1 \bullet \qquad \bar{A}_1 \bullet \quad \bullet A_2 \qquad \bullet \bar{A}_2$

$$\beta\colon A \to [\bullet \mathrm{a} \bullet x_1 \bullet b \bullet, \bullet c \bullet x_2 \bullet](A)$$

$A_1 \bullet \qquad \bar{A}_1 \bullet \quad \bullet A_2 \qquad \bullet \bar{A}_2$

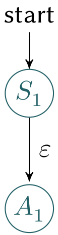
$$\beta\colon A \to [\bullet \mathrm{a} \bullet x_1 \bullet b \bullet, \bullet c \bullet x_2 \bullet](A)$$

$A_1 \bullet \bar{A}_1 \bullet \quad \bullet A_2 \bullet \bar{A}_2$

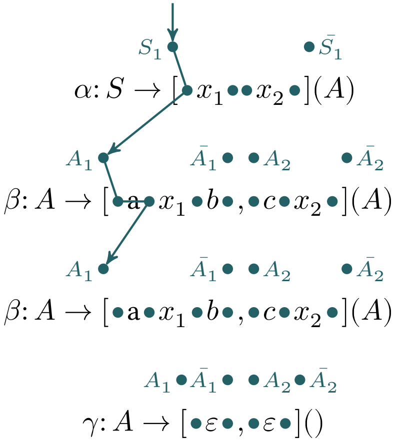
$$\gamma\colon A \to [\bullet \varepsilon \bullet, \bullet \varepsilon \bullet]()$$

word $w =$

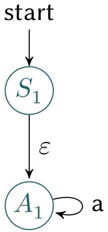
automaton for R:



Derivations and bracket words

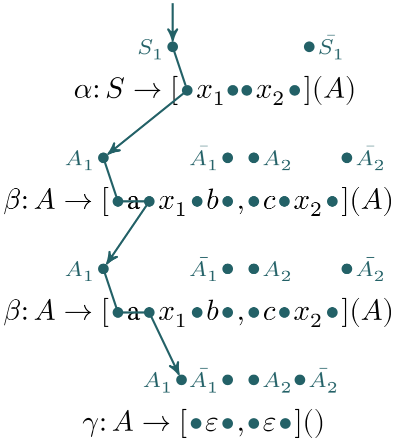


automaton for R:

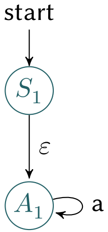


word $w = \mathrm{a}$

Derivations and bracket words

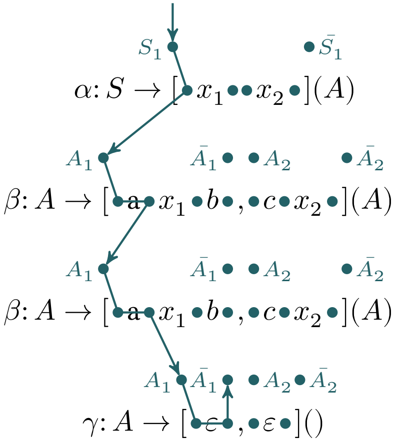


automaton for R:

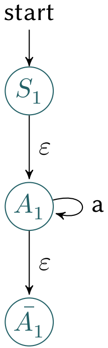


word $w = \mathrm{aa}$

Derivations and bracket words

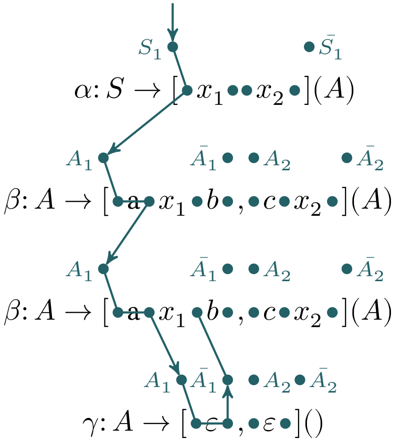


automaton for R:

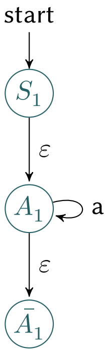


word $w = \mathrm{aa}$

Derivations and bracket words

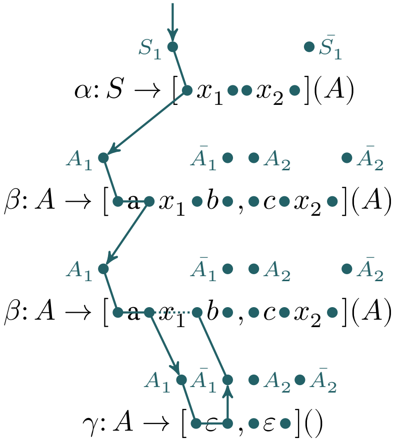


automaton for R:

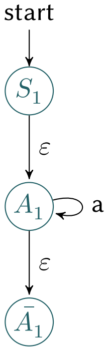


word $w = \mathrm{aa}$

Derivations and bracket words

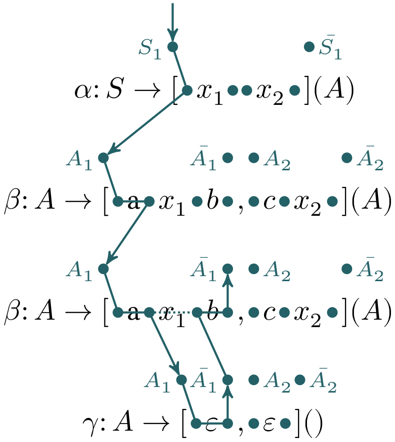


automaton for R:

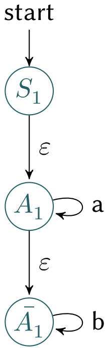


word $w = \text{aa}$

Derivations and bracket words

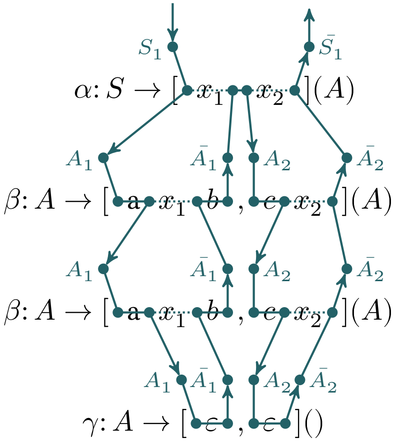


automaton for R:

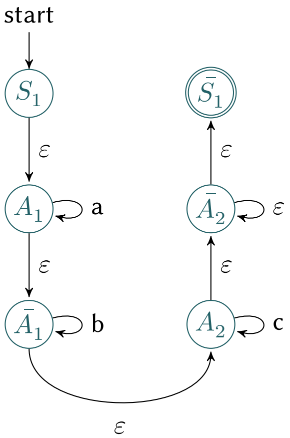


word $w = \text{aab}$

Derivations and bracket words

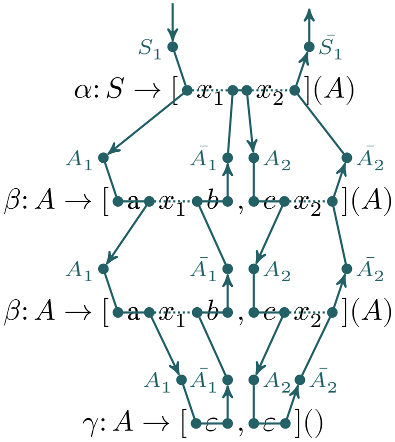


automaton for R:

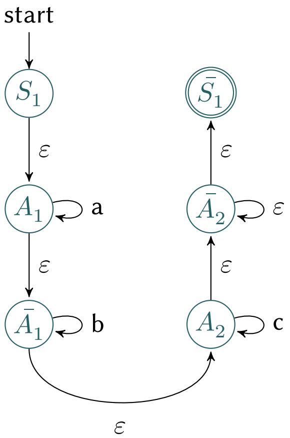


word $w = \text{aabbcc}$

Derivations and bracket words



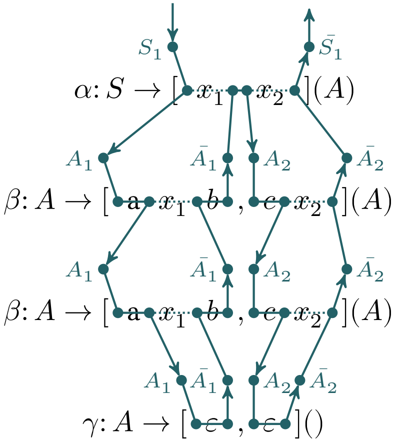
automaton for R:



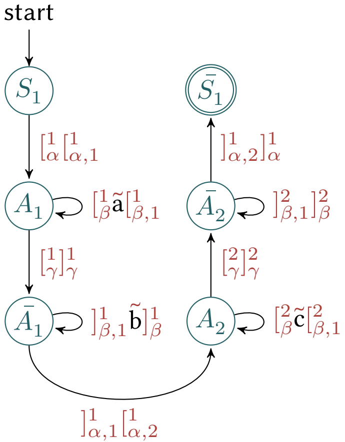
word $w = \text{aabbcc}$

$w' = \text{aaabbcc} \notin \mathrm{L}(G)$

Derivations and bracket words



automaton for R:



word $w = \mathrm{aabbcc}$ $\qquad$ $w' = \mathrm{aaabbcc} \notin \mathrm{L}(G)$

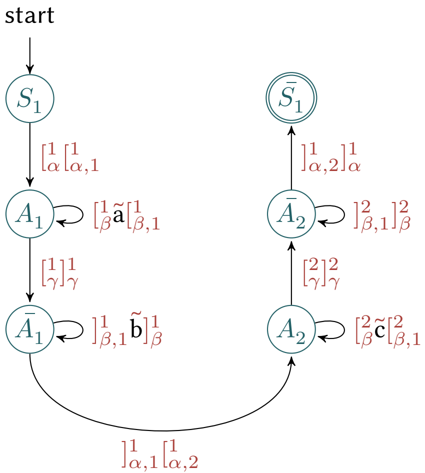
word $u = [^1_\alpha [^1_{\alpha,1}\ [^1_\beta \tilde{\mathrm{a}} [^1_{\beta,1}\ [^1_\beta \tilde{\mathrm{a}} [^1_{\beta,1}\ [^1_\gamma]^1_\gamma\ ]^1_{\beta,1} \tilde{\mathrm{b}}]^1_\beta\ ]^1_{\beta,1} \tilde{\mathrm{b}}]^1_\beta\ ]^1_{\alpha,1}$
$[^1_{\alpha,2}\ [^2_\beta \tilde{\mathrm{c}} [^2_{\beta,1}\ [^2_\beta \tilde{\mathrm{c}} [^2_{\beta,1}\ [^2_\gamma]^2_\gamma\ ]^2_{\beta,1}]^2_\beta\ ]^2_{\beta,1}]^2_\beta\ ]^1_{\alpha,2}]^1_\alpha$

Practical problems

... with the implementation of $\text{sort}^{wt}_{\trianglelefteq}(R \cap h^{-1}(w))$

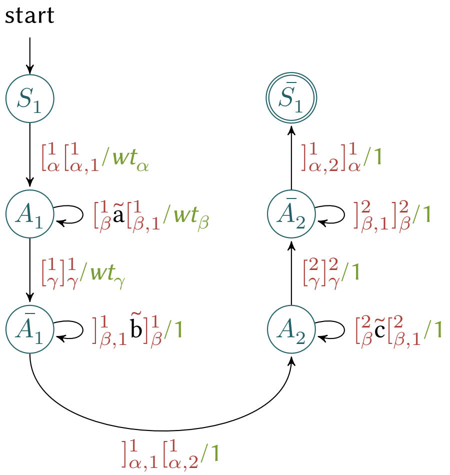
Practical problems

... with the implementation of $\text{sort}^{wt}_{\trianglelefteq}(R \cap h^{-1}(w))$



Practical problems

... with the implementation of $\mathrm{sort}^{wt}_{\trianglelefteq}(R \cap h^{-1}(w))$

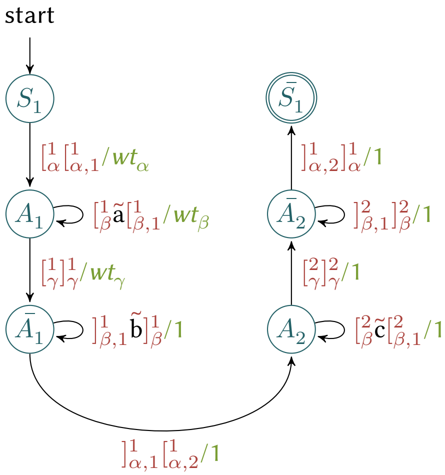


initial idea:

- attach weights to $[^1_\sigma$-brackets

Practical problems

... with the implementation of $\mathrm{sort}^{wt}_{\trianglelefteq}(R \cap h^{-1}(w))$



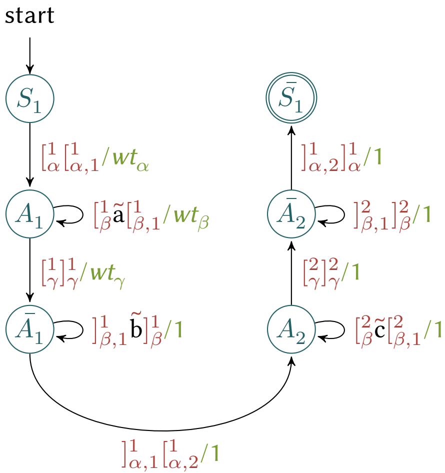
initial idea:

- attach weights to $[^1_\sigma$-brackets

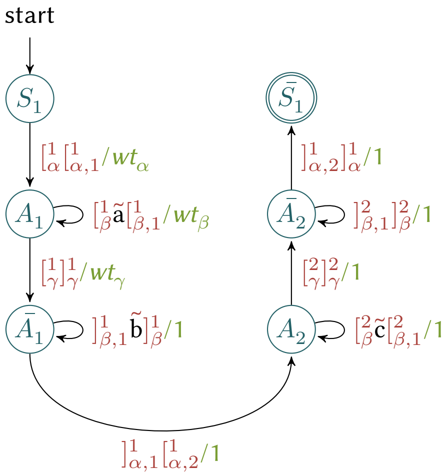
problem:

- loops with weight 1

Solutions and workarounds I

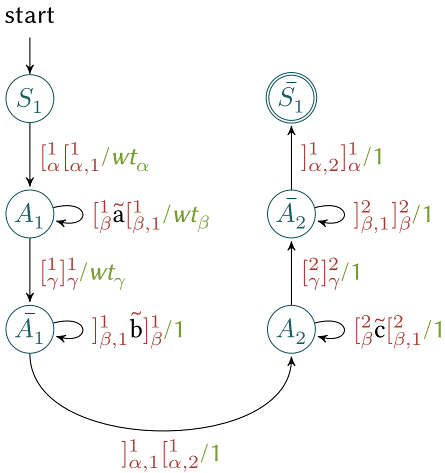


Solutions and workarounds I



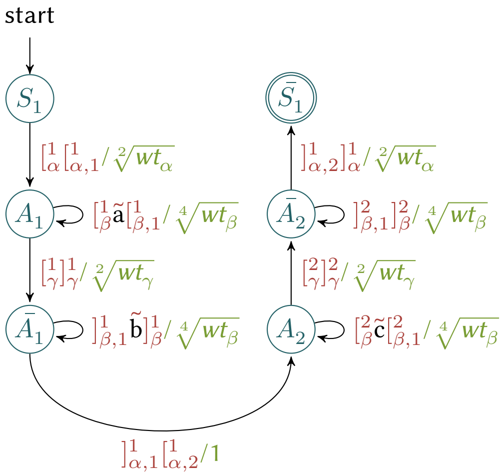
- **assume** that $wt_\sigma \neq 1$ in loops

Solutions and workarounds I



- **assume** that $wt_\sigma \neq 1$ in loops
- **assume** that weights can be *factorised*

Solutions and workarounds I



- **assume** that $wt_\sigma \neq 1$ in loops
- **assume** that weights can be *factorised*
- distribute factors of wt_σ among transitions with $[^1_\sigma,]^1_\sigma, [^2_\sigma,]^2_\sigma, \ldots$

Solutions and workarounds II

Assumption 1: weights wt_σ that occur in loops are $\neq 1$

Solutions and workarounds II

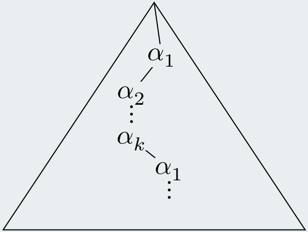
Assumption 1: weights wt_σ that occur in loops are $\neq 1$
$\implies$ restrict the grammar

Solutions and workarounds II

Assumption 1: weights wt_σ that occur in loops are $\neq 1$
$\Longrightarrow$ restrict the grammar

restricted weighted MCFGs:

may not have derivations of the form



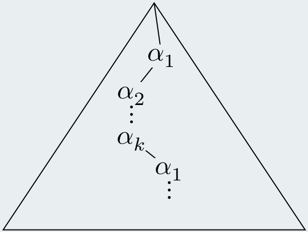
with $wt_{\alpha_1} = \ldots = wt_{\alpha_k} = 1$

Solutions and workarounds II

Assumption 1: weights wt_σ that occur in loops are $\neq 1$
$\Longrightarrow$ restrict the grammar

restricted weighted MCFGs:

may not have derivations of the form



with $wt_{\alpha_1} = \ldots = wt_{\alpha_k} = 1$

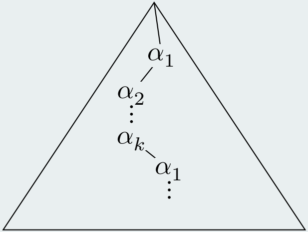
- proper implies restricted

Solutions and workarounds II

Assumption 1: weights wt_σ that occur in loops are $\neq 1$
$\implies$ restrict the grammar

restricted weighted MCFGs:

may not have derivations of the form



with $wt_{\alpha_1} = \ldots = wt_{\alpha_k} = 1$

- proper implies restricted
- $\mathbb{B}$-weighted MCFGs can be transformed to restricted $\mathbb{N}$-weighted MCFGs with the same support

Solutions and workarounds III

Assumption 2: weights can be factorised

Solutions and workarounds III

Assumption 2: weights can be factorised
$\Longrightarrow$ restrict weight algebra

Solutions and workarounds III

Assumption 2: weights can be factorised
$\Longrightarrow$ restrict weight algebra

factorisable (multiplicative) monoid with zero $\mathcal{A}$:

$\forall a \in \mathcal{A} \setminus \{0, 1\} \colon \exists a_1, a_2 \in \mathcal{A} \setminus \{1\} \colon a_1 \cdot a_2 = a$

Solutions and workarounds III

Assumption 2: weights can be factorised
$\Longrightarrow$ restrict weight algebra

factorisable (multiplicative) monoid with zero $\mathcal{A}$:

$\forall a \in \mathcal{A} \setminus \{0, 1\} \colon \exists a_1, a_2 \in \mathcal{A} \setminus \{1\} \colon a_1 \cdot a_2 = a$

example algebra	**factorisation**
$(\mathbb{R}_{\geq 0}, \cdot, 1, 0)$	$a = \sqrt[2]{a} \cdot \sqrt[2]{a}$
$(\mathbb{R}_{\geq 0}^{\infty}, \cdot, 1, \infty)$	$a = \sqrt[2]{a} \cdot \sqrt[2]{a}$
$(\mathbb{R}_{\geq 0}^{-\infty}, +, 1, -\infty)$	$a = a/2 + a/2$

Conclusion

- the CS-theorem can be used for parsing

Conclusion

- the CS-theorem can be used for parsing
- problems arise from 1-weighted rules

Conclusion

- the CS-theorem can be used for parsing
- problems arise from 1-weighted rules
  $\Longrightarrow$ can be circumvented by restricting
  the MCFG and the weight algebra

Conclusion

- the CS-theorem can be used for parsing
- problems arise from 1-weighted rules
  $\Longrightarrow$ can be circumvented by restricting
  the MCFG and the weight algebra
- restrictions are not problematic in practice

Conclusion

- the CS-theorem can be used for parsing
- problems arise from 1-weighted rules
  $\Longrightarrow$ can be circumvented by restricting
  the MCFG and the weight algebra
- restrictions are not problematic in practice

Thank you for your attention.

References

Chomsky, N. and M. P. Schützenberger (1963). "The algebraic theory of context-free languages". DOI: 10.1016/S0049-237X(09)70104-1.
Denkinger, T. (2017). "Chomsky-Schützenberger parsing for weighted multiple context-free languages". DOI: 10.15398/jlm.v5i1.159.
Huang, L. and D. Chiang (2005). "Better k-best Parsing".
Hulden, M. (2011). "Parsing CFGs and PCFGs with a Chomsky-Schützenberger Representation". DOI: 10.1007/978-3-642-20095-3_14.
Seki, H., T. Matsumura, M. Fujii, and T. Kasami (1991). "On multiple context-free grammars". DOI: 10.1016/0304-3975(91)90374-B.
Yoshinaka, R., Y. Kaji, and H. Seki (2010). "Chomsky-Schützenberger-type characterization of multiple context-free languages". DOI: 10.1007/978-3-642-13089-2_50.