Prerequisites: The C-S-Theorem (unweighted, CF)

Language $L$, t.f.a.e.
- $L$ is CF
- A string homomorphism $h$, regular language $R$, Dyck language $D$ s.t.
  \[ L = h(R \cap D) \]
  
  \[ w \in L \iff \exists u \in R \cap D: h(u) = w \]
  
  \[ \iff \exists u \in R \cap D: u \in h^{-1}(w) \]
  
  \[ \iff \exists u \in R \cap h^{-1}(w) \cap D \]

Cf. Chomsky & Schützenberger, 1963
Prerequisites: C-S parsing (unweighted, CF)

\[ w \in L \iff \exists v \in h^{-1}(w) : u \vdash D \]

- Construct FSAs for \( R, h^{-1}(w) \)
- Product construction of FSAs
- Extract Dyck words
- Read off derivation trees of some CFG
Prerequisites:
- Construct fsas for $R, \mu^{-1}(\omega)$

Grammar:
- $\alpha S \rightarrow S S$
- $\beta S \rightarrow A$
- $\gamma A \rightarrow a$

Diagram:
Prerequisites:

- Construct FSAs for $R$, $R_i^{-1}(w)$

Grammar

$\alpha S \rightarrow SS$
$\beta S \rightarrow A$
$\gamma A \rightarrow a$

Word

$w = \alpha \alpha_1 \alpha_2$

$\Delta = \{ (w_i, (\alpha_1, \alpha_2)) | \alpha \in \mathcal{P}, \alpha \in \mathcal{T}(\alpha_1) \} \\ \cup \{ \sigma | \sigma \in \Sigma \}$
Prerequisites:

- Extract Dyck words from product automaton

\[ \Rightarrow \text{put } p \xrightarrow{a} q \in T \text{ in agenda} \]

\[ \Rightarrow \text{for } p \xrightarrow{v} q \text{ in agenda} \]

- put \( p \xrightarrow{v} q \) into results

- for each \( p' \xleftarrow{v} p, q \xrightarrow{v} q' \in T \)
  
  put \( p' \xrightarrow{v} q' \) in agenda

- for each \( p' \xleftarrow{w} p \) in results
  
  put \( p' \xleftarrow{w} q \) in agenda

- for each \( q \xrightarrow{w} q' \) in results
  
  put \( p \xrightarrow{w} q' \) in agenda

- if \( p = q_0, q = q_6 \); return \( v \)
Prerequisites: Chart parsing

- build up chart (using deduction system)
as relation \((N \times N \times N) \times (P \times (N \times N)^*)\)

<table>
<thead>
<tr>
<th>NT + Range</th>
<th>Backtrace</th>
</tr>
</thead>
<tbody>
<tr>
<td>(S, 0, 2)</td>
<td>S \rightarrow SS, (0, 1), (1, 2)</td>
</tr>
<tr>
<td>(S, 0, 1)</td>
<td>S \rightarrow A, (0, 1)</td>
</tr>
<tr>
<td>(A, 0, 1)</td>
<td>A \rightarrow a</td>
</tr>
<tr>
<td>(S, 1, 2)</td>
<td>S \rightarrow A, (1, 2)</td>
</tr>
<tr>
<td>(A, 1, 2)</td>
<td>A \rightarrow a</td>
</tr>
</tbody>
</table>

C.E. Huang & Chiang, 2005
So, how is Chomsky–Schützenberger parsing the same as Chart parsing?
unweighted Chomsky-Schützenberger parsing for CFG is unweighted chart parsing for CFG
A closer look at the product FSA

- \( R \) w/l states \( \{ \overline{A}, A | A \in \mathbb{N} \} \) (and some uniques)
- \( h^{-1}(w) \) w/l states \( [0, |w|] \)

\( \rightarrow \) Consider runs from \((A, i)\) to \((\overline{A}, j)\) as NT A spanning range \( \langle i, j \rangle \)

\((i f \text{ brackets are Dyck})\)
A closer look at the product $\text{fsa}$
A closer look at the extraction of Dyck words

- Efficient extraction w/ dynamic programming
  → store information about how to produce intermediate results
  - initial \( \rho \xrightarrow{\alpha} \eta \)
  - by concatenation
    \( (\rho \xrightarrow{u} \rho', \rho' \xrightarrow{v} \eta) \Rightarrow \rho^{uv} \xrightarrow{} \eta \)
  - by enbracketing
    \( (\rho \xrightarrow{} \rho', \rho' \xrightarrow{u} \eta', \eta' \xrightarrow{} \eta) = \Rightarrow (\rho^{(u)} \xrightarrow{} \eta) \)
A closer look at the extraction of Dyck words

- build up chart

<table>
<thead>
<tr>
<th>State x State</th>
<th>Backtrace</th>
</tr>
</thead>
<tbody>
<tr>
<td>(p, q)</td>
<td>init(\tilde{a})</td>
</tr>
<tr>
<td></td>
<td>concat(p, p', q)</td>
</tr>
<tr>
<td></td>
<td>enbracket(&quot;&quot;,&quot;p', q', &quot;,&quot;)</td>
</tr>
</tbody>
</table>

- evaluate chart starting with (q_0, q_f)
### State x State

<table>
<thead>
<tr>
<th>State x State</th>
<th>Backtrace</th>
</tr>
</thead>
<tbody>
<tr>
<td>((s_0, \bar{s}_2))</td>
<td>enbracket((\alpha ), ((s_0, \bar{s}_2)), (\beta ))</td>
</tr>
<tr>
<td>((s_0, 0, 1))</td>
<td>enbracket((\gamma ), ((u_1, 0)), ((u_2, 1)), (\gamma ))</td>
</tr>
<tr>
<td>((u_0, 1)), ((s_2))</td>
<td>init((\bar{\alpha} ))</td>
</tr>
<tr>
<td>((u_0, 1)), ((0, 1))</td>
<td>enbracket((\beta ), ((u_0, 1)), ((0, 1)), (\beta ))</td>
</tr>
</tbody>
</table>

### Backtrace

- \(\alpha\)
- \((s_0, 0, 1)\)
- \((u_1, 0)\)
- \((u_2, 1)\)
- \((s_1, 0, 1)\)
- \((\bar{\alpha} \)
Putting both together

<table>
<thead>
<tr>
<th>$N \times \text{Range}$</th>
<th>$(\Delta \cup (N \times \text{Range}))^*$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S, \langle 0, 1 \rangle$</td>
<td>$(\gamma \tilde{a})^*_y$</td>
</tr>
<tr>
<td>$S, \langle 0, 1 \rangle$</td>
<td>$(^1 \Delta \langle 0, 1 \rangle)^*_y$</td>
</tr>
<tr>
<td>$A, \langle 0, 1 \rangle$</td>
<td>$(^\gamma \tilde{a})^*_y$</td>
</tr>
<tr>
<td>$S, \langle 1, 2 \rangle$</td>
<td>$(^1 \Delta \langle 1, 2 \rangle)^*_y$</td>
</tr>
<tr>
<td>$A, \langle 1, 2 \rangle$</td>
<td>$(^\gamma \tilde{a})^*_y$</td>
</tr>
</tbody>
</table>
Putting both together

<table>
<thead>
<tr>
<th>$N \times N \times N$</th>
<th>$P \times (N \times N)^k$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S, 0, 2$</td>
<td>$\kappa_i(S, 0, 1), (S, 1, 2)$</td>
</tr>
<tr>
<td>$S, 0, 1$</td>
<td>$\beta_i(A, 0, 1)$</td>
</tr>
<tr>
<td>$A, 0, 1$</td>
<td>$\gamma$</td>
</tr>
<tr>
<td>$S, 1, 2$</td>
<td>$\beta_i(A, 1, 2)$</td>
</tr>
<tr>
<td>$A, 1, 2$</td>
<td>$\gamma$</td>
</tr>
</tbody>
</table>
But wait, there's more!

<table>
<thead>
<tr>
<th></th>
<th>$\text{CFG}_i$</th>
<th>$\text{LCFRS}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>un-$W$</td>
<td>✓</td>
<td>$\text{CFGq}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\text{approx}$</td>
</tr>
<tr>
<td>$W$</td>
<td>✓</td>
<td>$\text{wCFGq}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\text{approx}$</td>
</tr>
</tbody>
</table>
Chomsky-Schützenberger parsing for CFG is Chart parsing for CFG.