Automata Theory :: CYK Parsing

Jörg Endrullis

Vrije Universiteit Amsterdam

Bottom-up Parsing

Bottom-up parsing applies rules backwards, it tries to construct the starting variable S from the input word.

Cocke-Younger-Kasami algorithm (1965)

The **CYK algorithm** is a bottom-up parsing technique for grammars in Chomsky normal form.

Cocke-Younger-Kasami Algorithm (1965)

Let G be a grammar in Chomsky normal form.

Goal: determine whether word $w \neq \lambda$ is in $L(G)$.

Idea: compute sets V_u of variables (u subword of w) such that

$$V_u = \{ A \in V \mid A \Rightarrow^+ u \}$$

as follows:

- if $|u| = 1$, then $V_u = \{ A \in V \mid A \to u \in P \}$
- if $|u| > 1$, then V_u is the set of all $A \in V$ such that
  - $u = u_1 u_2$ for some non-empty words u_1, u_2, and
  - $A \to BC \in P$ with $B \in V_{u_1}$ and $C \in V_{u_2}$.

Finally, $w \in L(G) \Leftrightarrow S \in V_w$.

Worst-case time complexity: $O(n^3)$
(There are $n(n+1)/2$ sets V_u, and computation of V_u is $O(n)$.)

Exercise

Use the CYK algorithm to check whether *abbb* is generated by

$$S \to AB \qquad A \to BB \mid a \qquad B \to AB \mid b$$

We have

$$V_a = \{A\} \quad \text{since } A \to a$$

$$V_b = \{B\} \quad \text{since } B \to b$$

$$V_{ab} = \{X \mid X \to V_a V_b = \{AB\}\} = \{S, B\}$$

$$V_{bb} = \{X \mid X \to V_b V_b = \{BB\}\} = \{A\}$$

$$V_{abb} = \{X \mid X \to V_a V_{bb} \cup V_{ab} V_b = \{AA, SB, BB\}\} = \{A\}$$

$$V_{bbb} = \{X \mid X \to V_b V_{bb} \cup V_{bb} V_b = \{BA, AB\}\} = \{S, B\}$$

$$V_{abbb} = \{X \mid X \to V_a V_{bbb} \cup V_{ab} V_{bb} \cup V_{abb} V_b\}$$

$$= \{X \mid X \to \{AS, AB, SA, BA\}\} = \{S, B\}$$

The word *abbb* is in the language since $S \in V_{abbb}$:

$$\underbrace{S}_{abbb} \to \underbrace{A}_{a}\underbrace{B}_{bbb} \to \underbrace{A}_{a}\underbrace{A}_{bb}\underbrace{B}_{b} \to \underbrace{A}_{a}\underbrace{B}_{b}\underbrace{B}_{b}\underbrace{B}_{b} \to^4 abbb$$