14.5.2
Formal description of algorithm

Recursive solution

1. Input: $w = w_1 w_2 \dots w_n$
2. Assume r non-terminals in G: $R_1, \dots, R_r$.
3. R_1: Start symbol.
4. $f(\ell, s, b)$: TRUE $\iff$ $w_s w_{s+1} \dots, w_{s+\ell-1} \in L(R_b)$.
   $=$ Substring w starting at pos ℓ of length s is deriveable by R_b.
5. **Recursive formula:** $f(1, s, a)$ is 1 $\iff$ $\left(R_a \to w_s\right) \in G$.
6. For $\ell > 1$: $f\left(\text{length}, \text{start pos}, \text{variable index}\right)$

$$f(\ell, s, a) = \bigvee_{\mu=1}^{\ell-1} \bigvee_{(R_a \to R_\beta R_\gamma) \in G} \Big(f(\mu, s, \beta) \wedge f(\ell - \mu, s + \mu, \gamma) \Big)$$

7. **Output:** $w \in L(G) \iff f(n, 1, 1) = 1$.

Recursive solution

1. Input: $w = w_1 w_2 \ldots w_n$
2. Assume r non-terminals in G: $R_1, \ldots, R_r$.
3. R_1: Start symbol.
4. $f(\ell, s, b)$: TRUE $\iff$ $w_s w_{s+1} \ldots, w_{s+\ell-1} \in L(R_b)$.
   $=$ Substring w starting at pos ℓ of length s is deriveable by R_b.
5. **Recursive formula:** $f(1, s, a)$ is $1 \iff \left(R_a \to w_s\right) \in G$.
6. For $\ell > 1$: $f\left(\text{length}, \text{start pos}, \text{variable index}\right)$

$$f(\ell, s, a) = \bigvee_{\mu=1}^{\ell-1} \bigvee_{(R_a \to R_\beta R_\gamma) \in G} \Big(f(\mu, s, \beta) \wedge f(\ell - \mu, s + \mu, \gamma) \Big)$$

7. **Output:** $w \in L(G) \iff f(n, 1, 1) = 1$.

Recursive solution

1. Input: $w = w_1 w_2 \ldots w_n$
2. Assume r non-terminals in G: $R_1, \ldots, R_r$.
3. R_1: Start symbol.
4. $f(\ell, s, b)$: TRUE $\iff$ $w_s w_{s+1} \ldots, w_{s+\ell-1} \in L(R_b)$.
   $=$ Substring w starting at pos ℓ of length s is deriveable by R_b.
5. **Recursive formula:** $f(1, s, a)$ is $1 \iff \left(R_a \to w_s\right) \in G$.
6. For $\ell > 1$: $f\left(\text{length}, \text{start pos}, \text{variable index}\right)$

$$f(\ell, s, a) = \bigvee_{\mu=1}^{\ell-1} \bigvee_{(R_a \to R_\beta R_\gamma) \in G} \left(f(\mu, s, \beta) \wedge f(\ell - \mu, s + \mu, \gamma) \right)$$

7. **Output:** $w \in L(G) \iff f(n, 1, 1) = 1$.

Analysis

Assume $G = \{R_1, R_2, \ldots, R_r\}$ with start symbol R_1

- $f\big(\underbrace{\text{length}}_{n}, \underbrace{\text{start pos}}_{n}, \underbrace{\text{variable index}}_{r}\big)$.
- Number of subproblems: $O(rn^2)$
- Space: $O(rn^2)$
- Time to evaluate a subproblem from previous ones: $O(|P|n)$
  P is set of rules
- Total time: $O(|P|rn^3)$ which is polynomial in both $|w|$ and $|G|$. For fixed G the run time is cubic in input string length.
- Running time can be improved to $O(n^3|P|)$.
- Not practical for most programming languages. Most languages assume restricted forms of CFGs that enable more efficient parsing algorithms.

Analysis

Assume $G = \{R_1, R_2, \ldots, R_r\}$ with start symbol R_1

- f(length, start pos, variable index).
- Number of subproblems: $O(rn^2)$
- Space: $O(rn^2)$
- Time to evaluate a subproblem from previous ones: $O(|P|n)$
  P is set of rules
- Total time: $O(|P|rn^3)$ which is polynomial in both $|w|$ and $|G|$. For fixed G the run time is cubic in input string length.
- Running time can be improved to $O(n^3|P|)$.
- Not practical for most programming languages. Most languages assume restricted forms of CFGs that enable more efficient parsing algorithms.

Analysis

Assume $G = \{R_1, R_2, \ldots, R_r\}$ with start symbol R_1

- $f(\text{length}, \text{start pos}, \text{variable index})$.
- Number of subproblems: $O(rn^2)$
- Space: $O(rn^2)$
- Time to evaluate a subproblem from previous ones: $O(|P|n)$
  P is set of rules
- Total time: $O(|P|rn^3)$ which is polynomial in both $|w|$ and $|G|$. For fixed G the run time is cubic in input string length.
- Running time can be improved to $O(n^3|P|)$.
- Not practical for most programming languages. Most languages assume restricted forms of CFGs that enable more efficient parsing algorithms.

$O(n^3)$

CYK Algorithm

Input string: $X = x_1 \ldots x_n$.
Input grammar G: r nonterminal symbols $R_1 \ldots R_r$, R_1 start symbol.

$P[n][n][r]$: Array of booleans. Initialize all to FALSE
for $s = 1$ to n **do**
    for each unit production $R_v \to x_s$ **do**
        $P[1][s][v] \leftarrow$ TRUE
for $\ell = 2$ to n **do**   // Length of span
    for $s = 1$ to $n - \ell + 1$ **do**   // Start of span
        for $\mu = 1$ to $\ell - 1$ **do**   // Partition of span
            for all $(R_a \to R_\beta R_\gamma) \in G$ **do**
                if $P[\overset{\mu}{\cancel{p}}][s][\beta]$ and $P[\ell - \mu][s + \mu][\gamma]$ **then**
                    $P[\ell][s][a] \leftarrow$ TRUE
if $P[n][1][1]$ is TRUE **then**
    return ``X is member of language''
else
    return ``X is not member of language''

THE END

...

(for now)