Equivalência entre GLC e AP

Pode-se demonstrar que GLCs e APs são formalismos com o mesmo poder computacional. Ou seja, ambos são equivalentes e capazes de computar linguagens livre de contexto.

Aqui vamos ver como obter um AP equivalente a uma GLC. A obtenção de uma GLC equivalente a um AP, embora esteja demonstrada no livro texto, é razoavelmente mais complexa e não discutiremos nessa aula.

Lema: Para toda GLC G, existe um AP que reconhece L(G).

Prova:

Vamos discutir somente a ideia por trás da prova.

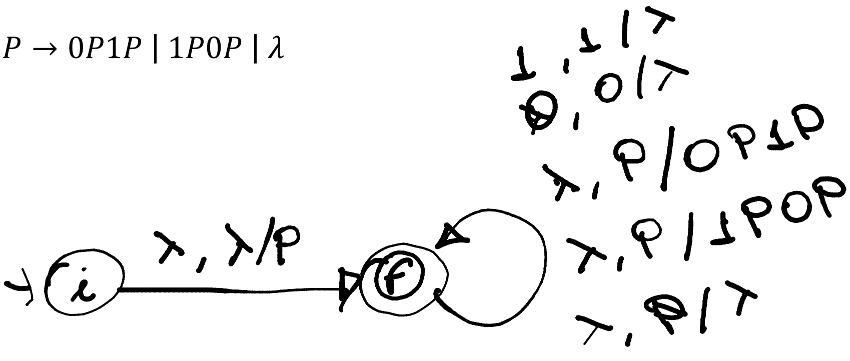
Intuitivamente, vamos criar um AP que simule as derivações de G. Nosso AP deve iniciar sua computação empilhando a variável de partida. Em seguida, entramos em um laço em que as transições correspondem às regras da gramática. A pilha será usada para armazenar as formas sentenciais. Sempre que um símbolo da entrada corresponder ao topo da pilha, esse é desempilhado e consumido da entrada.

De maneira mais formal, seja $G = (V, \Sigma, R, P)$ a gramática considerada, o AP $M = (\{i, f\}, \Sigma, V \cup \Sigma, \delta, \{i\}, \{f\})$ cuja função de transição é a seguinte:

- $\delta(i, \lambda, \lambda) = \{[f, P]$
- Para $X \in V$, $\delta(f, \lambda, X) = \{[f, z] \mid X \to z \in R\}$
- Para $a \in \Sigma$, $\delta(f, a, a) = \{[f, \lambda]\}$

Exemplo: Considere a gramática abaixo que gera a linguagem $\{w \in \{0,1\}^* \mid \eta_0(w) = \eta_1(w)\}$

- $P \to 0P1P \mid 1P0P \mid \lambda$



$[i, 01, \lambda] \vdash [f, 01, P]$

$\vdash [f, 01, 0P1P]$

01

$P \Rightarrow 0P1P \Rightarrow 01P \Rightarrow 01$

$\vdash [f, 1, P1P]$

$\vdash [f, 1, 1P]$

$\vdash [f, \perp, \perp p]$

$\vdash [f, \lambda, p]$

$\vdash [f, \lambda, \lambda]$