Autômatos com Pilha

Douglas O. Cardoso
douglas.cardoso@cefet-rj.br
docardoso.github.io

CEFET/RJ

Roteiro

Roteiro

Motivação: expressões aritméticas

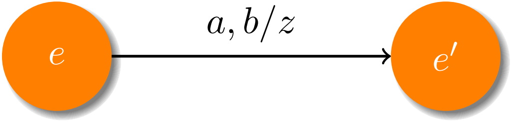
- Seja L a linguagem de todas as expressões aritméticas sem parênteses.
  - 123, 1+2, 2*3/5, etc.
- Determine o AF A, ER E e GR G t.q. $L(A) = L(E) = L(G) = L$.
- Seja L' a linguagem de todas as expressões aritméticas.
- Mostre usando o Lema do Bombeamento que L' não é regular.

Como reconhecer expressões aritméticas?

- É necessário lembrar, i.e. contar, quantos parênteses foram abertos.
- Num AF, a memória são os estados, finitos e assim insuficientes.
- Ideia: utilizar uma pilha infinita como memória auxiliar do AF.

Roteiro

Descrição informal



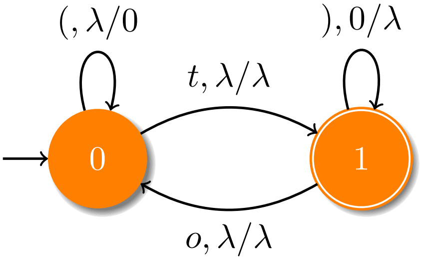
$$a \in \Sigma \cup \{\lambda\}$$

$b \in \Gamma \cup \{\lambda\}$ é tirado do topo da pilha

$z \in \Gamma^*$ é colocado no topo da pilha

$$\delta(e, a, b) = (e', z)$$

Exemplo: reconhecedor de expressões aritméticas



Exemplo de processamento:

$[0, (t-(t+t)), \lambda] \vdash [0, t-(t+t)), 0] \vdash [1, -(t+t)), 0] \vdash [0, (t+t)), 0] \vdash [0, t+t)), 00] \vdash [1, +t)), 00] \vdash [0, t)), 00] \vdash [1,)), 00] \vdash [1,), 0] \vdash [1, \lambda, \lambda]$

Definição formal

- Um APD $(E, \Sigma, \Gamma, \delta, i, F)$ é definido em parte como um AFD:
  - Alfabeto de entrada, Σ;
  - Conjuntos de estados, E;
  - Estado inicial, $i \in E$;
  - Conjunto de estados finais, $F \subset E$.
- As diferenças entre ambos são duas:
  - Um APD também possui um alfabeto da pilha, Γ, definido livremente;
  - A função (parcial) de transição de um APD é $\delta : E \times (\Sigma \cup \{\lambda\}) \times (\Gamma \cup \{\lambda\}) \to E \times \Gamma^*$.

Configuração instantânea e linguagem aceita

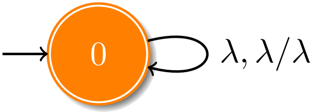
- A configuração instantânea de um AF é o par $[e, w] : e \in E, w \in \Sigma^*$.
- No caso de um AP, temos o trio $[e, w, p] : p \in \Gamma^*$.
- Uma computação $[e, ay, bz] \vdash [e', y, xz]$ é válida sss $\delta(e, a, b) = (e', x)$.
- $[e, x, y] \vdash^* [e', y, xz]$ é válida sss $[e, ay, bz] \vdash \cdots \vdash [e', w, z]$
- A linguagem aceita por um APD $(E, \Sigma, \Gamma, \delta, i, F)$ é

$$\{w \in \Sigma^* : [i, w, \lambda] \vdash^* [e, \lambda, \lambda], e \in F\}$$

.

Exercícios

- Determine os APDs que aceitem as linguagens:
    - $\{0^n 1^n : n \geq 0\}$;
    - $\{w \in \{0,1\}^* :$ o número de 0s e 1s em w é igual$\}$.
- Qual a linguagem aceita pelo APD a seguir?



Roteiro

Motivação

- Como é o APD que aceita a linguagem $\{0^m1^n : m < n\}$?
- E como é o APD que aceita a linguagem $\{w0w^R : w \in \{1,2\}^*\}$?
- E para as linguagem $\{0^m1^n : m > n\}$ e $\{ww^R : w \in \{1,2\}^*\}$, existem APDs correspondentes?

A importância do não-determinismo para APs

- Diferente do caso de AFs, um APN tem mais poder de reconhecimento que um APD.

- Logo, não é possível converter um APN em um APD correspondente.

- Considerando também que APNs tem um conjunto de estados iniciais, a linguagem aceita por um APN $(E, \Sigma, \Gamma, \delta, I, F)$ é:

$$\{w \in \Sigma^* : [i, w, \lambda] \vdash^* [e, \lambda, \lambda], i \in I, e \in F\}.$$

Compatibilidade de transições

- Considere um AP qualquer.
- Duas das sua transições $\delta(e, a, b)$ e $\delta(e, a', b')$ são ditas compatíveis se

$$(a = a' \vee a = \lambda \vee a' = \lambda) \wedge (b = b' \vee b = \lambda \vee b' = \lambda)$$

.

- A principal diferença entre APDs e APNs é que os últimos permitem transições compatíveis, e os primeiros não.

Exercícios

- Determine os APNs que aceitem as linguagens:
  - $\{w \in \{0,1\}^* :$ o número de 0s e 1s em w é igual$\}$ (menor que o APD!);
  - $\{w \in \{0,1\}^* : w = w^R\}$;
  - $\{0^m 1^n : m \geq n\}$.