UNIVERSIDADE DE ÉVORA
DEPARTAMENTO DE INFORMÁTICA

Linguagens Formais e Autómatos
Exame (normal) Resolvido
14 de Janeiro de 2015

Exercício 1 [4 valores] Aplique os algoritmos dados nas aulas para encontrar um autómato finito determinista mínimo equivalente ao AFND $A = (\{q_0, q_1, q_2, q_3\}, \{a, b\}, \delta, q_0, \{q_3\})$ com a transição δ definida pela tabela seguinte

q	a	b	λ
q_0	$\{q_1, q_2\}$		
q_1		$\{q_1, q_3\}$	
q_2	$\{q_2\}$	$\{q_2\}$	$\{q_1\}$
q_3	$\{q_3\}$	$\{q_3\}$	

AFD equivalente

p	a	b
$p_0 = \{q_0\}$	$\{q_1, q_2\}$	$\emptyset$
$p_1 = \{q_1, q_2\}$	$\{q_1, q_2\}$	$\{q_1, q_2, q_3\}$
$p_2 = \emptyset$	$\emptyset$	$\emptyset$
$p_3 = \{q_1, q_2, q_3\}$	$\{q_1, q_2, q_3\}$	$\{q_1, q_2, q_3\}$

AFD mínimo equivalente
O AFD acima já é mínimo.

Exercício 2 [2 valores] Considere o alfabeto $\Sigma = \{a, b\}$. Defina um AFND para reconhecer a linguagem $A \subseteq \Sigma^*$ das palavras que começam por aa e um AFND para reconhecer a linguagem $B \subseteq \Sigma^*$ das palavras que terminam em ab. **Use esses dois autómatos** para definir um terceiro AFND para reconhecer a linguagem das palavras que começam por aa e que terminam em ab.

Um AFND **bem preparado** para reconhecer A é $M_A = (Q_A = \{p_0, p_1, p_2, p_3\}, \{a, b\}, \delta_A, p_0, \{p_3\})$ com transição δ_A dada pela tabela

p	a	b	λ
p_0	$\{p_1\}$		
p_1	$\{p_2\}$		
p_2	$\{p_2\}$	$\{p_2\}$	$\{p_3\}$
p_3			

Um AFND **bem preparado** para reconhecer B é $M_B = (Q_B = \{q_0, q_1, q_2, q_3\}, \{a, b\}, \delta_B, q_0, \{q_3\})$ com transição δ_B dada pela tabela

q	a	b	λ
q_0			$\{q_1\}$
q_1	$\{q_1, q_2\}$	$\{q_1\}$	
q_2		$\{q_3\}$	
q_3			

O AFND pedido é $M_C = (Q_A \times Q_B, \{a, b\}, \delta_C, (p_0, q_0), \{(p_3, q_3)\})$ com transição δ_C dada pela regra

$$\begin{aligned} (p', q') &\in \delta_C((p, q), x) \\ &\Leftrightarrow \begin{cases} p' & \in \delta_A(p, x) \\ q' & \in \delta_B(q, x) \end{cases} \end{aligned}$$

Exercício 3 Considere a GIC $G = (\{S\}, \{a\}, \{S \to SS \mid a\}, S)$.

1. **[0,5 valores]** Mostre que G é ambígua;
2. **[1,5 valores]** Apresente uma GIC não ambígua equivalente a G;

A gramática dada é ambígua porque a palavra de terminais aaa tem duas derivações direitas distintas:

$$\begin{aligned} S &\Rightarrow_R SS \Rightarrow_R SSS \Rightarrow_R SSa \Rightarrow_R Saa \Rightarrow_R aaa \\ S &\Rightarrow_R SS \Rightarrow_R Sa \Rightarrow_R SSa \Rightarrow_R Saa \Rightarrow_R aaa \end{aligned}$$

Uma versão não ambígua desta gramática é

$$S \to aS \mid a$$

Exercício 4 Seja L a linguagem $\{a^n b^m c^{2n} \ : \ n, m \geq 0\}$.

1. **[1 valor]** Defina um autómato de pilha para reconhecer L;
2. **[1 valor]** Para a demonstração de que L não é regular, que palavra p usaria? Que decomposição (ou decomposições) de $p = uvw$ consideraria? Como, a partir daí, concluíria que L não é regular?

Um autómato de pilha para reconhecer a linguagem dada é

$$A = (Q = \{p_0, p\}, \Sigma = \{a, b, c\}, \Gamma = \{X\}, \delta, p_0, F = \{p\})$$

com a transição δ dada por

$$\begin{aligned} p_0 &\xrightarrow{a,\lambda/XX} p_0 \\ p_0 &\xrightarrow{b,\lambda/\lambda} p \\ p &\xrightarrow{c,X/\lambda} p \end{aligned}$$

Para demonstrar que L não é regular, com vista a uma contradição: supondo que L é regular existe um AFD A que reconhece L. Seja K o número de estados de controlo de A e

$$p = a^K c^{2K} \in L \qquad \text{com } n = K; m = 0$$

Então, pelo *pumping lemma*,

$$p = a^K c^{2K} = uvw \qquad \text{com } |uv| < K; |v| > 0$$

Portanto tem de ser

$$
\begin{aligned}
u &= a^U && \\
v &= a^V && \text{com } V > 0 \\
U + V < K \Rightarrow K &= U + V + W && \text{com } W = K - U - V \\
w &= a^W c^{2K} && \\
p &= a^U a^V a^W c^{2K} = a^{U+V+W} c^{2K} &&
\end{aligned}
$$

Ainda pelo *pumping lemma* tem de ser

$$p' = a^U a^{nV} a^W c^{2K} = a^{U+nV+W} c^{2K} \in L \qquad \forall n \geq 0$$

Fazendo, por exemplo, $n = 0$ obtém-se

$$a^{U+W} c^{2K} \in L$$

Mas $U + W < K$ porque $K = U + V + W$ e $V > 0$. Portanto a palavra $a^{U+W} c^{2K}$ não respeita a condição $a^n b^m c^{2n}$ que define L, o que é uma **contradição**.
Esta contradição resulta da suposição inicial que L seria uma linguagem regular. Portanto L não pode ser uma linguagem regular.

Exercício 5 [2 valores] Seguindo o processo dado nas aulas, obtenha a *Forma Normal de Greibach* para a gramática

$$G = (\{S, L\}, \{a, b, c\}, \{S \to ab \mid aLb, L \to c \mid LL \mid S\}, S).$$

O símbolo inicial da gramática não é recursivo e a gramática é não-contraível.
Existe uma produção unitária: $L \to S$. Aplicando o método de remoção de produções unitárias obtém-se a **gramática equivalente** com produções

$$
\begin{aligned}
S &\to ab \mid aLb \\
L &\to c \mid LL \mid ab \mid aLb
\end{aligned}
$$

Esta gramática está **limpa**.
Uma gramática equivalente na **forma normal de Chomsky** é

$$
\begin{aligned}
S &\to AB \mid DB \\
L &\to c \mid LL \mid AB \mid DB \\
D &\to AL \\
A &\to a \\
B &\to b
\end{aligned}
$$

Há um caso de **recursão directa à esquerda**. A gramática que se obtém removendo esse caso é

$$
\begin{aligned}
S &\to AB \mid DB \\
Z &\to LZ \mid L \\
L &\to cZ \mid ABZ \mid DBZ \\
&\quad \mid c \mid AB \mid DB \\
D &\to AL \\
A &\to a \\
B &\to b
\end{aligned}
$$

Ordenando os não-terminais por $S < Z < L < D < A < B$ as produções ficam como estão. Substituindo de baixo para cima os primeiros não-terminais de cada produção, obtém-se a gramática

$$
\begin{aligned}
S &\to aB \mid aLB \\
Z &\to cZZ \mid aBZZ \mid aLBZZ \mid cZ \mid aBZ \mid aLBZ \\
&\quad \mid cZ \mid aBZ \mid aLBZ \mid c \mid aB \mid aLB \\
L &\to cZ \mid aBZ \mid aLBZ \\
&\quad \mid c \mid aB \mid aLB \\
D &\to aL \\
A &\to a \\
B &\to b
\end{aligned}
$$

que está na **forma normal de Greibach**.

Exercício 6 [2 valores] Justifique se a gramática $G = (\{S, A\}, \{a, b, c\}, \{S \to aSb \mid A, A \to aAc \mid \lambda\}, S)$ é, ou não, $\mathcal{LL}(1)$.

O conjunto dos **produtores da palavra vazia** é

$$\Lambda = \{S, A\}$$

Os conjuntos dos **primeiros** e **seguintes** de cada não-terminal são

NT	PRIMEIROS	SEGUINTES
S	$\{a\}$	$\{b\}$
A	$\{a\}$	$\{b, c\}$

Os **directores** da produção $S \to aSb$ são $\{a\}$ e da produção $S \to A$ são $\{a, b\}$.
Como o não-terminal S tem produções distintas cujos símbolos directores se intersectam, **a gramática não é** $\mathcal{LL}(1)$.

Exercício 7 Considere a gramática $G = (\{S, A, B\}, \{a, b\}, \ldots, S)$ com produções

$$S \to AB \mid bAB \qquad A \to aA \mid \lambda \qquad B \to Bb \mid \lambda$$

- **[2,5 valores]** Mostre que esta gramática **não é** $\mathcal{LR}(0)$ e identifique os tipos de conflitos que acontecem;

- **[2,5 valores]** Mostre que esta gramática **não é** $\mathcal{LR}(1)$ e identifique os critérios que falham. **Sugestão:** *comece a construir o autómato finito dos itens $\mathcal{LR}(1)$ válidos e em cada estado que obtém confirme se são verificados os critérios $\mathcal{LR}(1)$;*

Para verificar que a gramática não é $\mathcal{LR}(0)$.
Na construção do autómato dos itens $\mathcal{LR}(0)$ válidos o estado inicial tem os itens

$$
\begin{aligned}
S &\to \cdot AB\\
S &\to \cdot bAB\\
A &\to \cdot aA\\
A &\to \cdot
\end{aligned}
$$

O último é um **item completo**. Portanto, nesse estado:

1. mais nenhum item pode ser completo (senão haveria um conflito redução/redução);
2. nos restantes itens o ponto, $\cdot$, não pode estar antes de um terminal (senão haveria um conflito transferência/redução);

Mas a segunda condição é violada pelos itens $A \to \cdot aA$ e $S \to \cdot bAB$. Portanto esta gramática não é $\mathcal{LR}(0)$ porque produz conflitos transferência/redução.
Para verificar que a gramática não é $\mathcal{LR}(1)$. Considere-se a gramática equivalente, estendida com a produção $S' \to S\#$.
Os **não-terminais que produzem a palavra vazia** são:

$$\Lambda = \{S, A, B\}$$

Os **primeiros** e **seguintes** de cada não-terminal são

NT	PRIMEIROS	SEGUINTES
S	$\{a, b, \#\}$	$\{\#\}$
A	$\{a\}$	$\{b, \#\}$
B	$\{b\}$	$\{b, \#\}$

Na construção do autómato dos itens $\mathcal{LR}(1)$ válidos o estado inicial tem os itens

$$
\begin{aligned}
S &\to \cdot AB &&; \#\\
S &\to \cdot bAB &&; \#\\
A &\to \cdot aA &&; b, \#\\
A &\to \cdot &&; b, \#
\end{aligned}
$$

O último é um item completo: $[A \to \cdot; b, \#]$. Para que esta gramática fosse $\mathcal{LR}(1)$ neste estado

1. havendo outro item completo os conjuntos dos símbolos da avanço (dos itens comopletos) teriam de ser disjuntos;
2. havendo um item $[B \to u \cdot av\,;\ K]$ (com a um terminal) então $a \notin \{b, \#\}$

Mas a segunda condição é violada pelo item $[S \to \cdot bAB; \#]$. Portanto esta gramática não é $\mathcal{LR}(1)$ porque produz um conflito transferência/redução.

Exercício 8 [1 valor] Suponha que programa MRI `m` resolve o problema de decisão M : "*O número x é múltiplo do número y*". Reduza o problema de decisão P : "*O número x é par*" a M.

A redução pedida é feita pelo programa MRI

$$\left\{\begin{array}{ll} 1. & \mathtt{Z}_2 \\ 2. & \mathtt{S}_2 \\ 3. & \mathtt{S}_2 \\ 4. & \mathtt{m}\,[1, 2 \to 1] \end{array}\right.$$