Linguagens Formais Autómatos
Resolução de Alguns Exercícios

Vasco Pedro
Departamento de Informática
Universidade de Évora

2008/2009

Aqui apresentam-se resoluções possíveis para alguns dos exercícios propostos em LFA.

Exercício 1.

(d) Definição recursiva de $C_4 = \{a^n b^n \mid n > 0\}$:

- $ab \in C_4$;
- se $w \in C_4$, então $awb \in C_4$;
- $w \in C_4$ somente se pode ser gerada através de um número finito de aplicações do passo recursivo a partir do elemento da base.

(f) Definição recursiva de $C_6 = \{w \mid w \in \{a, b\}^*$ e o número de a's em w é igual ao de b's$\}$:

- $\lambda \in C_6$;
- se $v, w \in C_6$, então awb, bwa e $vw \in C_6$;
- $w \in C_6$ somente se pode ser gerada através de um número finito de aplicações do passo recursivo a partir do elemento da base.

Exercício 6.

(a) $a^*b^*c^*$

(b) $aa^*b^*c^* \cup a^*bb^*c^* \cup a^*b^*cc^*$

(g) $(\lambda \cup a \cup aa)(\lambda \cup b(a \cup b)^*)$

(h) $((\lambda \cup a \cup aa)b)^*(\lambda \cup a \cup aa)$

Exercício 8.

(a) $\emptyset^* \cup a^* \cup b^*(a \cup b)^*$

$$\begin{aligned}
\emptyset^* \cup a^* \cup b^*(a \cup b)^* &= \lambda \cup a^* \cup b^*(a \cup b)^* && (\emptyset^* = \lambda)\\
&= a^* \cup b^*(a \cup b)^* && (\lambda \in L(a^*))\\
&= a^* \cup b^*(b \cup a)^* && (u \cup v = v \cup u)\\
&= a^* \cup (b \cup a)^* && (u^*(u \cup v) = (u \cup v)^*)\\
&= (b \cup a)^* && (L(a^*) \subseteq L((b \cup a)^*))
\end{aligned}$$

(c) $b^*(a \cup (b^*a^*)^*)ab^*(ab^*)^*b$

$$\begin{aligned}
b^*(a \cup (b^*a^*)^*)ab^*(ab^*)^*b &= b^*(a \cup (b \cup a)^*)ab^*(ab^*)^*b && ((u^*v^*)^* = (u \cup v)^*)\\
&= b^*(b \cup a)^*ab^*(ab^*)^*b && (L(a) \subseteq L((b \cup a)^*))\\
&= (b \cup a)^*ab^*(ab^*)^*b && (u^*(u \cup v)^* = (u \cup v)^*)\\
&= (b \cup a)^*a(b \cup a)^*b && (u^*(vu^*)^* = (u \cup v)^*)\\
&\textit{Poder-se-ia parar aqui ou continuar com:}\\
&= b^*(ab^*)^*a(b \cup a)^*b && ((u \cup v)^* = u^*(vu^*)^*)\\
&= b^*a(b^*a)^*(b \cup a)^*b && ((uv)^*u = u(vu)^*)\\
&= b^*a(b^*a)^*b^*(b \cup a)^*b && ((u \cup v)^* = u^*(u \cup v)^*)\\
&= b^*a(b \cup a)^*(b \cup a)^*b && ((u^*v)^*u^* = (u \cup v)^*)\\
&= b^*a(b \cup a)^*b && (\textit{mostrar que } u^*u^* = u^*)
\end{aligned}$$

Exercício 10.

(b) $(y \cup x(x \cup yx)^*yy)^*$ ou $(y^*x(x^*(yx)^*)^*yy)^*y^*$

(c) $(y \cup x(x \cup yx)^*yy)^*(\lambda \cup x(x \cup yx)^*)$

Exercício 19.

	λ-fecho
0	$\{0,1\}$
1	$\{1\}$
2	$\{2,3\}$
3	$\{3\}$

t	m	n
0	$\{1\}$	$\{1,2,3\}$
1	$\emptyset$	$\{1,3\}$
2	$\{1,3\}$	$\{2,3\}$
3	$\{1,3\}$	$\{2,3\}$

	δ_D	m	n
λ-fecho(0) =	$\{0,1\}$	$\{1\}$	$\{1,2,3\}$
	$\{1\}$	$\emptyset$	$\{1,3\}$
	$\{1,2,3\}$	$\{1,3\}$	$\{1,2,3\}$
	$\emptyset$	$\emptyset$	$\emptyset$
	$\{1,3\}$	$\{1,3\}$	$\{1,2,3\}$

O autómato finito determinista obtido é

$$N_D = (\{\{1\},\{1,2,3\},\emptyset,\{1,3\}\},\{m,n\},\delta_D,\{0,1\},\{\{1\},\{1,2,3\},\{1,3\}\})$$

Uma expressão regular que representa $L(N)$ é $(\lambda \cup m)(\lambda \cup n(m \cup n)^*)$.

Exercício 20.

(a) Um autómato finito não determinista que reconhece $(a \cup b)^* b(a \cup b)(a \cup b)$ é $M = (\{A, B, C, D\}, \{a, b\}, \delta, A, \{D\})$ com a função de transição

δ	a	b
A	$\{A\}$	$\{A, B\}$
B	$\{C\}$	$\{C\}$
C	$\{D\}$	$\{D\}$
D		

(b)

	λ-fecho
A	$\{A\}$
B	$\{B\}$
C	$\{C\}$
D	$\{D\}$

t	a	b
A	$\{A\}$	$\{A, B\}$
B	$\{C\}$	$\{C\}$
C	$\{D\}$	$\{D\}$
D		

	δ_D	a	b
λ-fecho$(A) =$	$\{A\}$	$\{A\}$	$\{A, B\}$
	$\{A, B\}$	$\{A, C\}$	$\{A, B, C\}$
	$\{A, C\}$	$\{A, D\}$	$\{A, B, D\}$
	$\{A, B, C\}$	$\{A, C, D\}$	$\{A, B, C, D\}$
	$\{A, D\}$	$\{A\}$	$\{A, B\}$
	$\{A, B, D\}$	$\{A, C\}$	$\{A, B, C\}$
	$\{A, C, D\}$	$\{A, D\}$	$\{A, B, D\}$
	$\{A, B, C, D\}$	$\{A, C, D\}$	$\{A, B, C, D\}$

O autómato finito determinista obtido é

$$M_D = (\{\{A\}, \{A, B\}, \{A, C\}, \{A, B, C\}, \{A, D\}, \{A, B, D\}, \{A, C, D\}, \{A, B, C, D\}\},$$
$$\{a, b\}, \delta_D, \{A\}, \{\{A, D\}, \{A, B, D\}, \{A, C, D\}, \{A, B, C, D\}\})$$

Exercício 23.

		a	b
I	B	II	II
	D	II	II
II	A	I	II
	C	I	II
	E	II	II
	F	II	II

		a	b
I	B	III	II
	D	III	II
II	A	I	III
	C	I	III
III	E	III	III
	F	III	III

δ_M	a	b
I	III	II
II	I	III
III	III	III

O autómato finito determinista mínimo equivalente a M é

$$M_M = (\{\mathrm{I}, \mathrm{II}, \mathrm{III}\}, \{a, b\}, \delta_M, \mathrm{II}, \{\mathrm{I}\})$$

A expressão regular $a(ba)^*$ representa a linguagem reconhecida por M.

Exercício 25.

(a) Um autómato finito não determinista que reconhece $(aa)^* \cup (aaa)^*$ é

$$M = (\{1,2,3,4,5,6\}, \{a\}, \delta, 1, \{2,4\})$$

com a função de transição

δ	a	λ
1		$\{2,4\}$
2	$\{3\}$	
3	$\{2\}$	
4	$\{5\}$	
5	$\{6\}$	
6	$\{4\}$	

(b)

	λ-fecho
1	$\{1,2,4\}$
2	$\{2\}$
3	$\{3\}$
4	$\{4\}$
5	$\{5\}$
6	$\{6\}$

t	a
1	$\{3,5\}$
2	$\{3\}$
3	$\{2\}$
4	$\{5\}$
5	$\{6\}$
6	$\{4\}$

	δ_D	a
λ-fecho(1) =	$\{1,2,4\}$	$\{3,5\}$
	$\{3,5\}$	$\{2,6\}$
	$\{2,6\}$	$\{3,4\}$
	$\{3,4\}$	$\{2,5\}$
	$\{2,5\}$	$\{3,6\}$
	$\{3,6\}$	$\{2,4\}$
	$\{2,4\}$	$\{3,5\}$

Renomeando os estados de acordo com as equivalências seguintes

$$124 \equiv \{1,2,4\} \quad 26 \equiv \{2,6\} \quad 25 \equiv \{2,5\} \quad 24 \equiv \{2,4\}$$
$$35 \equiv \{3,5\} \quad 34 \equiv \{3,4\} \quad 36 \equiv \{3,6\}$$

obtemos o autómato finito determinista

$$M_D = (\{124, 35, 26, 34, 25, 36, 24\}, \{a\}, \delta'_D, 124, \{124, 26, 34, 25, 24\})$$

com a função de transição

δ'_D	a
124	35
35	26
26	34
34	25
25	36
36	24
24	35

(c)

		a
I	124	II
	26	I
	34	I
	25	II
	24	II
II	35	I
	36	I

		a
I	124	III
	25	III
	24	III
II	26	II
	34	I
III	35	II
	36	I

		a
I	124	IV
	25	V
	24	IV
II	26	III
III	34	I
IV	35	II
V	36	I

		a
I	124	V
	24	V
II	25	VI
III	26	IV
IV	34	II
V	35	III
VI	36	I

O autómato finito determinista mínimo equivalente a M é

$$M_M = (\{\mathrm{I}, \mathrm{II}, \mathrm{III}, \mathrm{IV}, \mathrm{V}, \mathrm{VI}\}, \{a\}, \delta_M, \mathrm{I}, \{\mathrm{I}, \mathrm{II}, \mathrm{III}, \mathrm{IV}\})$$

com a função de transição δ_M abaixo

δ_M	a
I	V
II	VI
III	IV
IV	II
V	III
VI	I

Exercício 32.

Uma gramática que gera a linguagem pretendida é $G = (\{S, A, B\}, \{a, b\}, P, S)$, com P o conjunto com as produções:

$$\begin{aligned} &S \to aSb \mid A \mid B \\ &A \to aA \mid a \\ &B \to bB \mid b \end{aligned}$$

Exercício 36.

A gramática $G_{PL} = (\{T, A\}, \{v, f, (,), ,\}, P_{PL}, T)$, com P_{PL} o conjunto com as produções

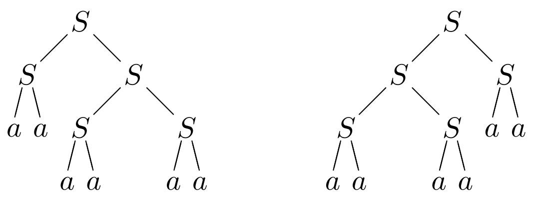
$$\begin{aligned} &T \to v \mid f \mid f(A) \\ &A \to T\mathbf{,} A \mid T \end{aligned}$$

gera a linguagem dos termos Prolog na forma pedida.

Exercício 38.

(a) Consideremos a palavra $aaaaaa \in L(G)$.

Esta palavra tem as seguintes árvores de derivação:



Como existe uma palavra da linguagem gerada por G com duas árvores de derivação distintas, G é ambígua.

(b) A gramática $G' = (\{S\}, \{a\}, \{S \to aa \mid aaS\}, S)$ é uma gramática independente do contexto não ambígua equivalente a G.

(c) A gramática $G'' = (\{S, X\}, \{a\}, \{S \to aX, X \to a \mid aS\}, S)$ é uma gramática regular equivalente a G.

(d) A expressão regular $aa(aa)^*$ representa $L(G)$.

Exercício 41.

O autómato de pilha seguinte reconhece $\{1^n + 1^m = 1^{m+n} \mid n, m \geq 0\}$:

$$M = (\{q_0, q_1, q_2\}, \{1, +, =\}, \{A\}, \delta, q_0, \{q_2\})$$

com a função de transição δ:

δ	1	$+$	$=$	λ
q_0, λ	q_0, A	q_1, λ		
q_1, λ	q_1, A		q_2, λ	
q_2, A	q_2, λ			

A tabela anterior tem a forma

δ	$\dots$	$a \in \Sigma$ *ou* λ	$\dots$
$\vdots$		$\vdots$	
q_i, α	$\dots$	q_j, β	$\dots$
$\vdots$		$\vdots$	

onde $\alpha, \beta \in \Gamma \cup \{\lambda\}$ e lê-se: a partir do estado q_i, quando no topo da pilha está α, existe uma transição com o símbolo a (ou sem símbolo, no caso da coluna λ) para o estado q_j, que substitui α por β no topo da pilha.

M é um autómato de pilha determinista porque todas as transições são com símbolo e nenhum estado tem mais que uma transição com algum dos símbolos. Em consequência, em qualquer configuração do autómato, a transição a efectuar é determinada univocamente pelo par (estado corrente, próximo símbolo da palavra a processar), nunca havendo lugar a escolha.

Exercício 50.

$G = (\{A, B\}, \{a, b\}, \{A \to aAb \mid B, B \to Bb \mid \lambda\}, A)$

1. Tornar o símbolo inicial não recursivo.

A' é o novo símbolo inicial e as novas produções são:

$$\begin{aligned}&A' \to A\\&A \to aAb \mid B\\&B \to Bb \mid \lambda\end{aligned}$$

2. Eliminar as produções-λ.

$\Lambda = \{A', A, B\}$ e as novas produções são:

$$\begin{aligned}&A' \to A \mid \lambda\\&A \to aAb \mid B \mid ab\\&B \to Bb \mid b\end{aligned}$$

3. Eliminar as produções unitárias.

As novas produções são:

	$CHAIN$
A'	$\{A', A, B\}$
A	$\{A, B\}$
B	$\{B\}$

$$\begin{aligned}&A' \to \lambda \mid aAb \mid ab \mid Bb \mid b\\&A \to aAb \mid ab \mid Bb \mid b\\&B \to Bb \mid b\end{aligned}$$

4. Eliminar símbolos inúteis.

a) PRODUTIVOS $= \{A', A, B\}$ Não há símbolos não terminais improdutivos a eliminar.

b) ACESSÍVEIS $= \{A', A, B\}$ Não há símbolos não terminais inacessíveis a eliminar.

5. Construir a forma normal de Chomsky.

a) Transformar as produções de modo a que os corpos que têm um símbolo do alfabeto não têm mais nenhum símbolo:

$$\begin{aligned}&A' \to \lambda \mid XAY \mid XY \mid BY \mid b\\&X \to a\\&Y \to b\\&A \to XAY \mid XY \mid BY \mid b\\&B \to BY \mid b\end{aligned}$$

b) As produções da gramática na forma normal de Chomsky são:

$$
\begin{aligned}
&A' \rightarrow \lambda \mid XZ \mid XY \mid BY \mid b \\
&Z \rightarrow AY \\
&X \rightarrow a \\
&Y \rightarrow b \\
&A \rightarrow XZ \mid XY \mid BY \mid b \\
&B \rightarrow BY \mid b
\end{aligned}
$$

5. Construir a forma normal de Greibach.

a) Ordenar os não terminais: $A'\ Z\ X\ Y\ A\ B$

b) Transformar as produções de modo a nenhum corpo começar por um não terminal menor do que ou igual ao da cabeça (começando no primeiro não terminal):

$$
\begin{aligned}
&A' \rightarrow \lambda \mid XZ \mid XY \mid BY \mid b \\
&Z \rightarrow AY \\
&X \rightarrow a \\
&Y \rightarrow b \\
&A \rightarrow aZ \mid aY \mid BY \mid b \\
&B \rightarrow bU \mid b \\
&U \rightarrow YU \mid Y
\end{aligned}
$$

c) Substituir os não terminais que são o primeiro símbolo do corpo de uma produção pelos corpos das suas produções (começando no último não terminal):

$$
\begin{aligned}
&B \rightarrow bU \mid b \\
&A \rightarrow aZ \mid aY \mid bUY \mid bY \mid b \\
&Y \rightarrow b \\
&X \rightarrow a \\
&Z \rightarrow aZY \mid aYY \mid bUYY \mid bYY \mid bY \\
&A' \rightarrow \lambda \mid aZ \mid aY \mid bUY \mid bY \mid b \\
&U \rightarrow bU \mid b
\end{aligned}
$$

A gramática na forma normal de Greibach é $G' = (\{A', A, B, X, Y, Z, U\}, \{a, b\}, P', A')$ com P' as produções:

$$
\begin{aligned}
&A' \rightarrow \lambda \mid aZ \mid aY \mid bUY \mid bY \mid b \\
&Z \rightarrow aZY \mid aYY \mid bUYY \mid bYY \mid bY \\
&X \rightarrow a \\
&Y \rightarrow b \\
&A \rightarrow aZ \mid aY \mid bUY \mid bY \mid b \\
&B \rightarrow bU \mid b \\
&U \rightarrow bU \mid b
\end{aligned}
$$

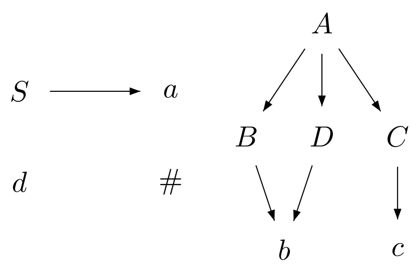
Os não terminais A, B e X são inacessíveis e poderiam ser eliminados.

Exercício 52. (c)

Símbolos que geram λ:

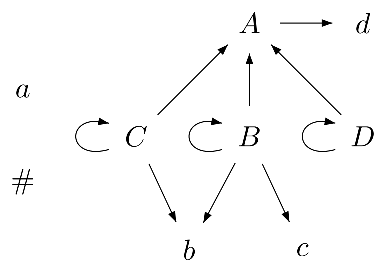
$$\begin{array}{l} S \to a\underline{A}d\# \\ \underline{\underline{A}} \to \underline{B}\,\underline{C}\,\underline{D} \\ \underline{\underline{B}} \to b\underline{B} \mid \underline{\lambda} \\ \underline{\underline{C}} \to c\underline{C} \mid \underline{\lambda} \\ \underline{\underline{D}} \to b\underline{D} \mid \underline{\lambda} \end{array} \qquad\qquad \Lambda = \{A, B, C, D\}$$

Grafo dos primeiros:



	S	A	B	C	D
PRIMEIROS	a	b, c	b	c	b

Grafo dos seguintes:



	A	B	C	D
SEGUINTES	d	b, c, d	b, d	d

Símbolos directores:

$$\begin{array}{ll} \text{DIR}(S \to aAd\#) & = \text{PRIMEIROS}(aAd\#) = \text{PRIMEIROS}(a) = \{a\} \\ \text{DIR}(A \to BCD) & = \text{PRIMEIROS}(BCD) \cup \text{SEGUINTES}(A) = \\ & = \text{PRIMEIROS}(B) \cup \text{PRIMEIROS}(C) \cup \text{PRIMEIROS}(D) \cup \\ & \quad \cup \text{SEGUINTES}(A) = \{b\} \cup \{c\} \cup \{b\} \cup \{d\} = \{b, c, d\} \\ \text{DIR}(B \to bB) & = \text{PRIMEIROS}(bB) = \{b\} \\ \text{DIR}(B \to \lambda) & = \text{PRIMEIROS}(\lambda) \cup \text{SEGUINTES}(B) = \emptyset \cup \{b, c, d\} = \{b, c, d\} \\ \text{DIR}(C \to cC) & = \{c\} \\ \text{DIR}(C \to \lambda) & = \{b, d\} \\ \text{DIR}(D \to bD) & = \{b\} \\ \text{DIR}(D \to \lambda) & = \{d\} \end{array}$$

A gramática não é LL(1) porque o símbolo b pertence aos símbolos directores das duas produções de B.

Exercício 58.

Seja $G = (\{S, A, B\}, \{a, b\}, P, S)$, em que P é o conjunto com as produções:

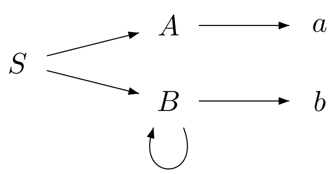
$$\begin{aligned} &S \to AB \\ &A \to aA \mid \lambda \\ &B \to Bb \mid \lambda \end{aligned}$$

(a) Construção do AFD dos itens LR(1) válidos

Símbolos que geram λ:

$$\begin{aligned} &\underline{S} \to \underline{AB} \\ &\underline{\underline{A}} \to a\underline{A} \mid \underline{\lambda} \\ &\underline{\underline{B}} \to \underline{B}b \mid \underline{\lambda} \end{aligned} \qquad\qquad \Lambda = \{S, A, B\}$$

Grafo dos primeiros:

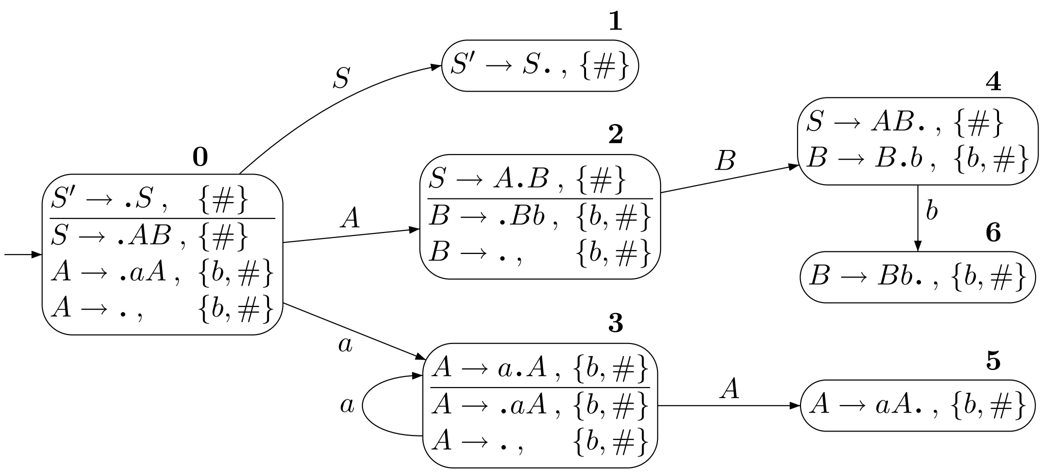


	S	A	B
PRIMEIROS	a, b	a	b

Como curiosidade, o tuplo correspondente ao AFD dos itens LR(1) válidos desta gramática seria:

$$(\{\mathbf{0}, \mathbf{1}, \mathbf{2}, \mathbf{3}, \mathbf{4}, \mathbf{5}, \mathbf{6}, \emptyset\}, \{S, A, B, a, b\}, \delta, \mathbf{0}, \{\mathbf{0}, \mathbf{1}, \mathbf{2}, \mathbf{3}, \mathbf{4}, \mathbf{5}, \mathbf{6}\})$$

com a função de transição δ representada no diagrama de estados seguinte, onde se omitiram o estado de erro $\emptyset$ e todas as transições de e para esse estado.



(b) Verificação das condições LR(1)

A gramática G é LR(1) porque o seu autómato dos itens válidos satisfaz as condições LR(1), nomeadamente:

- nenhum estado contém dois itens completos (*pelo que não pode haver conflitos redução/ redução*);
- os estados **0**, **2** e **3** contêm itens completos com conjuntos de símbolos de avanço $\{b, \#\}$ enquanto que nos outros itens, imediatamente a seguir ao ponto aparecem os símbolos S, A, a e B; e o estado **4** tem um item completo com conjunto de símbolos de avanço $\{\#\}$ enquanto que no outro item do mesmo estado, o ponto é seguido de b (*não há conflitos transferência/redução*).

(c) Verificação das condições LALR(1)

Como todos os estados do autómato dos itens válidos de G têm núcleos LR(0) distintos (o autómato amalgamado é igual a esse autómato) e a gramática é LR(1), a gramática também é LALR(1). (*Porquê?*)

(d) Tabela de análise sintáctica LR(1)

	S	A	B	a	b	a	b	$\#$
0	1	2		3		TRANSF	$A \to \lambda$	$A \to \lambda$
1								ACEITA
2			4				$B \to \lambda$	$B \to \lambda$
3		5		3		TRANSF	$A \to \lambda$	$A \to \lambda$
4					6		TRANSF	$S \to AB$
5							$A \to aA$	$A \to aA$
6							$B \to Bb$	$B \to Bb$

(e) Autómato de pilha LR(1)

O autómato de pilha LR(1) que reconhece $L(G)$ é

$$R = (\{q_I, q, q_a, q_b, q_\#\}, \{S, A, B, a, b, \#\}, \{S, A, B, a, b, \mathbf{0}, \mathbf{1}, \mathbf{2}, \mathbf{3}, \mathbf{4}, \mathbf{5}, \mathbf{6}\}, \delta', q_I, \{q_\#\})$$

com a função de transição δ' com as transições seguintes (onde $(q', \alpha) \xrightarrow{u} (q'', \beta)$ representa

$[q'', \beta] \in \delta'(q', u, \alpha))$:

inicialização

$(q_I, \lambda) \xrightarrow{\lambda} (q, \mathbf{0})$

leitura

$(q, \lambda) \xrightarrow{a} (q_a, \lambda)$

$(q, \lambda) \xrightarrow{b} (q_b, \lambda)$

$(q, \lambda) \xrightarrow{\#} (q_\#, \lambda)$

aceitação

$(q_\#, \mathbf{1}S\mathbf{0}) \xrightarrow{\lambda} (q_\#, \lambda)$

redução

$(q_b, \mathbf{0}) \xrightarrow{\lambda} (q_b, \mathbf{2}A\mathbf{0})$

$(q_\#, \mathbf{0}) \xrightarrow{\lambda} (q_\#, \mathbf{2}A\mathbf{0})$

$(q_b, \mathbf{2}) \xrightarrow{\lambda} (q_b, \mathbf{4}B\mathbf{2})$

$(q_\#, \mathbf{2}) \xrightarrow{\lambda} (q_\#, \mathbf{4}B\mathbf{2})$

$(q_b, \mathbf{3}) \xrightarrow{\lambda} (q_b, \mathbf{5}A\mathbf{3})$

$(q_\#, \mathbf{3}) \xrightarrow{\lambda} (q_\#, \mathbf{5}A\mathbf{3})$

$(q_\#, \mathbf{4}B\mathbf{2}A\mathbf{0}) \xrightarrow{\lambda} (q_\#, \mathbf{1}S\mathbf{0})$

$(q_b, \mathbf{5}A\mathbf{3}a\mathbf{0}) \xrightarrow{\lambda} (q_b, \mathbf{2}A\mathbf{0})$

$(q_b, \mathbf{5}A\mathbf{3}a\mathbf{3}) \xrightarrow{\lambda} (q_b, \mathbf{5}A\mathbf{3})$

$(q_\#, \mathbf{5}A\mathbf{3}a\mathbf{0}) \xrightarrow{\lambda} (q_\#, \mathbf{2}A\mathbf{0})$

$(q_\#, \mathbf{5}A\mathbf{3}a\mathbf{3}) \xrightarrow{\lambda} (q_\#, \mathbf{5}A\mathbf{3})$

$(q_b, \mathbf{6}b\mathbf{4}B\mathbf{2}) \xrightarrow{\lambda} (q_b, \mathbf{4}B\mathbf{2})$

$(q_\#, \mathbf{6}b\mathbf{4}B\mathbf{2}) \xrightarrow{\lambda} (q_\#, \mathbf{4}B\mathbf{2})$

transferência

$(q_a, \mathbf{0}) \xrightarrow{\lambda} (q, \mathbf{3}a\mathbf{0})$

$(q_a, \mathbf{3}) \xrightarrow{\lambda} (q, \mathbf{3}a\mathbf{3})$

$(q_b, \mathbf{4}) \xrightarrow{\lambda} (q, \mathbf{6}b\mathbf{4})$

(f) Computação para $aabb$

$$
\begin{array}{rrr}
[\,q_I, & aabb\#, & \lambda] \vdash \\
[\,q, & aabb\#, & \mathbf{0}] \vdash \\
[\,q_a, & abb\#, & \mathbf{0}] \vdash \\
[\,q, & abb\#, & \mathbf{3}a\mathbf{0}] \vdash \\
[\,q_a, & bb\#, & \mathbf{3}a\mathbf{0}] \vdash \\
[\,q, & bb\#, & \mathbf{3}a\mathbf{3}a\mathbf{0}] \vdash \\
[\,q_b, & b\#, & \mathbf{3}a\mathbf{3}a\mathbf{0}] \vdash \\
[\,q_b, & b\#, & \mathbf{5}A\mathbf{3}a\mathbf{3}a\mathbf{0}] \vdash \\
[\,q_b, & b\#, & \mathbf{5}A\mathbf{3}a\mathbf{0}] \vdash \\
[\,q_b, & b\#, & \mathbf{2}A\mathbf{0}] \vdash \\
[\,q_b, & b\#, & \mathbf{4}B\mathbf{2}A\mathbf{0}] \vdash \\
[\,q, & b\#, & \mathbf{6}b\mathbf{4}B\mathbf{2}A\mathbf{0}] \vdash \\
[\,q_b, & \#, & \mathbf{6}b\mathbf{4}B\mathbf{2}A\mathbf{0}] \vdash \\
[\,q_b, & \#, & \mathbf{4}B\mathbf{2}A\mathbf{0}] \vdash \\
[\,q, & \#, & \mathbf{6}b\mathbf{4}B\mathbf{2}A\mathbf{0}] \vdash \\
[q_\#, & \lambda, & \mathbf{6}b\mathbf{4}B\mathbf{2}A\mathbf{0}] \vdash \\
[q_\#, & \lambda, & \mathbf{4}B\mathbf{2}A\mathbf{0}] \vdash \\
[q_\#, & \lambda, & \mathbf{1}S\mathbf{0}] \vdash \\
[q_\#, & \lambda, & \lambda]
\end{array}
$$

Exercício 59.

Seja $G = (\{S, A, B\}, \{a, b\}, P, S)$, em que P é o conjunto com as produções:

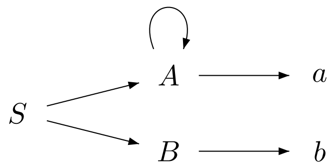
$$\begin{aligned} &S \to ABA \\ &A \to Aa \mid \lambda \\ &B \to bB \mid b \end{aligned}$$

(a) Construção do AFD dos itens LR(1) válidos

Símbolos que geram λ:

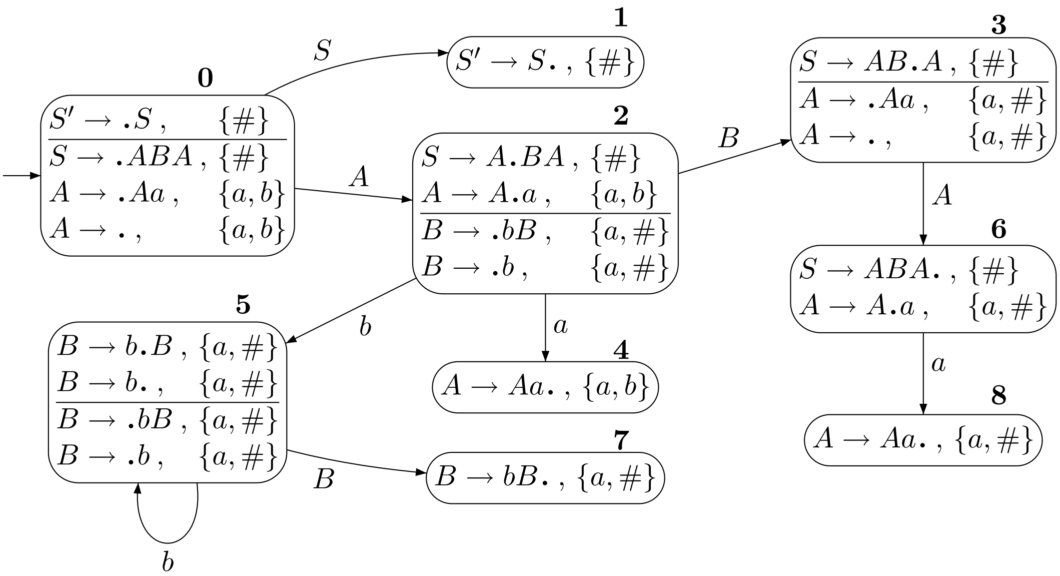
$$\begin{aligned} &S \to \underline{A}B\underline{A} \\ &\underline{\underline{A}} \to \underline{A}a \mid \underline{\lambda} \\ &B \to bB \mid b \end{aligned} \qquad\qquad \Lambda = \{A\}$$

Grafo dos primeiros:



	S	A	B
PRIMEIROS	a, b	a	b

Diagrama de estados do AFD dos itens válidos:



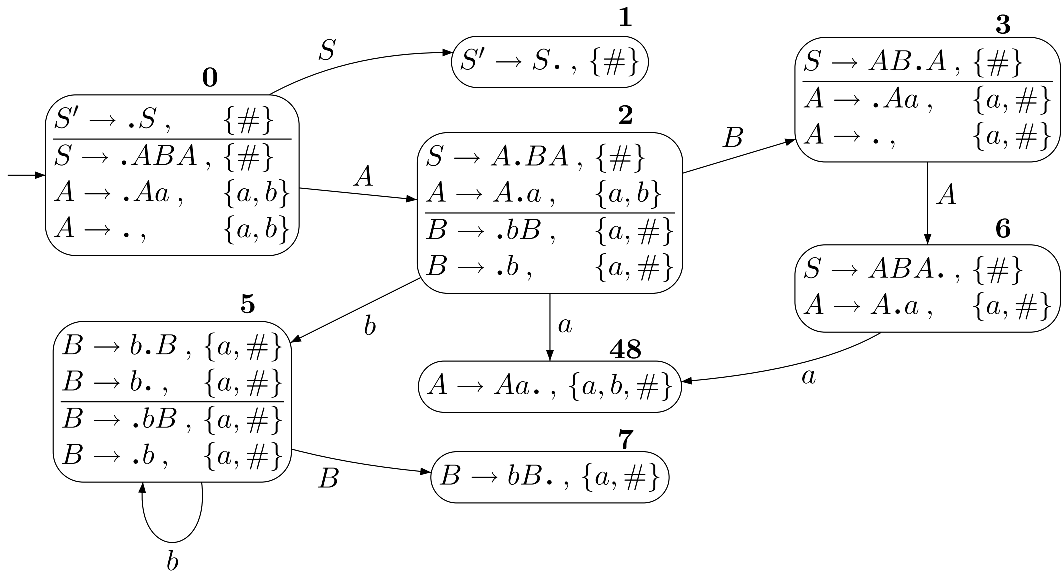
(b) Verificação das condições LR(1)

A gramática G é LR(1) porque o seu autómato dos itens válidos satisfaz as condições LR(1), nomeadamente:

- nenhum estado contém dois itens completos (*pelo que não pode haver conflitos redução/ redução*);
- os estados **0** e **3** contêm um item completo e outros itens em que o ponto está imediatamente à esquerda de um símbolo não terminal de G; o estado **5** contém um item completo com conjunto de símbolos de avanço $\{a, \#\}$ enquanto que nos outros itens, imediatamente a seguir ao ponto aparecem os símbolos b e B; e o estado **6** tem um item completo com conjunto de símbolos de avanço $\{\#\}$ enquanto que no outro item do mesmo estado, o ponto é seguido de a (*não há conflitos transferência/redução*).

(c) Verificação das condições LALR(1)

Os únicos estados do autómato dos itens válidos com o mesmo núcleo LR(0) são os estados **4** e **8**. Fundindo-os, obtém-se o autómato amalgamado:



O único estado novo é o estado **48** que contém um item completo isolado. Logo, a gramática é LALR(1). (*Porquê?*)

(d) Tabela de análise sintáctica LR(1)

	S	A	B	a	b	a	b	$\#$
0	1	2				$A \to \lambda$	$A \to \lambda$	
1								ACEITA
2			3	4	5	TRANSF	TRANSF	
3		6				$A \to \lambda$		$A \to \lambda$
4						$A \to Aa$	$A \to Aa$	
5			7		5	$B \to b$	TRANSF	$B \to b$
6				8		TRANSF		$S \to ABA$
7						$B \to bB$		$B \to bB$
8						$A \to Aa$		$A \to Aa$

(e) Autómato de pilha LR(1)

O autómato de pilha LR(1) que reconhece $L(G)$ tem as seguintes transições:

inicialização

$(q_I, \lambda) \xrightarrow{\lambda} (q, \mathbf{0})$

leitura

$(q, \lambda) \xrightarrow{a} (q_a, \lambda)$

$(q, \lambda) \xrightarrow{b} (q_b, \lambda)$

$(q, \lambda) \xrightarrow{\#} (q_\#, \lambda)$

aceitação

$(q_\#, \mathbf{1}S\mathbf{0}) \xrightarrow{\lambda} (q_\#, \lambda)$

redução

$(q_a, \mathbf{0}) \xrightarrow{\lambda} (q_a, \mathbf{2}A\mathbf{0})$

$(q_b, \mathbf{0}) \xrightarrow{\lambda} (q_b, \mathbf{2}A\mathbf{0})$

$(q_a, \mathbf{3}) \xrightarrow{\lambda} (q_a, \mathbf{6}A\mathbf{3})$

$(q_\#, \mathbf{3}) \xrightarrow{\lambda} (q_\#, \mathbf{6}A\mathbf{3})$

$(q_a, \mathbf{4}a\mathbf{2}A\mathbf{0}) \xrightarrow{\lambda} (q_a, \mathbf{2}A\mathbf{0})$

$(q_b, \mathbf{4}a\mathbf{2}A\mathbf{0}) \xrightarrow{\lambda} (q_b, \mathbf{2}A\mathbf{0})$

$(q_a, \mathbf{5}b\mathbf{2}) \xrightarrow{\lambda} (q_a, \mathbf{3}B\mathbf{2})$

$(q_\#, \mathbf{5}b\mathbf{2}) \xrightarrow{\lambda} (q_\#, \mathbf{3}B\mathbf{2})$

$(q_a, \mathbf{5}b\mathbf{5}) \xrightarrow{\lambda} (q_a, \mathbf{7}B\mathbf{5})$

$(q_\#, \mathbf{5}b\mathbf{5}) \xrightarrow{\lambda} (q_\#, \mathbf{7}B\mathbf{5})$

$(q_\#, \mathbf{6}A\mathbf{3}B\mathbf{2}A\mathbf{0}) \xrightarrow{\lambda} (q_\#, \mathbf{1}S\mathbf{0})$

$(q_a, \mathbf{7}B\mathbf{5}b\mathbf{2}) \xrightarrow{\lambda} (q_a, \mathbf{3}B\mathbf{2})$

$(q_\#, \mathbf{7}B\mathbf{5}b\mathbf{2}) \xrightarrow{\lambda} (q_\#, \mathbf{3}B\mathbf{2})$

$(q_a, \mathbf{7}B\mathbf{5}b\mathbf{5}) \xrightarrow{\lambda} (q_a, \mathbf{7}B\mathbf{5})$

$(q_\#, \mathbf{7}B\mathbf{5}b\mathbf{5}) \xrightarrow{\lambda} (q_\#, \mathbf{7}B\mathbf{5})$

$(q_a, \mathbf{8}a\mathbf{6}A\mathbf{3}) \xrightarrow{\lambda} (q_a, \mathbf{6}A\mathbf{3})$

$(q_\#, \mathbf{8}a\mathbf{6}A\mathbf{3}) \xrightarrow{\lambda} (q_\#, \mathbf{6}A\mathbf{3})$

transferência

$(q_a, \mathbf{2}) \xrightarrow{\lambda} (q, \mathbf{4}a\mathbf{2})$

$(q_b, \mathbf{2}) \xrightarrow{\lambda} (q, \mathbf{5}b\mathbf{2})$

$(q_b, \mathbf{5}) \xrightarrow{\lambda} (q, \mathbf{5}b\mathbf{5})$

$(q_a, \mathbf{6}) \xrightarrow{\lambda} (q, \mathbf{8}a\mathbf{6})$

(f) Computação para $aabb$

$$
\begin{array}{lrr}
[\,q_I, & aabb\#, & \lambda] \vdash \\
[\,q, & aabb\#, & \mathbf{0}] \vdash \\
[\,q_a, & abb\#, & \mathbf{0}] \vdash \\
[\,q_a, & abb\#, & \mathbf{2}A\mathbf{0}] \vdash \\
[\,q, & abb\#, & \mathbf{4}a\mathbf{2}A\mathbf{0}] \vdash \\
[\,q_a, & bb\#, & \mathbf{4}a\mathbf{2}A\mathbf{0}] \vdash \\
[\,q_a, & bb\#, & \mathbf{2}A\mathbf{0}] \vdash \\
[\,q, & bb\#, & \mathbf{4}a\mathbf{2}A\mathbf{0}] \vdash \\
[\,q_b, & b\#, & \mathbf{4}a\mathbf{2}A\mathbf{0}] \vdash \\
[\,q_b, & b\#, & \mathbf{2}A\mathbf{0}] \vdash \\
[\,q, & b\#, & \mathbf{5}b\mathbf{2}A\mathbf{0}] \vdash \\
[\,q_b, & \#, & \mathbf{5}b\mathbf{2}A\mathbf{0}] \vdash \\
[\,q, & \#, & \mathbf{5}b\mathbf{5}b\mathbf{2}A\mathbf{0}] \vdash \\
[q_\#, & \lambda, & \mathbf{5}b\mathbf{5}b\mathbf{2}A\mathbf{0}] \vdash \\
[q_\#, & \lambda, & \mathbf{7}B\mathbf{5}b\mathbf{2}A\mathbf{0}] \vdash \\
[q_\#, & \lambda, & \mathbf{3}B\mathbf{2}A\mathbf{0}] \vdash \\
[q_\#, & \lambda, & \mathbf{6}A\mathbf{3}B\mathbf{2}A\mathbf{0}] \vdash \\
[q_\#, & \lambda, & \mathbf{1}S\mathbf{0}] \vdash \\
[q_\#, & \lambda, & \lambda]
\end{array}
$$

Exercício 60.

Seja $G = (\{S\}, \{a\}, P, S)$, em que P é o conjunto com as produções:

$$S \to aSa \mid \lambda$$

(a) Construção do AFD dos itens LR(1) válidos

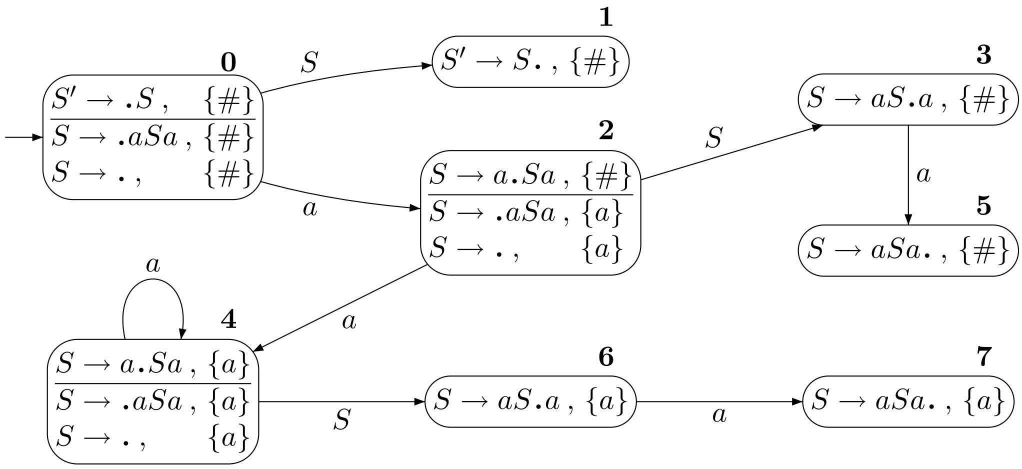
Símbolos que geram λ:

$$\underline{\underline{S}} \to a\underline{S}a \mid \underline{\lambda} \qquad\qquad \Lambda = \{S\}$$

Grafo dos primeiros:

$$S \longrightarrow a \qquad\qquad \mathsf{PRIMEIROS}(S) = \{a\}$$

Diagrama de estados do AFD dos itens válidos:



(b) Verificação das condições LR(1)

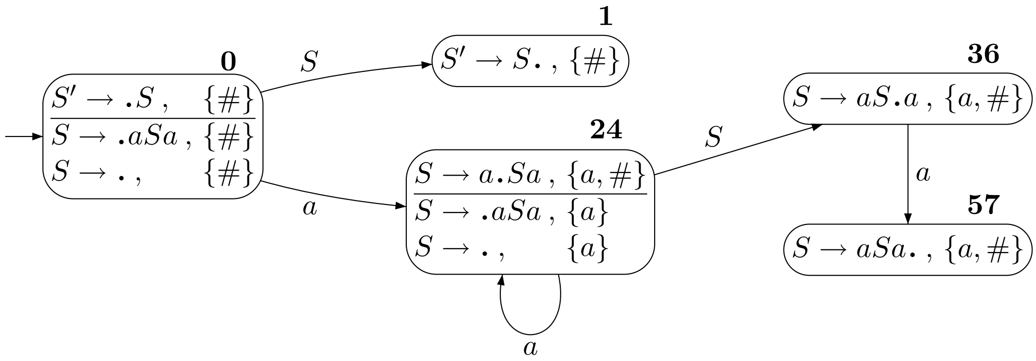
O estado **2** contém um item completo com conjunto de símbolos de avanço $\{a\}$ e um item em que o ponto é seguido de a, logo a gramática não é LR(1) (existe um conflito transferência/redução neste estado).

(O mesmo acontece no estado **4**.)

(c) Verificação das condições LALR(1)

A gramática não é LR(1), logo não é LALR(1). (*Porquê?*)

O autómato amalgamado seria:



(d) Tabela de análise sintáctica LR(1)

Como a gramática não é LR(1), não faz sentido falar na sua tabela de análise sintáctica LR(1). No entanto, pode-se fazer o exercício de a construir, incluindo todas as acções determinadas por cada estado.

	S	a	a	$\#$
0	1	2	TRANSF	$S \to \lambda$
1				ACEITA
2	3	4	TRANSF/$S \to \lambda$	
3		5	TRANSF	
4	6	4	TRANSF/$S \to \lambda$	
5				$S \to aSa$
6		7	TRANSF	
7			$S \to aSa$	

O conteúdo da parte da tabela contendo as acções mostra em que estados e para que símbolos de avanço existem conflitos, e que tipo de conflitos são.

Exercício 65.

Vamos demonstrar que o problema de decisão INSTR "O programa p executa a instrução i quando corre com dados `nil`?" é indecidível reduzindo o problema da terminação a este problema. (*Relembrando, o problema da terminação consiste em determinar se um programa termina quando corre com um* input *dado.*)

Para mostrar que a redução é possível, vamos assumir que existe uma solução para INSTR (um programa) a que recorremos para construir uma solução para o problema da terminação.

A (hipotética) solução para INSTR é o programa $\mathtt{p}_{\mathsf{INSTR}}(p, i)$, que diz se o programa p executa a instrução i, quando corrido com dados (*input*) `nil`. A solução para o problema da terminação será o programa $\mathtt{t}(p, d)$, construído da seguinte forma:

$$\mathtt{t(p,d)} = \mathtt{p}_{\mathsf{INSTR}}(f(\mathtt{p,d}))$$

com f(p,d) = (p',i) a função que, dados um programa WHILE p com a forma:

$$\texttt{p} \equiv \begin{array}{l} \texttt{read X}_p\texttt{;} \\ C_p\texttt{;} \\ \texttt{write Y}_p \end{array}$$

e dados d, constrói o programa:

$$\texttt{p'} \equiv \begin{array}{l} \texttt{read X}_p\texttt{;} \\ \texttt{X}_p\texttt{ := d;} \\ C_p\texttt{;} \\ \texttt{NV := nil;} \\ \texttt{write Y}_p \end{array}$$

onde NV é uma variável que não ocorre em p, e devolve o par (p', NV := nil).

Seja f o programa que implementa f. Dados um programa p e um valor d, f terá de examinar a representação interna de p para descobrir qual a variável que recebe o *input* e quais as variáveis usadas no programa. De seguida, terá de construir a representação interna das instruções "$\texttt{X}_p$:= d" e "NV := nil", e acrescentá-las, respectivamente, no início e no fim da lista com a representação interna do código C_p do programa p. Finalmente, deverá construir o par (p' . NV := nil) e devolvê-lo. (*A representação interna de "$\texttt{X}_p$:= d" será uma lista da forma (:= (var j) d), se j for o número da variável $\texttt{X}_p$, e a de "NV := nil" será a lista (:= (var $n+1$) (quote nil)), onde n é o maior número de uma variável de p.*)

O programa f, que manipula e efectua algumas operações simples sobre listas, existe e f é uma função computável.

O programa p' começa por ignorar o seu *input*, atribuindo à variável $\texttt{X}_p$, que contém o *input* de p, o valor d. De seguida, executa o código C_p de p. Se a execução deste código terminar, tem lugar a execução da instrução "NV := nil" — que consiste na avaliação da expressão nil e na atribuição do seu valor à variável NV, operações realizadas em tempo finito —, e o programa termina. Se a execução de C_p não terminar, então a instrução "NV := nil" nunca é executada. Como o comportamento do código "$\texttt{X}_p$:= d; C_p" é exactamente o comportamento de p quando o seu *input* é d, a instrução "NV := nil" é executada se e só se p(d) termina. Assim, ao correr $\mathsf{p_{INSTR}}$ com dados (p' . NV := nil), ele vai determinar se a instrução "NV := nil" é executada quando o *input* de p' é nil, o que é equivalente a determinar se o programa p termina quando corrido com dados d.

O programa t poderia, então, ser construído, a partir de f e de $\mathsf{p_{INSTR}}$, do modo que se segue:

$$\texttt{t} \equiv \begin{array}{ll} \texttt{read PD;} & \texttt{-- PD contém (p . d)} \\ \texttt{PI := f PD;} & \texttt{-- PI recebe (p' . NV := nil)} \\ \texttt{Y}_t\texttt{ := }\mathsf{p_{INSTR}}\texttt{ PI;} & \\ \texttt{write Y}_t & \end{array}$$

Como o problema da terminação é indecidível e o programa t não existe, mas tanto o programa f como as construções necessárias para obter t a partir de f e de $\mathsf{p_{INSTR}}$ existem, então $\mathsf{p_{INSTR}}$ não existe e INSTR também é um problema indecidível.