PROTÓTIPO PARA TRANSFORMAÇÃO DE UMA EXPRESSÃO REGULAR PARA UMA FUNÇÃO EQUIVALENTE EM PASCAL, UTILIZANDO DOIS ALGORITMOS BASEADOS NO TEOREMA DE KLEENE

RONALD GLATZ

ROTEIRO

ROTEIRO

SAIR

ANTERIOR

ROTEIRO

ROTEIRO

SAIR

ANTERIOR

OBJETIVO

- ✓ APRESENTAR A TRANSFORMAÇÃO DE UMA EXPRESSÃO REGULAR EM UM AFD
- ✓ SERVIR DE FERRAMENTA DE ENSINO-APRENDIZAGEM
- ✓ APRESENTAR UMA COMPARAÇÃO ENTRE AS DIFERENTES FORMAS DE TRANSFORMAÇÃO

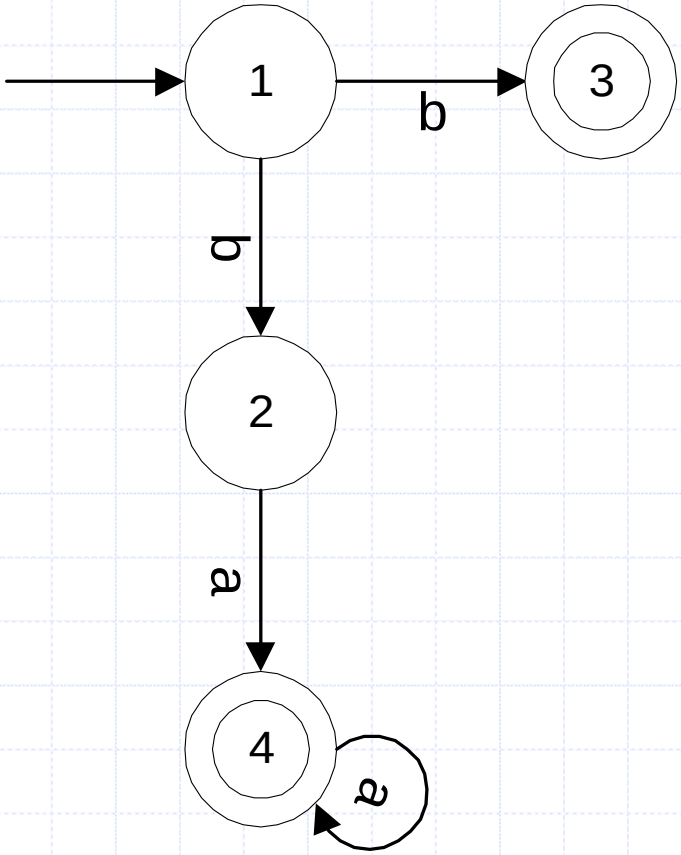
BÁSICO

- Aqui serão apresentados alguns conceitos mais importantes para este trabalho
- ALFABETO (Σ)
  - {A,B,C,D,E,...,X,Z}
  - {0,1}
  - {0,1,2,3,4,5,6,7,8,9}
- PALAVRA SOBRE O NOSSO ALFABETO
  - Abacate
  - Livro

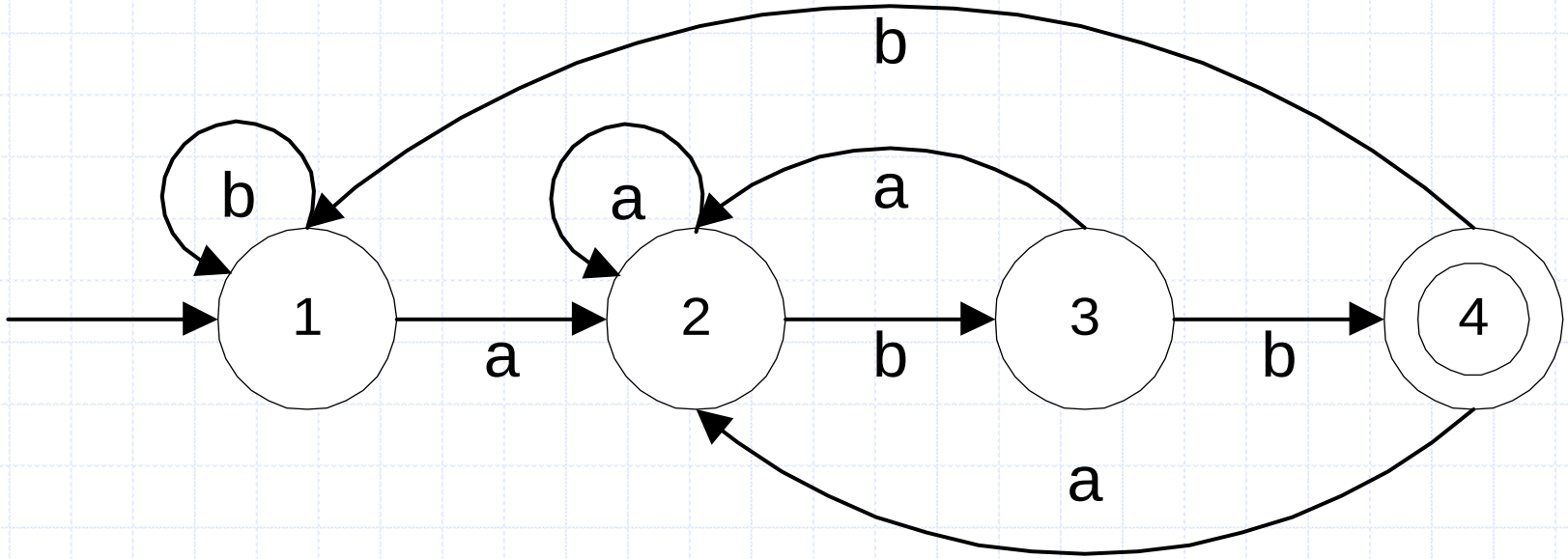
BÁSICO

- ✓ EXPRESSÃO REGULAR (ER)
  - ✓ a(b|c)*ca*|^
  - ✓ a(aa|bb)*
  - ✓ a*
  - ✓ a**
- ✓ CONJUNTO REGULAR
  - ✓ {0,1,01,001,011, ...}: cadeias de Σ={0,1}

BÁSICO-AFND



BÁSICO-AFD

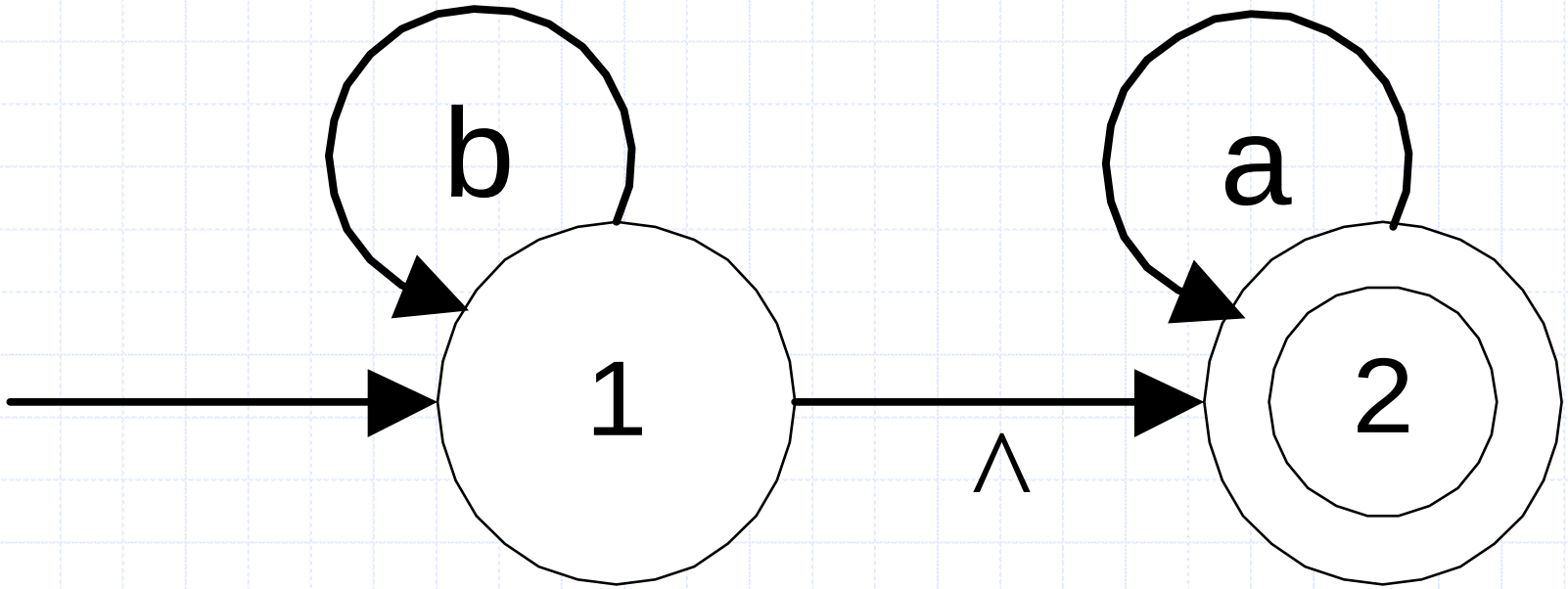


BÁSICO-AFε

ROTEIRO

SAIR

ANTERIOR



BÁSICO-LLC

ROTEIRO

SAIR

ANTERIOR

- ✓ LINGUAGEM PARA DEFINIR OUTRAS LINGUAGENS
- ✓ LINGUAGENS DE PROGRAMAÇÃO
- ✓ ANALISADORES LÉXICOS
- ✓ EDIÇÃO DE TEXTO

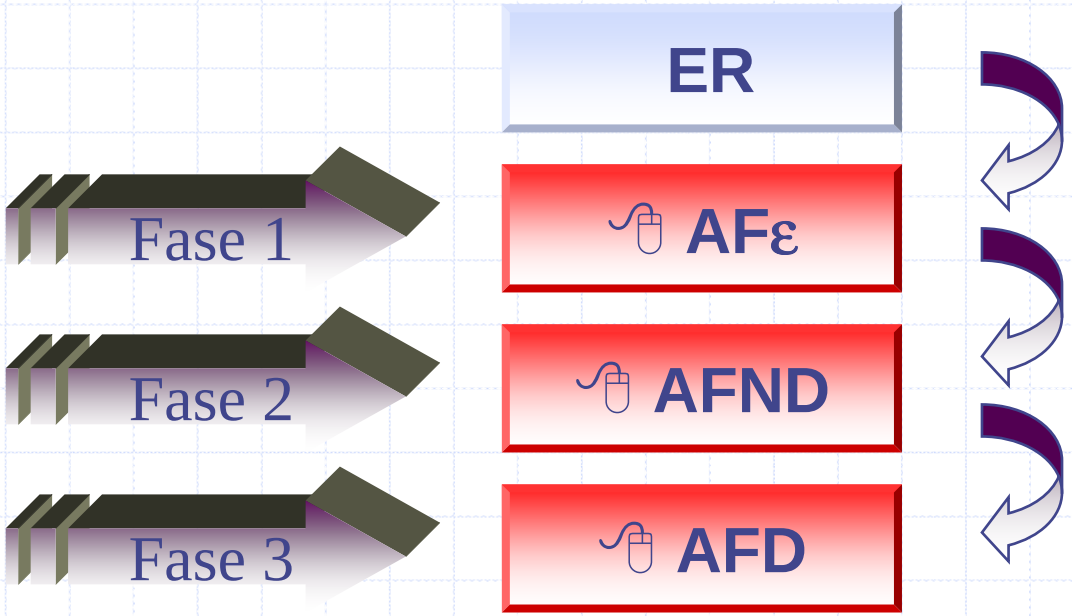
ALGORITMO DE [HOP1979]

FASES

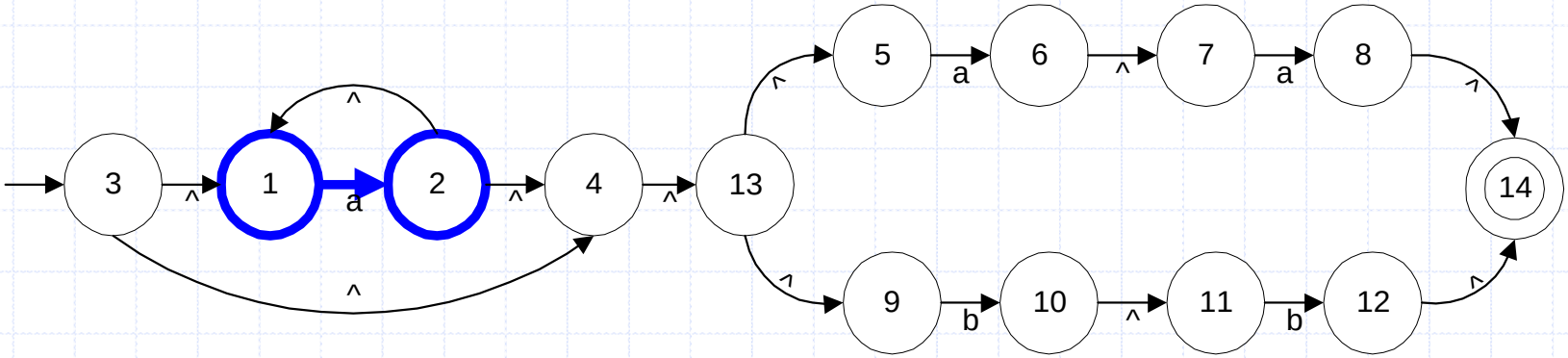
ROTEIRO

SAIR

ANTERIOR

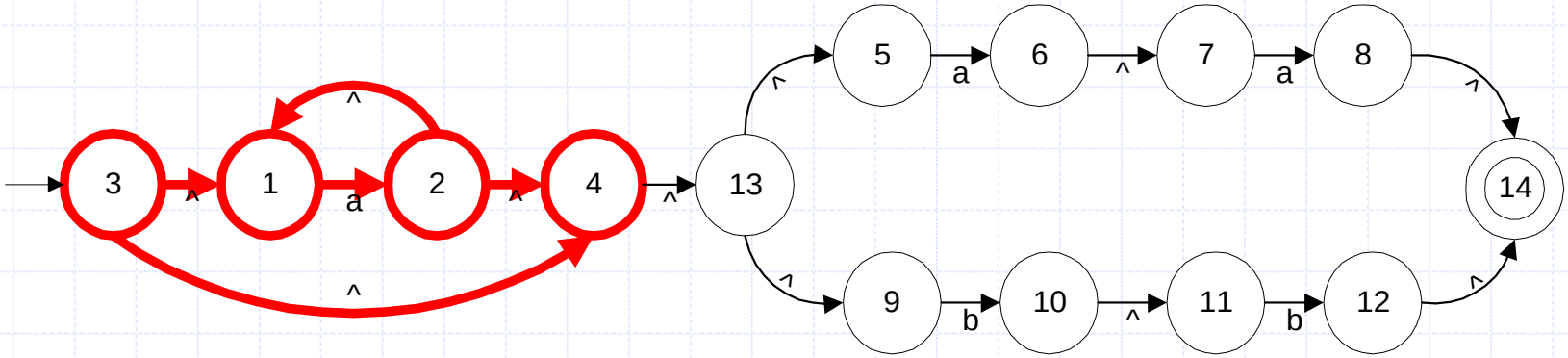


GERAÇÃO DO AFε



⇨ a*(aa|bb)

GERAÇÃO DO AFε



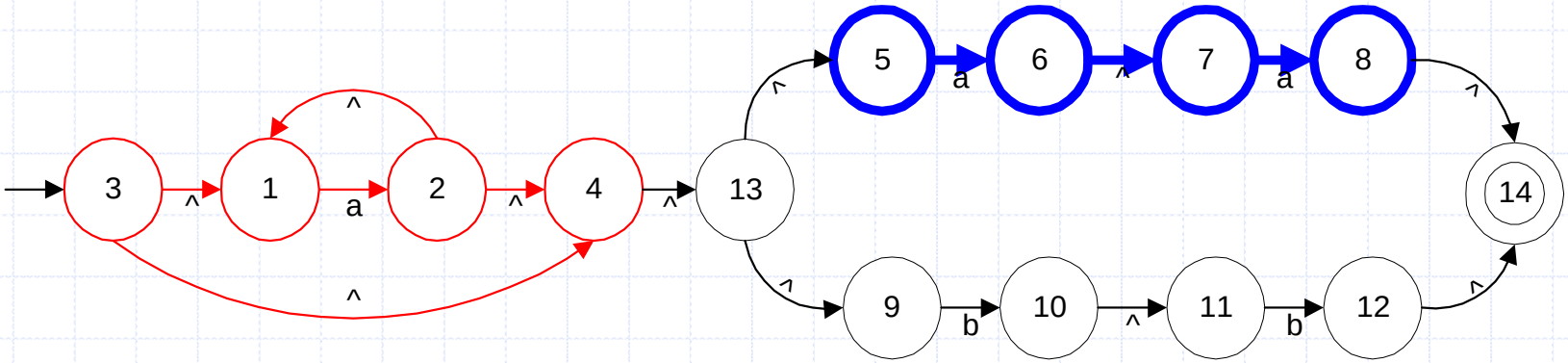
⇨ a*(aa|bb)

ROTEIRO

SAIR

ANTERIOR

GERAÇÃO DO AFε



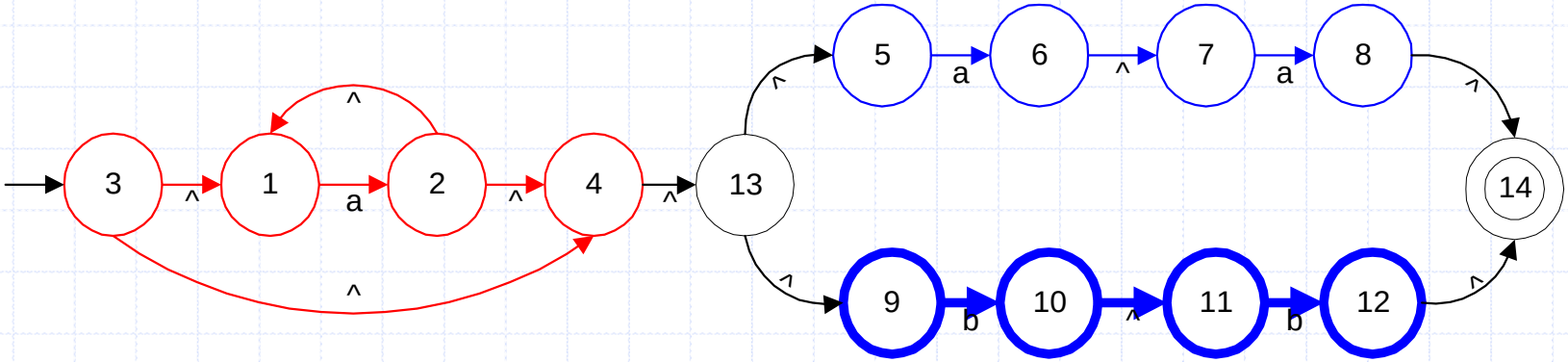
⇨a*(aa|bb)

ROTEIRO

SAIR

ANTERIOR

GERAÇÃO DO AFε



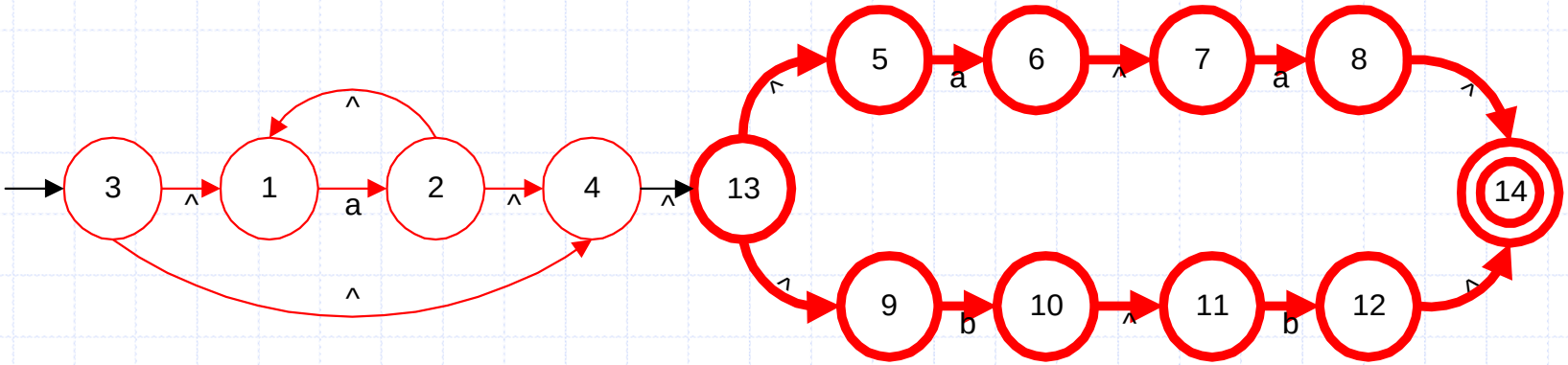
⇨a*(aa|bb)

ROTEIRO

SAIR

ANTERIOR

GERAÇÃO DO AFε



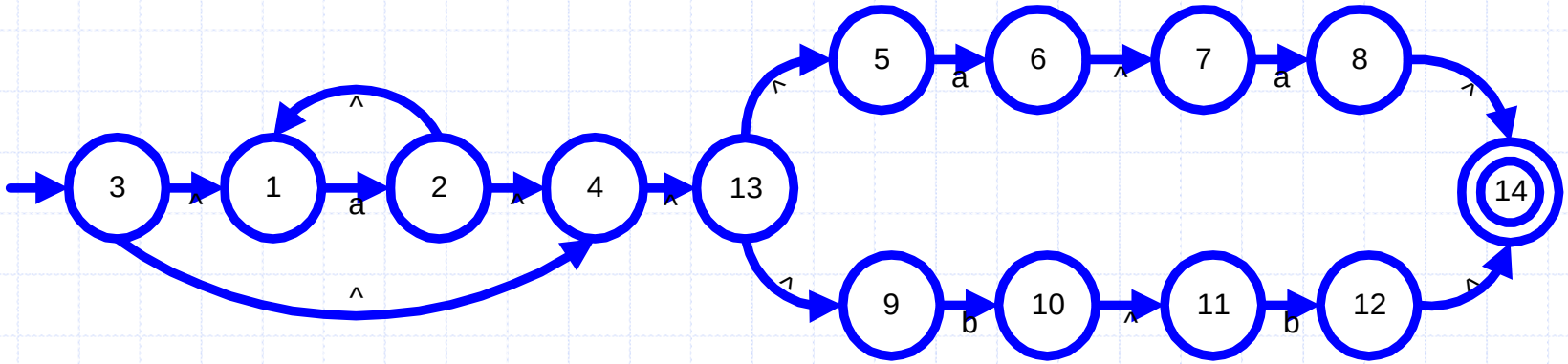
⇨a*(aa|bb)

ROTEIRO

SAIR

ANTERIOR

GERAÇÃO DO AFε



⇨ a*(aa|bb)

GERAÇÃO DO AFND A PARTIR DO AFε

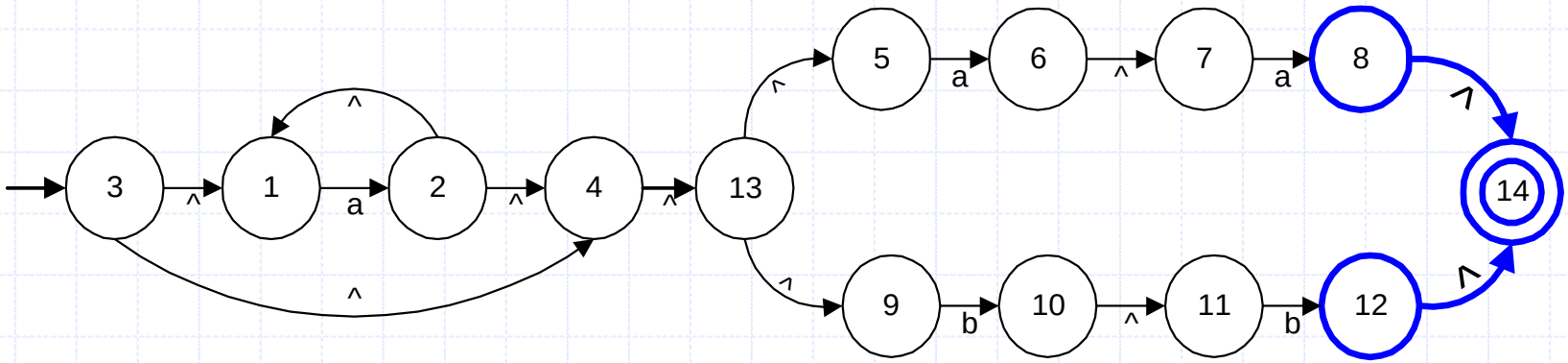
ROTEIRO

SAIR

ANTERIOR

- Consiste em gerar um AFND equivalente a partir do AFε da fase 1
- AFND M=(Σ,Q,δ',q0,F)
  - Σ = {a,b}
  - Q = {1,2,3,4, ..., 13, 14}
  - δ' = Fε(δ''(δ'''(q,^),x))
  - F = Conjunto de Estados Finais

GERAÇÃO DO AFND A PARTIR DO AFε
Conjunto de Estados Finais



- ✓ O conjunto de estados finais do AFND é formado pelo estado final do AFε e todos os estados que atingem o estado final com o símbolo vazio (^) (direta ou indiretamente) no AFε.
- ✓ F={8,12,14}

GERAÇÃO DO AFND A PARTIR DO AFε
Geração das Transições do AFND

ROTEIRO

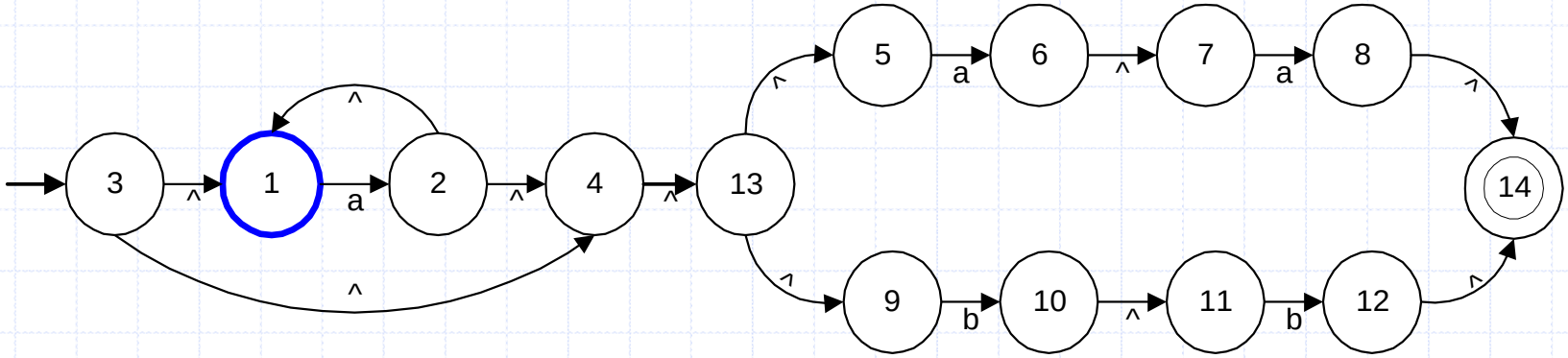
SAIR

ANTERIOR

- A geração das transições do AFND é realizada através da função de transição $\delta' = F\varepsilon(\delta''(\delta'''(q,\hat{}),x))$
- Esta função é dividida nas seguintes etapas:
  - $\delta'''(q,\hat{})$ = gera conjuntos de cada estado do AFND de todos os estados atingidos pelo símbolo vazio (^). O estado que gerou o conjunto também faz parte do conjunto – ETAPA 1
  - $\delta''(\delta'''(q,\hat{}),x)$ = para cada estado dos conjuntos da ETAPA 1 gera-se o conjunto de todos os estados atingidos pelos símbolos do alfabeto – ETAPA 2
  - $F\varepsilon(\delta''(\delta'''(q,\hat{}),x))$ = para todos os estados atingidos na ETAPA 2 gera-se o conjunto de todos os estados atingidos (direta ou indiretamente) pelo símbolo vazio (^) – ETAPA 3

GERAÇÃO DO AFND A PARTIR DO AFε

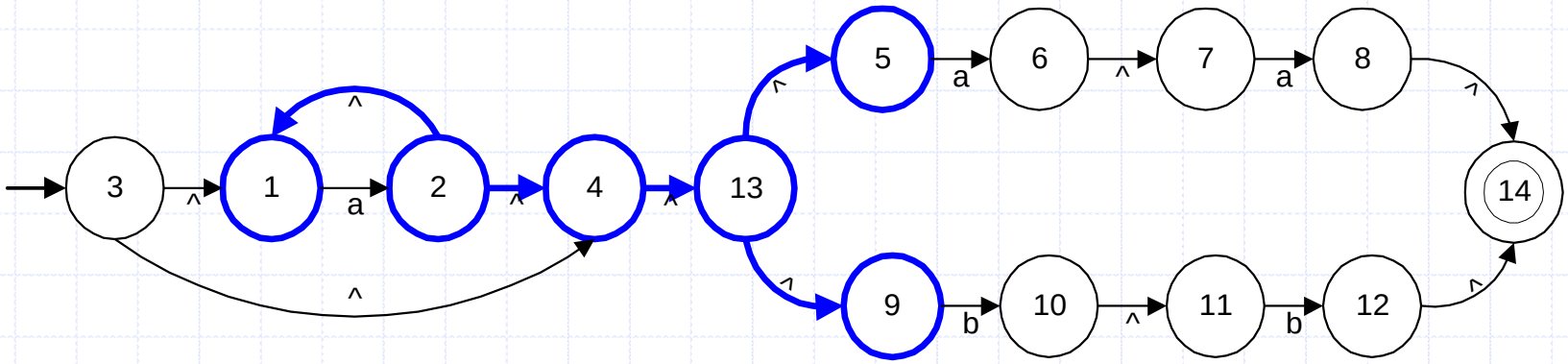
$$\delta' = F\varepsilon(\delta''(\delta'''(q,\hat{}),x))$$



- Sendo q = 1 aplicado sobre $\delta'''(q,\hat{})$ temos:
  - $\delta'''(1,\hat{}) = \{1\}$

GERAÇÃO DO AFND A PARTIR DO AFε

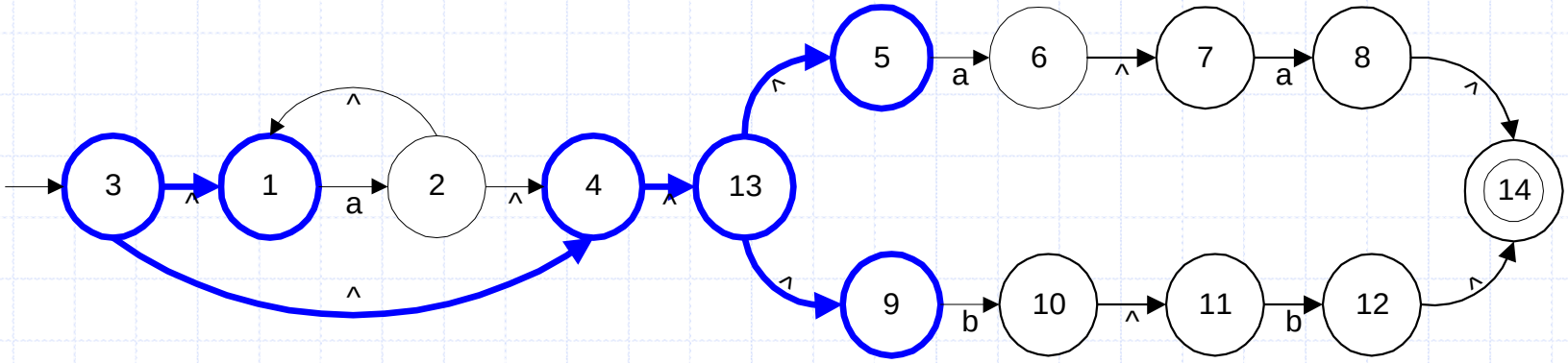
$$\delta' = F\varepsilon(\delta''(\delta'''(q,\hat{}),x))$$



- Sendo q = 2 aplicado sobre $\delta'''(q,\hat{})$ temos:
  - $\delta'''(2,\hat{}) = \{1,2,4,5,9,13\}$

GERAÇÃO DO AFND A PARTIR DO AFε

$$\delta' = F\varepsilon(\delta''(\delta'''(q,\hat{}),x))$$



- Sendo q = 3 aplicado sobre $\delta'''(q,\hat{})$ temos:
  - $\delta'''(3,\hat{}) = \{1,3,4,5,9,13\}$

GERAÇÃO DO AFND A PARTIR DO AFε

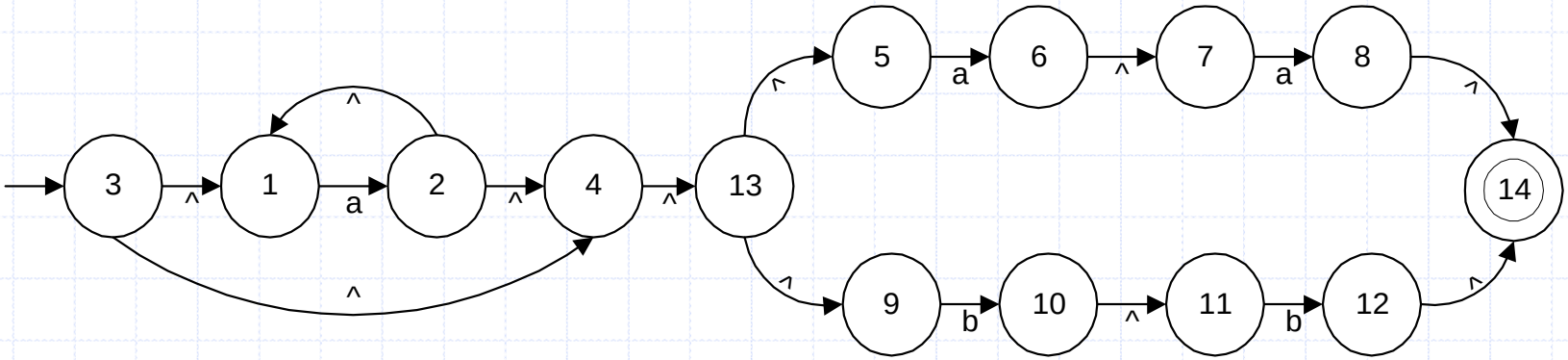
$$\delta' = F\varepsilon(\delta''(\delta'''(q,\hat{}),x))$$

conjuntos obtidos:

- ✓ $\delta'''(1,\hat{}) = \{1\}$
- ✓ $\delta'''(2,\hat{}) = \{1,2,4,5,9,13\}$
- ✓ $\delta'''(3,\hat{}) = \{1,3,4,5,9,13\}$
- ✓ $\delta'''(4,\hat{}) = \{4,5,9,13\}$
- ✓ $\delta'''(5,\hat{}) = \{5\}$
- ✓ $\delta'''(6,\hat{}) = \{6,7\}$
- ✓ $\delta'''(7,\hat{}) = \{7\}$
- ✓ $\delta'''(8,\hat{}) = \{8,14\}$
- ✓ $\delta'''(9,\hat{}) = \{9\}$
- ✓ $\delta'''(10,\hat{}) = \{10,11\}$
- ✓ $\delta'''(11,\hat{}) = \{11\}$
- ✓ $\delta'''(12,\hat{}) = \{12,14\}$
- ✓ $\delta'''(13,\hat{}) = \{5,9,13\}$
- ✓ $\delta'''(14,\hat{}) = \{14\}$

GERAÇÃO DO AFND A PARTIR DO AFε

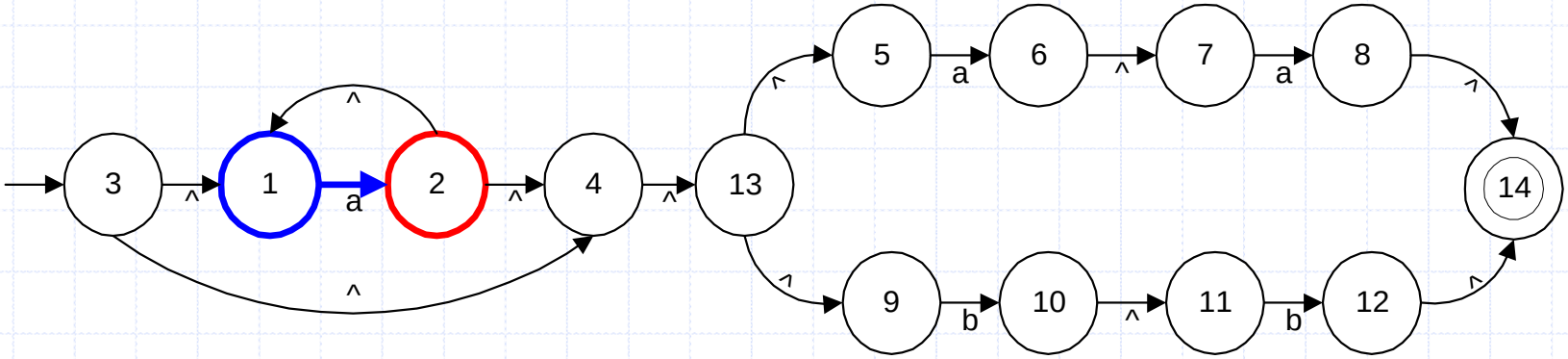
$$\delta' = F\varepsilon(\delta''(\delta'''(q,\hat{}),x))$$



- Elementos da ETAPA 2:
  - $\delta''(\delta'''(q,\hat{}),x)$
  - $\Sigma=\{a,b\}$
  - x = símbolo do Σ
  - Conjunto de cada estado da ETAPA 1

GERAÇÃO DO AFND A PARTIR DO AFε

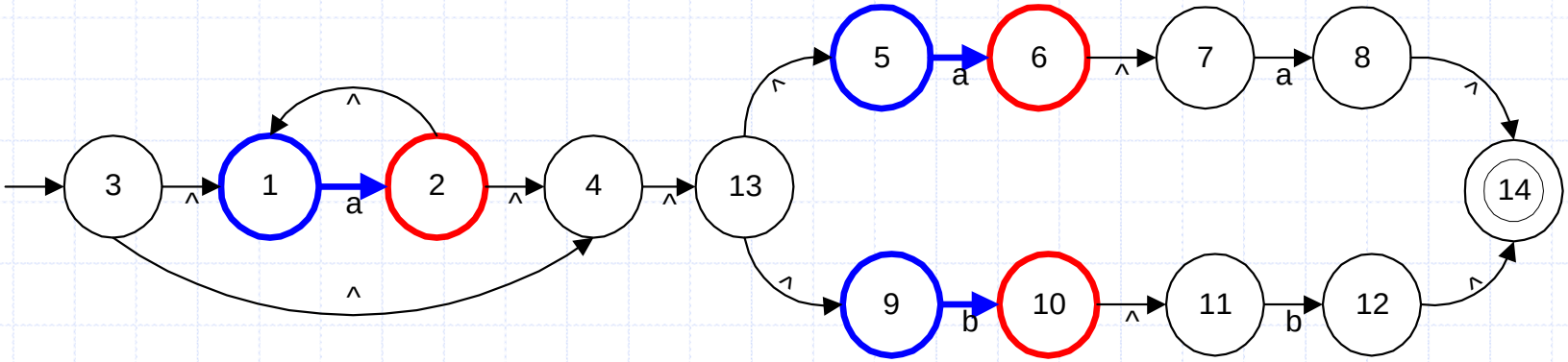
$$\delta' = F\varepsilon(\delta''(\delta'''(q,\hat{}),x))$$



- ✓ $\Sigma=\{a,b\}$
- ✓ $\delta'''(1,\hat{}) = \{1\}$ - ETAPA 1
- ✓ x="a"
- ✓ $\delta''(\{1\},"a")) = \{2\}$
- ✓ x="b"
- ✓ $\delta''(\{1\},"b")) = \{\}$

GERAÇÃO DO AFND A PARTIR DO AFε

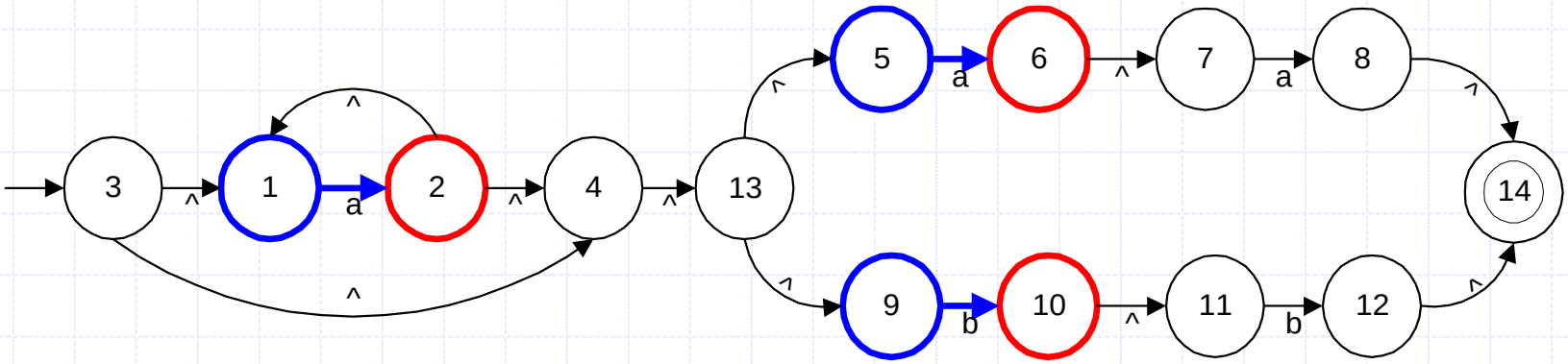
$$\delta' = F\varepsilon(\delta''(\delta'''(q,\hat{}),x))$$



- ✓ $\Sigma=\{a,b\}$
- ✓ $\delta'''(2,\hat{}) = \{1,2,4,5,9,13\}$ - ETAPA 1
- ✓ x="a"
- ✓ $\delta''(\{1,2,4,5,9,13\},"a") = \{2,6\}$
- ✓ x="b"
- ✓ $\delta''(\{1,2,4,5,9,13\},"b") = \{10\}$

GERAÇÃO DO AFND A PARTIR DO AFε

$$\delta' = F\varepsilon(\delta''(\delta'''(q,\hat{}),x))$$



- ✓ $\Sigma=\{a,b\}$
- ✓ $\delta'''(3,\hat{}) = \{1,3,4,5,9,13\}$ - ETAPA 1
- ✓ x="a"
- ✓ $\delta''(\{1,3,4,5,9,13\},"a") = \{2,6\}$
- ✓ x="b"
- ✓ $\delta''(\{1,3,4,5,9,13\},"b") = \{10\}$

GERAÇÃO DO AFND A PARTIR DO AFε

$$\delta' = F\varepsilon(\delta''(\delta'''(q,\hat{}),x))$$

ROTEIRO

SAIR

ANTERIOR

RESULTADOS DA ETAPA 2 PARA Σ={a,b}

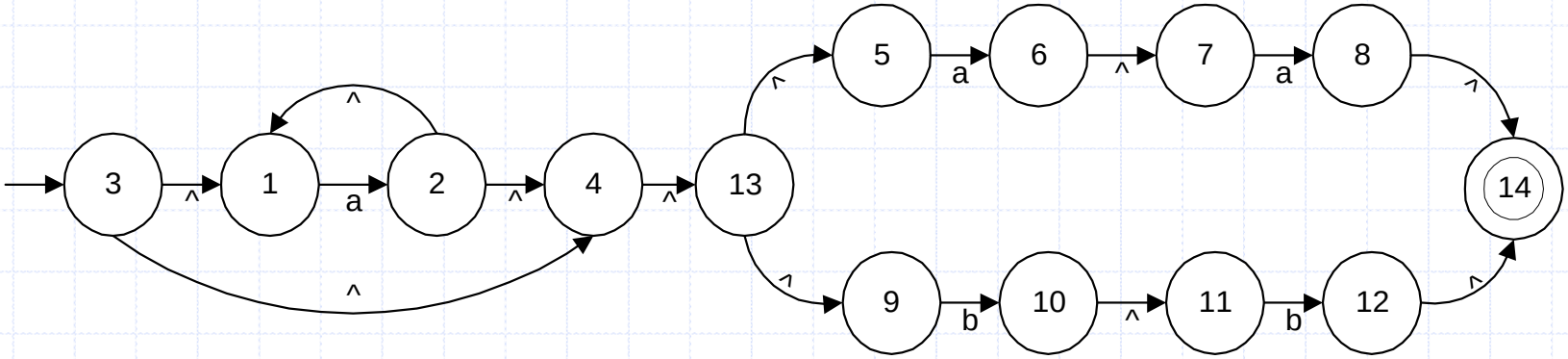
- ✓ δ″({1},″a″) = {2} (Estado 1)
- ✓ δ″({1},″b″) = {}
- ✓ δ″({1,2,4,5,9,13},″a″) = {2,6} (Estado 2)
- ✓ δ″({1,2,4,5,9,13},″b″) = {10}
- ✓ δ″({1,3,4,5,9,13},″a″) = {2,6} (Estado 3)
- ✓ δ″({1,3,4,5,9,13},″b″) = {10}
- ✓ δ″({4,5,9,13} ,″a″) = {6} (Estado 4)
- ✓ δ″({4,5,9,13} ,″b″) = {10}

...

- ✓ δ″({14} ,″a″) = {} (Estado 14)
- ✓ δ″({14} ,″b″) = {}

GERAÇÃO DO AFND A PARTIR DO AFε

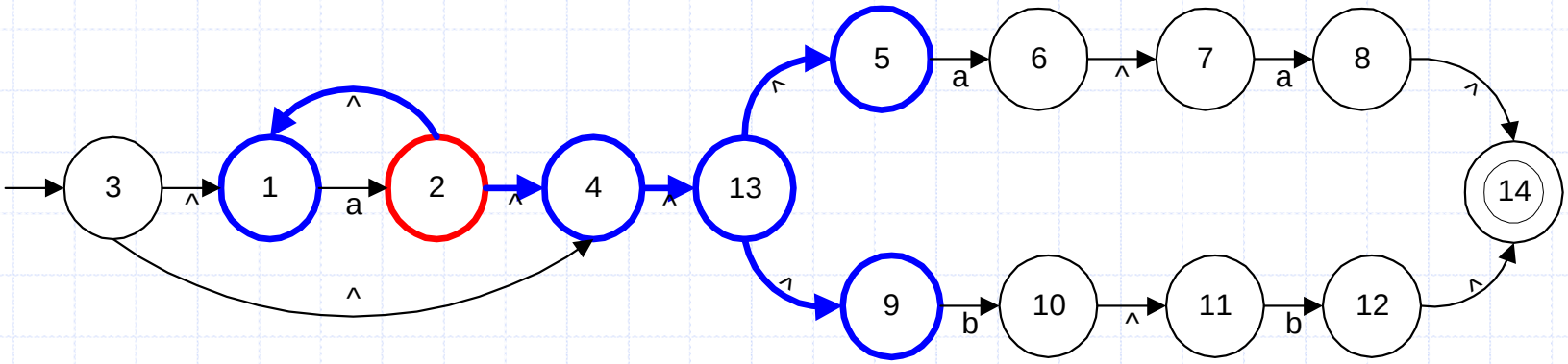
$$\delta' = F\varepsilon(\delta''(\delta'''(q,\hat{}),x))$$



- Elementos da ETAPA 3:
  - $\Sigma=\{a,b\}$
  - $F\varepsilon(\delta''(\delta'''(q,\hat{}),x))$ onde $F\varepsilon=\delta(q,\hat{})$
  - Conjunto de cada estado da ETAPA 2

GERAÇÃO DO AFND A PARTIR DO AFε

$$\delta' = F\varepsilon(\delta''(\delta'''(q,\hat{}),x))$$



TRANSIÇÕES DO ESTADO 1

- ✓ $\Sigma=\{a,b\}$
- ✓ $\delta''(\{1\},"a")$ = {2} (Estado 1)
- ✓ $\delta''(\{1\},"b")$ = {}
- ✓ $F\varepsilon(\{2\})$ = **{1,2,4,5,9,13}** (símbolo "a" do Σ)
- ✓ $F\varepsilon(\{\})$ = **{}** (símbolo "b" do Σ)

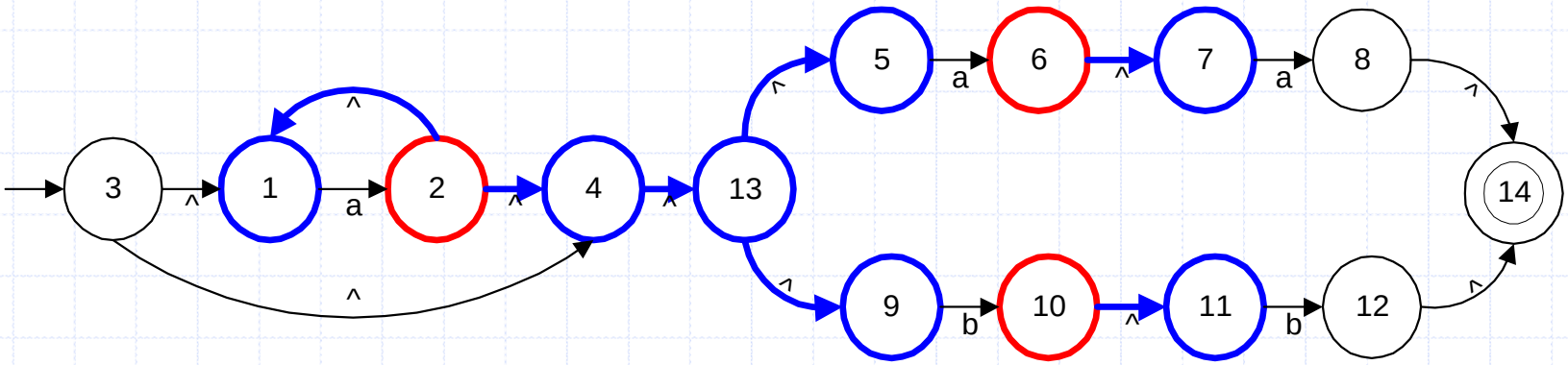
ROTEIRO

SAIR

ANTERIOR

GERAÇÃO DO AFND A PARTIR DO AFε

$$\delta' = F\varepsilon(\delta''(\delta'''(q,\hat{}),x))$$



TRANSIÇÕES DO ESTADO 2

- ✓ $\Sigma=\{a,b\}$
- ✓ $\delta''(\{1,2,4,5,9,13\},"a") = \{2,6\}$ (Estado 2)
- ✓ $\delta''(\{1,2,4,5,9,13\},"b") = \{10\}$
- ✓ $F\varepsilon(\{2,6\}) = \mathbf{\{1,2,4,5,7,9,13\}}$ (símbolo "a" do Σ)
- ✓ $F\varepsilon(\{10\}) = \mathbf{\{10,11\}}$ (símbolo "b" do Σ)

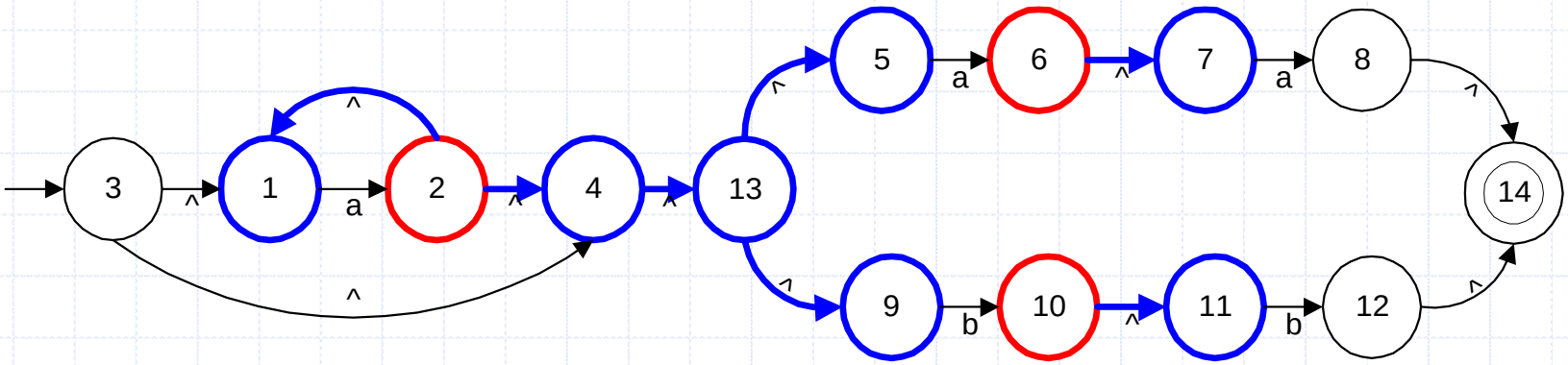
ROTEIRO

SAIR

ANTERIOR

GERAÇÃO DO AFND A PARTIR DO AFε

$$\delta' = F\varepsilon(\delta''(\delta'''(q,\hat{}),x))$$



TRANSIÇÕES DO ESTADO 3

- ✓ $\Sigma=\{a,b\}$
- ✓ $\delta''(\{1,3,4,5,9,13\},"a") = \{2,6\}$ (Estado 3)
- ✓ $\delta''(\{1,3,4,5,9,13\},"b") = \{10\}$
- ✓ $F\varepsilon(\{2,6\}) = \mathbf{\{1,2,4,5,7,9,13\}}$ (símbolo "a" do Σ)
- ✓ $F\varepsilon(\{10\}) = \mathbf{\{10,11\}}$ (símbolo "b" do Σ)

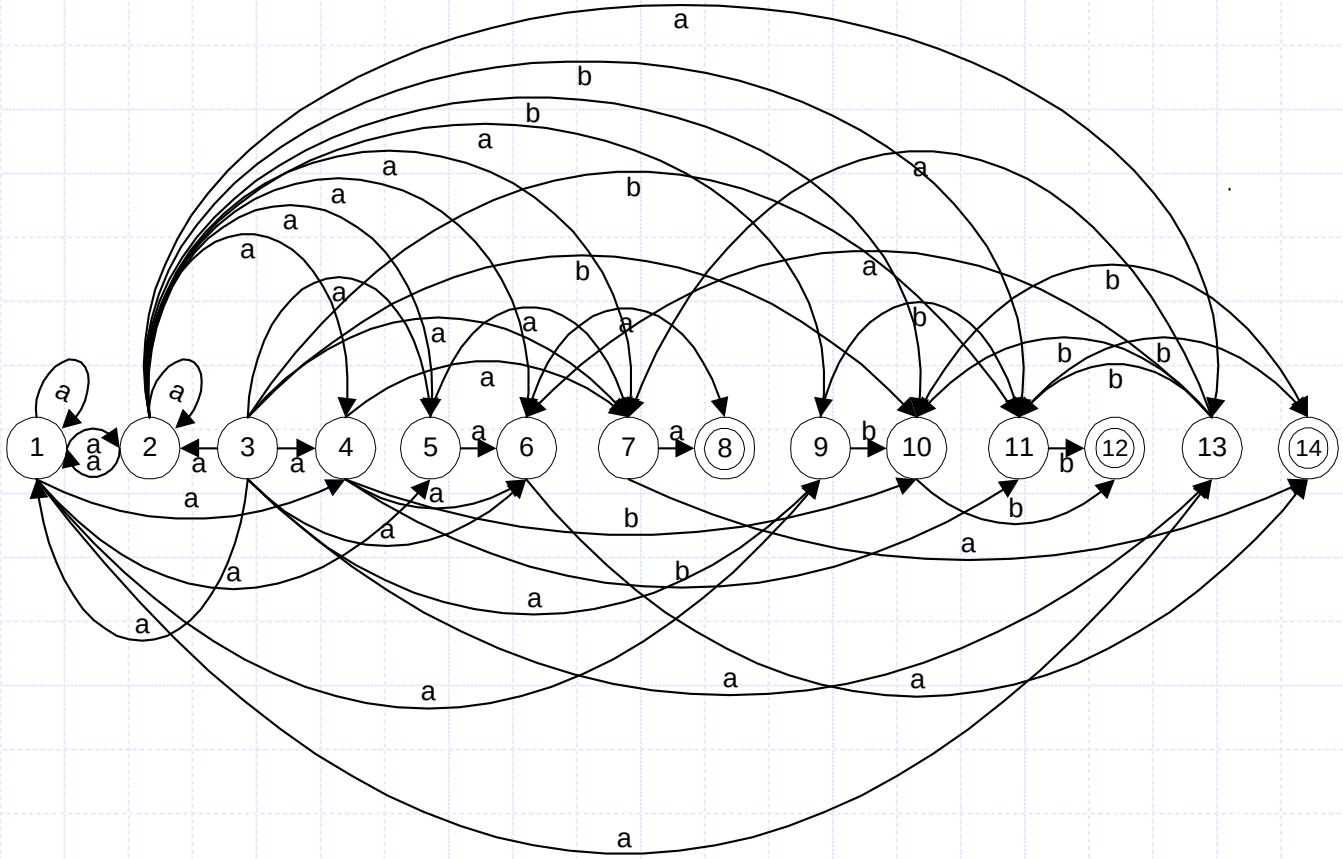
ROTEIRO

SAIR

ANTERIOR

GERAÇÃO DO AFND A PARTIR DO AFε

AFND RESULTANTE



ALGORITMO DE [SIL2000]
FASES

ROTEIRO

SAIR

ANTERIOR

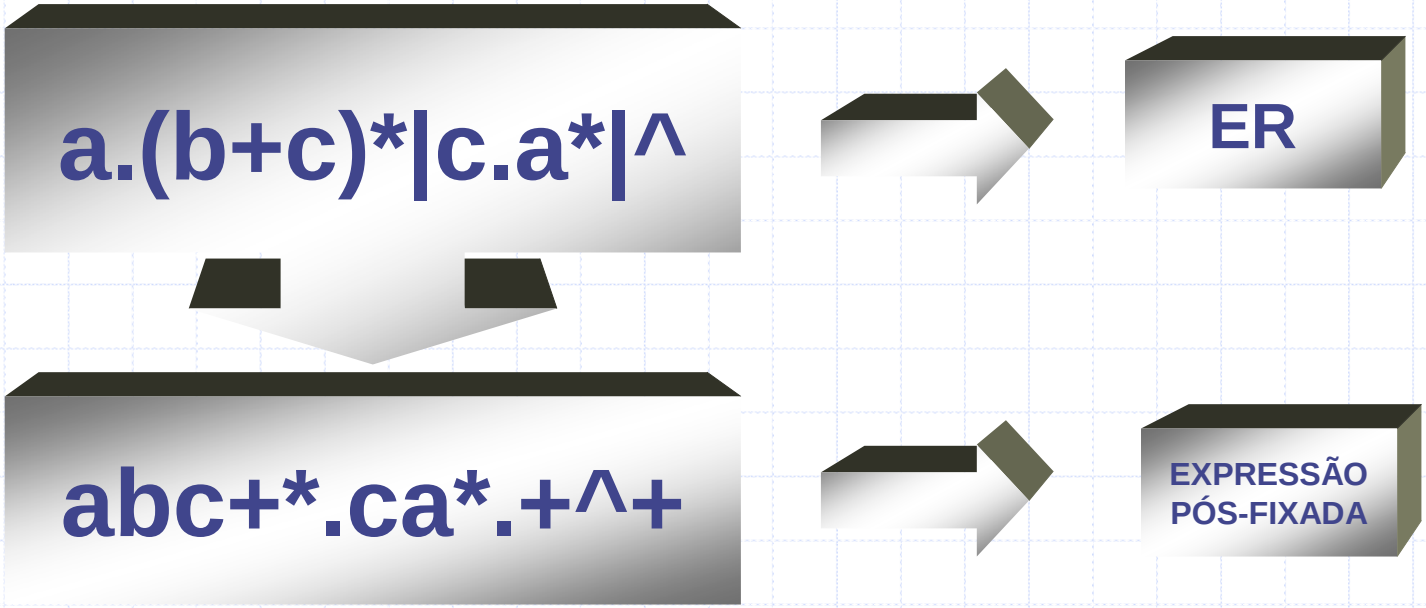
GERAÇÃO DA EXPRESSÃO PÓS-FIXADA

ROTEIRO

SAIR

ANTERIOR

- A EXPRESSÃO PÓS-FIXADA É GERADA ATRAVÉS DE UMA DEFINIÇÃO LLC DESCRITA EM [SIL2000].



EXPRESSÃO PÓS-FIXADA LLC DE [SIL2000]

ROTEIRO

SAIR

ANTERIOR

^ → símbolo vazio
R → expressão regular
SR → simples expressão regular
RR → resto da expressão regular

R → SR RR;
RR → '+' SR | '.' SR | ^;
SR → T SR1;
SR1 → '+' T SR1 | ^; {EMITIR '+'}
T → F T1;
T1 → '.' F T1 | ^; {EMITIR '.'}
F → '(' R ')' F1 | #S F1; {EMITIR Símbolo S}
F1 → '*' F1 | ^; {EMITIR '*'}

EXPRESSÃO PÓS-FIXADA
Geração pela LLC

ROTEIRO

SAIR

ANTERIOR

- ✓ DADA A EXPRESSÃO REGULAR:
  - ✓ **a.(b+c)*+c.a*+^**
- ✓ PROCESSO: LER TODOS OS SÍMBOLOS DA EXPRESSÃO REGULAR DA ESQUERDA PARA A DIREITA E SUBMETÊ-LOS A DEFINIÇÃO EM LLC APRESENTADA POR [SIL2000]

VEJA PROCESSO

TABELA DO DESMONTE

- ✓ CONSISTE EM GERAR UMA TABELA A PARTIR DA EXPRESSÃO PÓS-FIXADA OBTIDA NA FASE 1
- ✓ A EXPRESSÃO PÓS-FIXADA É SUBMETIDA A UM CONJUNTO DE OPERAÇÕES QUE DARÃO ORIGEM A TABELA
- ✓ A EXPRESSÃO PÓS-FIXADA É PROCESSADA SÍMBOLO A SIMBOLO DA DIREITA PARA A ESQUERDA. CADA SÍMBOLO É SUBMETIDO AO CONJUNTO DE PASSOS QUE FORMARÃO A TABELA DO DESMONTE.

TABELA DO DESMONTE

- ✓ESTADO INICIAL: 1
- ✓ESTADO FINAL: 6
  - ✓abc+*.ca*.+^+
    - ✓**1 PARA ESTADO INICIAL**
    - ✓**1 PARA ESTADO FINAL**
    - ✓**2 PARA O OPERADOR DE CONCATENAÇÃO**
    - ✓**2 PARA O OPERADOR DE FECHAMENTO**

TABELA DO DESMONTE

abc+*.ca*.+^+

ROTEIRO

SAIR

ANTERIOR

SAÍDA:	SÍMBOLO:	CHEGADA:
1	+	6
1	+	6
1	^	6
1	.	6
1	.	6
1	c	2
...	...	...

VEJA PROCESSO

GRAFO DE TRANSIÇÕES (GT)
Geração a partir da FASE 2

ROTEIRO

SAIR

ANTERIOR

- É GERADO INICIALMENTE UMA NOVA TABELA A PARTIR DA TABELA DO DESMONTE
- NESTA NOVA TABELA ESTÃO ELIMINADAS TODAS AS LINHAS QUE CONTÉM OPERADORES DE UNIÃO, FECHAMENTO E CONCATENAÇÃO
- A NOVA TABELA RESULTA NO GRAFO DE TRANSIÇÕES.

GRAFO DE TRANSIÇÕES (GT)
Geração a partir da FASE 2

ROTEIRO

SAIR

ANTERIOR

SAÍDA:	SÍMBOLO:	CHEGADA:
1	^	6
1	c	2
2	^	3
3	^	6
3	a	3
...	...	...

GRAFO DE TRANSIÇÕES (GT)
Geração a partir da FASE 2

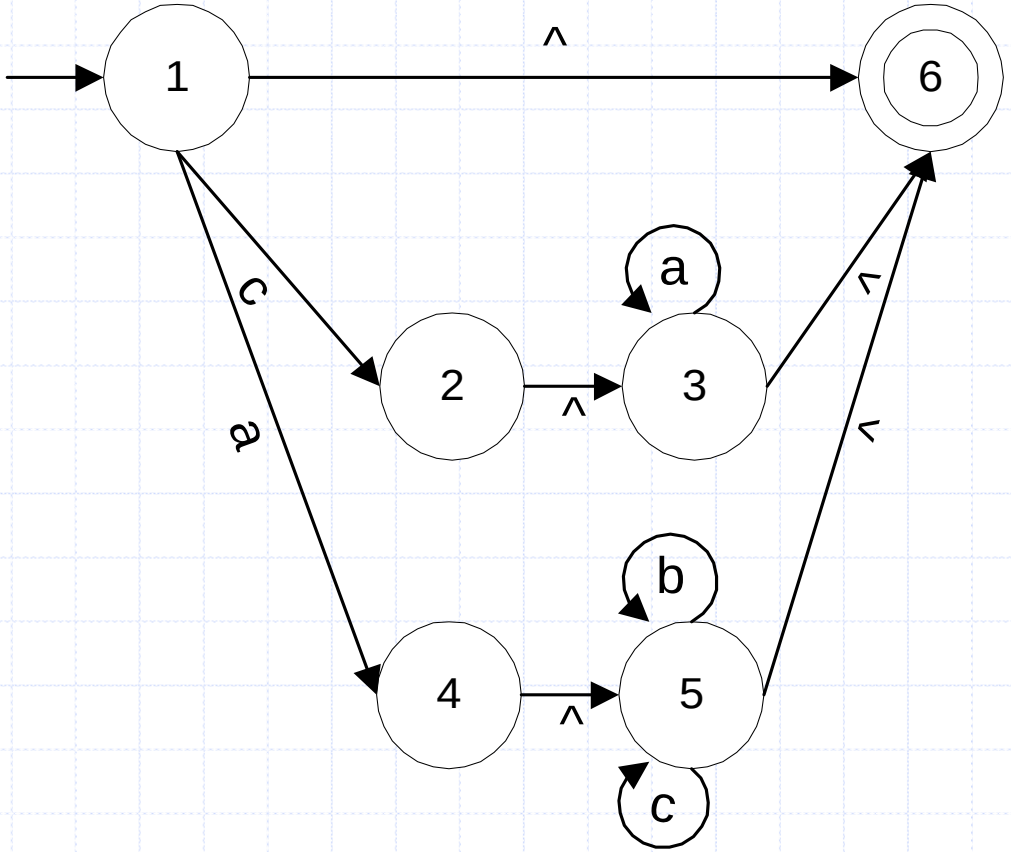


TABELA DE TRANSIÇÕES DO AFD

ROTEIRO

SAIR

ANTERIOR

- ✓ CONSISTE NA GERAÇÃO DE UMA TABELA DE TRANSIÇÕES A PARTIR DO GT
- ✓ ESTADO INICIAL: ESTADOS INICIAIS DO GT E TODOS OS ESTADOS ATINGIDOS A PARTIR DOS ESTADOS INICIAIS PELO SÍMBOLO VAZIO (^)
- ✓ O PROCESSO DE GERAÇÃO DA TABELA DE TRANSIÇÕES A PARTIR DO GT É IDÊNTICO AO PROCESSO DE GERAÇÃO DA TABELA DE TRANSIÇÕES A PARTIR DO AFND NO ALGORITMO DE [HOP1979]
- ✓ O GT É UMA REPRESENTAÇÃO GRÁFICA DO MODELO MATEMÁTICO AUTÔMATO FINITO

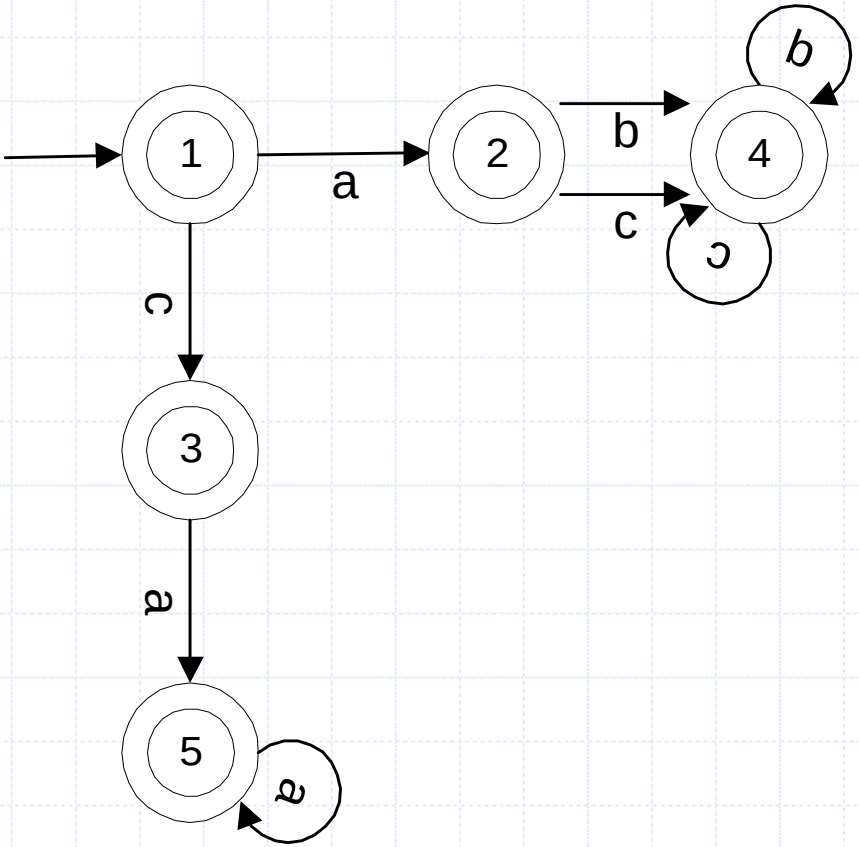
TABELA DE TRANSIÇÕES DO AFD

- ESTADOS FINAIS: CONJUNTO DE ESTADOS FORMADOS A PARTIR DO GT E QUE CONTÉM ALGUM DOS ESTADOS FINAIS DO GT
- PROCESSAMENTO DOS SÍMBOLOS ALFABETO COM OS ESTADOS DO GT E SUAS TRANSIÇÕES DE ACORDO COM O ALGORITMO APRESENTADO POR [MAN1974]
- UMA EXPLICAÇÃO DETALHADA DOS PASSOS DO ALGORITMO DESCRITO POR [MAN1974] PODE SER ENCONTRADO NA MONOGRAFIA

TABELA DE TRANSIÇÕES DO AFD

ESTADOS DO AFD	Mi	Mj		
		a	b	c
1 (-/+)	{1,6}	{4,5,6}	{}	{2,3,6}
2 (+)	{4,5,6}	{}	{5,6}	{5,6}
3 (+)	{2,3,6}	{3,6}	{}	{}
4 (+)	{5,6}	{}	{5,6}	{5,6}
5 (+)	{3,6}	{3,6}	{}	{}

AFD OBTIDO A PARTIR DA TABELA DE TRANSIÇÕES



TRANSFORMAÇÃO DE UM AFD EM PROGRAMA

ROTEIRO

SAIR

ANTERIOR

- ✓ CONSISTE EM TRANFORMAR UM AFD EM UM PROGRAMA
- ✓ A IMPLEMENTAÇÃO DE UM AFD EM UM COMPUTADOR SEGUE DUAS LINHAS:
  - ✓ IMPLEMENTAÇÃO POR CÓDIGO DIRETO
  - ✓ CONTROLE DE TABELA DE TRANSIÇÕES

TRANSFORMAÇÃO DE UM AFD EM PROGRAMA

ROTEIRO

SAIR

ANTERIOR

- ✓ IMPLEMENTAÇÃO POR CÓDIGO DIRETO: cada estado e suas transições são implementadas diretamente no programa
- ✓ CONTROLE DE TABELA DE TRANSIÇÕES: um interpretador universal interpreta a Tabela de Transições do AFD

VEJA O INTERPRETADOR UNIVERSAL DE TABELAS

TRANSFORMAÇÃO DE UM AFD EM PROGRAMA

ROTEIRO

SAIR

ANTERIOR

- ✓ Os algoritmos vistos neste trabalho realizam a transformação de uma ER em um AFD. Como parte complementar, foi implementada a transformação do AFD resultante em uma função na linguagem de programação Pascal. Estas transformação é equivalente nos dois protótipos que implementam os algoritmos.
- ✓ A função é gerada para uma *unit* da linguagem de programação Pascal e esta *unit* poderá ser utilizada em um programa desta linguagem

EXEMPLO DE PROGRAMA QUE DECLARA FUNÇÃO

ESPECIFICAÇÃO DOS PROTÓTIPOS

ROTEIRO

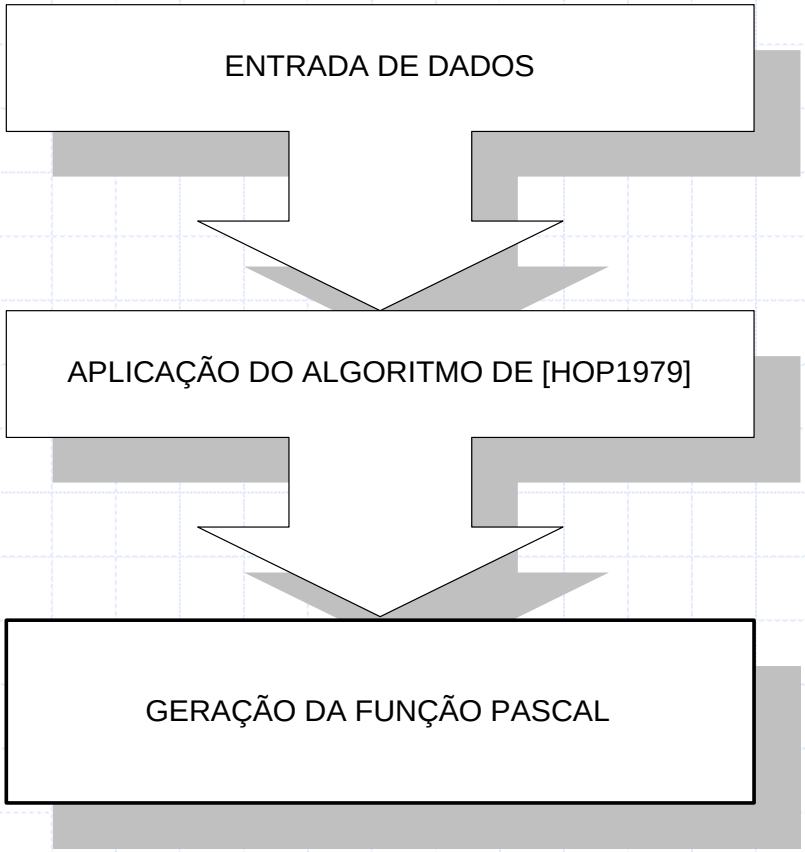
SAIR

ANTERIOR

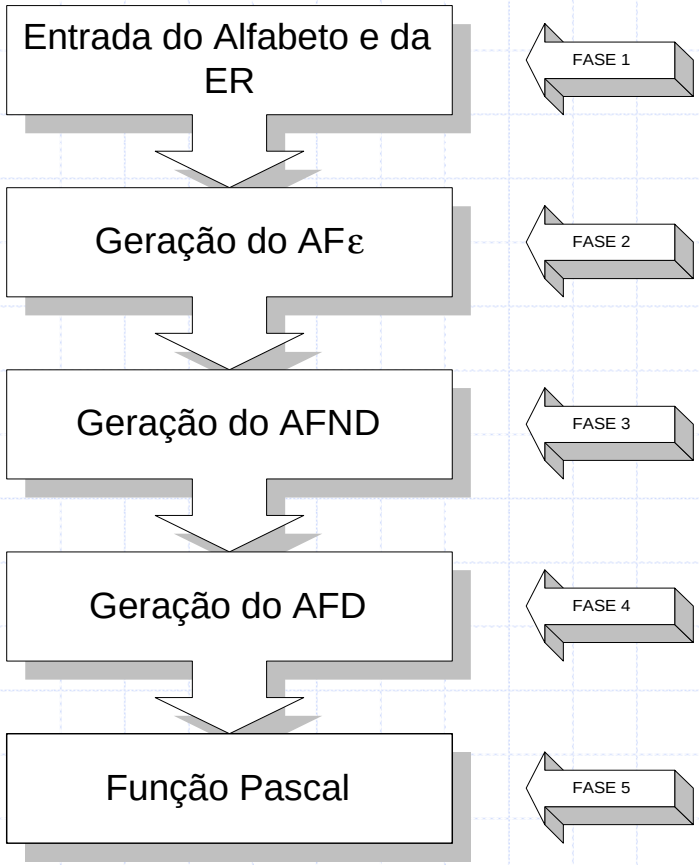
- PROTÓTIPO DE [HOP1979]
- PROTÓTIPO DE [SIL2000]

- ✓ OS ALGORITMOS FORAM IMPLEMENTADOS SEPARADAMENTE COM O OBJETIVO DE SIMPLIFICAR O ENTENDIMENTO DE CADA UM.

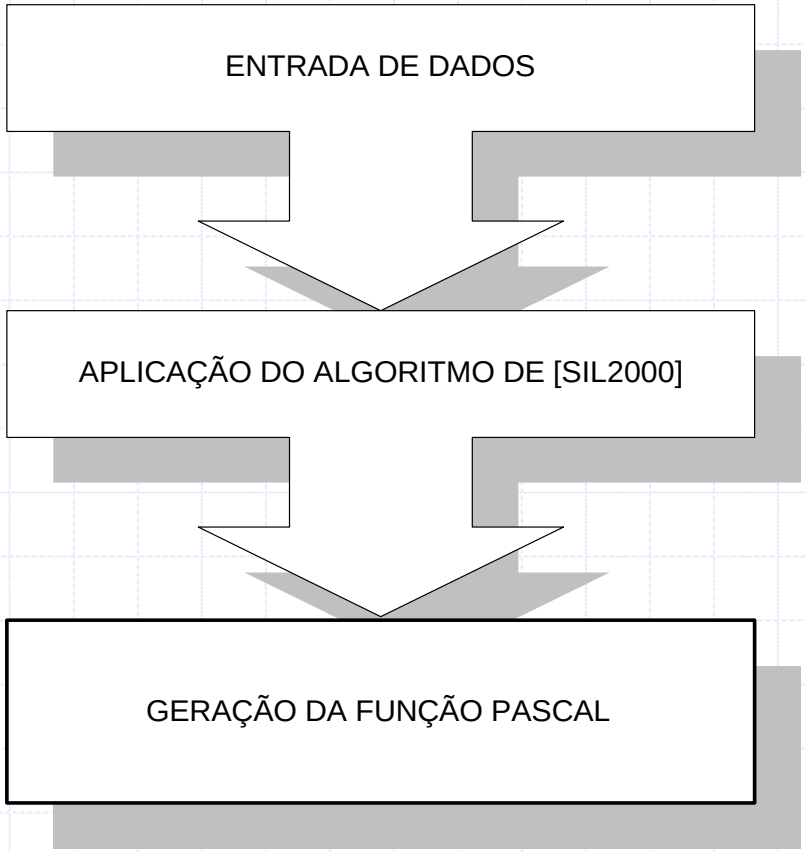
ESPECIFICAÇÃO DOS PROTÓTIPOS – HOP1979



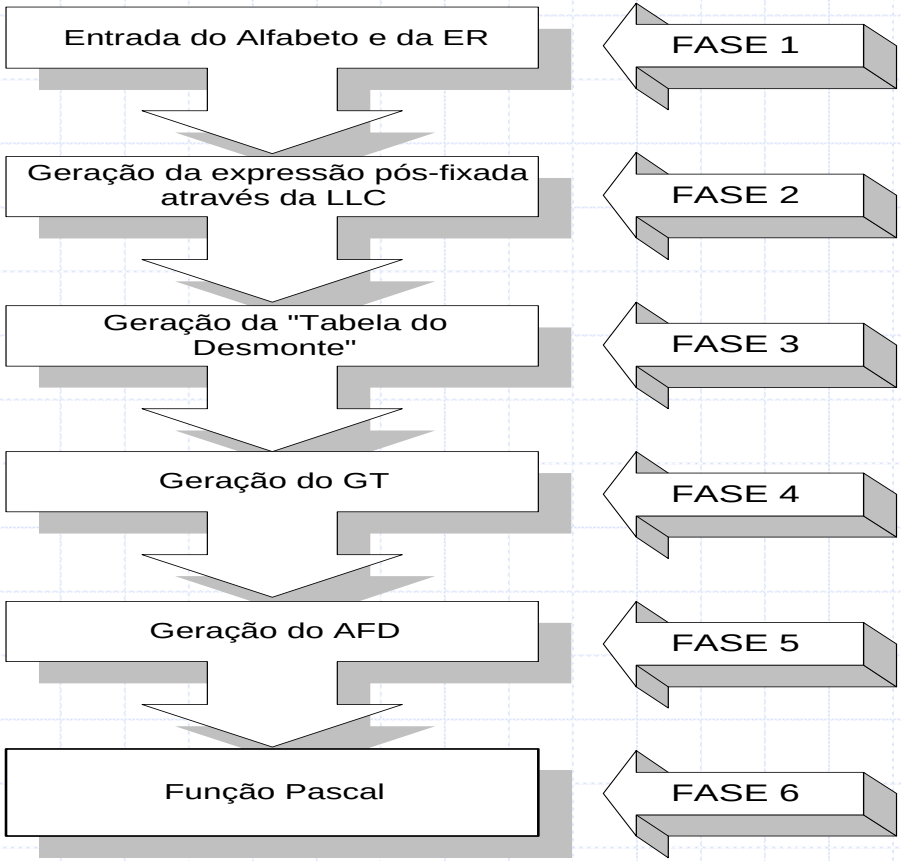
ESPECIFICAÇÃO DOS PROTÓTIPOS – HOP1979



ESPECIFICAÇÃO DOS PROTÓTIPOS – SIL2000



ESPECIFICAÇÃO DOS PROTÓTIPOS SIL2000

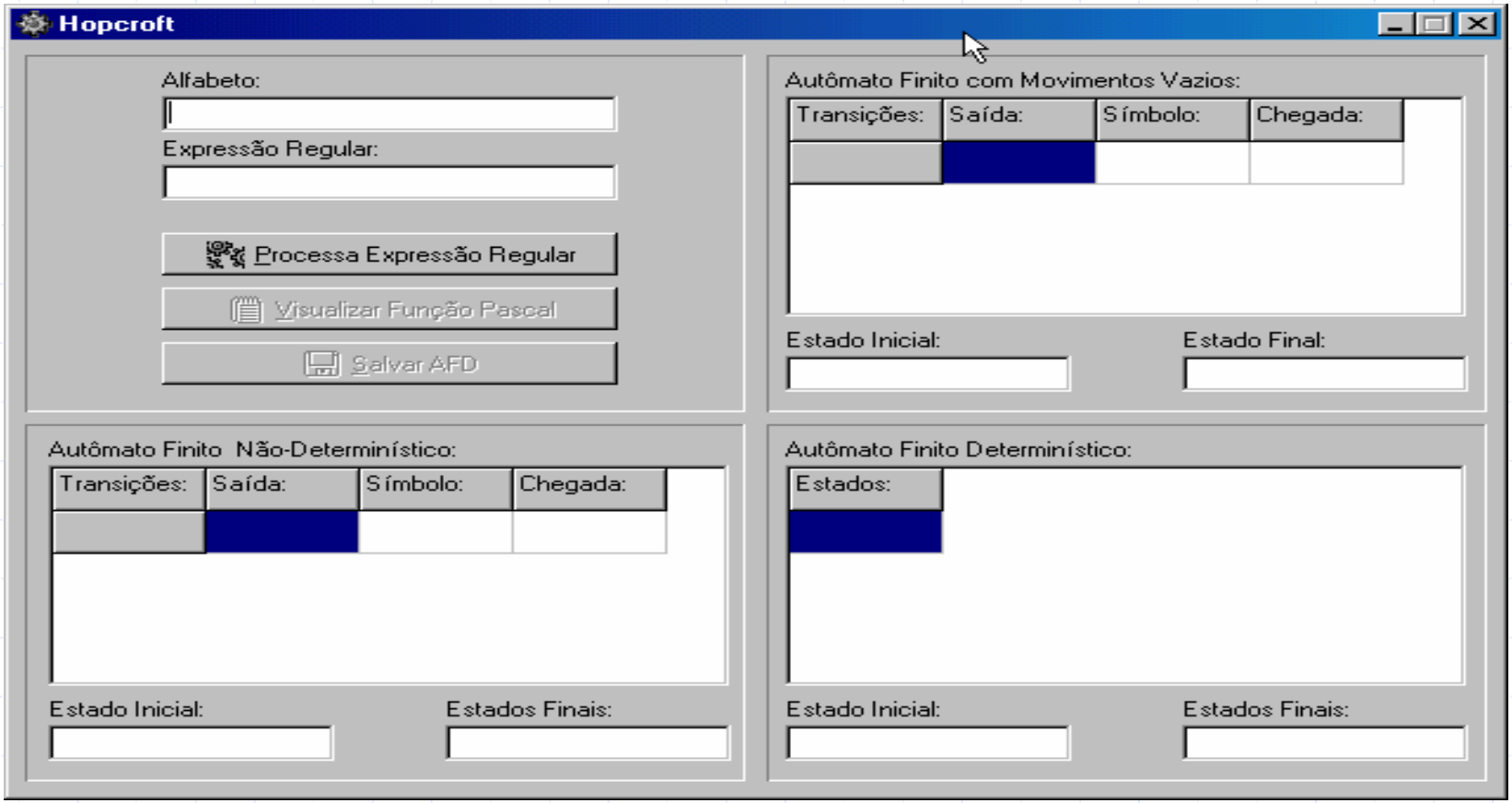


APRESENTAÇÃO DOS PROTÓTIPOS
TELA PRINCIPAL ([HOP1979])

ROTEIRO

SAIR

ANTERIOR

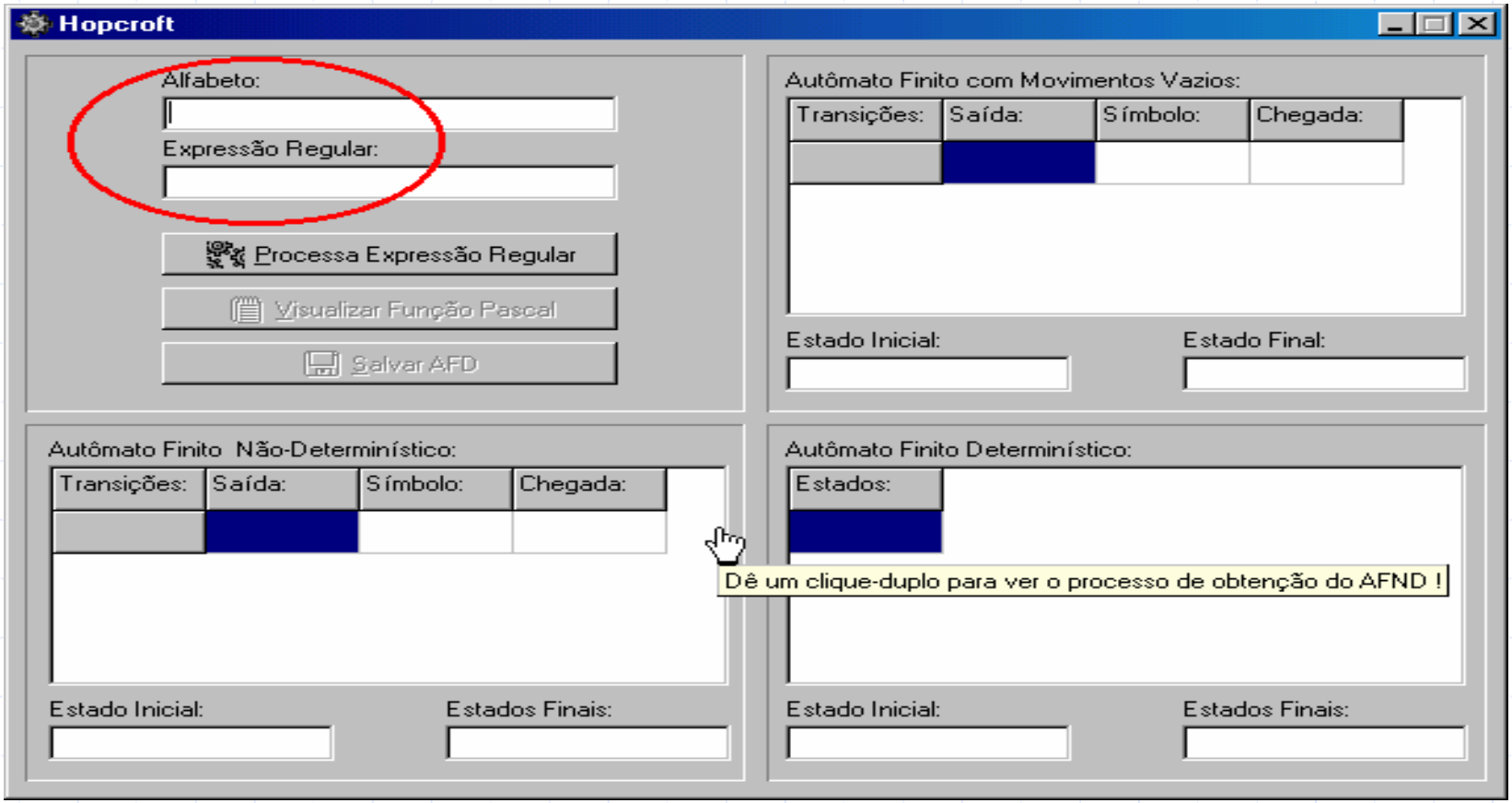


APRESENTAÇÃO DOS PROTÓTIPOS
ENTRADA DE DADOS

ROTEIRO

SAIR

ANTERIOR

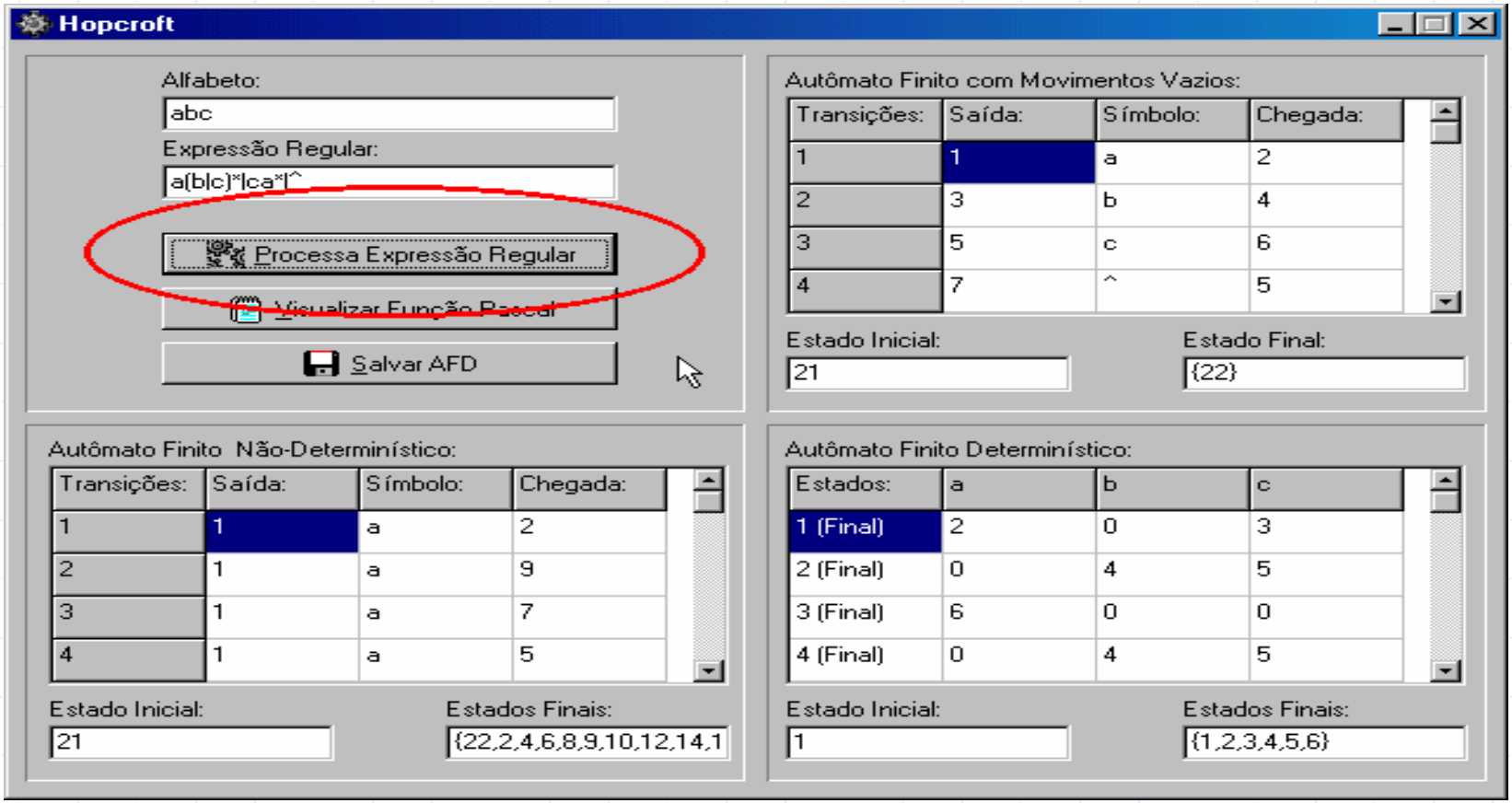


APRESENTAÇÃO DOS PROTÓTIPOS
PROCESSAR EXPRESSÃO REGULAR

ROTEIRO

SAIR

ANTERIOR

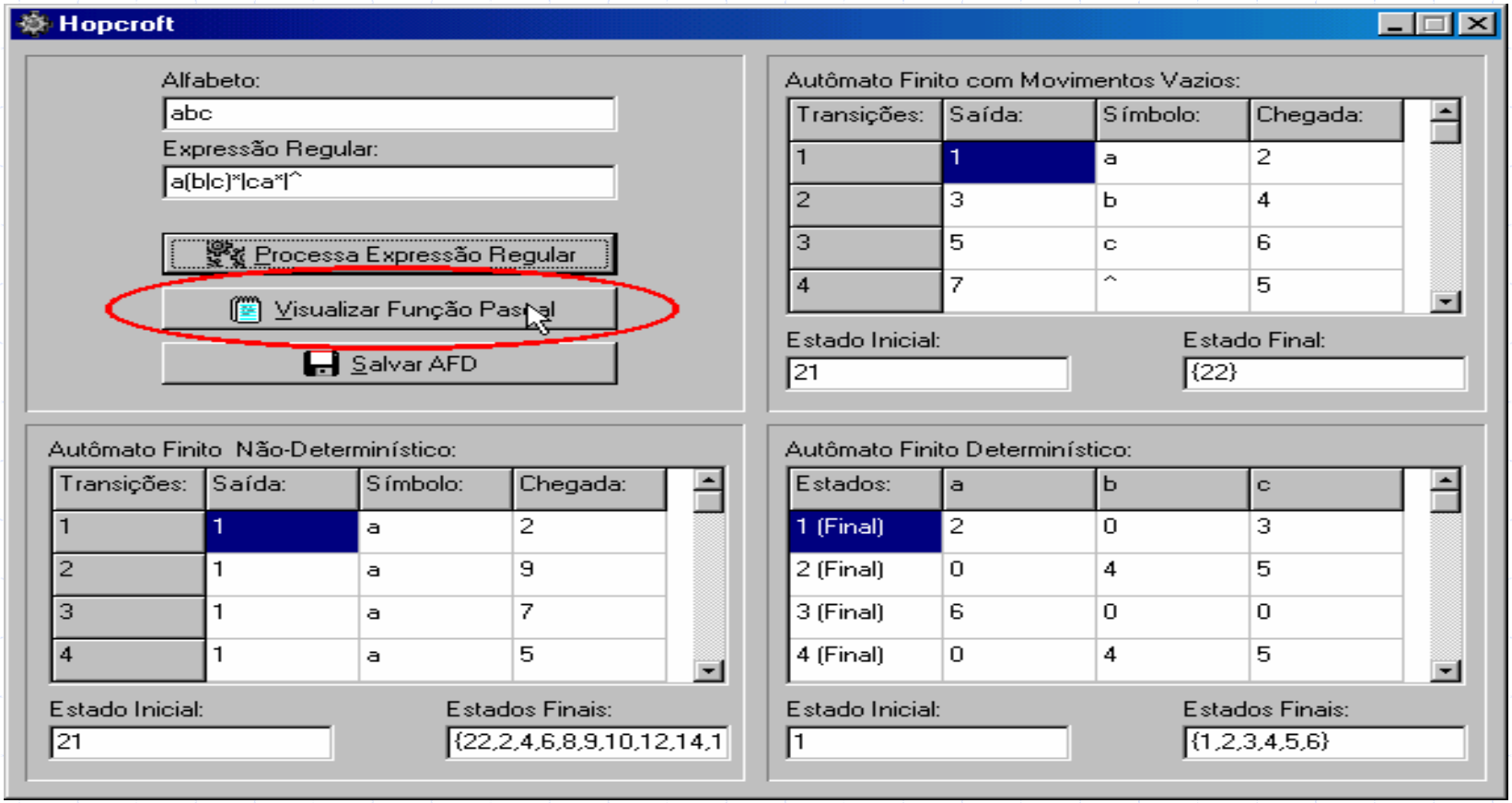


APRESENTAÇÃO DOS PROTÓTIPOS
VISUALIZAR FUNÇÃO

ROTEIRO

SAIR

ANTERIOR

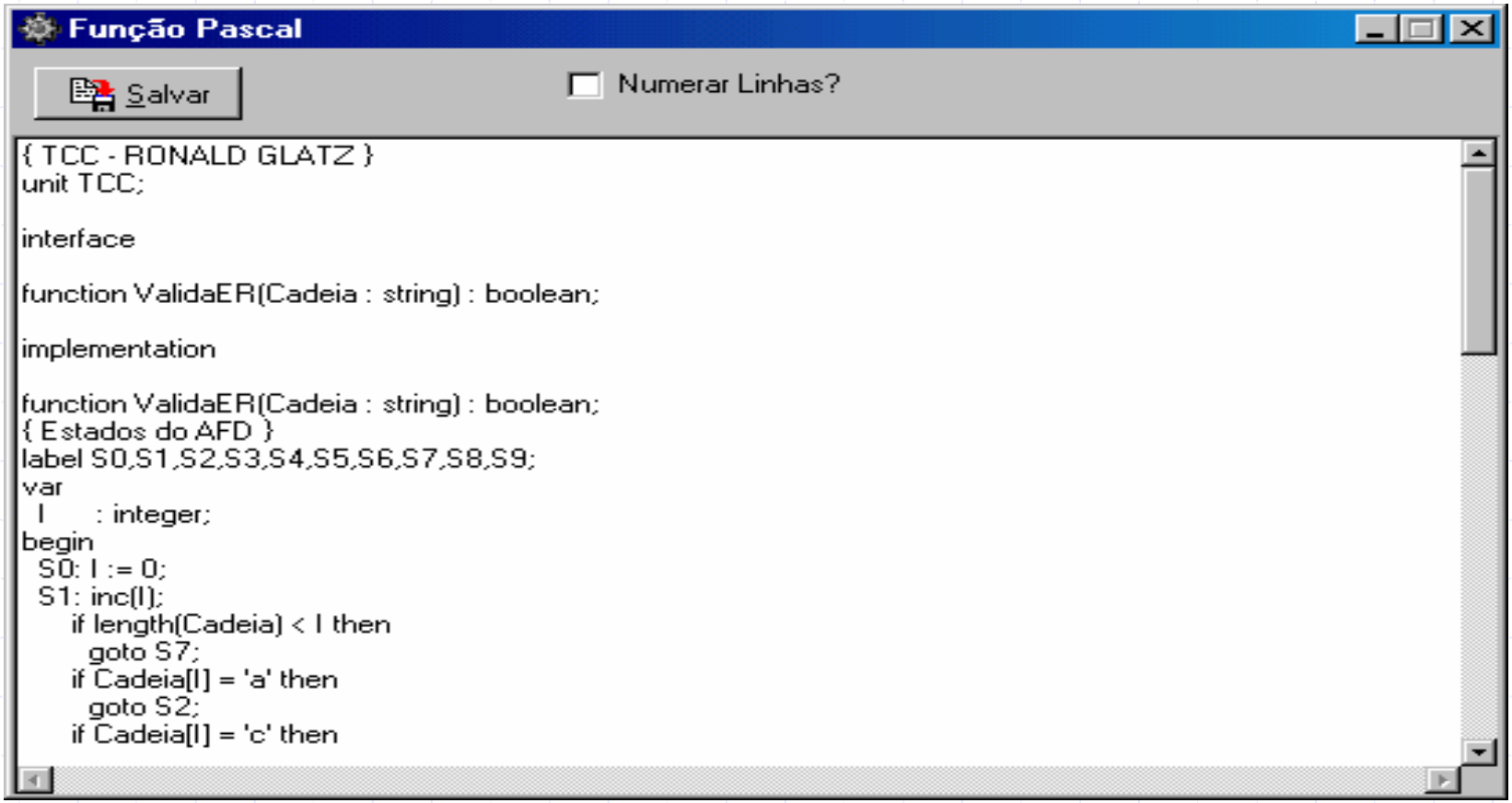


APRESENTAÇÃO DOS PROTÓTIPOS
FUNÇÃO PASCAL

ROTEIRO

SAIR

ANTERIOR



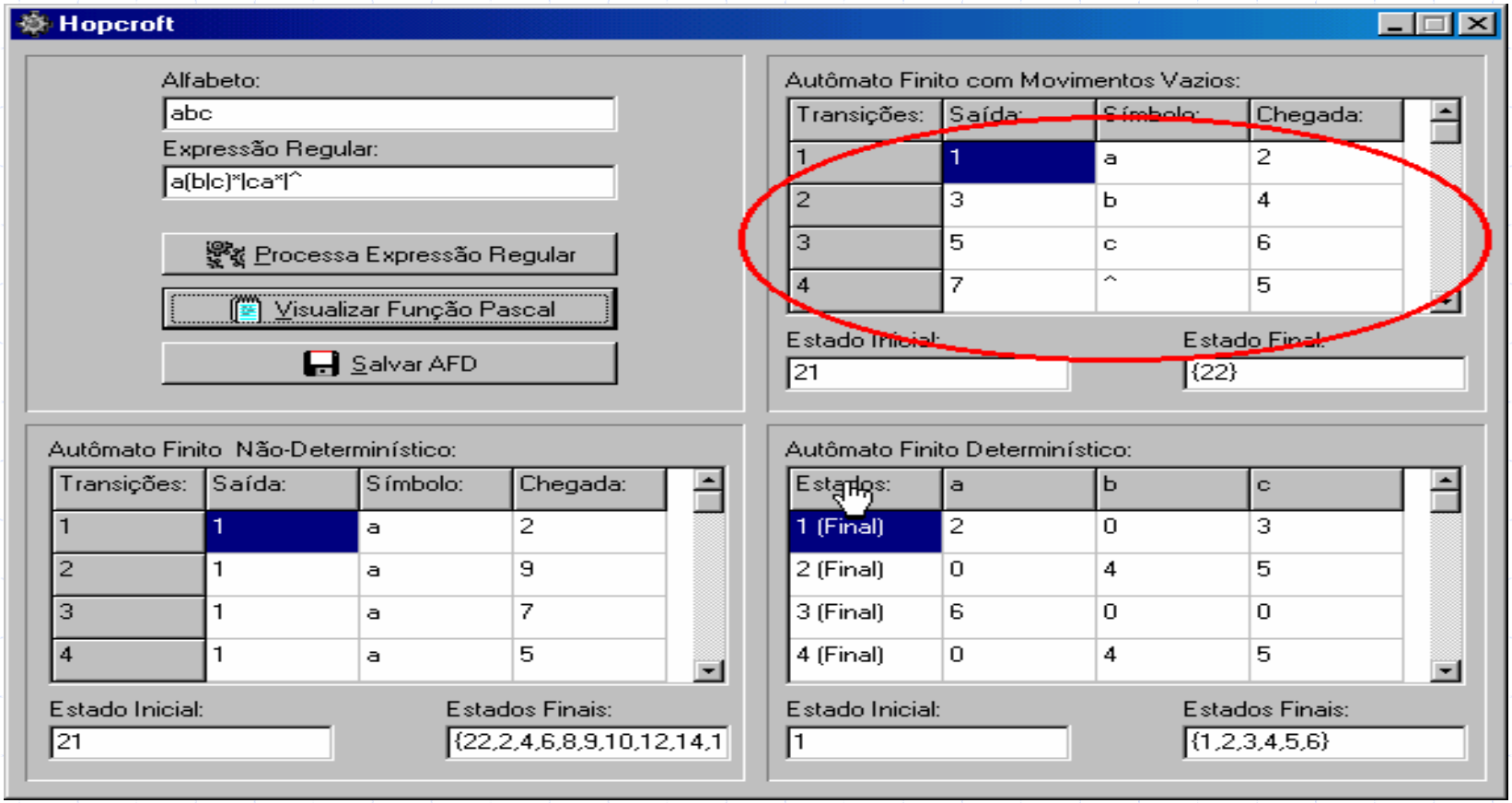
APRESENTAÇÃO DOS PROTÓTIPOS

AFε

ROTEIRO

SAIR

ANTERIOR

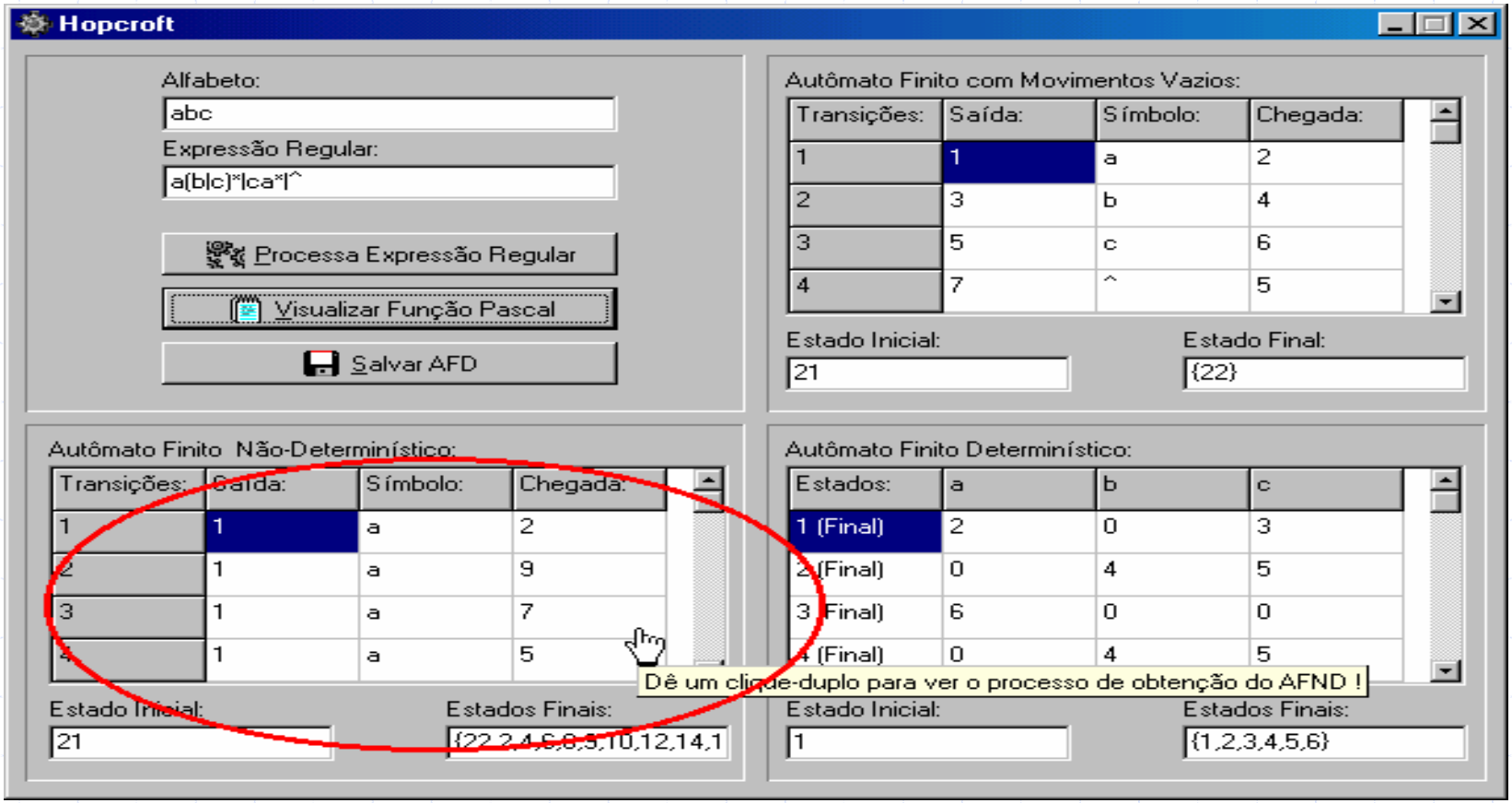


APRESENTAÇÃO DOS PROTÓTIPOS AFND

ROTEIRO

SAIR

ANTERIOR

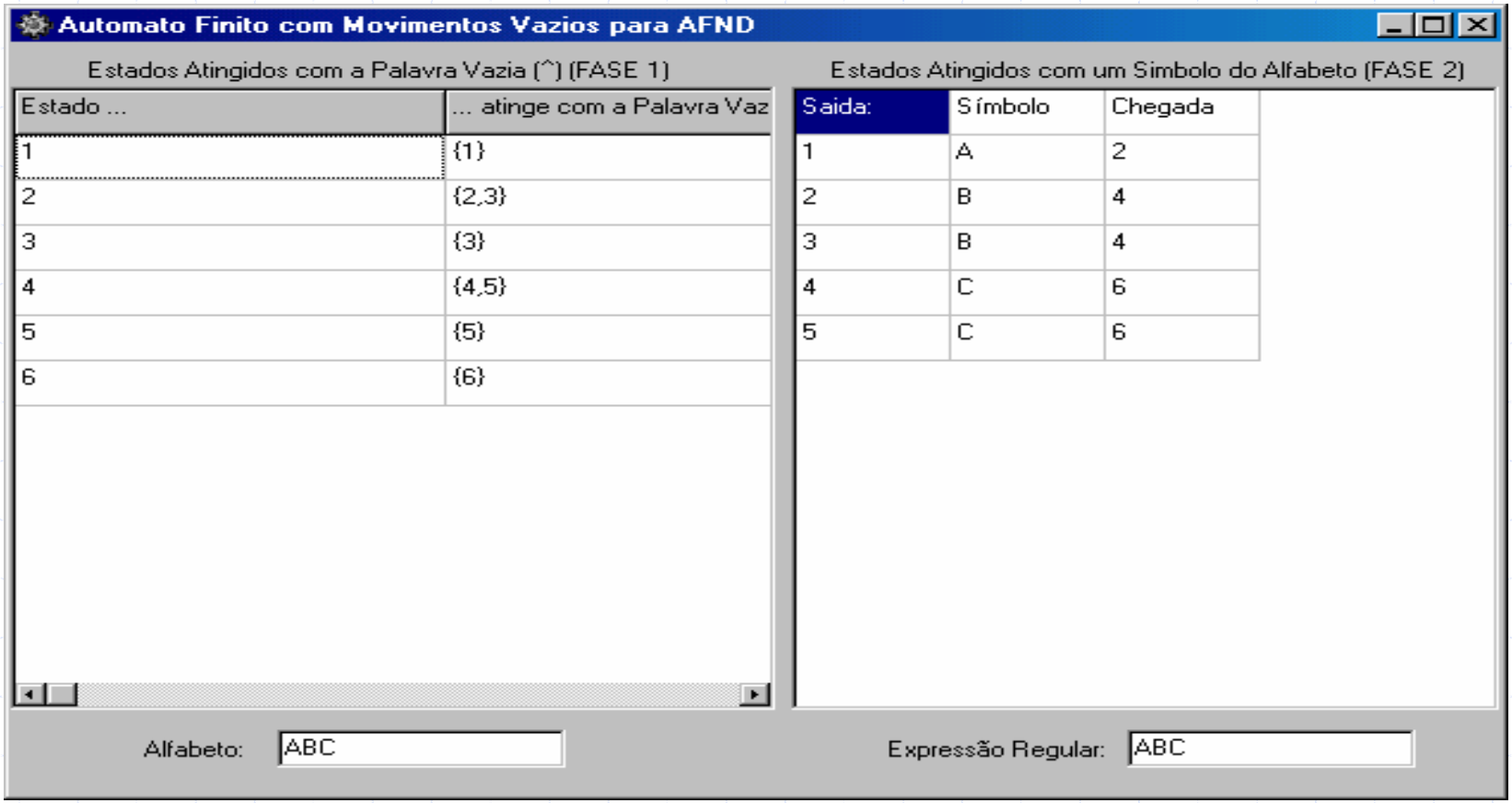


APRESENTAÇÃO DOS PROTÓTIPOS
AFND – FASES DA δ

ROTEIRO

SAIR

ANTERIOR

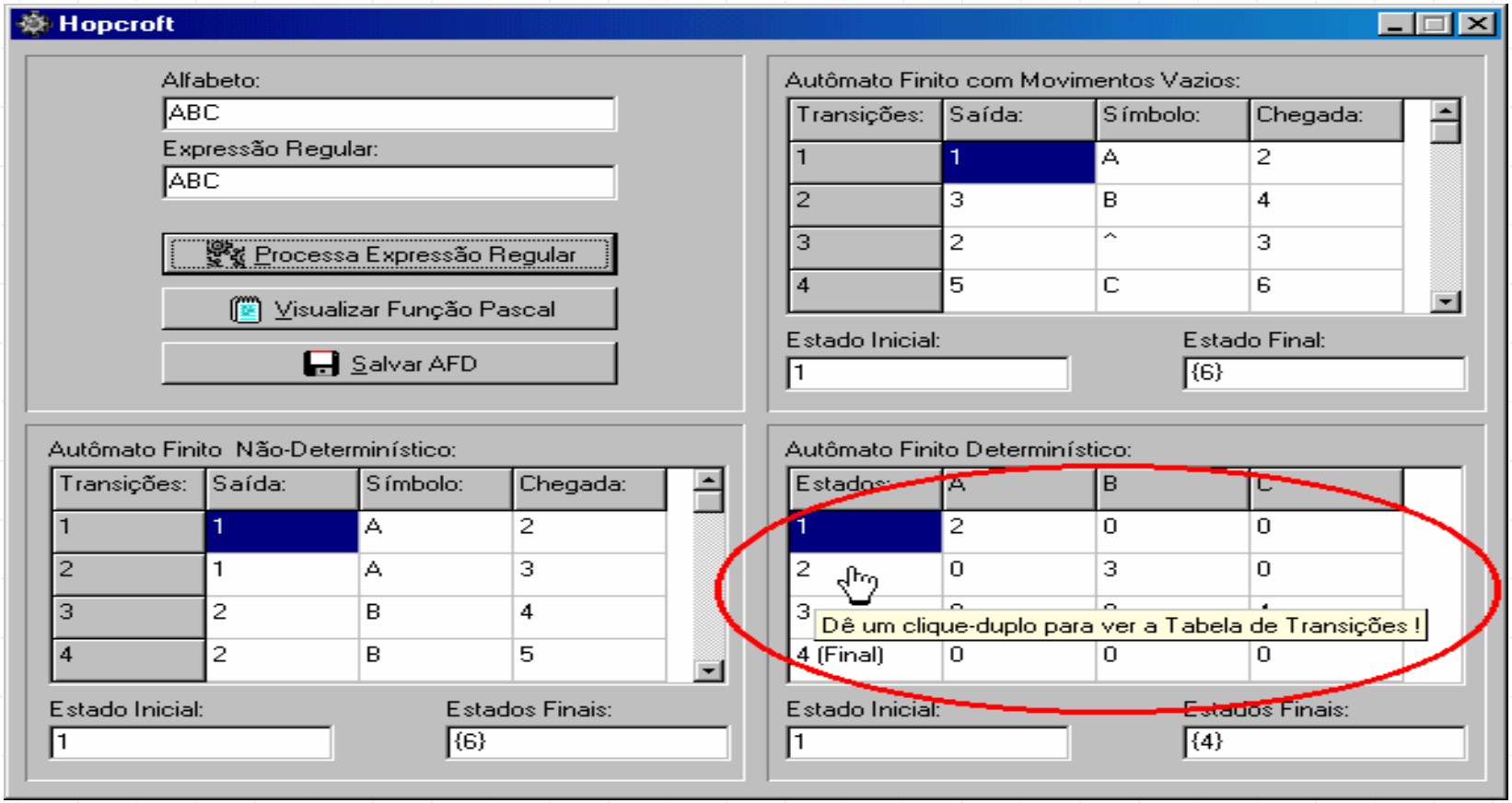


APRESENTAÇÃO DOS PROTÓTIPOS AFD

ROTEIRO

SAIR

ANTERIOR

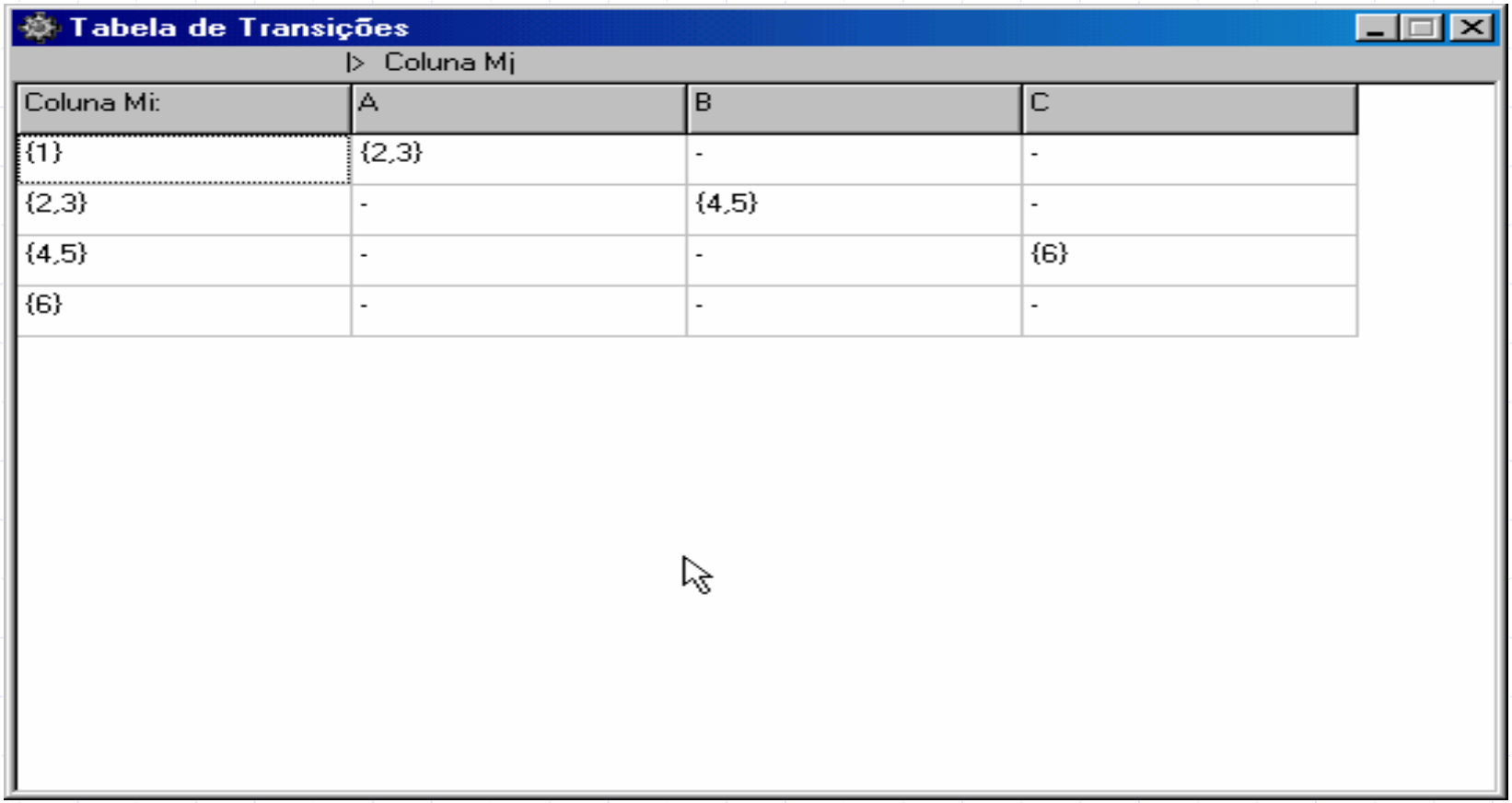


APRESENTAÇÃO DOS PROTÓTIPOS
TABELA DE TRANSIÇÕES DO AFD

ROTEIRO

SAIR

ANTERIOR

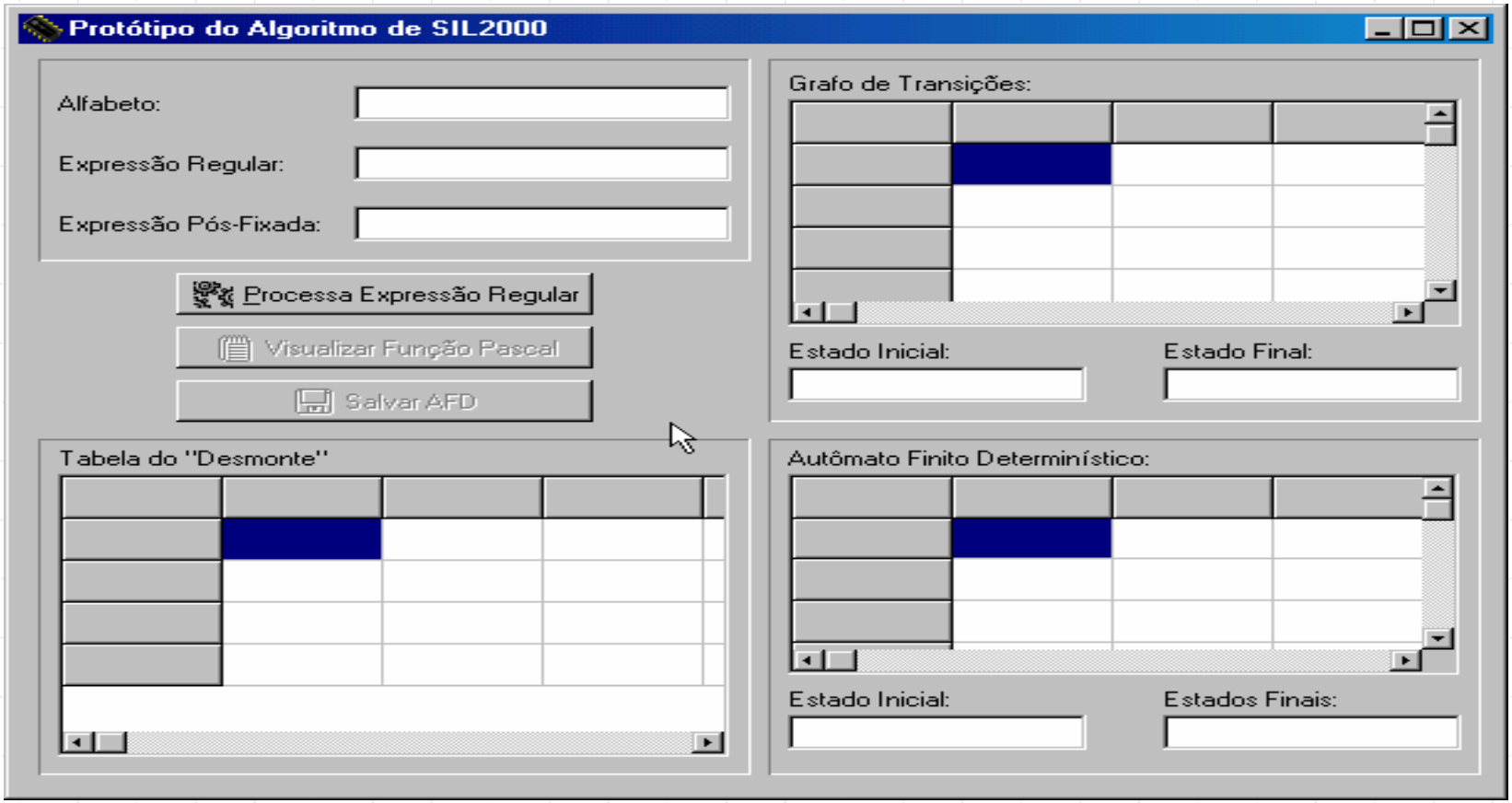


APRESENTAÇÃO DOS PROTÓTIPOS
TELA PRINCIPAL ([SIL2000])

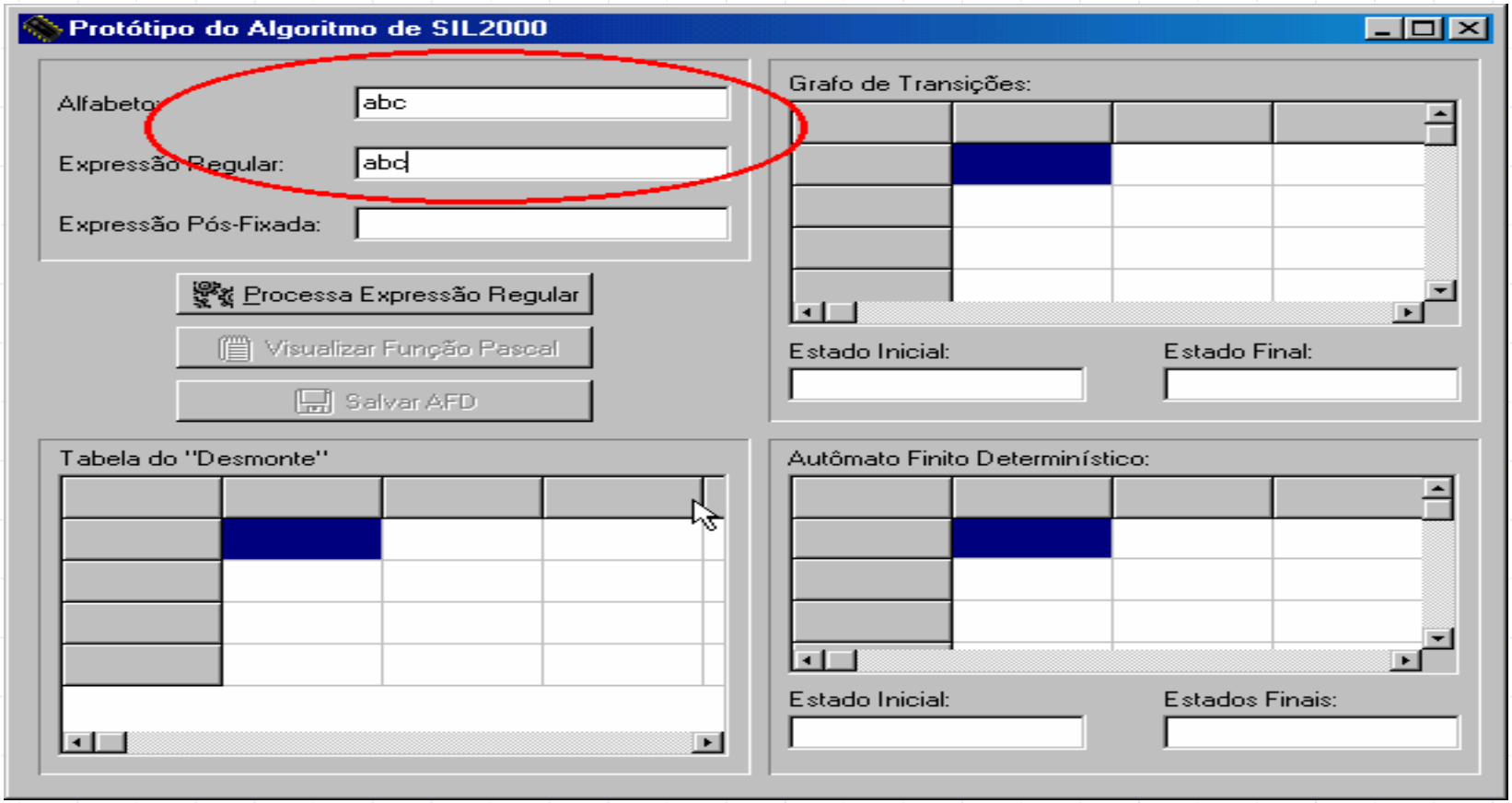
ROTEIRO

SAIR

ANTERIOR



APRESENTAÇÃO DOS PROTÓTIPOS
ENTRADA DE DADOS

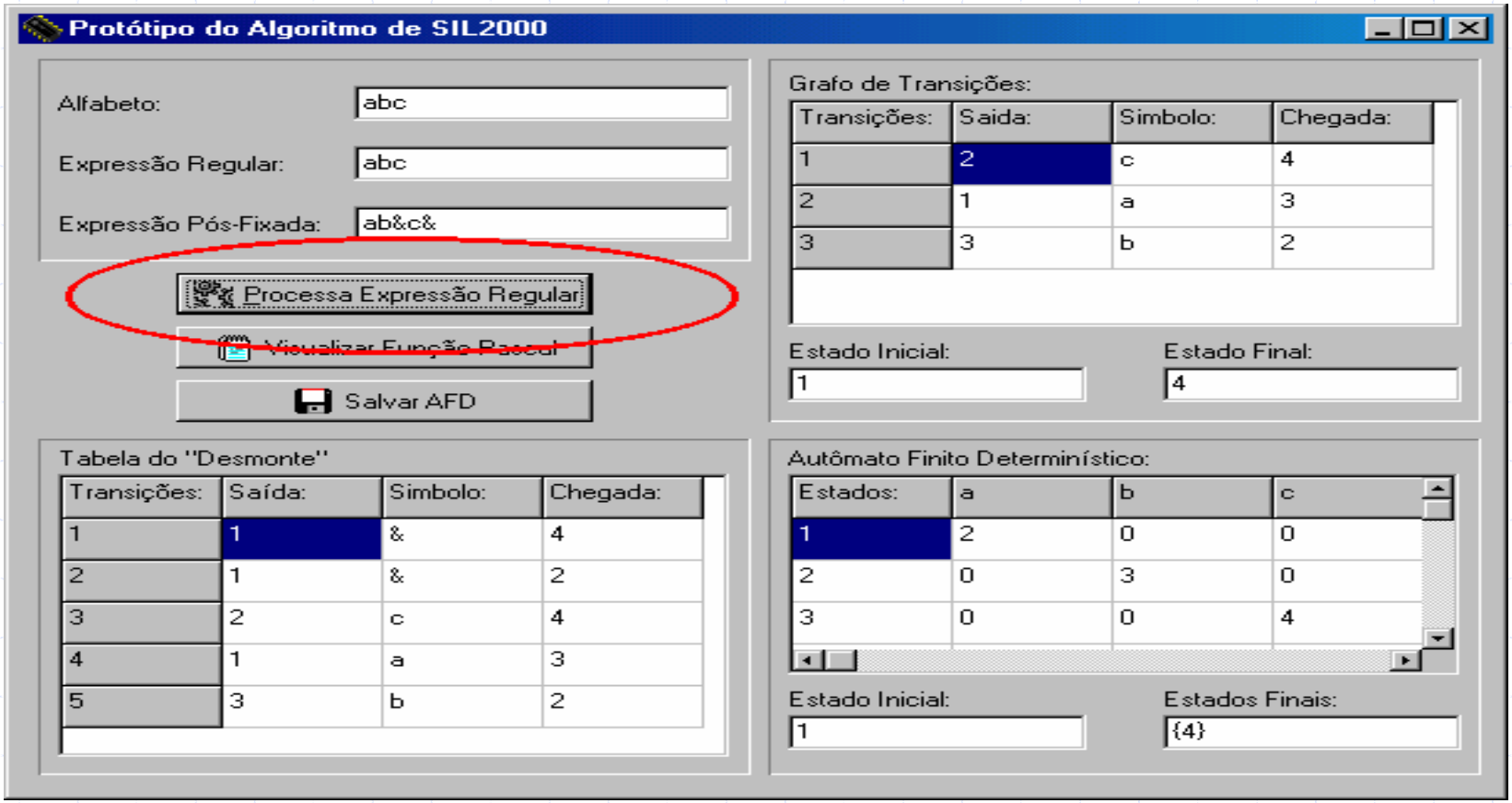


APRESENTAÇÃO DOS PROTÓTIPOS
PROCESSAR EXPRESSÃO

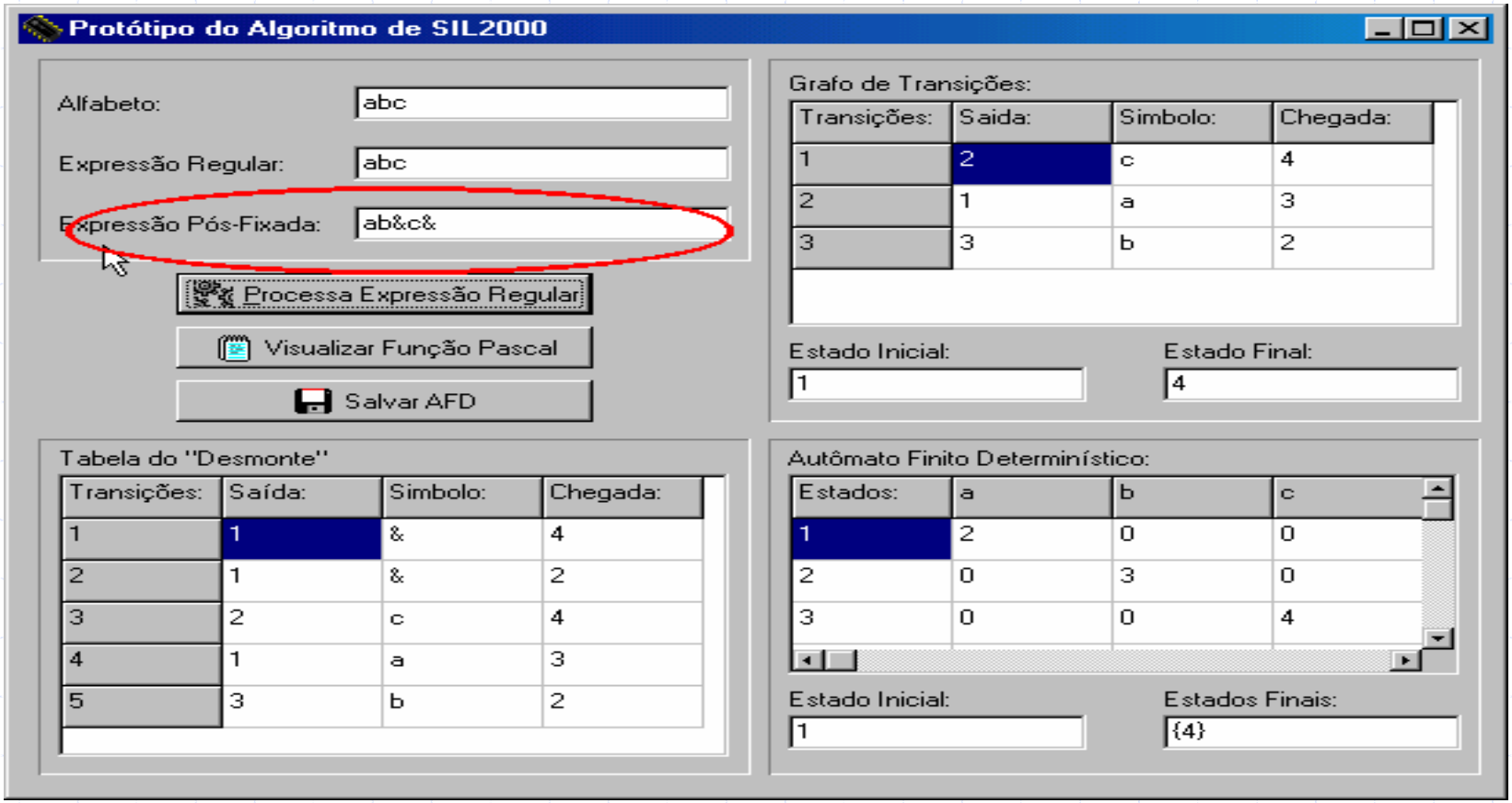
ROTEIRO

SAIR

ANTERIOR



APRESENTAÇÃO DOS PROTÓTIPOS
EXPRESSÃO PÓS-FIXADA



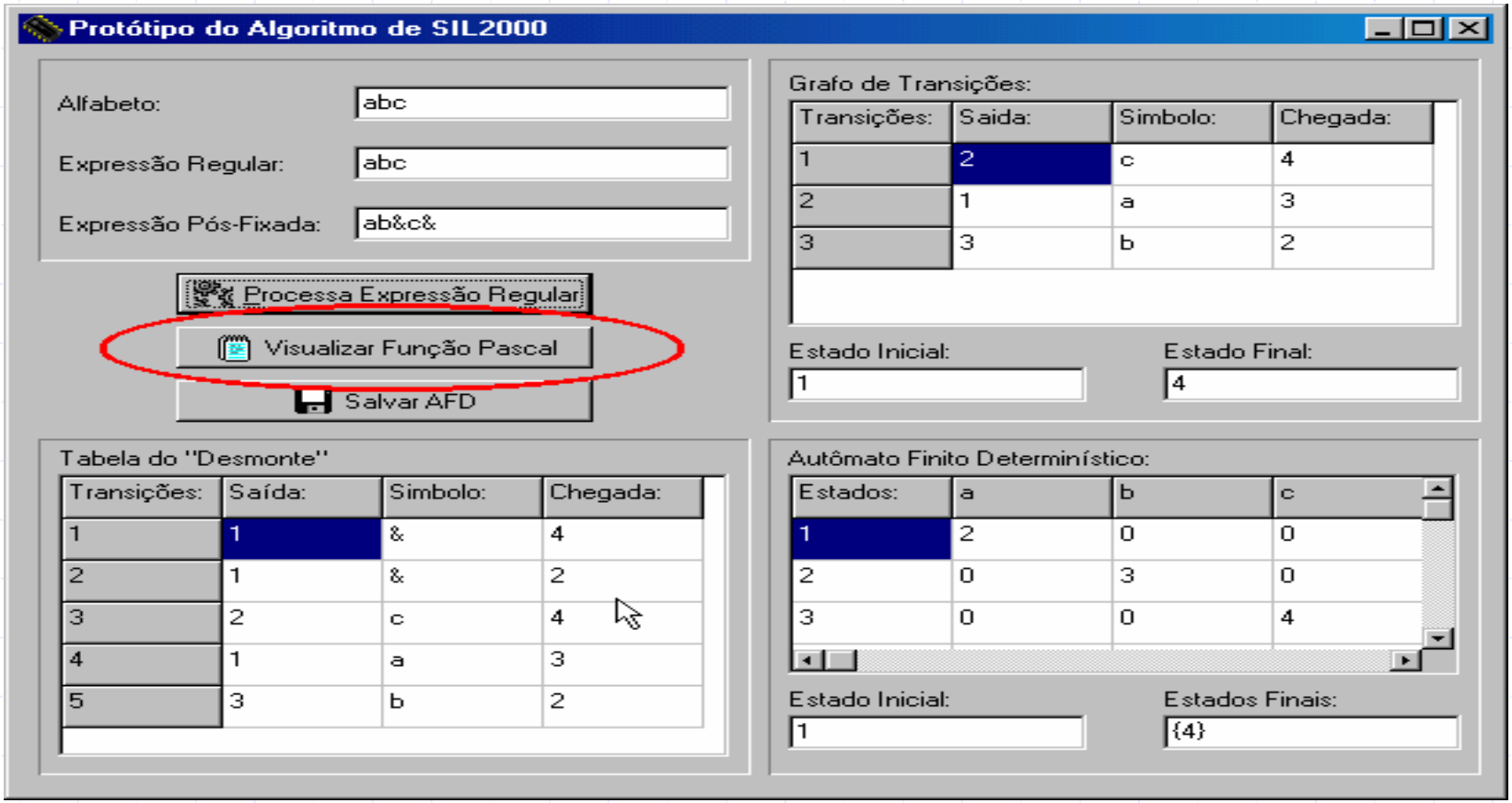
ROTEIRO
SAIR
ANTERIOR

APRESENTAÇÃO DOS PROTÓTIPOS
VISUALIZAR FUNÇÃO GERADA

ROTEIRO

SAIR

ANTERIOR

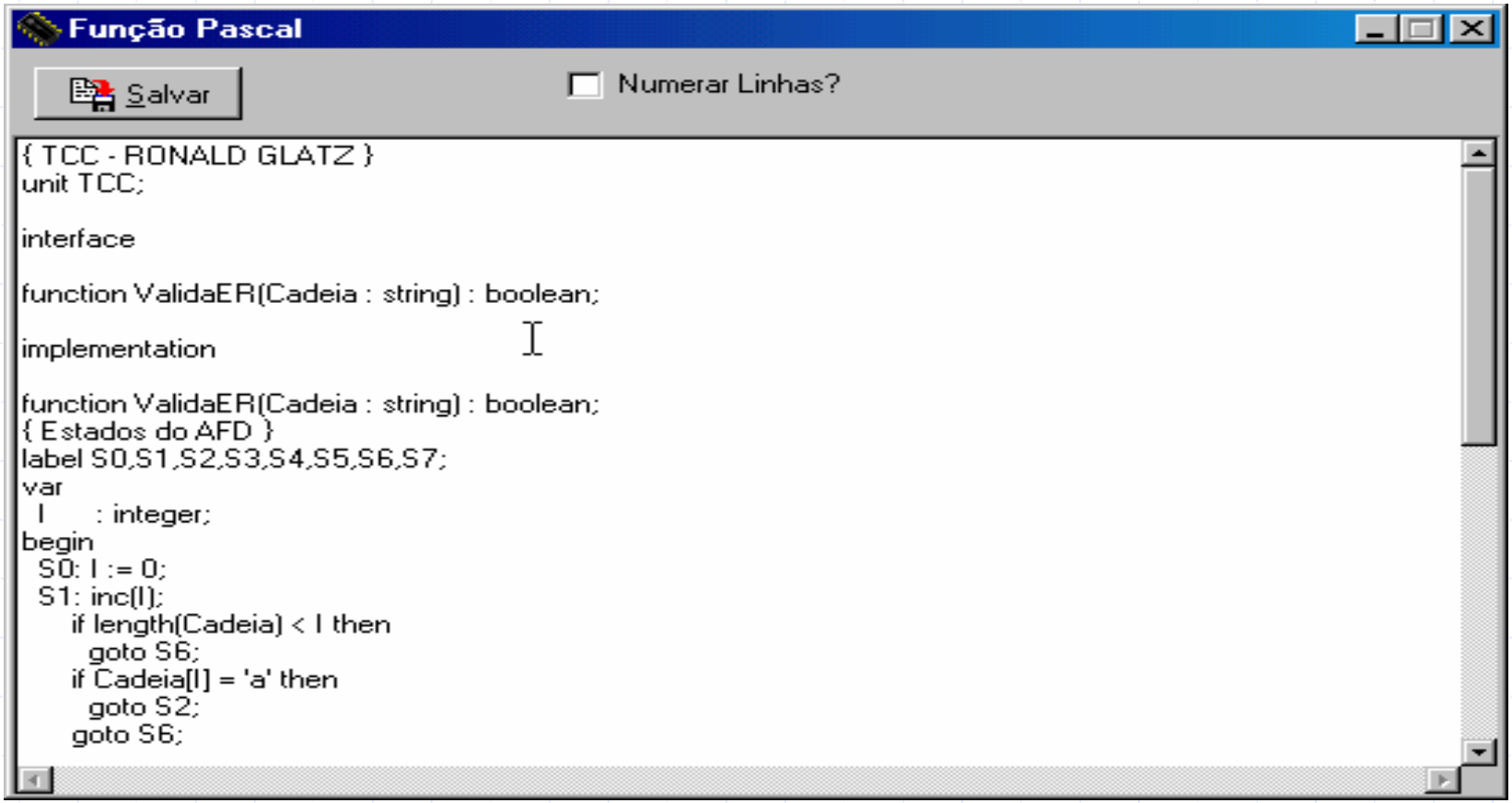


APRESENTAÇÃO DOS PROTÓTIPOS
FUNÇÃO PASCAL

ROTEIRO

SAIR

ANTERIOR

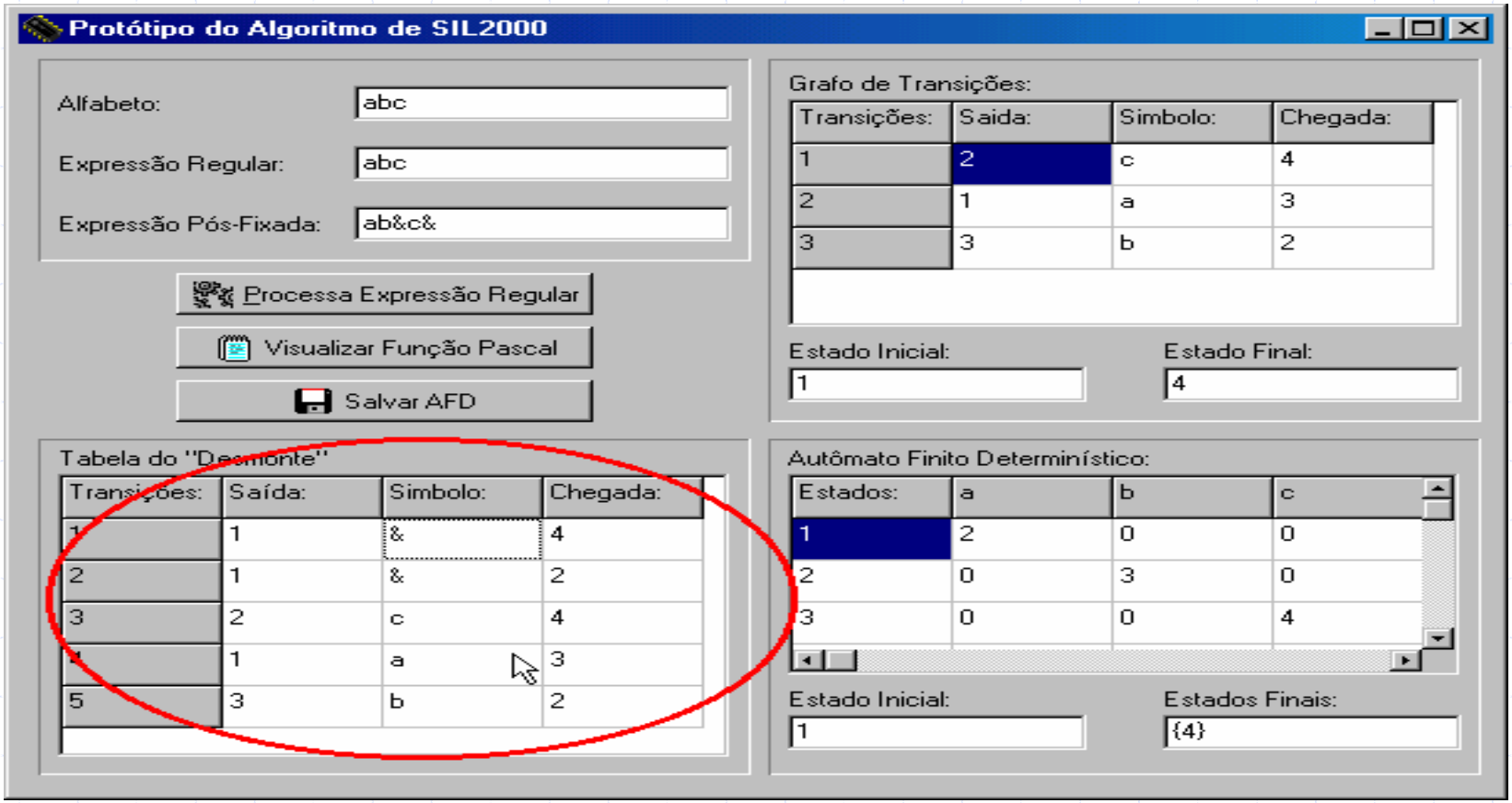


APRESENTAÇÃO DOS PROTÓTIPOS
TABELA DO DESMONTE

ROTEIRO

SAIR

ANTERIOR

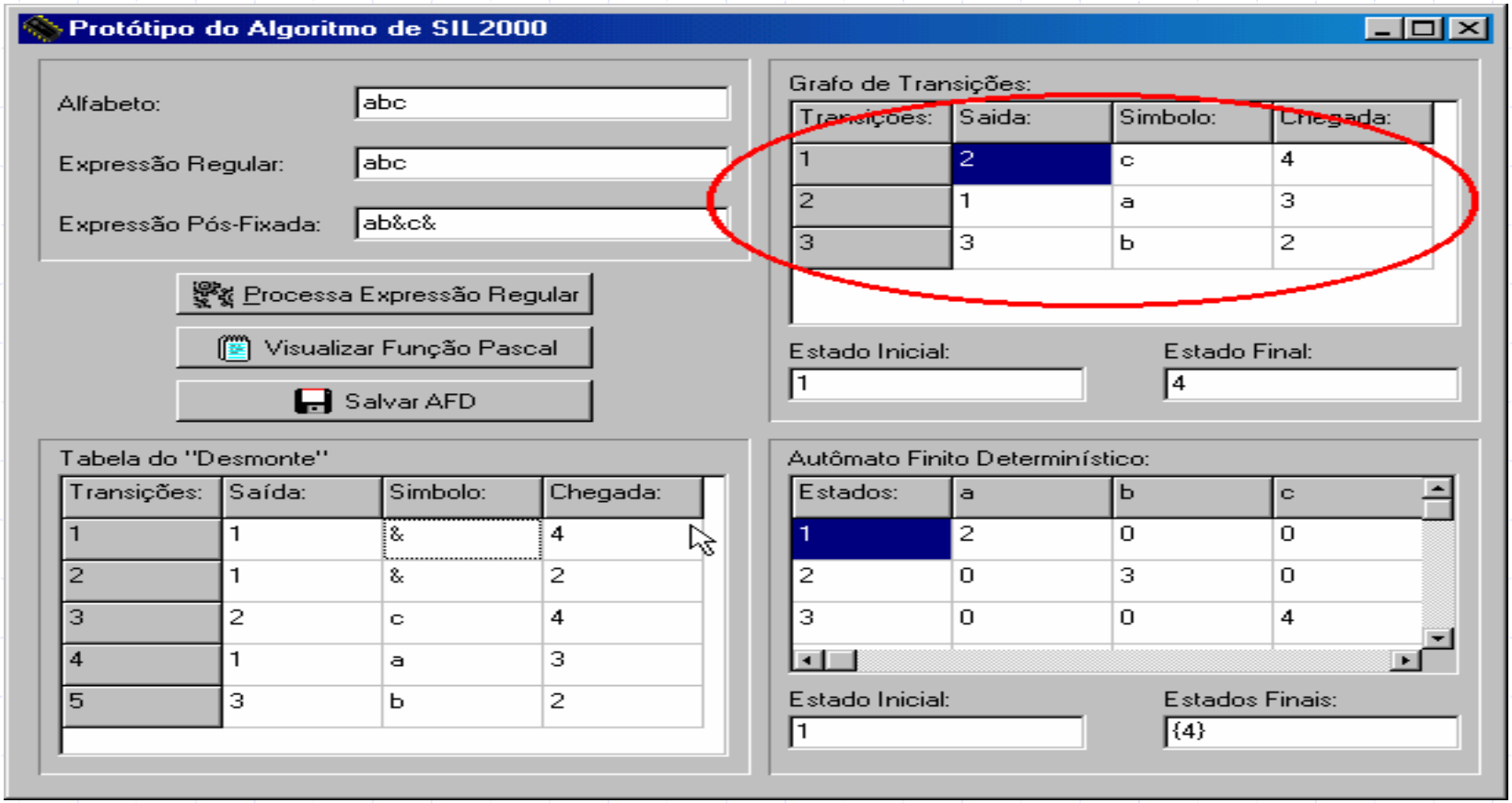


APRESENTAÇÃO DOS PROTÓTIPOS GT

ROTEIRO

SAIR

ANTERIOR

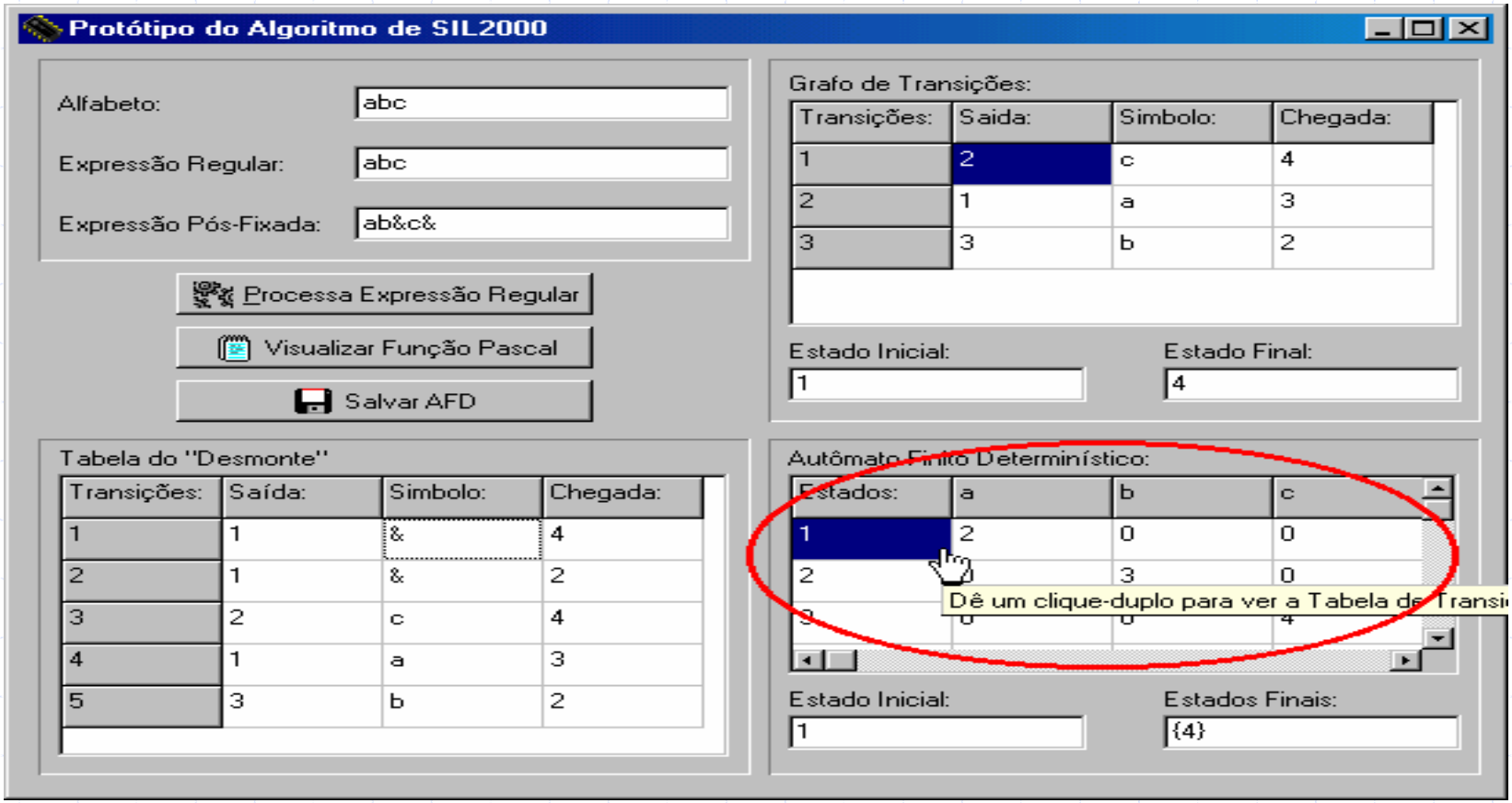


APRESENTAÇÃO DOS PROTÓTIPOS AFD

ROTEIRO

SAIR

ANTERIOR

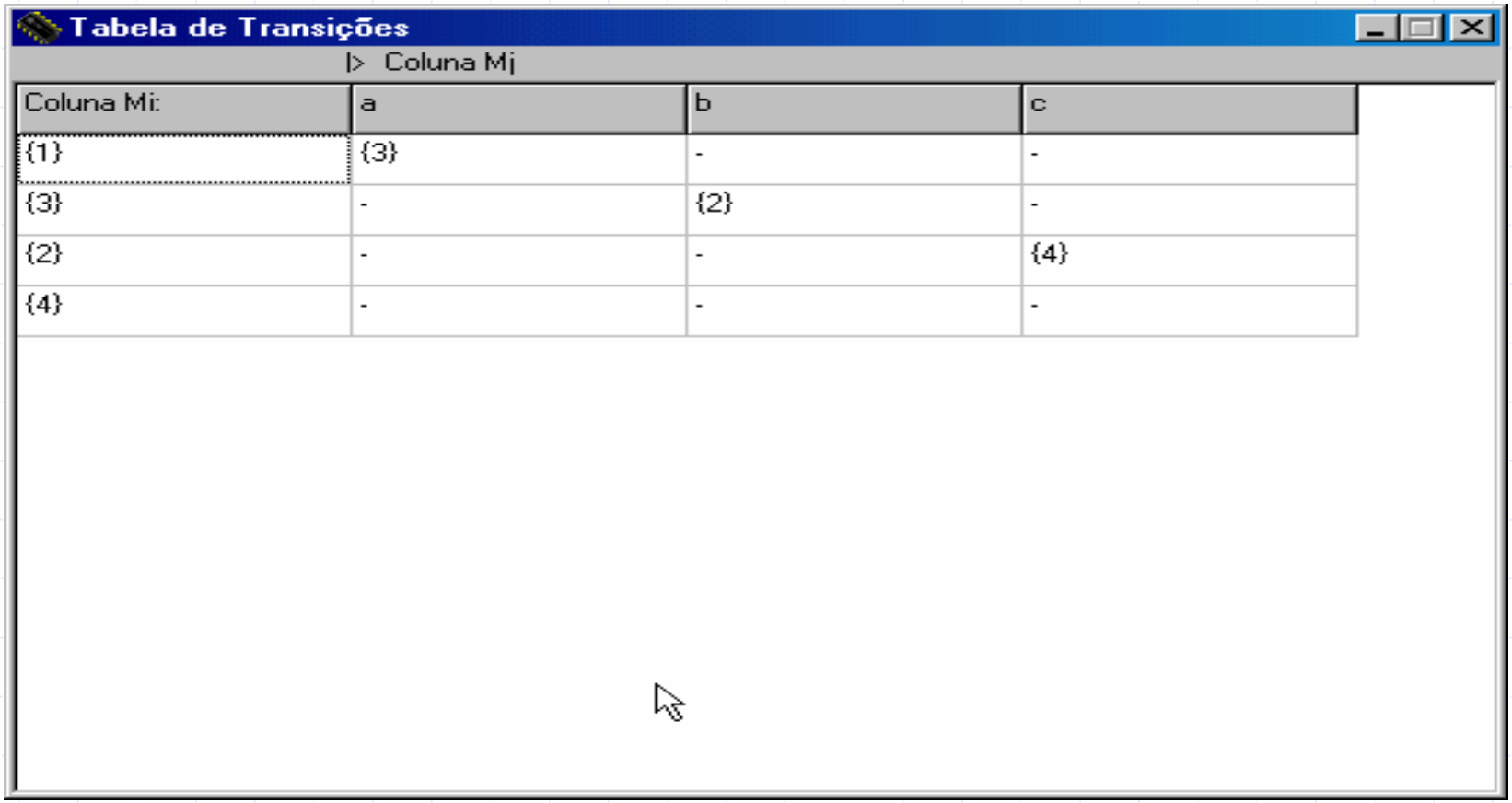


APRESENTAÇÃO DOS PROTÓTIPOS
TABELA DE TRANSIÇÕES DO AFD

ROTEIRO

SAIR

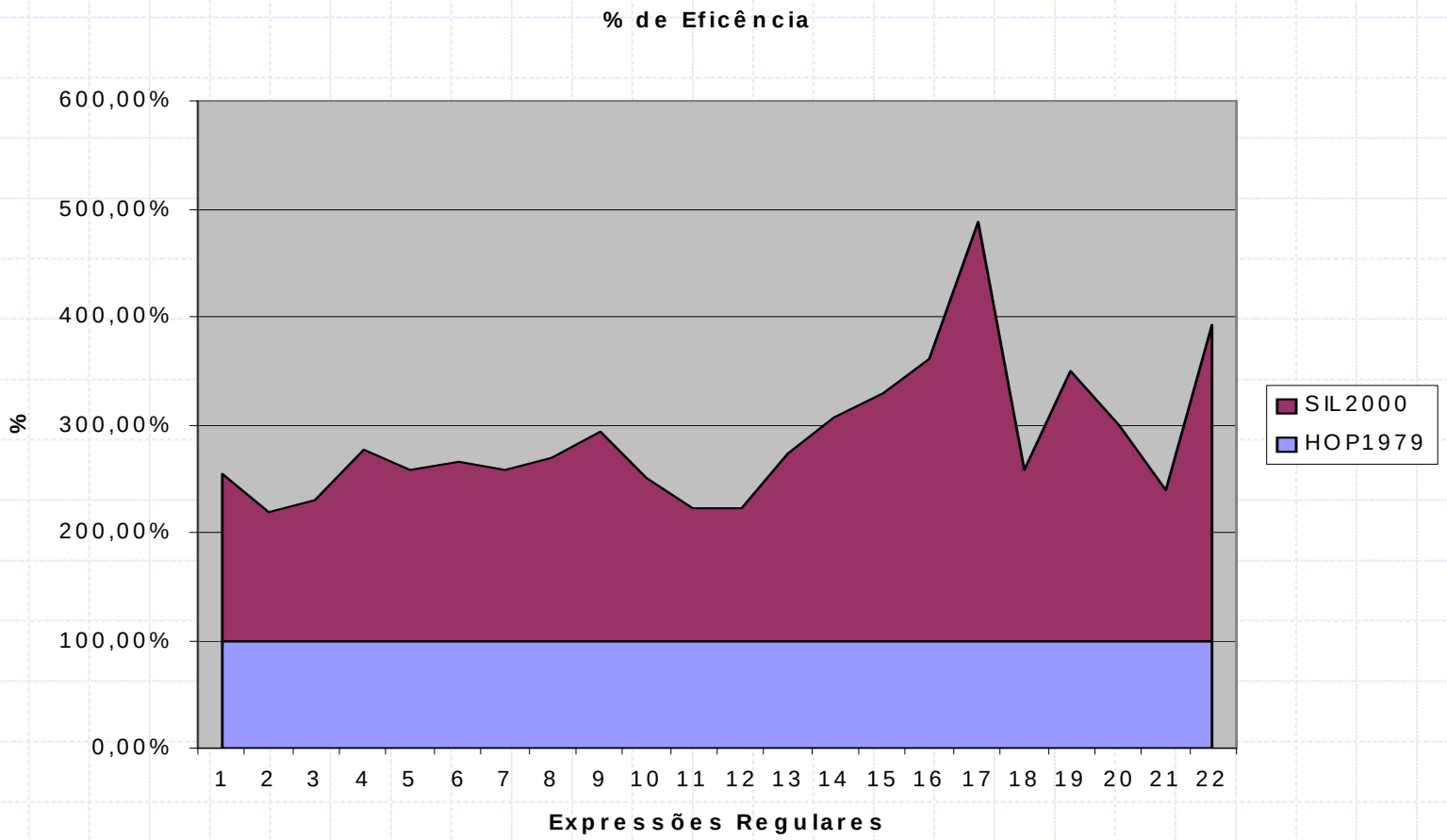
ANTERIOR



COMPARAÇÃO DOS ALGORITMOS

- CRITÉRIOS UTILIZADOS:
  - TEMPO DE EXECUÇÃO (% DE EFICIÊNCIA DE UM ALGORITMO EM RELAÇÃO AO OUTRO)
  - UTILIZAÇÃO DE MEMÓRIA (% DE EFICIÊNCIA DE UM ALGORITMO EM RELAÇÃO AO OUTRO)
  - OTIMIZAÇÃO DE RESULTADOS

COMPARAÇÃO DOS ALGORITMOS
TEMPO DE EXECUÇÃO

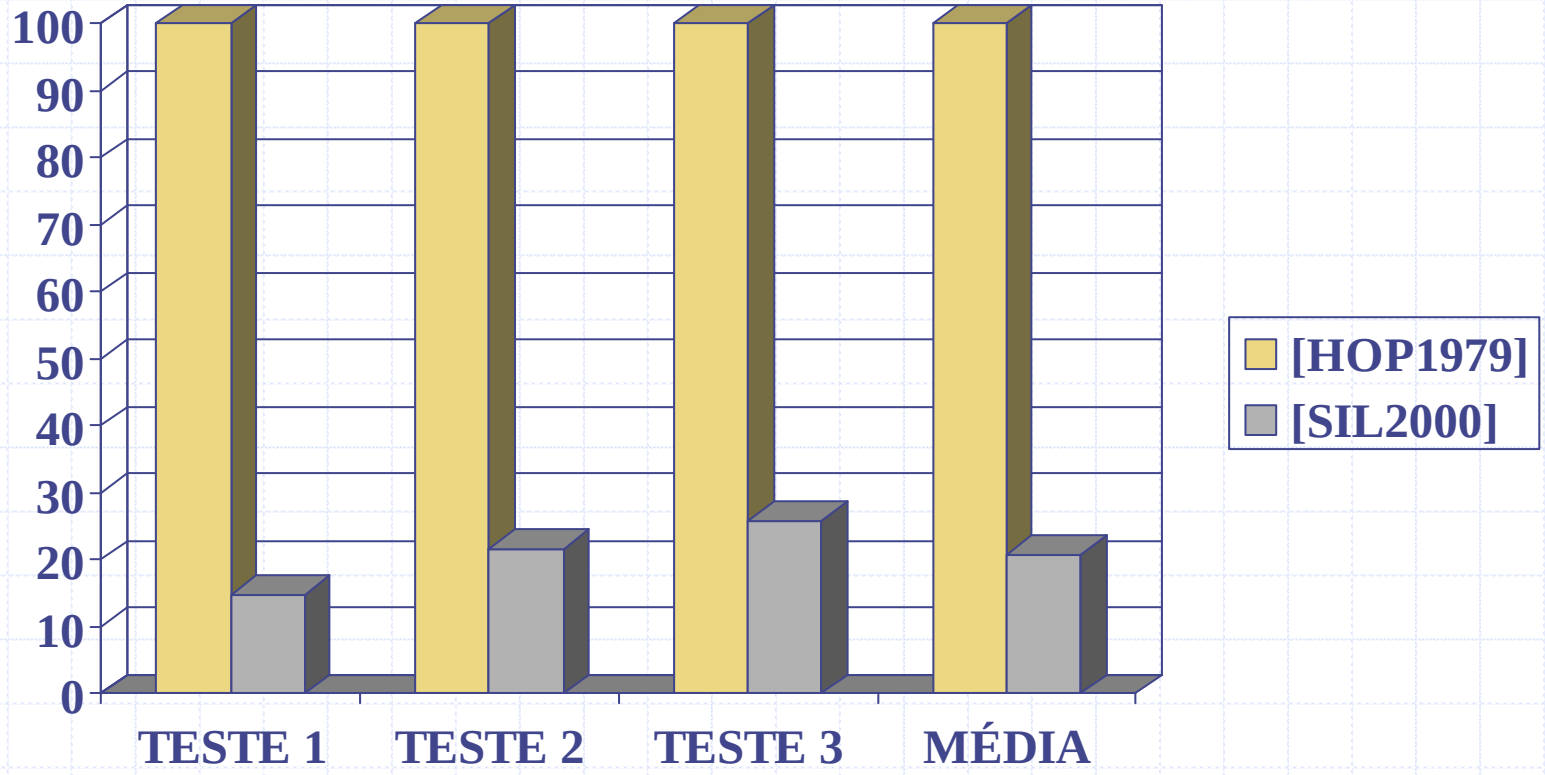


ROTEIRO

SAIR

ANTERIOR

COMPARAÇÃO DOS ALGORITMOS
UTILIZAÇÃO DE MEMÓRIA

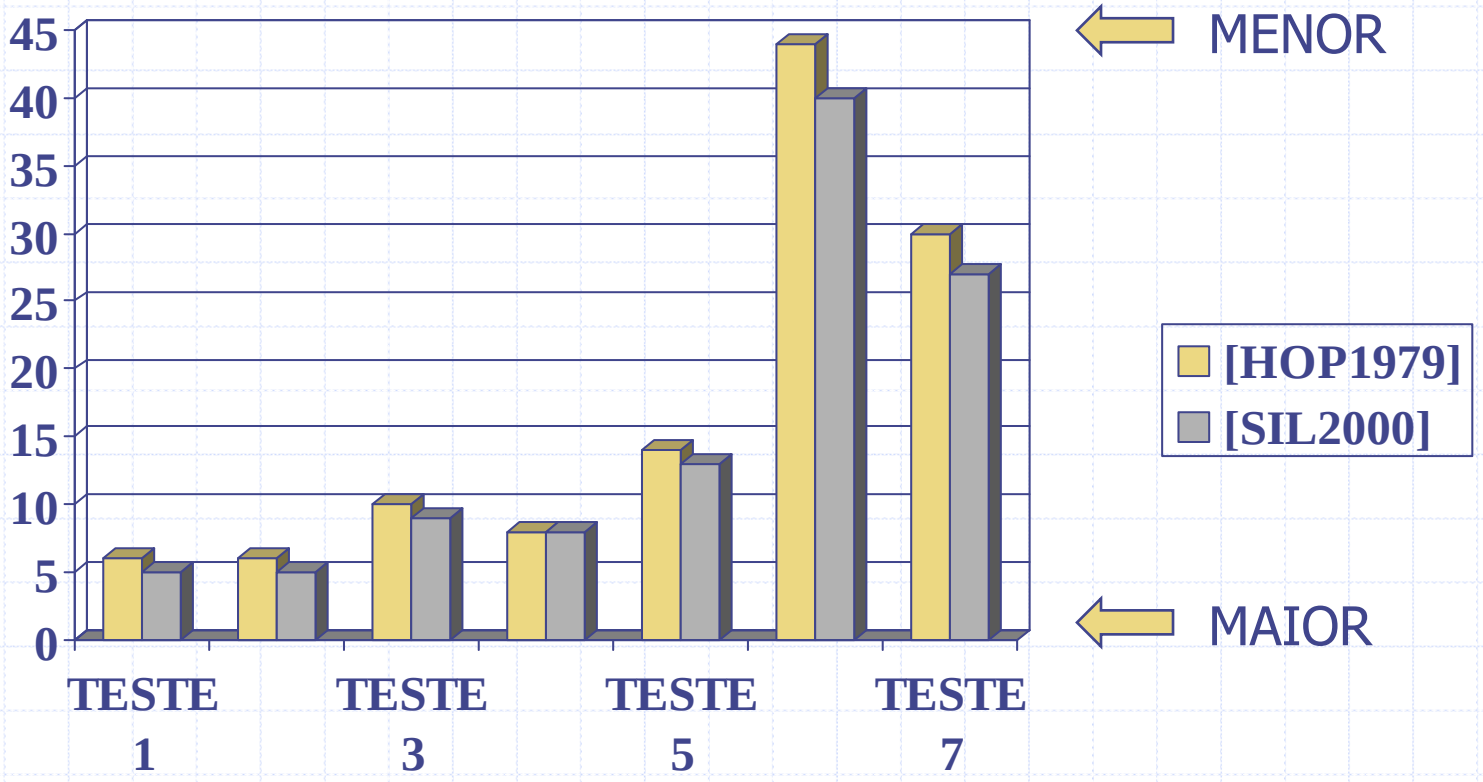


ROTEIRO

SAIR

ANTERIOR

COMPARAÇÃO DOS ALGORITMOS OTIMIZAÇÃO DE RESULTADOS



ROTEIRO

SAIR

ANTERIOR

COMPARAÇÃO DOS ALGORITMOS

- ✓ Para o critério TEMPO DE EXECUÇÃO, o algoritmo de [SIL2000] mostrou-se amplamente superior ao algoritmo de [HOP1979]
- ✓ Idem para o critério UTILIZAÇÃO DE MEMÓRIA
- ✓ No critério OTIMIZAÇÃO DE RESULTADOS foi utilizada a contagem dos estados do AFD resultante. Na quase totalidade dos testes o algoritmo de [SIL2000] se mostrou superior.

CONCLUSÃO

- ✓ Durante este trabalho foram vistos alguns conceitos e técnicas para transformação de uma ER em AFD
- ✓ Também foram vistos os algoritmos de [HOP1979] e [SIL2000]
- ✓ Foram feitas considerações sobre cada algoritmo e comparações entre eles