

Fundamentos Teóricos da Computação
Prof. Andrei Rimsa Álvares

Trabalho Prático

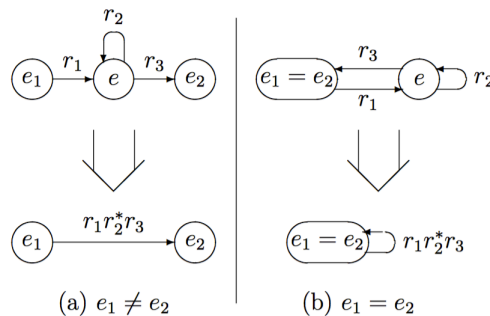
1. Objetivo

O objetivo desse trabalho é permitir que os alunos apliquem os conceitos assimilados na disciplina em um trabalho prático de implementação. A ideia é desenvolver um dos algoritmos em uma das máquinas/gramáticas vistas na disciplina em um programa de computador.

2. Descrição

Implementar um programa que recebe um Autômato Finito Não Determinístico com transições λ (AFN λ) e o converte em uma Expressão Regular (ER). Para obter uma expressão regular a partir de um AFN λ $M = (E, \Sigma, \delta, I, F)$:

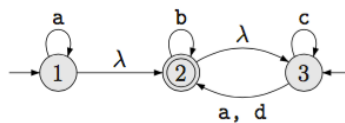
1. Obter um AFN λ $M' = (E \cup \{i, f\}, \Sigma, \delta', i, f)$ a partir do AFN λ M , onde $i, f \notin E$:
 - a. $\delta'(i, \lambda) = e, \forall e \in I$
 - b. $\delta'(e, \lambda) = f, \forall e \in F$
 - c. $\delta'(e, a) = \delta(e, a)$
2. Obter diagrama ER a partir de M' substituindo transições de e para e' sob s_1, s_2, \dots, s_n por uma só transição de e para e' sob $s_1 + s_2 + \dots + s_n$
3. Eliminar um a um os estados do diagrama ER com exceção de i e f :
 - a. Simular todas as passagens por e para cada par $[e_1, e_2]$ tais que há transição de e_2 para e e e para e_1 ($e_1 \neq e$ e $e_2 \neq e$)



- b. Se havia transição de e_1 para e_2 sob s , substituir por transição de e_1 para e_2 sob $s + r_1r_2^*r_3$

2.1. Exemplo

Considere o seguinte AFN λ :



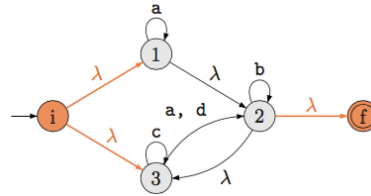
Seguindo o algoritmo listado acima para transformar o AFN λ em ER:



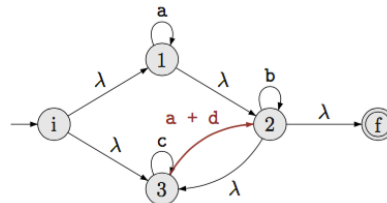
Fundamentos Teóricos da Computação

Prof. Andrei Rimsa Álvares

- Adicionar apenas um estado inicial e final, indicados em laranja:

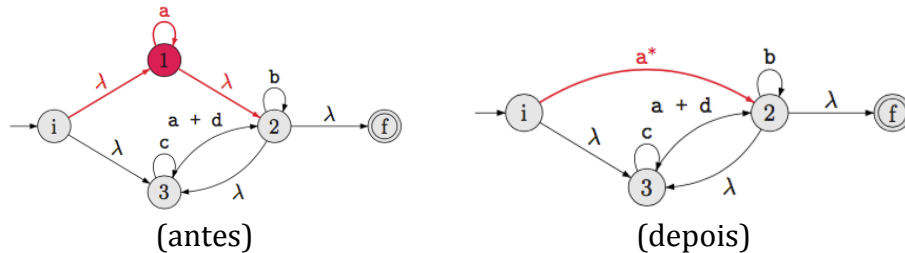


- Substituir todas as transições de ϵ para ϵ' por conjunções em ER, indicados em marrom:

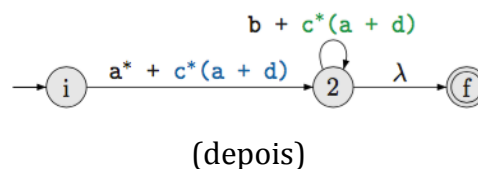
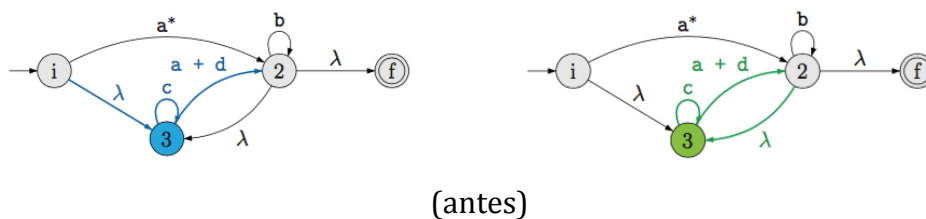


- Remover os estados na ordem 1, 3 e 2:

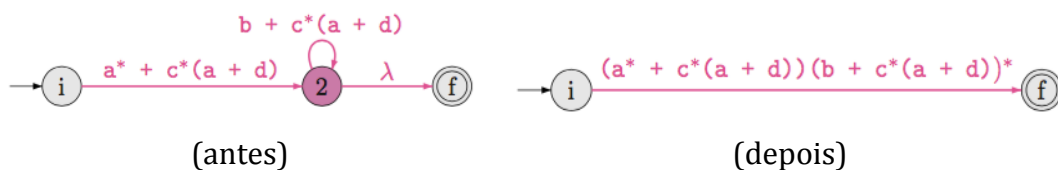
- Remover o estado 1, indicado em vermelho:



- Remover o estado 3 com suas transições, indicado pelas cores azul e verde:



- Remover o estado 2, indicado pela cor magenta:





Fundamentos Teóricos da Computação
Prof. Andrei Rimsa Álvares

3. Instruções

O programa deve receber uma especificação de um AFN λ $M = (E, \Sigma, \delta, I, F)$ e a lista de estados R na ordem que devem ser removidos no formato JSON conforme a seguinte especificação:

$\{ \text{"af": } [[e, \forall e \in E], [a, \forall a \in \Sigma], [[e, a, e'], \delta(e, a) = e'], [i, \forall i \in I], [f, \forall f \in F]],$
 $\text{"r": } [r, \forall r \in R] \}$

O exemplo da seção anterior com a remoção dos estados na ordem 1, 3 e 2 pode ser definida da seguinte forma, onde # indica o símbolo λ :

```
{ "af": [
  ["1", "2", "3"],
  ["a", "b", "c", "d"],
  [
    ["1", "a", "1"],
    ["1", "#", "2"],
    ["2", "b", "2"],
    ["2", "#", "3"],
    ["3", "c", "3"],
    ["3", "a", "2"],
    ["3", "d", "2"]
  ],
  ["1", "3"],
  ["2"]
],
  "r": ["1", "3", "2"]
}
```

teste.afn

Para uma determinada especificação com a ordem de eliminação dos estados só existe uma expressão regular válida se considerar as seguintes regras:

Tipo	Regras	Exemplos
-	Somente um símbolos ou λ se mantém	a λ
Conjunção	Termos em conjunção devem sempre ser envoltos por parênteses (externamente)	(a+b) (a+b+c)
	λ não deve ser ignorado em conjunções; a conjunção de múltiplos λ gera λ ; λ deve sempre ser o primeiro	(λ +a) (λ +b+c)
Composição	Termos em composição não devem ser envoltos por parênteses; λ deve ser ignorado, exceto se resultar no próprio λ	ab abc a(b+c)
Fecho de Kleene	Se for um símbolo: símbolo seguido de asterisco	a* b*
	Se for uma conjunção: conjunção (já entre parênteses) seguido de asterisco	(a+b)* (a+b+c)*
	Se for uma composição: composição entre parênteses seguido de asterisco	(ab)* (a(b+c))*

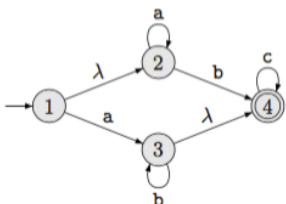
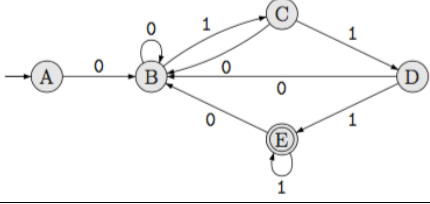
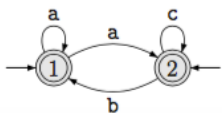
Fundamentos Teóricos da Computação

Prof. Andrei Rimsa Álvares

O programa deve receber o arquivo de entrada de acordo com a especificação em linha de comando e exibir a expressão regular na tela. Repare que a expressão regular não possui espaços.

```
$ ./af2er
Usar: verificador [AFN]
$ ./af2er teste.afn
(a*+c*(a+d))(b+c*(a+d))*
```

Mais alguns exemplos de AFN λ cuja a única expressão regular é dada conforme a seguir:

AFN λ	Ordem	Expressão Regular
	1, 2, 3 e 4	$(a^*b+ab^*)c^*$
	A, C, D, E e B	$0(0+10+110+1111^*0)^*1111^*$
	1 e 2	$(a^*+(\#+a^*a)(c+ba^*a)^*(\#+ba^*))$

4. Avaliação

O trabalho deve ser feito em grupo de até dois alunos, sendo esse limite superior estrito. O trabalho será avaliado em 10 pontos, onde essa nota será multiplicada por um fator entre 0.0 e 1.0 para compor a nota de cada aluno individualmente. Esse fator poderá estar condicionado a apresentações presenciais a critério do professor.

Trabalhos copiados, parcialmente ou integralmente, serão avaliados com nota **ZERO** do valor da prática, sem direito a contestação. Você é responsável pela segurança de seu código, não podendo alegar que outro grupo o utilizou sem o seu consentimento.

5. Submissão

O trabalho deverá ser submetido até as 23:55 do dia 25/05/2017 (terça-feira) via sistema acadêmico (Moodle) em pasta específica. Não serão aceitos, em hipótese alguma, trabalhos enviados por e-mail ou por qualquer outra fonte.