

UNIVERSIDADE FEDERAL DA BAHIA – UFBA
INSTITUTO DE MATEMÁTICA
DISCIPLINA – MATA 50
MONITOR: JANDSON SANTOS RIBEIRO
PROFESSORA: DÉBORA ABDALLA
SEMESTRE:2011.1

1º trabalho prático

1. Descrição Geral :

A classe das linguagens regulares é fechada sobre a operação de união. O trabalho consiste na implementação de um programa que realiza a união sobre duas linguagens regulares e faz o teste de pertinência de um conjunto de cadeias sobre a nova linguagem gerada.

A entrada das linguagens regulares será feita pela leitura de dois autômatos que as reconhecem.

Os formatos de entrada e saída são descritos a seguir e devem ser seguidos.

O programa deve ler da entrada padrão e escrever na saída padrão.

2. Formatos de Entrada:

A entrada é composta por um conjunto de linhas contendo as informações do Autômato de cada linguagem e das cadeias.

Inicialmente o programa deve ler os dois autômatos referentes a cada linguagem. Logo após a leitura do primeiro autômato inicia-se, na próxima linha, a leitura do outro autômato.

2.1 – Representação do Automato

1. A primeira linha deve conter a assinatura do autômato:

$$A = \{\{q_1, q_2, q_3, q_4\}, \{a, b, c, d\}, f, q_1, \{q_3, q_4\}\}$$

Um identificador qualquer representando o autômato, podendo o mesmo ocorrer entre os símbolos dos estados, alfabeto, identificador da função de transição, estado inicial ou estados finais, seguido de espaço seguido do símbolo '=', seguido de espaço seguido de '{', seguido dos estados do autômato, seguido de '}', seguido de

virgula ',', seguido '{', seguido dos símbolos do alfabeto, seguido de '}', seguido de virgula ',', seguido de '{', seguido de um símbolo que representa a função transição, seguido de virgula ',', seguido de '{', seguido dos estados finais, seguido de '}'.

2. A segunda linha deve ter o seguinte formato : $f = \{$
O símbolo identificador da função transição, seguido de espaço, seguido de '=',
, seguido de espaço, seguido de '{'.
A segunda linha deve conter somente essas informações e as próximas linhas
devem conter transições do autômato .

3. Transições do autômato : $(q_1, a) = \{q_2\}$

'(' seguido de um estado seguido de virgula ',', seguido de um símbolo do
alfabeto, seguido de ')', seguido de '=', seguido de '{', seguidos dos estados
para aquela transição , seguido de '}'.

4. Ultima linha das transições : };
Esta linha sinaliza o fim das transições e do autômato.

'}' seguido de ';'.

- Após a leitura do primeiro autômato o mesmo padrão se repete para a leitura do
segundo autômato. Após a leitura dos dois, inicia-se na próxima linha a leitura das
cadeias que devem ser testadas.

2.2 Cadeias : Cada linha contem uma cadeia w.

Ex:

aaab

bbaa

3. Formatos de saída

A saída é descrita por um conjunto de linhas. Primeiro o autômato obtido da união dos
dois primeiros, depois as cadeias reconhecidas.

3.1 Autômato:

- 3.1.1 Deve-se seguir o mesmo padrão do conjunto de entrada para
autômato, na saída. A assinatura do autômato segue o mesmo padrão
da entrada ,onde dois símbolos passam a ser obrigatórios : M e q0.
Para mais detalhes veja o tópico palavras reservadas para a saída.
- 3.1.2 As transições dos autômatos devem ser listadas em uma determinada
ordem explicada na sessão Transições e estados do autômato gerado.
O nome dos estados do autômato gerado deve seguir a mesma

nomenclatura utilizada no algoritmo indicado [1]. Para mais informações veja o tópico Transições e estados do autômato gerado.

3.2 Cadeias:

Cada linha contem a cadeia que foi lida da entrada seguindo o padrão abaixo:

W = true

Cadeia w seguido de espaço, seguido de '=', seguido de true. Essa linha deve ser impressa se a cadeia w pertence a união das duas linguagens.

W = false

Cadeia w seguido de espaço, seguido de '=' seguido de false. Essa linha deve ser impressa se a cadeia w não pertence a união das duas linguagens.

As cadeias devem ser impressas na mesma ordem que foram lidas.

Palavras reservadas para a saída:

O símbolo inicial deverá ser a concatenação do q1 de M1 w q2 de M2, ou seja, q1q2 que não será usado em nenhuma entrada como símbolo de estado de um autômato. Entretanto tal símbolo é o estado inicial do autômato M gerado pela união dos autômatos lidos. Assim toda saída terá q1q2 no conjunto de estados e o mesmo não aparece na entrada como estado.

O identificador do autômato gerado será obrigatoriamente M.

Ex:

Assinatura de autômato válida na saída :

$M = \{\{q_0, q_2, d, f, h\}, \{a, b, c\}, P, q_0, \{q_2\}\}$

Assinaturas erradas:

$L = \{\{q_2, d, f, h\}, \{a, b, c\}, P, q_0, \{q_2\}\}$

O identificador deve ser obrigatoriamente M.

$M = \{\{q_2, q_3, d, f, h\}, \{a, b, c\}, P, q_2, \{q_2\}\}$

O símbolo q0 não foi utilizado como o símbolo inicial do autômato.

Transições e estados do autômato gerado

1. Nomenclatura para estados gerados pelo algoritmo [1].

Uma das etapas do algoritmo consiste em fazer combinações dos estados de M1 com M2. Seja r1 um estado de M1 e r2 um estado de M2. (r1,r2) é um estado de M e o seu nome deverá ser exatamente r1r2, ou seja, a concatenação do estado r1 com o estado r2. Para padronizar a saída, o estado r2r1 não poderá existir, pois

iremos admitir que a concatenação segue a ordem do nome do estado do M1 com o estado de M2, onde M1 é o primeiro autômato da entrada e M2 o segundo autômato.

Exemplo de transição válida:

$(r1r2,a)=\{q2q3\}$

2. Ordem das transições

Na assinatura do M1 (primeiro autômato da entrada) existe um conjunto de estados $\{q1,q2,...,qn\}$, nessa ordem. Observe que os símbolos podem ser outros como $\{a,b,c\}$. O importante aqui é a ordem em que eles aparecem no conjunto. Assim ao criar o autômato pelo algoritmo indicado [1], a função transição é criada e a mesma deve ter suas transições escritas na saída como determinado. Aqui iremos impor uma restrição na ordem da saída. Assim primeiro deve aparecer todas as transições do estado $(q0,x)$, depois todas as transições de $(q1,x)$ e assim até (qn,x) , onde x é um símbolo qualquer do alfabeto. A ordem em que os símbolos do alfabeto aparecem também deve ser seguido assim se o alfabeto é $\{a,b\}$, a saída deve ser :

$(q0,a)=(qr1qr2)$

$(q0,b)=(qr1qr2)$

A ordem deve ser seguida e faz parte do padrão de saída. Trabalhos que não sigam o padrão não serão considerados.

IMPORTANTE:

Para a implementação do trabalho o algoritmo utilizado deve ser obrigatoriamente o que se encontra no livro do SIPSER[1]. Trabalhos que não usem este algoritmo não serão corrigidos.

A notação para gerar as transições e estados deve ser a indicada no tópico Transições e estados do autômato gerado.

Linguagens aceitas: C/C++ e Java

Compiladores:

C: gcc

C++:G++

Java : Compilador Java do JDK

Trabalhos implementados em outras linguagens não serão aceitas. O trabalho não será compilado em outros compiladores. Erros de compilação serão

considerados erros do trabalho. Trabalhos que não lerem da entrada padrão e escreverem na saída padrão não serão corrigidos. Trabalhos que não seguirem o padrão de entrada e de saída não serão considerados.

Obs: gcc e g++ fazem parte do GNU compiler. Aconselhavel que pesquisem sobre o GNU compiler se tiverem quaisquer duvida sobre os compiladores citados para C/C++.

[1] SIPSER, Michael. **Introdução à teoria da computação**. 2.ed. São Paulo, SP: Thomson, 2007. Algoritmo da sessão Operações regulares em Fechamento sobre a união de linguagem regulares.