

# SCE 0722 – Métodos e técnicas de análise e projeto de sistemas reativos

## Trabalho 1 – para ser feito em trios

Escreva um programa que leia a descrição de uma ou mais MEFs de um arquivo e que dê algumas informações sobre essas MEF's:

- diga se é determinística
- diga se é fortemente conectada
- diga se é completamente especificada
- diga se é minimal e, se não for, apresente a MEF minimal correspondente

O formato do arquivo é exemplificado a seguir:

```
MEF "exemplo 1" // Esse é o primeiro exemplo
Entradas: a b c
Saídas: 0 1
Estados: A B C
Transicoes: A -- a / 0 --> B // Comportamento na forma de trans.
A -- b / 0 --> A
A -- c / 1 --> C
B -- a / 1--> A
B -- b / 1 --> C
B -- c / 1 --> C
C -- a / 0--> A
C -- b / 0 --> B
C -- c / 1 --> B
```

```
MEF "exemplo 2" // Esse é o segundo exemplo
entradas: a0 b8 00h
saidas: falso positivo talvez
estados: S1 S2
tabela: //Comportamento na forma de tabela
S1      falso falso talvez ; S1 S2 S1 \\
S2      talvez positivo falso; S2 S1 S1 \\
```

Algumas observações:

- as entradas, as saídas e os nomes de estados podem ser qualquer sequência de dígitos e letras apenas como: a, ab, ab5t, 12by78, 128.
- o comportamento pode ser dado na forma de transições ou de tabela, como no exemplo acima. Um mesmo arquivo pode misturar as duas formas.
- em ambos os casos, a ordem em que cada transição ou cada linha da tabela aparece é irrelevante.
- no caso da tabela, as colunas seguem a ordem em que as entradas foram definidas
- as palavras chaves como MEF, entradas, saidas, etc não são sensíveis à caixa. Já os nomes de entradas, saídas e estados são. Porém não permita que nomes similares como ABC, abc ou aBc apareçam numa mesma categoria (saídas, entradas, estados).
- a cadeia "\\" (duas barras invertidas) deve ser usada para terminar cada linha da tabela de transição.
- use // para prover comentários de uma linha, estilo Java.

- Para poder definir uma MEF não completamente especificada usando a notação de tabela, use um “-” (menos) na entrada da tabela que não for definida. Por exemplo:

```
MEF "exemplo 2" // Esse é o segundo exemplo
entradas: a0 b8 00h
saídas: falso positivo talvez
estados: S1 S2
tabela: //Comportamento na forma de tabela
S1      falso - talvez ; S1 - S1 \\
S2      talvez positivo falso; S2 S1 S1 \\
```

Para facilitar a tarefa de ler o arquivo, foi disponibilizado um arquivo que pode ser processado pelo programa JavaCC para gerar um parser que lê as descrições das MEFS como descrito a seguir.

Como segunda parte deste trabalho, o programa deverá, para cada MEF lida em um arquivo, gerar as seqüências básicas estudadas em classe.  
Ou seja:

- para cada estado, qual seria uma seqüência de sincronização;
- para a MEF, qual seria uma seqüência de distinção;
- para cada estado, qual seria uma seqüência única de entrada/saída;
- para cada uma das seqüências geradas, o programa deve, também, exibir a árvore que foi criada para sua geração.

Procure sempre determinar as menores seqüências possíveis.

Preste também atenção que em alguns casos, as propriedades determinadas na primeira fase do exercício devem ser levadas em consideração. Por exemplo, nenhuma dessas seqüências faz sentido numa MEF não determinística. Ou seja, se a MEF for não determinística o aluno não deve calcular as seqüências. A resposta seria algo como “seqüência de sincronização não pode ser calculada porque MEF é não determinística”.

Se a MEF dada não for minimal, calcule a MEF minimal e use essa MEF minimal para gerar as seqüências.