

**PMR-5237 Modelagem e Design de Sistemas Discretos
em Redes de Petri**

Prof. Dr. José Reinaldo Silva

Lista de exercícios 1 (entrega : até quarta-feira, 16 de março até 24:00)

Exercício 1: Encontrar a representação em linguagem regular do seguinte TG,

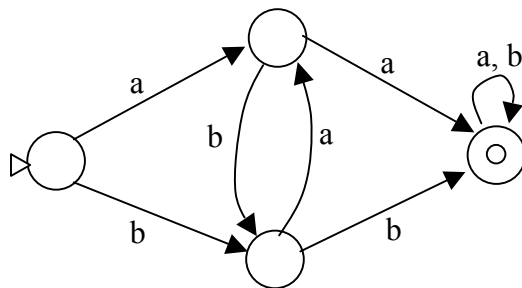


Figura 1.

Exercício 2: Encontre os TG's das seguintes expressões regulares,

- a) $(a + b)^2 b (a + b)^*$
- b) $(a + ba^* ba^*b)^+$

Exercício 3: Os exemplos acima compõem dos autômatos determinísticos, finitos. Um problema ligeiramente diferente seria o da linguagem definida por

$$a^n b^n$$

para um dado n inteiro. O problema neste caso seria contar a quantidade de a 's para garantir que teremos a mesma quantidade de b 's. Nenhum TG se aplica a este caso sem a inclusão artificial de contadores. Procure referências sobre os autômatos de pilha nos livros de Teoria de Autômatos ou de Teoria de Computação. Apresente uma solução para este problema.

Um palíndromo pode ser definido como uma palavra que pode ser lida da mesma forma no sentido normal e de trás pra frente. Por exemplo *abaabbaaba*.

Exercício 4 : Mostre que,

- a) a string vazia λ é um palíndromo
- b) se a é um símbolo, então a é um palíndromo
- c) se a é um símbolo e x é um palíndromo, então “ axa ” é um palíndromo
- d) nenhuma palavra que não satisfaça os itens anteriores pode ser um palíndromo
- e) faça um autômato finito que processa um palíndromo sobre o conjunto de símbolos $\{a, b, c\}$. Note que este NÃO é um autômato determinístico mas sim um autômato dependente de contexto. Portanto um DFA não é suficiente para fazer o parser.

Exercício 5: Faça o download do software PIPE 2 no site do Petri Nets World (veja transparências de aula) e use o editor de redes de Petri para modelar o “Winter Train Problem” discutido em sala. Este exercício é importante (embora puramente operacional) para se familiarizar com o PIPE 2 e com editores de redes de Petri em geral.

Exercício 6: Suponha que depois de fazer a modelagem e a simulação (jogo de marcas) em Redes de Petri, você deva fazer uma apresentação dos resultados para o chefe da companhia e para os supervisores garantindo que o sistema automatizado funcionará “sem erros”. Qual seria o argumento básico (mas intuitivo)?

Exercício 7: Se ainda não tiver com o modelo completo (sem “gates”), complete o modelo, e com ele,

- a) Identifique na matriz de incidência, o conflito e a forma como é “resolvido”, isto é, o sistema não tem nenhum estado cujo sucessor não esteja plenamente definido (pelo estado do gate G).
- b) Explique o funcionamento do controle do trem. Note que este poderia ser totalmente automatizado.

Exercício 8: Escreva a equação de estado para este problema e veja que é possível usar esta equação para calcular sequencialmente os próximos estados do problema dos trens. Claro que para isso em cada estado é preciso determinar o vetor de habilitação (a única incógnita da equação além do estado final).

- a) mostre que este vetor de habilitação é inconsistente se insistirmos em representar fielmente uma situação de conflito. Por exemplo, na figura 2 abaixo está representado um mutex clássico (o semáforo). A situação que chamamos conflito é definida pela marcação mostrada na figura, onde as transições t_1 e t_2 estão habilitadas em um mesmo estado, mas não são independentes (veja transparências da Aula1).

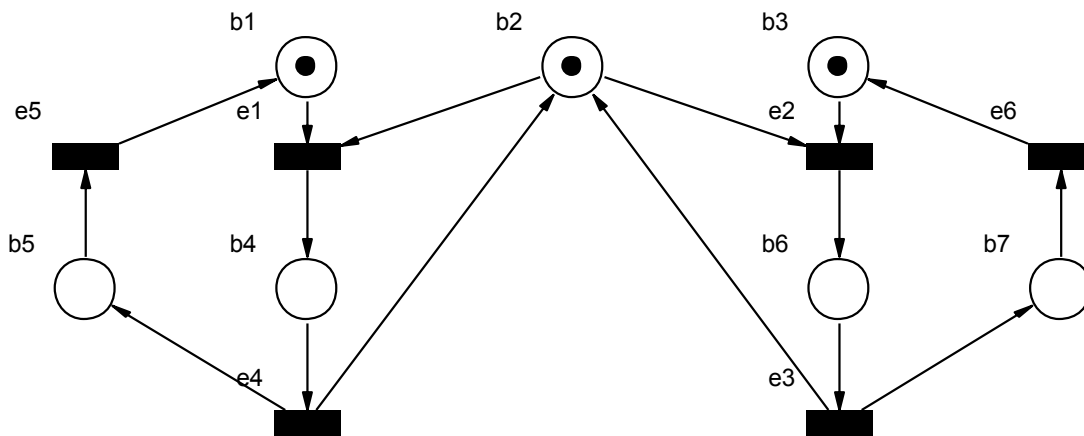


Fig. 2: Semáforo clássico

Se usarmos o vetor de habilitação

$$\sigma = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

é possível que se chegue a uma marcação inconsistente, isto é, onde existem uma marcação negativa em um dos lugares (o que significaria isso fisicamente?).

- b) Temos portanto que fazer uma escolha em casos de conflito (ou inserir esta escolha na modelagem. Portanto a implementação não pode ser esta da figura 2. Como seria? Faça as mudanças necessárias pra tornar a Fig. 2 em algo que possa ser implementado.
- c) Outra coisa importante é que costumamos acreditar nos software que “simulam” ou jogam marcas e enfrentam situações como esta. Como estes softwares não podem analisar caso a caso, uma solução seria que resolvessem a equação de estado. Para isso precisariam em cada passo calcular (especialmente para redes grandes) o vetor de habilitação (as limitações seriam menores, embora ainda existam). Proponha um algoritmo pra isso, e lembre que é preciso identificar os conflitos e “escolher” uma transição para ser disparada.