ECM253 – Linguagens Formais, Autômatos e Compiladores

Lista de Exercícios

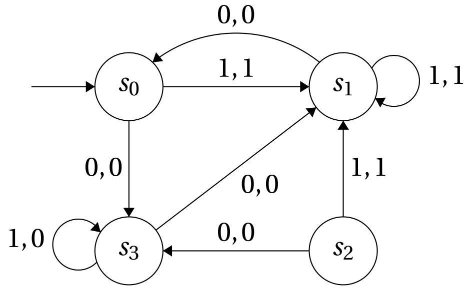
Máquinas de Estados Finitos e Autômatos Finitos

Marco Furlan

15 de maio de 2019

1 Máquinas de estados finitos

1.1. A partir da máquina de estados finitos a seguir, pede-se:



(a) Elaborar uma tabela de transição de estados para a máquina.

(b) Simular a máquina com a entrada `10001`.

1.2. Construir uma máquina de estados finitos tipo Mealy que permita a soma de dois números binários. Os números deverão ser informados como um par na entrada da máquina. Por exemplo, considerar os números $x =$ `101011` e $y =$ `000110`, sabe-se que sua soma deverá ser (os “vai-um” estão indicados na figura a seguir):

$$
\begin{array}{rl}
 & \texttt{111} \\
 & \texttt{101011}\ (x) \\
+ & \texttt{000110}\ (y) \\
\hline
 & \texttt{110001}
\end{array}
$$

Alimentar a máquina com pares de bits do tipo $x_k y_k$, para $0 \leq k \leq n-1$, onde $n > 0$ é o tamanho da cadeia que representa o número binário. O par $x_0 y_0$ representa os bits mais à direita dos números e o par $x_{n-1} y_{n-1}$ representa os bits mais à esquerda dos números. No exemplo apresentado a entrada deveria ser: `101101100010` ($x_0 y_0 x_1 y_1 x_2 y_2 x_3 y_3 x_4 y_4 x_5 y_5$).

Sugestão: a partir de um estado inicial e de acordo com os bits do par, transitar para um estado que represente “vai-um” igual a um ou para um estado que represente “vai-um” igual a zero. De acordo com os bits da entrada é produzida uma saída que, dependendo do estado do “vai-um” pode ser somada a um ou zero e depois transitar ou não para o estado de “vai-um” adequado. Prefixar a cadeia de entrada dos números com um par de zeros adicionais à esquerda para considerar a possibilidade de um último “vai-um”. É possível resolver este problema com três estados. A sequência de saída é o número resultante em binário escrito de modo reverso.

1.3. Construir uma máquina de estados finitos do tipo Mealy que emitirá saída com valor `1` se o número de símbolos de entrada lidos for divisível por `3`; caso contrário, emitir saída com valor `0`.

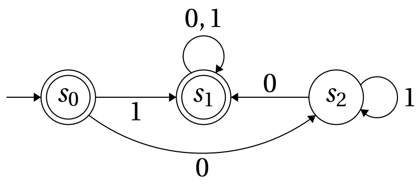
1.4. Construir uma máquina de estados finitos do tipo Moore que emitirá saída com valor `1` se o número de símbolos de entrada lidos for divisível por `4`; caso contrário, emitir saída com valor `0`.

1.5. Construir uma máquina de estados finitos do tipo Moore que determine se uma cadeia de entrada contém um número par ou impar de `1`’s, emitindo saída `1` se o número for par e `0`, caso contrário.

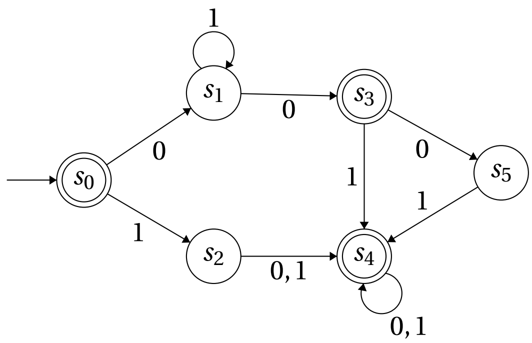
2 Autômatos finitos

2.1. Determinar as linguagens reconhecidas pelos autômatos finitos determinísticos apresentados a seguir:

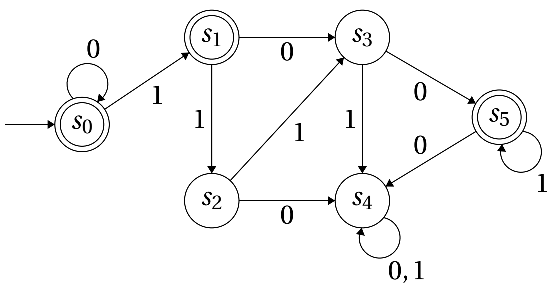
(a)



(b)



(c)



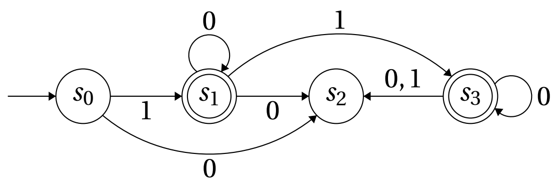
2.2. Construir um autômato de estados finito determinístico que reconheça o conjunto de todas as cadeias binárias contendo a sequência `101`.

2.3. Construir um autômato de estados finito determinístico que reconheça o conjunto de todas as cadeias binárias que começam por `0` ou `11`.

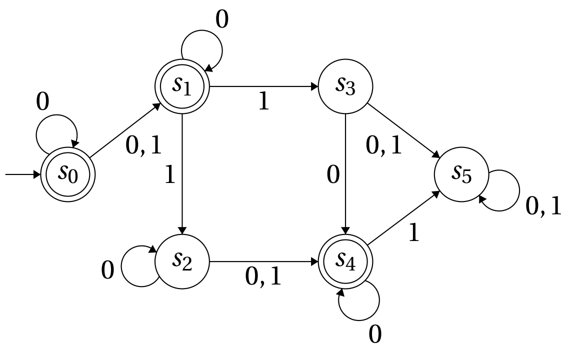
2.4. Construir um autômato de estados finito determinístico que reconheça o conjunto de todas as cadeias binárias consistindo que tenham o símbolo `0` seguido por uma cadeia de um número ímpar do símbolo `1`.

2.5. Determinar as linguagens reconhecidas pelos autômatos finitos não-determinísticos apresentados a seguir:

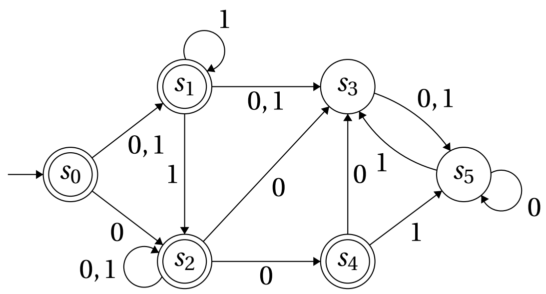
(a)



(b)



(c)



2.6. Converter os autômatos não-determinísticos do exercício anterior em autômatos determinísticos.