Operações sobre autômatos

Vimos na Aula 02 que algumas linguagens podem ser construídas a partir de outras, usando-se operações de conjuntos. Dado que os AFDs são formalismos que também expressam linguagens, seria possível obter novas linguagens aplicando-se algumas operações sobre os autômatos. Por exemplo, dados M_1 e M_2 abaixo, que reconhecem respectivamente as linguagens $\{0\}\{0,1\}^*$ e $\{0,1\}^*\{1\}$, seria possível obter um AFD que reconhecesse $L(M_1) \cap L(M_2) = \{0\}\{0,1\}^*\{1\}$ sem ter que iniciar o projeto do zero? Veremos que a resposta é sim!

co

Note que, apesar de ser um exemplo trivial, já que o AFD poderia ser facilmente obtido do zero, existem casos em que a utilização da técnica que veremos será mais fácil que desenhar o autômato por completo. Ou seja, da mesma forma que as operações de conjunto auxiliam na definição de linguagens mais complexas, as operações sobre os autômatos que veremos a seguir auxiliam na obtenção de autômatos para linguagens mais complexas.

A primeira operação que veremos será para a obtenção de um AFD para o complemento da linguagem. Para isso, considere um AFD $M = (Q, \Sigma, \delta, i, F)$. Um AFD M' que reconhece $L(\overline{M})$ é obtido invertendo-se os estados finais e não-finais de M. Ou seja, $M' = (Q, \Sigma, \delta, i, Q - F)$.

Exemplo: Para os dois autômatos da figura acima, basta tornar p e $c1$ estados finais em M_1 e $c0$ não-final para reconhecer $L(\overline{M_1})$. O mesmo vale para M_2.

Agora partiremos para as operações para obtenção de união e interseção de linguagens. O método usado para ambos os casos será o mesmo. A diferença será na especificação dos estados finais. Esse método é chamado de **produto de autômatos**.

Considere o caso de reconhecimento da interseção das linguagens de dois AFDs. No nosso exemplo, suponha que estivéssemos interessados em reconhecer $L(M_1) \cap L(M_2)$ como posto acima. Intuitivamente, poderíamos simular a computação de uma palavra em paralelo em ambos os autômatos. Se a palavra fosse reconhecida por ambos, i.e., ambos terminam suas computações em estados finais, então a palavra seria reconhecida. A técnica do produto de autômatos materializa exatamente essa ideia.

Para isso, vamos criar um autômato cujos estados serão todos os pares possíveis dos estados dos outros autômatos. Ou seja, considerando $M_1 = (Q_1, \Sigma, \delta_1, i_1, F_1)$ e $M_2 = (Q_2, \Sigma, \delta_2, i_2, F_2)$, os estados do novo autômato seriam $Q_p = Q_1 \times Q_2$. A computação nos autômatos originais começava em seus estados iniciais. Logo, a computação no autômato que simulará as computações em paralelo deve ser iniciar no estado $[i_1, \quad i_2]$. Como a ideia é simular as computações dos outros autômatos, as transições nesse novo autômato executa exatamente essa operação. Sendo assim, $\delta_p([e_1, e_2], a) = [\delta_1(e_1, a), \delta_2(e_2, a)]$. Por fim, como dito anteriormente, os estados finais dependerão do objetivo. No caso de interseção, uma palavra deve ser reconhecida somente se ambos os autômatos a reconhecerem. Portanto, ela deve ser aceita pelo novo autômato somente se o estado em que a computação terminou $[e_1, e_2] \in F_1 \times F_2$. Já no caso da união, basta que um dos dois reconheça a palavra. Logo, a computação deve terminar em um estado $[e_1, e_2] \in (F_1 \times Q_2 \cup Q_1 \times \quad F_2)$.

Exemplo: Vamos computar o produto dos autômatos M_1 e M_2 da figura anterior.