Universidade Federal do Maranhão

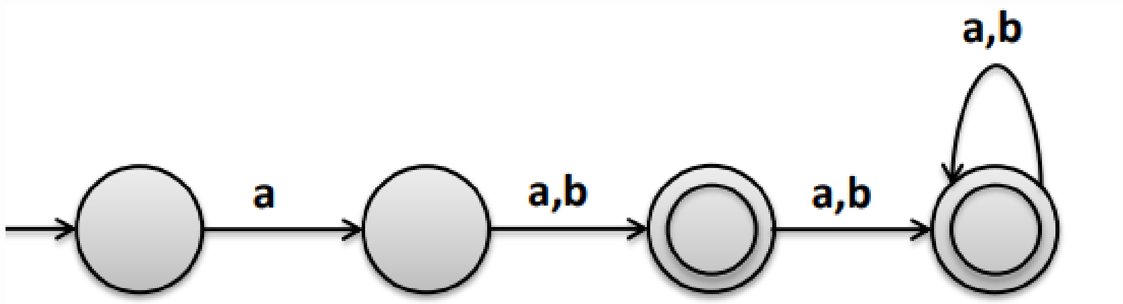
Minimização de Autômatos Finitos

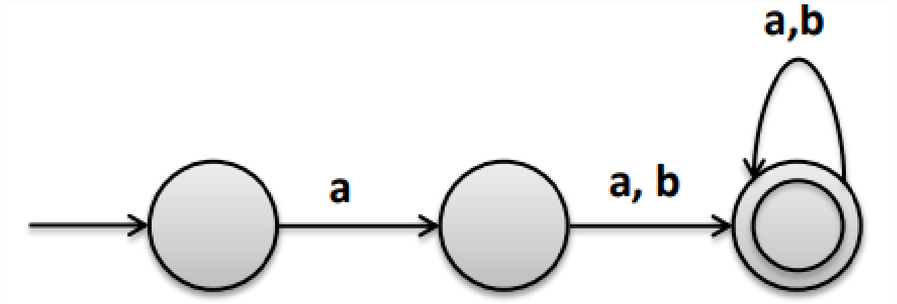
Katarina Ires
Núbia Valvan Alves Freitas
Pedro Lucas Cardoso Correa

SUMÁRIO

Introdução

- Considere a palavra w = abababab. Em termos de processamento, qual dos autômatos abaixo é mais eficiente?





- Os dois possuem a mesma eficiência, já que o processamento depende do tamanho da palavra e não do tamanho do autômato. Então por que minimizar?

Introdução

- Autômatos são maquinas reconhecedoras de palavras sobre uma linguagem, formados por uma quíntupla M={Q, Σ, δ, q0, qf};
- O processo de minimização visa unificar os estados equivalentes de um autômato, tornando o autômato final mínimo e único. Toda linguagem regular possui um autômato finito determinístico, mínimo e único que a reconhece;
- Minimizar um AFD não o torna mais eficiente, pois o tempo de processamento não depende do número de estados, muito menos do autômato de reconhecimento considerado, mas sim do tamanho da palavra a ser testada, ou seja, qualquer autômato finito determinístico que reconheça a linguagem terá a mesma eficiência;
- Um AFD M é dito ser mínimo para a linguagem L(M) se nenhum AFD para L(M) contém menor número de estados que M.

Comentários Iniciais

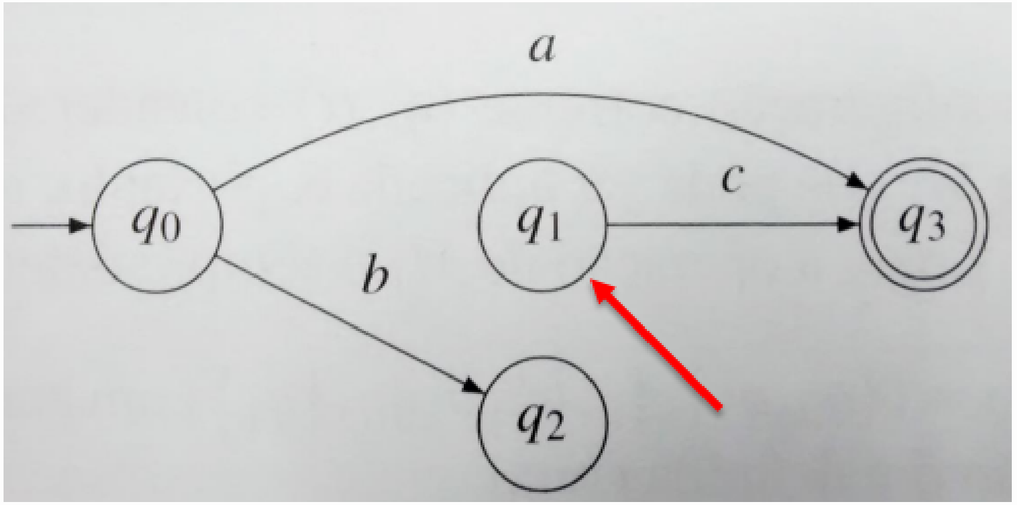
- Foi demonstrado que esse resultado é válido apenas para a classe das linguagens regulares;
- Possibilita a construção de reconhecedores sintáticos extremamente compactos e eficientes;
- É possível automatizar a minimização de autômatos finitos;
- Autômato finito mínimo é único para cada linguagem regular, possibilitando a elaboração de novos métodos no estudo de linguagens formais.

Método de Minimização de Estados

- Partimos do pressuposto de que o autômato a ser minimizado é determinístico. Não possui transições com cadeia vazia;
- Os estados devem ser alcançáveis a partir do estado inicial;
- Agrupamento e fusão de estados equivalentes.

Estados Inacessíveis

- Um estado qi é dito inacessível se não existe no autômato qualquer caminho, formado por transições válidas, que leve do estado inicial até qi;
- *Estados inacessíveis não contribuem para o poder de reconhecimento do autômato, já que nenhuma cadeia $w \in \Sigma^*$ pode levar o autômato do estado inicial até esses estados.*



Estados Inacessíveis

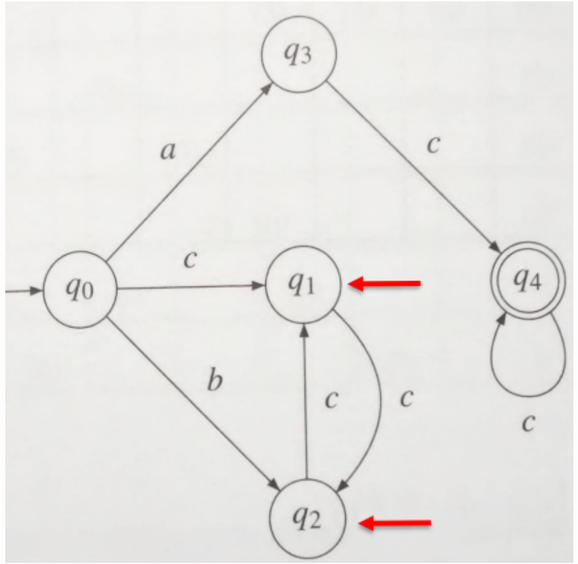
- Algoritmo para eliminação de estados inacessíveis:

Duas marcas para cada estado: acessível e finalizado; o método inicia com todos os estados em branco

1. Marque o estado inicial como acessível (mas não como finalizado);
2. Enquanto houver um estado qi marcado como acessível mas não finalizado:
    - Para cada estado qj ainda não marcado, tal que haja uma transição $qi \acute{} a\ qj$, marque qj como acessível
    - Quando todos os estados vizinhos de qi tiverem sido inspecionados, marque qi como finalizado.
3. Elimine todos os estados não marcados

Estados Inúteis

- Um estado qi é dito inútil se não existe no autômato qualquer caminho, formado por transições válidas, que leve de qi até algum estado de aceitação;
- Nenhuma cadeia $w \in \Sigma *$ conduz o autômato de um estado inútil até um dos estados finais;



Estados Inúteis

- Algoritmo para eliminação de estados inúteis:

Duas marcas para cada estado: útil e finalizado; o método inicia com todos os estados em branco

1. Marque todos os estados de aceitação como estados úteis (mas não como finalizados)
2. Enquanto houver um estado qi marcado como útil mas não finalizado:
    - Para cada estado qj ainda não marcado, tal que haja uma transição $qj \acute{} a\ qi$, marque qj como útil
    - Quando todos os estados satisfazendo a condição (a) tiverem sido inspecionados, marque qi como finalizado.
3. Elimine todos os estados não marcados

Equivalência entre Estados

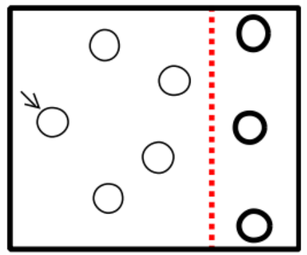
- Definição: Dois estados $q1$, $q2$ são ditos k-indistinguíveis (denotado por $q1 \equiv_k q2$) se e apenas se não houver cadeia x, $|x| \leq k$, que permita distinguir $q1$ de $q2$.
- De acordo com a definição acima, para quaisquer pares de estados qi , $qj \in Q$, valem:
  - $qi \equiv_0 qj$ se e somente se ambos forem de aceitação ou nenhum for de aceitação:
    - $qi \equiv_0 qj \Leftrightarrow (qi , qj \in F \vee qi , qj \in Q - F)$.
  - $qi \equiv_k qj$ se e somente se:
    - $qi \equiv_{k-1} qj$;
    - $\forall\ a \in \Sigma, \delta\ (qi , a) \equiv_{k-1} \delta\ (qj , a)$.

Equivalência entre Estados

- Teorema: Seja M = (Q, Σ, δ, $q0$, F) um AFD com n estados, e considere dois estados quaisquer $q1$, $q2$ de M. Então, $q1 \equiv q2$ se e somente se $q1 \equiv_{n-2} q2$.
  - (n - 2) indistinguibilidade.
- Ideia da demonstração:
  - $q1 \equiv q2 \Rightarrow q1 \equiv_{n-2} q2$: imediato, pois $q1 \equiv k\ q2 \Rightarrow q1 \equiv k\text{-}1\ q2$
  - $q1 \equiv_{n-2} q2 \Rightarrow q1 \equiv q2$:
    - Trivial se o autômato tiver estados que são somente de aceitação ou de rejeição.
      - Assumindo que a função de transição $\delta: Q \times \Sigma \acute{} Q$ seja total.

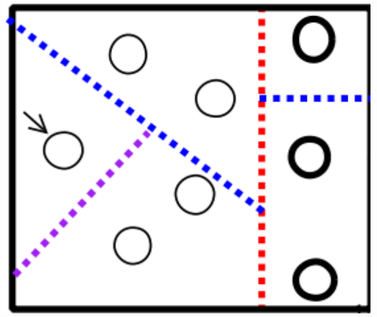
Equivalência entre Estados

- Consideremos agora o caso mais geral em que há tanto estados finais como não-finais em M.
- De acordo com o critério $\equiv 0$, o conjunto Q pode ser particionado inicialmente em dois grandes grupos:
  - O primeiro formado pelos estados finais (F)
  - O segundo pelos estados não finais $(Q - F)$
  - Trata-se, portanto, do primeiro de uma série de sucessivos refinamentos do o objetivo de determinar as classes de equivalências de estados de M.



Equivalência entre Estados

- Executa-se, em seguida, para cada um dos dois subconjuntos obtidos através de $\equiv 0$, seu particionamento através de relações $\equiv i$, i = 1, 2, 3, etc.
- Como M possui n estados, sendo alguns finais e outros não, o maior subconjunto de Q criado através de $\equiv 0$ possui no máximo n - 1 estados, portanto, haverá no máximo n - 2 refinamentos sucessivos de $\equiv 0$ gerando conjuntos de classes de equivalência, distintas umas das outras.



Equivalência entre Estados

- Para completar a demonstração, basta provar que cada um dos n - 2 particionamentos distintos sucessivos (no máximo) refere-se ao uso correspondente de cadeias de comprimento 1, 2, ... , n - 2, para efetuar o teste de distinguibilidade do par de estados.
  - Consequentemente, não há possibilidade de ocorrer um novo particionamento distinto dos anteriores para cadeias de comprimento k se os particionamentos obtidos para cadeias de comprimento k - 1 e k - 2 se mostrarem idênticos.
- Para provar essa afirmação, considere-se o conjunto de todas as classes de equivalência de M que satisfazem simultaneamente às relações $\equiv k$ e $\equiv k + 1$. Nesse caso, essas mesmas classes de equivalência satisfazem a $\equiv k + 2$, $\equiv k + 3$ e assim sucessivamente.

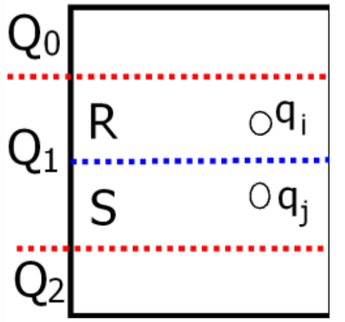
Equivalência entre Estados

- Considere-se, por exemplo, uma situação hipotética em que:

i. a relação $\equiv k$ particiona um certo conjunto Q em três subconjuntos $Q0$, $Q1$ e $Q2$;

ii. a relação $\equiv k + 1$ preserva o particionamento da relação $\equiv k$ inalterado;

iii. a relação $\equiv k + 2$ produz uma partição diferente, digamos $Q0$, R, S e $Q2$, com $R \cup S = Q1$, $R \cap S = \varnothing$: $\equiv k$: $Q0, Q1, Q2$ // $\equiv k+1$: $Q0, Q1, Q2$ // $\equiv k+2$: $Q0, R, S, Q2$.



Equivalência entre Estados

- Admitindo-se, por hipótese, que $Q1$ seja particionado em duas novas classes de equivalência R, S, isso significa que existem $q1$, $q2 \in Q1$ tais que $q1 \not\equiv_{k+2} q2$. Mas para que isso fosse verdade, seria necessário, de acordo com a definição, que:
  - i. $q1 \not\equiv_{k+1} q2$, ou
  - ii. $\delta(q1, a) \not\equiv_{k+1} \delta(q2, a)$ para algum $a \in \Sigma$.
- A condição i é falsa, pois de acordo com a hipótese original, $q1$, $q2 \in Q1$ e portanto $q1 \equiv_{k+1} q2$.
- A condição ii também é falsa, pois se $q1 \equiv_{k+1} q2$ então $\delta(q1, a) \equiv_k \delta(q2, a)$, como por hipótese, as partições produzidas pelas relações $\equiv_k$ e $\equiv_{k+1}$ são idênticas, então $\delta(q1, a) \equiv_{k+1} \delta(q2, a)$.

Equivalência entre Estados

- Fica, portanto, demonstrado que:
  - Na hipótese de serem obtidos dois conjuntos idênticos de classes de equivalência para k e k + 1, não haverá mais necessidade de se analisar a equivalência de tais classes para valores maiores do que k.
  - Para um autômato finito com n estados, haverá no máximo n - 1 conjuntos distintos de classes de equivalência ($\equiv$ 0 e os demais n - 2), cada qual associado a cadeias de comprimento 0 até n - 2, não havendo, portanto, necessidade de se examinar a equivalência de tais classes para cadeias de comprimento superior a n - 2.

Etapas para Minimização de Estados

- Remoção dos estados inacessíveis e inúteis;
- Identificação dos pares de estados equivalentes entre si;
- Agrupamentos dos estados em classes de equivalência, cada uma identificada pelo seu representante;
- Criação de um novo AFD mínimo:
  - Novos estados correspondem aos representantes das classes de equivalências do AFD original;
  - Novas transições são obtidas do AFD original, substituindo a referência aos estados originais pelos respectivos representantes.

Identificação dos Pares de Estados Equivalentes - Algoritmo

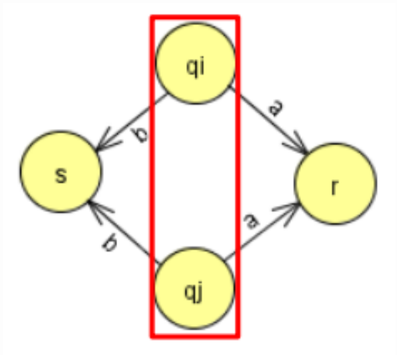
- Entrada: um AFD $M = (Q, \Sigma, \delta, q0, F)$
- Saída: Uma partição $Q0, Q1, \ldots, QK$ do conjunto Q de estados, de tal forma que seus elementos correspondem às mais amplas classes de equivalências de estados existentes em Q.
  - Divide-se o conjunto original de estados de M nos dois subconjuntos que compõem sua partição inicial:
    - Subconjunto dos estados finais;
    - Subconjunto dos estados não finais;
    - Justificativa: Em um AFD, um estado final, qualquer que seja ele, é sempre distinguível de um estado não final (já que a cadeia vazia os distingue).
  - Essa partição inicial corresponde, portanto, ao resultado da aplicação da relação $\equiv 0$ ao conjunto Q.

Identificação dos Pares de Estados Equivalentes - Algoritmo

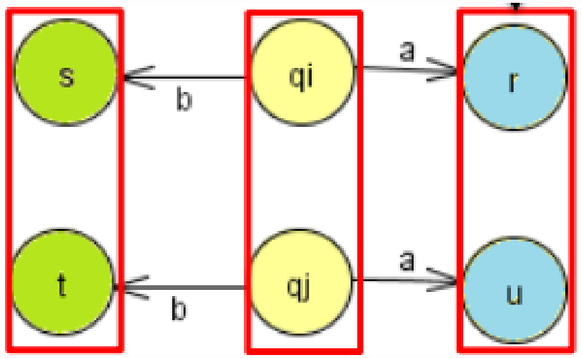
- Para cada um dos subconjuntos obtidos em (1), refiná-los em novas partições, segundo o critério:
  - Dois estados qi , qj de um mesmo subconjunto Qi , obtido de uma partição prévia do conjunto Q de estados, são equivalentes se e somente se:
    - qi e qj têm transições definidas sobre o mesmo conjunto de símbolos $S \subseteq \Sigma$, e
    - Para cada um desses símbolos a $\in S$:
      - $\delta\ (qi\ ,\ a) = \delta\ qj\ ,\ a$ ou
      - $\delta\ (qi\ ,\ a) \neq \delta\ (qj\ ,\ a)$ mas $\delta\ (qi\ ,\ a)$ e $\delta\ (qj\ ,\ a)$ são equivalentes.
      - Caso contrário, qi e qj não são equivalentes e devem, portanto, ensejar uma partição de Qi .

Identificação dos Pares de Estados Equivalentes - Algoritmo

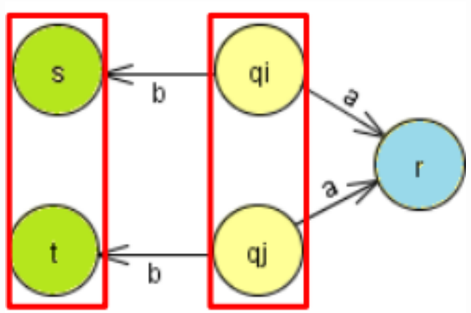
- Representações dos casos que satisfazem à condição 2b do algoritmo:



Transições com as mesmas entradas para estados idênticos



Transições com as mesmas entradas para estados equivalentes



Transições com as mesmas entradas para estados idênticos e equivalentes

Agradecemos