Linguagens Formais e Autômatos

Aula 15 - Formatos e formas normais

Prof. Dr. Daniel Lucrédio
Departamento de Computação / UFSCar
Última revisão: ago/2015

Referências bibliográficas

- **Introdução à teoria dos autômatos, linguagens e computação / John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman** ; tradução da 2.ed. original de Vandenberg D. de Souza. - Rio de Janeiro : Elsevier, 2002 (Tradução de: Introduction to automata theory, languages, and computation - ISBN 85-352-1072-5)
  - Capítulo 7 - Seção 7.1
- **Introdução à teoria da computação / Michael Sipser** ; tradução técnica Ruy José Guerra Barretto de Queiroz ; revisão técnica Newton José Vieira. -- São Paulo : Thomson Learning, 2007 (Título original : Introduction to the theory of computation. "Tradução da segunda edição norte-americana" - ISBN 978-85-221-0499-4)
  - Capítulo 2 - Seção 2.1

Formatos

- A notação utilizada é a BNF ou Backus-Naur Form
- Basicamente é a mesma utilizada até agora, porém no seguinte formato:
  - <símbolo> ::= <expressão>
  - Onde <símbolo> é sempre um não-terminal ou variável, e expressão é uma sequência de terminais e/ou não terminais
  - O símbolo “|” significa a união de duas produções
  - Ex: S ::= E “+” E | E “*” E | “(“ E “)”
- Existe também a BFN estendida, ou EBNF
  - Basicamente existem algumas notações especiais
    - [ ] = opcional
    - { } = repetição
    - ( ) = agrupamento
    - Etc...

Formatos e formas normais

- Formatos são úteis para utilização prática
  - Ex: BNF surgiu para analisar programas em ALGOL
- Mas também existem as formas normais
  - São “maneiras” de escrever as produções, seguindo algumas propriedades
  - Ex: ao invés de escrever
    - $S \rightarrow abc$
  - Escrevo:
    - $S \rightarrow aB$
    - $B \rightarrow bC$
    - $C \rightarrow c$

Formais normais

- Possuem algumas propriedades úteis, permitindo
  - Simplificar gramáticas
  - Analisar aspectos da gramática
  - Estudo teórico e formal
- Apresentaremos duas
  - Forma Normal de Chomsky
  - Forma Normal de Greibach

Forma Normal de Chomsky

- CNF – Chomsky Normal Form
- É uma forma normal simplificada, onde todas as produções estão em uma entre duas formas simples:
  - $A \rightarrow BC$, onde A, B e C são todas variáveis, ou
  - $A \rightarrow a$, onde A é uma variável e a é um terminal
- Qualquer linguagem livre de contexto é gerada por uma gramática livre de contexto na CNF
  - Ou seja, é possível converter uma gramática para a CNF

Forma Normal de Chomsky

- Antes de fazer a conversão, porém, é necessário:
  - Eliminar símbolos inúteis
  - Eliminar produções vazias ($A \to \varepsilon$)
  - Eliminar produções unitárias ($A \to B$, onde B é uma variável)

Eliminação de símbolos inúteis

- Formalmente:
  - Um símbolo X é útil para uma gramática G=(V,T,P,S), se existe alguma derivação da forma $S \Rightarrow^{*} \alpha X \beta \Rightarrow^{*} w$, onde w está em T*
- Informalmente, X é útil se:
  - É gerador, ou seja, $X \Rightarrow^{*} w$ para alguma cadeia de terminas w
  - E é alcançável, ou seja, $S \Rightarrow^{*} \alpha X \beta$ para algum α e β
- Um símbolo útil é ao mesmo tempo gerador e alcançável, caso contrário ele é inútil
  - Símbolos inúteis podem ser removidos da gramática, junto com as produções que os envolvem
- Procedimento: primeiro eliminam-se os símbolos não geradores, e depois os símbolos não alcançáveis
  - Só sobram os símbolos úteis

Eliminação de símbolos inúteis

- Exemplo:
  - $S \rightarrow AB \mid a$
  - $A \rightarrow b$
- Todos os símbolos com exceção de B são geradores
  - a e b geram a si mesmos
  - S gera a, e A gera b
- Elimina-se o B, e as produções que o envolvem ($S \rightarrow AB$)
- Resultado:
  - $S \rightarrow a$
  - $A \rightarrow b$

Eliminação de símbolos inúteis

- Apenas S e a são alcançáveis a partir de S. Eliminando A e b, fica:
  - $S \rightarrow a$
- Essa gramática denota a mesma linguagem que a gramática original
- Observe que se tivéssemos primeiro removido os símbolos inalcançáveis e depois os geradores, o resultado seria errado

Cálculo de símbolos geradores e alcançáveis

- Algoritmo para calcular os símbolos geradores
  - Base: todo símbolo de T é sem dúvida gerador, pois ele gera a si próprio
  - Indução: suponha que exista uma produção $A \rightarrow \alpha$, e todo símbolo de α já seja conhecido como gerador. Então, A é gerador.
    - Obs: Se $A \rightarrow \varepsilon$, então A é gerador

Cálculo de símbolos geradores e alcançáveis

- Algoritmo para calcular os símbolos alcançáveis
  - Base: S é sem dúvida alcançável (pois é o símbolo inicial)
  - Indução: suponha que descobrimos que alguma variável A é alcançável. Então para todas as produções com A na cabeça, os símbolos dos corpos dessas produções também são alcançáveis.

Exercício

- Encontre uma gramática equivalente à seguinte, sem símbolos inúteis
    - $S \rightarrow AB \mid CA$
    - $A \rightarrow a$
    - $B \rightarrow BC \mid AB$
    - $C \rightarrow aB \mid b$
- Resposta:
    - Símbolos geradores: {a,b,A,C,S}
        - B não é gerador, portanto eliminamos as produções envolvendo B:
            - $S \rightarrow CA$
            - $A \rightarrow a$
            - $C \rightarrow b$
    - Símbolos alcançáveis: {S, A, C}
        - Não há símbolos não alcançáveis, portanto a gramática acima não possui símbolos inúteis

Eliminação de produções vazias

- Produções do tipo $A \rightarrow \varepsilon$ são convenientes no projeto de gramáticas, mas não são essenciais
- Podem ser removidas para simplificar a gramática
  - Por exemplo, a CNF não permite este tipo de produção
- Porém, há um efeito colateral
  - Se a linguagem da gramática inclui a cadeia vazia (ou seja, $S \Rightarrow^* \varepsilon$), a remoção das produções vazias obviamente elimina a cadeia vazia da linguagem

Eliminação de produções vazias

- Estratégia:
  - Descobrir quais variáveis são "anuláveis"
    - Ou seja, quais variáveis podem ser substituídas (em uma ou mais derivações) pela cadeia vazia
    - A é anulável se $A \Rightarrow^{*} \varepsilon$
- Algoritmo:
  - Base: se $A \rightarrow \varepsilon$ é uma produção, então A é anulável
  - Indução: se existe uma produção $B \rightarrow C_1C_2...C_k$, onde cada C_i é anulável, então B é anulável
    - Obs: cada C_i deve ser uma variável, pois terminais não são anuláveis

Eliminação de produções vazias

- Uma vez descobertos os símbolos anuláveis
  - Iremos modificar a gramática, mas sem modificar a linguagem
    - Com exceção, é claro, do fato de que a nova linguagem não mais aceita a cadeia vazia
- Algoritmo:
  - Para cada produção $A \rightarrow X_1X_2...X_k$, onde $k \geq 1$, suponha que m dos k valores de X_i sejam símbolos anuláveis
    - Criaremos $2^m - 1$ novas produções, de forma que todas as combinações possíveis em que cada X_i anulável está presente ou ausente façam parte da gramática

Eliminação de produções vazias

- Exemplo: Produção B → CADA, A e D são anuláveis
- Devemos produzir as seguintes produções (7):
  - B → C (A e D ausentes)
  - B → CAD (Segundo A ausente)
  - B → CAA (D ausente)
  - B → CDA (Primeiro A ausente)
  - B → CA (D e segundo A ausentes)
  - B → CD (Ambos os As ausentes)
  - B → CA (Primeiro A e D ausentes)
- Neste caso, a regra B → CA aparece duas vezes, e uma delas pode ser eliminada

Exercício

- Elimine as produções vazias da seguinte gramática
  - $S \rightarrow AB$
  - $A \rightarrow aAA \mid \varepsilon$
  - $B \rightarrow bBB \mid \varepsilon$
- Resposta:
  - Símbolos anuláveis: {A,B,S}
  - $S \rightarrow AB \mid A \mid B$
  - $A \rightarrow aAA \mid aA \mid a$
  - $B \rightarrow bBB \mid bB \mid b$

Eliminação de produções unitárias

- Produção unitária: $A \rightarrow B$, onde A e B são variáveis
  - Obs: $A \rightarrow b$ não é uma produção unitária, se b for um terminal
- Produções unitárias podem complicar certas provas, e introduzir etapas extras em derivações
- Método simples: expandir produções unitárias até que desapareçam

Eliminação de produções unitárias

- Exemplo:
  - $I \rightarrow a \mid b \mid Ia \mid Ib \mid I0 \mid I1$
  - $F \rightarrow I \mid (E)$
  - $T \rightarrow F \mid T * F$
  - $E \rightarrow T \mid E + T$
- Há três produções unitárias:
  - $F \rightarrow I$, $T \rightarrow F$ e $E \rightarrow T$
- Eliminando $E \rightarrow T$:
  - $E \rightarrow F \mid T * F \mid E + T$
- Agora apareceu outra: $E \rightarrow F$, eliminando:
  - $E \rightarrow I \mid (E) \mid T * F \mid E + T$
- Apareceu mais uma: $E \rightarrow I$, eliminando:
  - $E \rightarrow a \mid b \mid Ia \mid Ib \mid I0 \mid I1 \mid (E) \mid T * F \mid E + T$
- Assim fazemos sucessivamente

Eliminação de produções unitárias

- Esse método tem um problema
  - Se houver um ciclo de produções unitárias, como $A \rightarrow B$, $B \rightarrow C$ e $C \rightarrow A$
- Existe uma segunda técnica, mas não veremos aqui
  - Consulte a bibliografia da disciplina

Simplificações

- É preciso um certo cuidado na ordem de aplicação das simplificações:
  - Primeiro, eliminam-se as produções vazias
  - Segundo, eliminam-se as produções unitárias
  - Terceiro, elminam-se os símbolos inúteis
    - Primeiro eliminando-se os símbolos não-geradores
    - Depois eliminando-se os símbolos não-alcançáveis
- Essa ordem garante que não “sobra” nenhuma simplificação por fazer

Forma Normal de Chomsky

- Uma gramática na Forma Normal de Chomsky:
  - Não possui nenhum símbolo inútil
  - Não possui produções vazias
  - Não possui produções unitárias
    - Obs: neste ponto, com certeza toda produção terá a forma $A \rightarrow a$, ou terá um corpo de comprimento 2 ou mais
  - E todas as produções estão em uma dentre duas formas simples:
    - $A \rightarrow BC$, onde A, B e C são variáveis, ou
    - $A \rightarrow a$, onde A é uma variável e a é um terminal.
      - Obs: esta condição já é parcialmente aceita caso os três itens acima sejam satisfeitos

Forma Normal de Chomsky

- São necessárias duas tarefas:
  - a) Organizar todos os corpos de comprimento 2 ou mais que consistem apenas em variáveis
  - b) Desmembrar os corpos de comprimento 3 ou mais em uma cascata de produções, cada uma com um corpo consistindo em duas variáveis
- Para satisfazer a), basta fazer o seguinte:
  - Para todo terminal a que aparecer em um corpo de comprimento 2 ou mais, crie uma nova variável, digamos A, com apenas uma produção: $A \rightarrow a$
    - Usamos A em lugar de a em todo lugar que a aparecer em um corpo de comprimento 2 ou mais
  - Ex: $S \rightarrow XaYY \mid aaT$
  - Transformando:
    - $S \rightarrow XAYY \mid AAT$
    - $A \rightarrow a$

Forma Normal de Chomsky

- Neste ponto, toda produção terá um corpo que será um único terminal ou pelo menos duas variáveis e nenhum terminal
  - Agora basta desmembrar as produções $A \rightarrow B_1B_2...B_k$ para $k \geq 3$, em um grupo de produções com duas variáveis em cada corpo
  - Exemplo: $A \rightarrow B_1B_2B_3B_4B_5$
  - Resultado:
    - $A \rightarrow B_1C_1$
    - $C_1 \rightarrow B_2C_2$
    - $C_2 \rightarrow B_3C_3$
    - $C_3 \rightarrow B_4B_5$

Forma Normal de Chomsky

- Exemplo
- Considere a seguinte gramática:

$$S \to ASA \mid aB$$

$$A \to B \mid S \mid \varepsilon$$

$$B \to b \mid \varepsilon$$

Forma Normal de Chomsky

- Passo 1: Remover as produções vazias

S → ASA | aB
A → B | S | ε
B → b | ε

S → ASA | aB **| a**
A → B | S | ε
B → b ~~**| ε**~~

S → ASA | aB | a
S → SA | AS | S
A → B | S ~~**| ε**~~
B → b

Forma Normal de Chomsky

- Passo 2: Remover produções unitárias (A→B) (neste caso não há ciclos)

S → ASA | aB | a | SA | AS ~~| **S**~~

A → B | S

B → b

S → ASA | aB | a | SA | AS

A → ~~**B** |~~ S **| b**

B → b

S → ASA | aB | a | SA | AS

A → ~~**S** |~~ b **| ASA | aB | a | SA | AS**

B → b

Forma Normal de Chomsky

- Passo 3: Converter as regras remanescentes para a forma apropriada, criando novas variáveis quando necessário

$$
\begin{aligned}
S &\rightarrow ASA \mid aB \mid a \mid SA \mid AS \\
A &\rightarrow b \mid ASA \mid aB \mid a \mid SA \mid AS \\
B &\rightarrow b
\end{aligned}
$$

$$
\begin{aligned}
S &\rightarrow AA_1 \mid UB \mid a \mid SA \mid AS \\
A &\rightarrow b \mid AA_1 \mid UB \mid a \mid SA \mid AS \\
B &\rightarrow b \\
A_1 &\rightarrow SA \\
U &\rightarrow a
\end{aligned}
$$

Exercícios

- Comece com a gramática

  $S \to ASB \mid \varepsilon$

  $A \to aAS \mid a$

  $B \to SbS \mid A \mid bb$

- Elimine as produções vazias
- Elimine quaisquer produções unitárias na gramática resultante
- Elimine símbolos inúteis
- Coloque a gramática resultante na forma normal de Chomsky

Exercícios

- Eliminando produções vazias
  - Símbolos anuláveis: {S}
- Nova gramática:
  - $S \rightarrow ASB \mid AB$
  - $A \rightarrow aAS \mid aA \mid a$
  - $B \rightarrow SbS \mid bS \mid Sb \mid b \mid A \mid bb$

Exercícios

- Eliminando produções unitárias
  - A única produção unitária é $B \rightarrow A$, basta substituir o corpo
- Nova gramática
  - $S \rightarrow ASB \mid AB$
  - $A \rightarrow aAS \mid aA \mid a$
  - $B \rightarrow SbS \mid bS \mid Sb \mid b \mid aAS \mid aA \mid a \mid bb$

Exercícios

- Eliminando símbolos inúteis
  - Símbolos não-geradores
  - Símbolos não-alcançáveis
- Não existem, a gramática permanece a mesma

Exercícios

- Transformando para CNF (passo 1)
  - S → ASB | AB
  - A → XAS | XA | a
  - B → SYS | YS | SY | b | XAS | XA | a | YY
  - X → a
  - Y → b
- Transformando para CNF (passo 2)
  - S → AE | AB
  - A → CF | CA | a
  - B → SG | DS | SD | b | CF | CA | a | DD
  - C → a
  - D → b
  - E → SB
  - F → AS
  - G → YS

Forma Normal de Greibach

- Toda produção é da forma $A \rightarrow a\alpha$
- Onde a é um terminal e α é uma cadeia de zero ou mais variáveis
- Conversão é complexa, mas existem algumas aplicações teóricas
  - Pelo fato de que cada produção introduz exatamente um único terminal
  - Ou seja, uma cadeia de comprimento n tem n etapas de derivação
- Mas não entraremos no assunto

Fim

Aula 15 - Formatos e formas normais