Linguagens Livres de Contexto

Continuação

Hierarquia de Chomsky para Linguagens Recursivamente Enumeráveis

Tipo 0 (recursivamente enumeráveis)
Tipo 1 (sensíveis ao contexto)
Tipo 2 (livres de contexto)
Tipo 3 (regulares)

Gramáticas

Dispositivos geradores de linguagens, obedecendo à hierarquia de Chomsky, definidas como: G=(N, T, P, S), onde:
N – Alfabeto de símbolos não-terminais (ou auxiliares);
T – Alfabeto de símbolos terminais;
P – Conjunto de regras de produção, cada regra é representada da forma $\alpha \rightarrow \beta$, em que se aplicará uma substituição dos símbolos em α pelos símbolos de β, ao se efetuar uma derivação (α possui ao menos um símbolo não-terminal);
S – Símbolo inicial (a partir do qual derivam-se as cadeias da linguagem);

Hierarquia de Chomsky

Gramáticas Livres de Contexto (Tipo 2), restrição:
$\alpha \in N$

Derivação

Processo através do qual as cadeias de uma linguagem. Representa-se através do símbolo $\Rightarrow$. O comprimento de uma derivação corresponde ao número de substituições efetuadas até se produzir uma cadeia de terminais. A linguagem gerada por G é representada por L(G), e, tem-se:
$L(G) = \{w \mid S \Rightarrow^* w \in T^*\}$

Derivação mais à Esquerda/Direita

Representação da derivação através de substituições efetuadas sempre no não-terminal mais à esquerda.

Árvore de Derivação

Representação do processo de derivação de uma cadeia. Ilustra a derivação das cadeias de uma linguagem. Os terminais são representados pelas folhas, e cada nó interno representa um não-terminal.

Autômato com Pilha

Um autômato com pilha é um modelo computacional composto de uma fita de entrada dividida em células, nas quais estarão os símbolos das cadeias (cada símbolo em uma célula), uma pilha de trabalho e um controle finito por estados. A principal parte do modelo é um controle finito, o qual indica em qual estado, dentre um número finito de estados distintos, o autômato estará. Há ainda um cabeçote de leitura, que é capaz de capturar o símbolo presente na posição corrente da fita, e, ao capturá-lo, avança o cabeçote para a próxima célula da fita.

Um modelo de autômato com pilha possui um estado inicial e um conjunto de estados finais (todos pertencentes ao conjunto de estados). O estado inicial designa o início de funcionamento do modelo, dependendo do símbolo presente na fita e do símbolo presente no topo da pilha o próximo estado será escolhido, enquanto que os estados finais indicam a finalização do processo computacional com sucesso, isto é, se a cadeia presente na fita de entrada tiver sido completamente lida, a pilha estiver vazia e o modelo não se encontrar em um estado final, considera-se a cadeia como não aceita, e, em caso contrário, como aceita.

A **linguagem aceita** por um modelo de autômato com pilha é composta pelas cadeias aceitas por este (isto é, aquelas cujo processo computacional termina em um estado final).

$M=(K, \Sigma, \Gamma, \Delta, s, F)$, onde, K é conjunto de estados do autômato, Σ é o alfabeto aceito, Γ é o alfabeto da pilha, s é o estado inicial (que é único), F é conjunto de estados finais ($F \subseteq K$), e Δ é a relação de transferência, $\Delta \subseteq (K \times \Sigma^* \times \Gamma^*) \times (K \times \Gamma^*)$. A relação de transferência determina o próximo estado do autômato e a situação do topo da pilha, a partir do estado corrente, do topo da pilha corrente e do símbolo encontrado na fita de entrada.

Um par $((p, u, \beta), (q, \gamma)) \in \Delta$ é chamado de transição do autômato e descreve o estado anterior à aplicação da transição como sendo p, a cadeia de entrada como sendo u e a cadeia no topo da pilha como sendo β, assim como, q descreve o próximo estado após u ter sido consumida e γ representa a nova cadeia inserida no topo da pilha após a retirada de β.

O modelo de autômato pode ser determinístico se há no máximo uma transição aplicável a cada configuração do autômato.

Configuração

Representa o estado do controle finito, da fita de entrada e do cabeçote de leitura, e do topo da pilha, em momentos sucessivos, isto é, permite acompanhar a evolução do processo computacional. Uma configuração é um elemento do conjunto $K \times \Sigma^* \times \Gamma^*$. Como é possível acompanhar a evolução do processo a cada momento, define-se sobre o conjunto das configurações a relação $\vdash_M$, que indica um par de configurações sucessivas durante o processo computacional, isto é, indica um passo computacional (um único movimento do modelo de autômato sobre o conjunto de configurações). Se $x \in \Sigma^*$, $\alpha \in \Gamma^*$, $((p, u, \beta), (q, \gamma)) \in \Delta$, define-se $(p, ux, \beta\alpha) \vdash_M (q, x, \gamma\alpha)$ como um passo computacional de M. O fecho transitivo e reflexivo da relação de passo é denotado por $\vdash_M^*$.

Usando a definição anterior pode-se definir a linguagem aceita por um autômato com pilha $M=(K, \Sigma, \Gamma, \Delta, s, F)$ como: $L(M) = \{w \in \Sigma^* \mid (s, w, \varepsilon) \vdash_M^* (q, \varepsilon, \varepsilon), q \in F\}$, isto é, todas as cadeias que M aceite.

A representação pode ser feita através de um grafo orientado com as indicações dos estados finais e do estado inicial, como também pode ser através da descrição dos componentes da sêxtupla, com uma tabela representando a relação Δ.

Autômato com Pilha e Linguagens Livres de Contexto

A classe de linguagens aceita por autômato com pilha é exatamente a classe de linguagens geradas pelas gramáticas livres de contexto.

Teorema: A classe de linguagens aceitas pelos autômatos com pilha é exatamente a classe das linguagens livres de contexto.

Demonstra-se dividindo o teorema em duas partes: a primeira construindo um autômato de pilha para simular as ações de geração de uma gramática livre de contexto qualquer, e, a seguir, usando-se o princípio da indução sobre o comprimento da derivação da cadeia. A outra parte fazendo o contrário, e, a seguir, usando-se o princípio da indução sobre o tamanho da cadeia aceita.

A demonstração do teorema anterior é construtiva, isto é, permite a construção de um modelo de autômato com pilha a partir de uma gramática livre de contexto e vice-versa, através de um algoritmo.

Propriedades das Linguagens Livres de Contexto

Há três tipos de propriedades a serem consideradas:

1. Propriedades de fechamento;
2. Propriedades de periodicidade;
3. Propriedades algorítmicas.

Teorema: As linguagens livres de contexto são fechadas sobre a união, a concatenação e a operação estrela de Kleene.

Demonstra-se através da construção de gramáticas livres de contexto apropriadas, usando a possibilidade de considerar os conjuntos de símbolos disjuntos entre duas gramáticas livres de contexto quaisquer.

Teorema: A interseção entre uma linguagem livre de contexto e uma linguagem regular é uma linguagem livre de contexto.

Demonstra-se usando dois modelos de autômato, um autômato com pilha para a gramática livre de contexto e um autômato finito para a linguagem regular. Construindo-se uma nova máquina através da combinação dos dois modelos anteriores (usando o produto cartesiano dos conjuntos de estados), na qual uma transição no novo modelo só existe quando pertence tanto à relação de transição do autômato com pilha quanto à função de transição do autômato finito.

Teorema do bombeamento: Seja G uma gramática livre de contexto. Então existe um número K, dependente de G, tal que qualquer cadeia w em L(G) cujo comprimento é maior do que K pode ser reescrita como w=uvxyz de modo que ou v ou y é não vazio e uv^nxy^nz está em L(G) para todo $n \geq 0$.

Demonstra-se usando propriedades da árvore de derivação e o princípio da casa de pombos ao se atingir um número que corresponde ao limite no uso de símbolos não-terminais sem repetição.

Teorema: $L = \{a^nb^nc^n : n \geq 0\}$ não é livre de contexto.

Demonstra-se usando o teorema do bombeamento.

Teorema: As linguagens livres de contexto não são fechadas sobre a interseção e a complementação.

Demonstra-se usando o bombeamento para mostrar que as linguagens $L_1= a^mb^nc^n$ e $L_2= a^nb^nc^m$ são livres de contexto e que $L_1 \cap L_2= a^nb^nc^n$ não é livre de contexto pelo teorema anterior. E, tendo como base que $L_1 \cap L_2=\Sigma^*-((\Sigma^*-L_1)\cup(\Sigma^*-L_2))$, as linguagens livres de contexto também não são fechadas sobre a complementação.

Autômatos com Pilha Determinísticos

Pode-se definir um autômato com pilha para aceitar uma linguagem L$ tal que $ represente sempre o final das cadeias. Foi definido anteriormente que um autômato com pilha determinístico é aquele que possui no máximo uma transição aplicável a cada configuração, isto é, se há duas ou mais transições aplicáveis a cada configuração então elas deverão ser iguais para que o modelo de autômato seja determinístico.

Teorema: Seja $L \subseteq \Sigma^*$ uma linguagem livre de contexto tal que:

a) Para cada $u \in \Sigma^*$, $\exists\ v \in \Sigma^*$ tal que $uv \in L$;
b) Σ^* - L não é livre de contexto.

Então L não é determinística livre de contexto.

Demonstra-se através da construção de modelo de autômato com pilha.

Reconhecimento de Cadeias de Forma Ascendente

Um modelo de autômato construído diretamente de uma gramática livre de contexto pode conduzir a um modelo não-determinístico. Entretanto, no caso da linguagem ser determinística, é possível resolver o não-determinismo em algumas situações. Quando a linguagem é determinística e basta conhecer-se um único símbolo antes de lê-lo ("*lookahead*") para eliminar o não-determinismo, então constrói-se um novo modelo, a partir do modelo não-determinístico, no qual acrescentam-se estados para desvio após a leitura de cada símbolo . Nestes estados empilha-se a regra com não-terminais adequada e retorna-se ao estado anterior. Constrói-se também um estado final especial, atingido somente quando a cadeia acaba ($).