

Análise Ascendente

Construção de compiladores I

Objetivos

Objetivos

- Apresentar uma motivação para o uso de analisadores ascendentes.
- Introduzir os as funções de fechamento e goto em analisadores LR(0).

Introdução

Introdução

- Até o presente, vimos técnicas de análise sintática top-down.
- Intuitivamente, essas técnicas produzem uma derivação iniciando no símbolo de partida da gramática.

Introdução

- Analisadores descendentes recursivos
 - Simples implementação manual.
 - Restrições sobre a forma da gramática.

Introdução

- Parsing expression grammars
 - Não suportam recursão à esquerda.
 - Semântica não intuitiva.

Introdução

- Analisador LL(1)
 - Constrói uma derivação mais a esquerda.
 - Utiliza uma tabela construída utilizando os conjuntos first e follow.

Introdução

- Problemas com gramáticas LL(1):
 - Parser deve tomar decisão sobre qual produção usar com base nos primeiro token da entrada.

Introdução

- Mas, existe outra alternativa?

Introdução

- Outra possibilidade: Análise ascendente.
 - Construir uma derivação mais a direita ao **inverso**.

Introdução

- Derivação mais à direita
 - Expandir o não terminal mais a direita em cada passo.

Introdução

- Gramática de exemplo

$$E \rightarrow E + E \mid E * E \mid id$$

Introdução

- Derivação mais a direita para $id * id$:

$$E \Rightarrow^R$$

Introdução

- Derivação mais a direita para $id * id$:

$$\begin{array}{l} E \Rightarrow^R \\ E * E \Rightarrow^R \end{array}$$

Introdução

- Derivação mais a direita para $id * id$:

$$\begin{array}{l} E \Rightarrow^R \\ E * E \Rightarrow^R \\ E * id \Rightarrow^R \end{array}$$

Introdução

- Derivação mais a direita para $id * id$:

$$\begin{array}{l} E \Rightarrow^R \\ E * E \Rightarrow^R \\ E * id \Rightarrow^R \\ id * id \end{array}$$

Introdução

- Mas, como um analisador ascendente funciona?

Introdução

- O analisador utiliza:
 - Sequência de tokens de entrada.
 - Pilha.

Introdução

- O analisador pode utilizar 4 tipos de ação:
 - shift: mover itens da entrada para a pilha
 - reduce: escolher regra $A \rightarrow \beta$
 - * remover β do topo da pilha
 - * empilhar A .
 - aceitar e falhar.

Introdução

- Idealmente, o analisador deve reduzir uma palavra de entrada até o símbolo de partida da gramática.
- Vamos mostrar o funcionamento, usando um exemplo.

Introdução

- Exemplo:
 - entrada: id * id\$
 - pilha: \$

Introdução

- Ação: shift id para a pilha.
- Exemplo:
 - entrada: * id\$
 - pilha: \$ id

Introdução

- Ação: reduzir id usando $E \rightarrow id$.
- Exemplo:
 - entrada: * id\$
 - pilha: \$ E

Introdução

- Ação: shift * para a pilha.
- Exemplo:
 - entrada: id\$
 - pilha: \$ E *

Introdução

- Ação: shift id para a pilha.
- Exemplo:
 - entrada: \$
 - pilha: \$ E * id

Introdução

- Ação: reduce id usando $E \rightarrow id$.
- Exemplo:
 - entrada: \$
 - pilha: \$ E * E

Introdução

- Ação: reduce E * E usando $E \rightarrow E * E$.
- Exemplo:
 - entrada: \$
 - pilha: \$ E

Introdução

- Tendo a entrada sido consumida e a pilha é formada apenas pelo símbolo de partida e o marcador de final de pilha, temos que a palavra é aceita.

Introdução

- Observe que o analisador produz uma derivação mais a direita **invertida**

Introdução

- Porém, como determinar qual ação deve ser executada?

Introdução

- Analisadores ascendentes usam um AFD sobre a pilha para determinar qual ação executar.
- AFD representado por uma tabela que armazena quais ações devem ser executadas pelo analisador.

Introdução

- Diferentes técnicas de análise ascendente usam diferentes estratégias para construção das tabelas.
- Nesta aula, veremos o analisador LR(0)

Analisador LR(0)

Analisador LR(0)

- Analisador sintático ascendente que usa apenas a pilha para decidir ações.
- Não aplicável em gramáticas de linguagens de programação.
 - Útil para compreensão do mecanismo de construção de tabelas.

Analisador LR(0)

- A construção de tabelas LR(0) utiliza o conceito de **item**
 - Item: regra de uma gramática contendo uma marcação em seu lado direito.
 - Marcação representada por um “.”

Analisador LR(0)

- A ideia do algoritmo é construir um AFD sobre coleções de itens.
- Cada estado do AFD representa um conjunto de itens.

Analizador LR(0)

- Como obter o conjunto de itens?
- Primeiro, precisamos modificar a gramática de entrada e calculamos o fechamento e transição entre conjuntos de itens.

Analizador LR(0)

- Fechamento do conjunto de itens I .
 - $I \subseteq closure(I)$.
 - Se $A \rightarrow \alpha.B\beta \in I$, incluir toda regra $B \rightarrow \gamma$ em I .
 - Repetir passo anterior enquanto possível.

Analizador LR(0)

- Outro ponto da construção do autômato LR(0) é o cálculo da função de transição.
- $goto(I, a)$
 - I : conjunto de itens.
 - $a \in \Sigma$.

Analizador LR(0)

- Definimos $goto(I, X)$:
 - $J \leftarrow \emptyset$
 - Para cada item $A \rightarrow \alpha.X\beta \in I$
 - * Adicione $A \rightarrow \alpha X.\beta$ a J .
 - retorne $closure(J)$.

Analizador LR(0)

- Construção da tabela
 - Inicialize T com $\{closure(\{S' \rightarrow .S \$\})\}$.
 - Inicialize E com \emptyset

Analizador LR(0)

- Repita enquanto T e E mudarem
 - para cada $I \in T$
 - * para cada $A \rightarrow \alpha . X \beta \in I$
 - $J = \text{goto}(I, X)$
 - $T = T \cup \{J\}$
 - $E = E \cup \{(I, X, J)\}$

Analizador LR(0)

- Ilustraremos essas etapas considerando a seguinte gramática.

$$\begin{aligned} S &\rightarrow (L) | \mathbf{x} \\ L &\rightarrow L, S | S \end{aligned}$$

Analizador LR(0)

- Algoritmo de análise
- Inicialize a entrada com w\$
- Inicialize a pilha com \$0

Analizador LR(0)

- repita enquanto possível
 - Seja a o 1o token da entrada
 - Seja t o topo da pilha.
 - Se $A[t, a] = \text{shift } n$
 - * Empilhe n
 - * a passa a ser o próximo token

Analizador LR(0)

- se $A[t, a] = \text{reduce } A \rightarrow \gamma$
 - Desempilhe $|\gamma|$ itens.
 - Seja p o topo da pilha
 - empilhe $G[t, A]$

Analizador LR(0)

- Se $A[t, a] = \text{accept}$, aceite senão erro.

Exercícios

Exercícios

- Apresente o conjunto de itens obtidos para a versão estendida da seguinte gramática:
 - Considere como conjunto inicial o formado pela nova variável de partida.

$$E \rightarrow T + E \mid T$$

$$T \rightarrow \mathbf{x}$$