

Compiladores

Análise Sintática Descendente:
Analisador Preditivo Dirigido por Tabela

Gramática LL(1)

Analísadores Preditivos que não precisam de retrocesso podem ser construídos para uma classe de gramáticas chamada LL(1). O primeiro L em LL(1) significa que a cadeia de entrada é percorrida da esquerda para a direita (*Left-to-right*); o segundo L representa derivação mais à esquerda (*leftmost*) e o 1 indica o uso de um símbolo a frente na entrada em cada passo para a tomada de decisão.

Gramática LL(1)

Uma gramática G é LL(1) se e somente se, sempre que $A \rightarrow a \mid b$ forem duas produções distintas de G , e as seguintes condições forem atendidas:

1. para um terminal a , tanto a quanto b não derivam cadeias começando com a
2. No máximo um dos dois, a ou b , pode derivar a cadeia vazia.
3. Se $b \rightarrow^* \varepsilon$, então a não deriva nenhuma cadeia começando com um terminal em $\text{Follow}(A)$. De modo semelhante se $a \rightarrow^* \varepsilon$

Analizador Sintático Descendente Preditivo sem recursão

Composto por :

- Um programa controlador,
- Uma pilha de execução,
- Uma fita de entrada, e
- Uma tabela sintática

Analizador Sintático Descendente Preditivo sem recursão

Funcionamento:

O analisador preditivo dirigido por tabela na configuração inicial tem :

- na pilha o \$ final de cadeia; e
- no topo o símbolo inicial da Gramática.

Em cada iteração age baseando-se no atual token ***a*** da cadeia de entrada e no símbolo **A** no topo da pilha e de acordo com esses executa uma das seguintes ações:

Analizador Sintático Descendente Preditivo sem recursão

- Se A é terminal e $a = A = \$$ então a análise é encerrada com sucesso.
- Se A é terminal e $a = A \neq \$$ então o analisador “casou” o símbolo da entrada com um símbolo no corpo da produção escolhida para a derivação, ele avança para o próximo terminal da entrada,
- Se A é terminal e $a \neq A$ um erro ocorreu e deve ser tratado.

Analizador Sintático Descendente Preditivo sem recursão

- Se A é não terminal então o controlador consulta a tabela sintática em $M[A,a]$ que deve conter:
 - Uma produção do tipo $A \rightarrow \gamma_1\gamma_2\gamma_3$: que determina que A seja desempilhado e que $\gamma_1\gamma_2\gamma_3$ seja empilhado de trás para frente, deixando γ_3 no topo; ou
 - Uma produção do tipo $A \rightarrow \varepsilon$: que determina que A seja desempilhado; ou
 - Nada : o que se trata de um erro sintático e deve ser tratado.

Exemplo de Funcionamento

Para a gramática

$E \rightarrow TE'$

$E' \rightarrow +TE' \mid e$

$T \rightarrow FT'$

$T' \rightarrow *FT' \mid e$

$F \rightarrow (E) \mid id$

Temos a seguinte tabela sintática

	id	+	*	(	)	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow \varepsilon$	$E' \rightarrow \varepsilon$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow \varepsilon$	$T' \rightarrow *FT'$		$T' \rightarrow \varepsilon$	$T' \rightarrow \varepsilon$
F	$F \rightarrow id$			$F \rightarrow (E)$		

Exemplo

Pilha	Cadeia Entrada	Ação
\$ E	Id + id *id \$	$E \rightarrow TE'$
\$ E' T	Id + id *id \$	$T \rightarrow FT'$
\$ E' T' F	Id + id *id \$	$F \rightarrow id$
\$ E' T' id	Id + id *id \$	casou id
\$ E' T'	+ id *id \$	$T' \rightarrow \varepsilon$
\$ E'	+ id *id \$	$E' \rightarrow +TE'$
\$ E' T +	+ id *id \$	casou +
\$ E' T	id *id \$	$T \rightarrow FT'$
\$ E' T' F	id *id \$	$F \rightarrow id$
\$ E' T' id	id *id \$	casou id
\$ E' T'	*id \$	$T' \rightarrow *FT'$
\$ E' T' F *	*id \$	casou *
\$ E' T' F	id \$	$F \rightarrow id$
\$ E' T' id	id \$	casou id
\$ E' T'	\$	$T' \rightarrow \varepsilon$
\$ E'	\$	$E' \rightarrow \varepsilon$
\$	\$	sucesso

Construção da Tabela Sintática

Para cada produção $A \rightarrow \alpha$ da gramática faça

Para cada terminal a no $\text{First}(\alpha)$ faça

$$M[A, a] = A \rightarrow \alpha$$

Se ε pertence ao $\text{First}(\alpha)$ então

Para cada terminal b no $\text{Follow}(A)$ faça

$$M[A, b] = A \rightarrow \alpha$$

Se ε pertence ao $\text{First}(\alpha)$ e

$\$$ pertence ao $\text{Follow}(A)$ então

$$M[A, \$] = A \rightarrow \alpha$$

Se, depois de realizar todos esses passos, não houver produção alguma em $M[A, a]$, então defina $M[A, a]$ como erro (que normalmente é representado por uma entrada vazia na tabela)

Tratamento de Erro

Um erro é detectado no analisador sintático descendente preditivo sem recursão quando o terminal no topo da pilha A é diferente do terminal atual na cadeia de entrada a ; ou quando ao consultar a tabela sintática $M[A,a]$ o parser encontra uma entrada vazia.

Existem duas técnicas para construir o tratamento do erro:

- Modo Pânico; e
- Recuperação em Nível de Frase

Tratamento de Erro: Modo Pânico

A ideia é ignorar símbolos da entrada até encontrar um token no conjunto de tokens de sincronismo.

Sua eficácia depende da escolha do conjunto de tokens de sincronismo, que deve permitir que o analisador se recupere rapidamente dos erros.

Algumas das heurísticas para criação do conjunto de tokens de sincronismo são:

Tratamento de Erro: Modo Pânico

1. Incluir todos os símbolos do $\text{Follow}(A)$ no conjunto de sincronização de A
2. Adicionar os símbolos que iniciam construções sintáticas de um nível superior, além do $\text{Follow}(A)$
3. Incluir o $\text{First}(A)$ no conjunto de sincronização
4. Se um não-terminal A produz a cadeia vazia, então a produção $A \rightarrow \varepsilon$ pode ser usada com padrão e adiar a detecção do erro.
5. Se o terminal A no topo da pilha não casa com o token da entrada, uma solução simples é desempilhar o símbolo e emitir uma mensagem. Na verdade neste caso estamos assumindo que todos os outros tokens estão no conjunto de sincronização.

Tratamento de Erro: Modo Pânico

A recuperação de erro no modo pânico não resolve o problema das mensagens de erro, que devem ser precisas ao descrever o erro encontrado e indicar o local onde ocorre o erro.

Tratamento de Erro: em Nível de Frase

É implementado preenchendo cada entrada vazia da tabela com um apontador para uma rotina de tratamento do erro.

Essas rotinas podem :

- substituir, inserir e remover símbolos da cadeia de entrada e emitir mensagens de erros apropriadas.
- remover símbolos da pilha. (Não é recomendável inserir ou substituir símbolos na pilha pois isto pode levar a uma construção que não existe na gramática)