



Construção de Compiladores

Alterado por: prof. Marlon Oliveira
Elaborado por: Prof. Ricardo Silveira

Tratamento de Erros

- Se um compilador tivesse que processar somente programas corretos, seu projeto e sua implementação seriam grandemente simplificados.
- Mas os programadores frequentemente escrevem programas incorretos, e um bom compilador deveria assistir o programador na identificação e localização de erros.
- É gritante que, apesar dos erros serem lugar-comum, poucas linguagens sejam projetadas tendo-se o tratamento de erros em mente.
- A maioria das especificações das linguagens de programação não descreve como um compilador deveria responder aos erros;
- Tal tarefa é deixada para o projetista do compilador.

Tratamento de Erros

Felizmente, a maioria dos erros são simples

Pesquisa com estudantes de Pascal:

- 80% dos enunciados contém apenas um erro; 3% tem dois
- 90% são erros em um único token
- 60% são erros de pontuação, ex.: uso do ponto e vírgula (;)
- 20% são erros de operadores e operandos, ex.: omissão de :
no símbolo :=
- 15% são erros de palavras-chave, ex: erros ortográficos
(wrteln)

Tratamento de Erros

- Em muitos compiladores, ao encontrar uma construção mal formada o erro é reportado e a tarefa da Análise Sintática é dada como concluída.
- Na prática, o compilador pode, e até deve, reportar o erro e tentar continuar a Análise Sintática para detectar outros erros e assim diminuir a necessidade de recomeçar a compilação a cada relato de erro.
- A realização efetiva do tratamento de erros pode ser uma tarefa difícil.
- O tratamento inadequado de erros pode introduzir uma avalanche de erros que não foram cometidos pelo programador, mas pelo tratamento de erros realizado.

Tratamento de Erros

- Na recuperação de erros, tenta-se colocar o analisador em um estado tal que o restante da sentença de entrada ainda possa ser analisada. Este processo pode envolver: a modificação da entrada (excluir tokens), a modificação da pilha, ou de ambos.

Tipos de Erros

- Léxicos
 - Errar a grafia de um identificador, palavra-chave ou operador.
 - Usar um caracter invalido Ex: pon#eiro = 5;
- Sintáticos
 - Uma expressão aritmética com parênteses não-balanceados, inserção de um símbolo inválido para a estrutura. Ex:
result=var++var2
- Semânticos
 - Um operador aplicado a um operando incompatível. Ex:
boolean i; i=9/6;
- Lógicos
 - Uma chamada infinitamente recursiva.

Comportamento

- Pane
 - travar a máquina OU gerar código objeto errado OU abortar a compilação
- Notificação
 - abortar a compilação e emitir mensagem de erro
- Recuperação
 - acusar o erro da forma mais precisa possível e recuperar-se para poder continuar a compilação até o fim.

Comportamento

- Reparar
 - reparar o erro, transformando um símbolo de entrada errado em algo similar que seja aceitável (ex.: ao ver $2 = 3 * 4$ acusa o erro de atribuição inválida, substitui o 2 por um identificador virtual e continua a análise)
- Corrigir
 - prever (adivinhar) o objetivo do usuário de modo a continuar a compilação e geração de código para produzir um módulo executável (onde alguns erros de execução possam ser observados). Por exemplo, ao ver $IF\ A > B\ C = D$, acusa o erro de falta de THEN, insere o símbolo automaticamente e segue em frente.

Tratamento de Erros

Sabemos que ocorre um erro quando:

- o símbolo corrente da entrada não corresponde ao *terminal* contido no topo da pilha
OU
- o símbolo corrente da entrada não possui produção correspondente a partir do *não-terminal* contido no topo da pilha.

Tipos de recuperação

- Modo Pânico: forma mais simples de detecção na qual o analisador lê e descarta símbolos da entrada até encontrar um token de sincronização;
- Modo Recuperação Local: recupera o erro fazendo alterações sobre um símbolo, desprezando um token de entrada, ou substituindo-o por outro, ou inserindo um novo token, ou ainda removendo um símbolo da pilha.

Modalidade Pânico

A eficiência da recuperação de erros desta forma depende da escolha de um conjunto adequado de *tokens* de sincronização.

- Não existem regras formais para tal escolha que pode ser baseada em técnicas heurísticas.
- A recuperação de erros é provável na maioria dos casos embora não possa ser assegurada completamente.

Modalidade Pânico

Considerando a ocorrência de um erro durante a expansão do *não-terminal* A:

Técnica 1:

- Como *tokens* de sincronizacao usam-se todos os simbolos em FOLLOW(A).
- Descartam-se os símbolos de entrada até encontrar-se um elemento de FOLLOW(A), quando também descartamos A da pilha
- É provável que a análise sintática possa prosseguir.

Exemplo

Dada a gramática:

$$S \rightarrow aAcd \mid bAef$$
$$A \rightarrow gh$$

e a sentença *agabcd*

- O conjunto $\text{FOLLOW}(A) = \{c, e\}$
- Uma chamada ao método *S* consome *a* da entrada
- Uma chamada ao método *A* consome o *g*
- O método *A* tenta achar na entrada um *h*, mas não encontra, acusando um erro sintático.
- O método *A* deve consumir todos os caracteres da entrada até encontrar um membro de $\text{FOLLOW}(A)$ quando então retorna para o método *S* e consome *c*

Modalidade Pânico

Técnica 2:

- Como somente os símbolos em FOLLOW(A) nem sempre são suficientes para a sincronização, adicionam-se os símbolos FIRST das produções que se expandem em A.
- Descartam-se os símbolos de entrada até encontrar-se do conjunto de sincronização, quando também descartamos A.
- É provável que a análise sintática possa prosseguir.

Modalidade Pânico

Técnica 3:

- Se os símbolos em $\text{FIRST}(A)$ são adicionados ao conjunto de sincronização, poderá ser possível retomar a análise a partir de A se um símbolo que figura em $\text{FIRST}(A)$ estiver na entrada.

Técnica 4:

- Se o não-terminal A puder gerar ϵ , tal produção pode ser usada como *default*. Pode-se assim adiar a detecção do erro sem haver possibilidade de perdê-lo.

Modalidade Pânico

Técnica 5:

- Se um terminal no topo da pilha não pode ser reconhecido, pode-se removê-lo. Sugere-se a emissão de mensagem de aviso nestes casos.
- A análise provavelmente poderá prosseguir.
- Tal enfoque corresponde a adicionar todos os *tokens* possíveis como símbolos do conjunto de sincronização.

Tratamento de Erros

- Toda e qualquer tentativa de recuperação de erros não deve provocar um laço infinito na análise sintática da entrada. Uma forma de proteção e assegurar-se que a pilha foi encurtada após a ação de correção.
- É sempre questionável a inserção ou alteração de símbolos na pilha quando tais operações “criam” construções inexistentes na linguagem sendo analisada.

Mensagem de Erros

- Todas as mensagens de erro produzidas por um compilador devem ser informativas o suficiente para permitir a rápida localização e correção do erro.
- Sugere-se que cada entrada vazia da tabela sintática preditiva contenha apontadores para rotinas de tratamento de erro onde mensagens apropriadas serão fornecidas conforme o erro que as provocou.