Análise léxica

O papel do analisador léxico

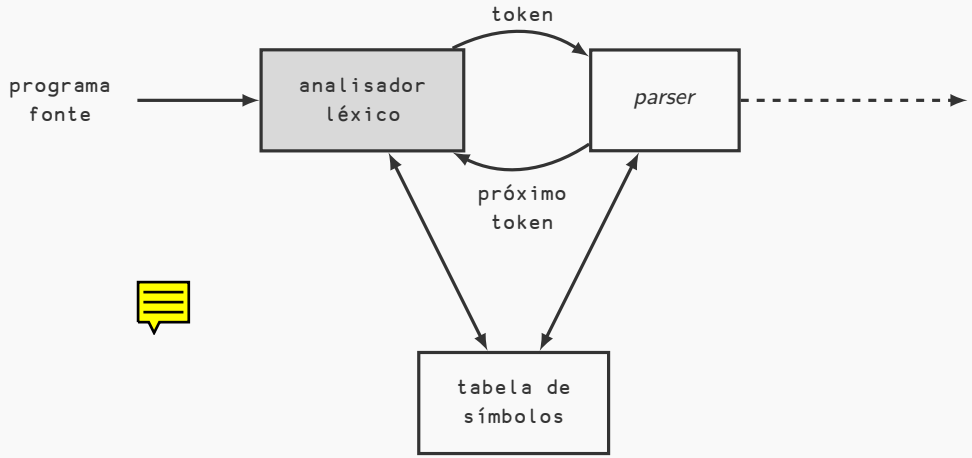
Prof. Edson Alves

Faculdade UnB Gama

Analisador léxico

- A análise léxica é a primeira fase de um compilador
- Um analisador léxico deve ler os caracteres da entrada e produzir uma sequência de tokens, os quais serão usados pelo *parser* durante a análise sintática
- Uma forma de se construir um analisador léxico é escrever um diagrama que ilustre a estrutura dos tokens da linguagem fonte e o traduzir manualmente em um programa que os identifique
- As técnicas de construção de um analisador léxico podem ser utilizadas em outras áreas
- Como o analisador léxico é responsável pela leitura do programa fonte, ele pode também realizar tarefas secundárias a nível de interface com o usuário, como a remoção e espaços e comentários, por exemplo

Interação entre o analisador léxico e o *parser*



Separação entre a análise léxica e a análise gramatical

Há quatro principais motivos para se separar a análise léxica da análise gramatical (*parsing*):

1. A separação entre estas duas fases pode simplificar uma das duas (ou ambas)
2. A eficiência do compilador é melhorada, uma vez que a separação permite o uso de técnicas especializadas, como buferização, para melhorar o desempenho da leitura da entrada e extração de tokens
3. A separação permite uma melhor portabilidade do compilador, uma vez que diferenças entre a captura da entrada e codificação de caracteres, em diferentes plataformas, podem ser tratadas de forma isolada na análise léxica
4. A separação entre as fases permite a criação de ferramentas especilizadas para a automação da construção de analisadores léxicos e de *parsers*

Tokens, padrões e lexemas

- Tokens, padrões e lexemas são conceitos correlacionados e onipresentes na análise léxica
- Token é um símbolo terminal da gramática da linguagem fonte (em geral, grafados em negrito)
- Nas maioria das linguagens de programação, são tokens: palavras-chave, operadores, identificadores, constantes, pontuações, etc
- Um lexema é um conjunto de caracteres que é reconhecido como um token
- Um mesmo token pode ser representado por lexemas distintos (por exemplo, 1 e 42 são lexemas distintos para o token `NUM`)
- Um padrão descreve o conjunto de lexemas que podem representar um token em particular

Atributos para tokens

- Quando dois ou mais lexemas estão associados a um mesmo token, o analisador léxico deve prover informações adicionais para as fases subsequentes, para que elas possam distinguí-los
- Estas informações são os atributos do token
- Deste modo, o analisador léxico deve identificar os tokens e seus respectivos atributos, caso existam
- Os tokens influenciam as decisões da análise gramatical
- Os atributos influenciam a tradução dos tokens
- Em tokens numéricos, o valor do número representado pelo lexema pode ser o atributo token
- No caso de identificadores, o próprio lexema pode ser o atributo do token

Erros léxicos

- Determinados erros não podem ser detectados em nível léxico
- Por exemplo, na expressão em C++

```
fi (a == f(x)) {
    ...
}
```

  o analisador léxico identificaria `fi` como um identificador válido, e só na análise gramatical é que seria detectado o erro de digitação da palavra-chave `if`
- Os erros léxicos mais comuns são aqueles onde o analisador léxico não consegue associar o prefixo lido a nenhum dos padrões associados aos tokens da linguagem
- Neste ponto, o analisador léxico pode abordar a leitura, emitindo uma mensagem de erro
- Outra alternativa é tentar tratar o erro de alguma maneira

Ações de recuperação de erros

Há quatro ações que configuram tentativas de recuperação de erros léxicos:

- remover um caractere estranho da entrada
- inserir um caractere ausente
- substituir um dos caracteres incorretos por um caractere correto
- transpor dois caracteres adjacentes

Se uma ou mais ações conseguem tornar o prefixo em um token válido, o analisador podem indicar ao usuário a sequência de ações como sugestão de correção do programa fonte, ou mesmo prosseguir assumindo esta correção.

Referências

1. **AHO**, Alfred V, **SETHI**, Ravi, **ULLMAN**, Jeffrey D. *Compiladores: Princípios, Técnicas e Ferramentas*, LTC Editora, 1995.
2. GeeksForGeeks. Flex (Fast Lexical Analyzer Generator), acesso em 04/06/2022.