

Atividades Práticas

Análise léxica da linguagem *Torben*

Departamento de Computação
Universidade Federal de Ouro Preto
Prof. José Romildo Malaquias
13 de abril de 2015

Resumo

Nesta atividade vamos implementar um analisador léxico para uma pequena linguagem de programação usando um gerador de analisador léxico.

Sumário

1	A linguagem <i>Torben</i>	1
2	Atividades	2

1 A linguagem *Torben*

Torben Ægidius Mogensen apresenta no capítulo 4 de seu livro *Introduction to Compiler Design* uma pequena linguagem de programação para fins didáticos.

Vamos considerar uma linguagem, que chamaremos de *Torben*, baseada nessa linguagem, porém estendida para incluir:

- comentários de linha,
- literais booleanos,
- o tipo `string`,
- literais `string`,
- operadores aritméticos - (subtração), `*` (multiplicação) e `/` (divisão),
- operadores relacionais `<>` (diferente), `>` (maior que), `>=` (maior ou igual a), `<` (menor que), e `<=` (menor ou igual a), e
- operadores lógicos `&&` (e lógico) e `||` (ou lógico).

```

Program → Funs
Funs → Fun
Funs → Fun Funs
Fun → TypeId ( TypeIds ) = Exp
TypeId → bool id
TypeId → int id
TypeId → string id
TypeIds →
TypeIds → TypeId TypeIdsRest
TypeIdsRest →
TypeIdsRest → , TypeId TypeIdsRest
Exp → litbool
Exp → litnum
Exp → litstring
Exp → id
Exp → Exp + Exp
Exp → Exp - Exp
Exp → Exp * Exp
Exp → Exp / Exp
Exp → Exp = Exp
Exp → Exp <> Exp
Exp → Exp > Exp
Exp → Exp >= Exp
Exp → Exp < Exp
Exp → Exp <= Exp
Exp → Exp && Exp
Exp → Exp || Exp
Exp → if Exp then Exp else Exp
Exp → id ( Exps )
Exp → let id = Exp in Exp
Exps →
Exps → Exp ExpsRest
ExpsRest →
ExpsRest → , Exp ExpsRest

```

Comentários de linha são iniciados com a sequência de caracteres `//`.

Os identificadores são sequências de letras maiúsculas ou minúsculas, dígitos decimais e sublinhados (`_`), começando com uma letra minúscula.

Os literais numéricos são formados por uma sequência de um ou mais dígitos decimais opcionalmente precedidos dos símbolos `+` ou `-`.

Os literais booleanos são `#t` (verdadeiro) e `#f` (falso).

Os literais string são formados por uma sequência de caracteres delimitados por aspas (`"`). Na sequência de caracteres o caracter `\` é especial e inicia uma sequência de escape. As únicas sequência de escape válidas são:

- `\\` (correspondente ao caracter `\`), e
- `\"` (correspondente ao caracter `"`).

2 Atividades

Tarefa 1: Implementação

Implemente um analisador léxico usando um gerador de analisador léxico para a linguagem *Torben*.

A sua aplicação deverá aceitar o nome do arquivo a ser analisado como argumento na linha de comando, e exibir a sequência de *tokens* obtidas pela análise léxica do arquivo.

Para cada **token** o seu analisador léxico deverá informar:

- o tipo do **token**,
- o valor semântico do **token**, quando relevante, e
- a posição (número da linha e coluna) em que o **token** aparece no programa fonte.

Todos os possíveis erros léxicos devem ser reportados corretamente pelo seu analisador léxico. Ao reportar um erro, deve-se exibir a posição (linha e coluna) em que o erro foi detectado, e uma mensagem de diagnóstico.

Por exemplo, a análise léxica do programa fonte seguinte:

```
int fatorial(int n) =  
  if n > 0 then  
    1  
  else  
    n * fatorial(n-1)  
  
int main() =  
  print_int(fatorial(7))
```

produz a lista de símbolos léxicos:

Tarefa 2: Teste

Teste o seu analisador léxico com os programas fontes seguintes.

1.

```
int fatorial(int n) =  
  if n > 0 then  
    1  
  else  
    n * fatorial(n-1)  
  
int main() =  
  print_int(fatorial(7))
```

2.

```
bool f(string s) =  
  s = "pedro" || #t
```

3.

```
int main() =  
  let a = 873 in  
    let b = a ^ 3 in  
      (a + b)/2
```

4.

```
int main() =  
  // program execution starts here  
  let PesoPessoa := 45 in  
    print_int(PesoPessoa + 2)
```

5.

```
int main() =  
    print(-2342,  
        56.7,  
        "aspas\"internas\" e \\ barra",  
        "ouro \n preto",  
        "bom dia)
```