

Descrição da Linguagem DECAF

Eduardo Ferreira dos Santos¹

¹Centro Universitário UniCEUB
QS 1, Lote 1/17, Rua 214
Taguatinga, Brasília — DF, CEP 72025-120

eduardo.santos@uniceub.br

Resumo. *O documento trata da tradução do manual da linguagem chamada Decaf [Amarasinghe and Rinard 2010]. Trata-se de uma linguagem imperativa simples similar a C ou Pascal.*

Abstract. *The project for the course is to write a compiler for a language called Decaf. Decaf is a simple imperative language similar to C or Pascal [Amarasinghe and Rinard 2010].*

Considerações Léxicas

Todas as palavras reservadas em Decaf são em caixa baixa (minúsculas). As palavras reservadas e identificadores são *case sensitive*: enquanto `if` é uma palavra reservada, `IF` é um nome de variável; `foo` e `Foo` são dois nomes diferentes que se referem a duas variáveis distintas.

Algumas considerações sobre a análise léxica:

- As palavras reservadas para a linguagem são:

```
boolean break callout class continue else false for
int return true void
```

- `Program` não é uma palavra reservada, mas um identificador com significado especial, dependendo da circunstância.
- Os comentários começam com `//` e terminam no fim da linha.
- O espaço em branco (*white space*) pode aparecer entre quaisquer *tokens*. É definido como:

- Um ou mais espaços;
 - *Tabs*;
 - Caracteres de quebra de página e de linha;
 - Comentários.
- Palavras reservadas e identificadores devem ser separados ou por espaço em branco, ou um *token* que não seja uma palavra reservada, ou um identificador. Por exemplo: `thisfortrue` representa um único identificador, e não três palavras reservadas distintas. Se a palavra começa com um caractere do alfabeto ou *underscore* (`_`), então a maior sequência de caracteres seguidos forma um *token*.
 - Literais (*string literals*) são compostos por `<char>` entre aspas duplas. Um caractere (*char literal*) é um `<char>` entre aspas simples;
 - Os números em Decaf são assinados em 32 bits (*32 bits signed*), ou seja, valores inteiros entre -2147483648 e 2147483647. Se uma palavra começa com `0x`, esses dois primeiros caracteres e a maior sequência na lista `[0-9a-fA-F]` formam um número inteiro hexadecimal. Se a palavra começa com um dígito decimal (que não seja `0x`), o maior prefixo de dígitos decimais formam um inteiro decimal. Perceba que a validação do *range* será realizada depois: uma sequência longa de dígitos como `123456789123456789` ainda é reconhecida como um único *token*.
 - Um tipo `<char>` é qualquer caractere da tabela ASCII (valores decimais entre 32 e 126, ou 40 e 176 em octal) que não seja aspas duplas (`"`), aspas simples (`'`), contrabarra (`\`), mais as sequências de dois caracteres para do tipo `"\""` para representar aspas duplas, `"\'"` para representar aspas simples, `"\\"` para representar a contrabarra `"\t"` para representar o `tab` e `"\n"` para representar uma quebra de linha.

Referências

Amarasinghe, S. and Rinard, M. (2010). Computer language engineering. Disponível em <http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/>

6-035-computer-language-engineering-spring-2010/ Acessado
em 02/08/2016.