

Compiladores

Linguagem *setlang*

Departamento de Electrónica, Telecomunicações e Informática
Universidade de Aveiro

Junho de 2024

Objectivos

O objectivo geral deste trabalho é o desenvolvimento de uma linguagem de programação compilada – i.e. que crie programas numa linguagem de programação genérica (Java, C++, Python, ...) – que permita a manipulação de conjuntos¹ de elementos de vários tipos (inteiros, reais, texto).

Esta linguagem deve poder manipular números inteiros, números reais, texto e conjuntos. Um conjunto é declarado como contendo elementos de um determinado tipo (ver exemplos).

Por exemplo:

```
— line comments
— ; is a mandatory command separator

set<integer> s1 := {1,2,3};

— basic types: integer, real, text
— set types: set<basic-type>

writeln "Set: " + s1;
integer i := 5;
writeln "Another: " + {i*3//2-3, i};

— normal arithmetic operators (with standard precedences):
— — real operations: *, /, +, -
— — integer operations: *, //, \, +, - (integer division: // and \)
— — unary arithmetic operators: +, -
— — text operation: + (concatenation)
— set operations: + (set union), - (set difference), & (set intersection),
— — # (result: integer with then number of set elements)

set<text> st; — st undefined
st := set<text>(read "set: ");
writeln "Set: " + st
```

A descoberta da sintaxe desta linguagem deve ser feita recorrendo os programas de exemplo.

¹Relembra-se que os conjuntos não têm uma ordem nem repetem elementos.

Características da solução

Apresentam-se a seguir um conjunto de características que a solução desenvolvida pode ou deve contemplar. Essas características estão classificadas a 2 níveis:

- mínima – característica que a solução tem obrigatoriamente que implementar;
- desejável – característica não obrigatória, mas fortemente desejável que seja implementada pela solução (apenas considerada se as mínimas forem cumpridas);

Características mínimas

O exemplo `p1.s1` indica algum código fonte que tem de ser aceite (e devidamente compilado) pela linguagem a desenvolver.

A linguagem deve implementar:

- Os tipos de dados básicos inteiro, texto e real.
- O tipo de dados conjunto com elementos dum determinado tipo básico.
- Aceitar expressões aritméticas standard para os tipos de dados (ver comentários no exemplo).
- Aceitar operadores sobre conjuntos (ver comentários no exemplo).
- Instrução de escrita no *standard output*.
- Instrução de leitura de texto a partir do *standard input*.
- Operadores de conversão entre tipos de dados (por exemplo, `text(10)` para converter para texto; ou `integer("10")` para converter para inteiro).
- Verificação semântica do sistema de tipos.

Características desejáveis

O exemplo `p2.s1` indica algum código fonte que se enquadra nas características desejáveis.

- O tipo de dados básico booleano (`boolean`, com os valores literais `true` e `false`).
- Permitir a definição de expressões booleanas (predicados) contendo, pelo menos relações de ordem e operadores booleanos (conjunção, disjunção, negação, igual, maior, ...).
- Incluir a instrução condicional (operando sobre expressões booleanas).
- Incluir uma instrução iterativa (ver exemplo).