Compiladores

Linguagem *pdraw*

Departamento de Electrónica, Telecomunicações e Informática
Universidade de Aveiro

Abril 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 o desenho de imagens, usando como abstracção o desenho livre com canetas/marcadores.

A ideia segue a abstracção inerente à biblioteca *turtle* do python. Assim, temos uma ou mais canetas que podem estar levantadas ou pousadas sobre o 'papel', cada uma com uma orientação (i.e. estão 'viradas' num determinado sentido) e que podem ser movimentadas (se pousadas, irão desenhar) para a frente, ou trás, e ao quais se pode mudar as suas propriedades (orientação, cor, espessura, padrão, etc.). A espessura do traço da caneta é determinado por uma propriedade e pela 'pressão' exercida sobre o papel (pressão zero, traço de 1 pixel, pressão de 1 traço com o valor da propriedade).

Por exemplo, podemos definir o seguinte tipo de caneta:

```
define pen PenType1 {
   color = green;
   position = (10,10);
   orientation = 45°; % literal orientation in degrees (internally is always stored in radians)
   % orientation = 0; % literal orientation in radians
   % real PI constant predefined
   thickness = 10; % default thickness is 1
   pressure = -1; % up (pressure=-1); down (pressure>=0 and pressure <=1)
   % pressure = 0; % down thickness 1
   % pressure = 1/3; % down 1/3 of thickness
}
```

Depois podemos executar o seguinte desenho:

```
pen p1 = new PenType1; % variable p1 can refer to any pen (or none if no creation/assignment involved)
p1 down; % pen in canvas with pressure 0
% draw square
p1 forward 10; % pen p1 advances 10
p1 left 90°; % pen p1 rotates 90° counterclockwise
p1 forward 10;
p1 left 90°;
```

```
p1 forward 10;
p1 left 90°;
p1 forward 10;
p1 left 90°;
```

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

A linguagem secundária (interpretada) vai ser uma versão interpretada simplificada da linguagem principal. Nesta linguagem, como em turtle, existe apenas uma caneta (implícita).

Como exemplo de um programa simples nesta linguagem:

```
position (10,10); % set pen position
down;
forward 10;
left 45°;
backward 5;
```

Características da solução

Apresentam-se a seguir um conjunto de características que a solução desenvolvida pode ou deve contemplar. Essas características estão classificadas a 3 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);
- avançada – característica adicional apenas considerada para avaliação se as obrigatórias e as desejáveis tiverem sido contempladas na solução.

Características mínimas

Os exemplos `p1.pdraw`, `p2.pdraw` e `p1.ipdraw` indicam algum código fonte que tem de ser aceite (e devidamente compilado e interpretado) pelas linguagens a desenvolver.

A linguagem deve implementar:

- Instrução para definir os atributos de um tipo de caneta: **pen**.

  Cada definição de caneta tem de ter uma identificação única (no programa).
- Os tipo de de dados inteiro, string, real e ponto.
- Aceitar expressões aritméticas standard para os tipos de dados numéricos. Deve também aceitar a operação de concatenação de texto (sem nenhum operador, isto é, dois textos seguidos correspondem à sua concatenação).

- 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, **string**(10) para converter para texto; ou **int**("10") para converter para inteiro).
- Instruções para movimentar e rodar canetas.
- Instruções para mudar atributos de canetas.
- Instrução de pausa (valor em microssegundos).
- Verificação semântica do sistema de tipos.

A linguagem interpretada deve aceitar os programas `ipdraw` acima referidos. Deve também incluir:

- A definição de expressões booleanas (predicados) contendo, pelo menos relações de ordem e operadores booleanos (conjunção, disjunção, etc.).
- Instrução condicional.
- Instrução de iteração (loop).

Características desejáveis

O exemplo `p3.pdraw` indica algum código fonte que se enquadra nas características desejáveis.

- Tratar todas as operações sobre canetas (movimento, rotação, ...) assim como a instrução de pausa como parte integrante de uma expressão, permitindo múltiplas operações na mesma instrução.
- Permitir a definição de expressões booleanas (predicados) contendo, pelo menos relações de ordem e operadores booleanos (conjunção, disjunção, etc.).
- Incluir a instrução condicional (operando sobre expressões booleanas).
- Incluir instruções iterativas tipo **for** e tipo **while**/**until** (operando sobre expressões booleanas).
- Aceitar vários canvas (com instrução de selecção do canvas activo). Os atributos das canetas aplicam-se a qualquer canvas (i.e. não são especificas para cada um).

Características avançadas

- Implementar funções e variáveis locais às mesmas.
- Implementar uma tabela de símbolos que resolva o problema dos contextos de declaração.
- ...