

Linguagens Formais e Autómatos / Compiladores

Linguagem *quiz*

Departamento de Electrónica, Telecomunicações e Informática
Universidade de Aveiro

abril de 2021

Objetivos

O objetivo geral deste trabalho é a criação de um ambiente de programação que permita a criação de programas numa linguagem de programação genérica (Java, C++, Python, ...) que, quando executados, realizem um questionário interativo.

Este objetivo geral pode ser decomposto em 4 componentes:

- Definição da linguagem principal *quiz* que permita construir questionários interativos.
- Construção do compilador que permite transformar uma descrição nessa linguagem num programa na linguagem genérica.
- Definição de uma linguagem secundária que permita definir bancos de perguntas.
- Dotação da linguagem principal de meios para extrair perguntas de bancos de perguntas.

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:

- obrigatória – característica que a solução tem obrigatoriamente que implementar;
- desejável – característica não obrigatória, mas fortemente desejável que seja implementada pela solução;
- opcional – característica adicional apenas considerada para avaliação se as obrigatórias e as desejáveis tiverem sido contempladas na solução.

Características obrigatórias

- O conceito de pergunta deve ser nativo à linguagem, com a existência de um tipo de dados primitivo que o represente (por exemplo, `question`).
- Além do tipo `question` já referido, a linguagem deve suportar nativamente outros tipos de dados, nomeadamente `number` e `text`.
- O tipo de pergunta ‘escolha múltipla’ (`multiple-choice`) deve ser suportado pela linguagem.
 - podendo ter um número variável de propostas de resposta;
 - podendo ter indicação da resposta certa para poder ser usada em questionários com avaliação automática;
 - podendo alterar-se aleatoriamente a ordem das propostas de resposta.
- O tipo de pergunta ‘correspondência’ (`matching`) deve ser suportado pela linguagem.
 - podendo ter um número diferente de elementos nos dois conjuntos;
 - podendo ter indicação dos pares válidos para poder ser usada em questionários com avaliação automática;
 - podendo alterar-se aleatoriamente a ordem dos elementos dos dois conjuntos.
- A linguagem auxiliar usada para a escrita de bancos de perguntas deve ser simples e diferente da linguagem principal. Sobre o seu uso na linguagem principal, considere o seguinte:
 - As perguntas extraídas de bancos de perguntas têm de ser embutidas no questionário. Durante a execução de um questionário os bancos de perguntas não estão disponíveis. Apenas o estão durante a compilação.
 - Deve ser possível criar-se um programa *quiz* sem se usar bancos de perguntas.
 - Deve ser possível a extração de perguntas de bancos de perguntas por ID. Isto significa que as perguntas nos bancos de perguntas têm de ter um ID único.
 - As perguntas nos bancos de perguntas não têm cotação.
- Deve haver uma separação clara entre a definição de perguntas e a sua apresentação ao respondedor e a respetiva recolha de respostas. Relativamente à apresentação e recolha de respostas:
 - Todas as respostas dadas pelo respondedor do questionário têm de ser guardadas, de forma a se produzir um relatório final.
 - Cada tipo de pergunta contemplada pela solução deve ter um modo de apresentação pré-definido.

Características desejáveis

- Deve suportar tipo de pergunta com ‘resposta curta’ (`short-answer`), constituída por poucas palavras, eventualmente apenas uma.
 - podendo ter indicação da resposta certa (por exemplo, através de uma expressão regular), para poder ser usada em questionários com avaliação automática;
- Deve suportar tipo de pergunta ‘resposta numérica’ (`numeric-answer`), com validação sintática da resposta,
 - podendo ter indicação da resposta certa, para poder ser usada em questionários com avaliação automática.
- Deve ser possível alterar o modo pré-definido de apresentação de uma pergunta.
- Deve suportar listas.
 - A separação, já referida, entre declaração e apresentação de perguntas é adequada ao uso de listas, de forma a se poder iterar sobre listas de perguntas criadas previamente.
 - Deve possuir o operador unário `shuffle`, que realize a baralhamento dos elementos de uma lista. Além do baralhamento das propostas de respostas das perguntas de escolha múltipla, pode ser interessante para a apresentação de perguntas de forma baralhada.
- Deve suportar o tipo de dados `time`, que deve aceitar um sufixo com as unidades.
 - Deve permitir a adição e subtração de tempos com unidades diferentes.
 - Deve permitir a multiplicação e divisão de tempos com grandezas adimensionais.
- Deve haver a possibilidade de definir grupos de perguntas. Nesse caso:
 - Deve permitir que o respondedor navegue entre as perguntas de um grupo.

Características opcionais

- O tipo de pergunta com ‘resposta longa’ (`essay`) deve ser suportado pela linguagem.
- O questionário pode ser de avaliação automática. Neste caso:
 - As perguntas têm de ter cotação e as respostas válidas têm de ser conhecidas.
 - Deve possuir um mecanismo (através de fórmula, porventura) que permita calcular a classificação, desejavelmente com suporte para expressões condicionais.
- Deve ser possível a extração de perguntas de um banco de perguntas por tópico/tema. Isto significa que as perguntas nos bancos de perguntas têm de ter uma categorização de acordo com uma abordagem qualquer.

- Tendo sido introduzido a noção de tempo, podem incorporar *timeout*:
 - no tempo de resposta a uma pergunta;
 - no tempo de resposta a um grupo de perguntas;
 - no tempo de resposta ao questionário como um todo.
- Podem ser considerados mecanismos de segurança:
 - As perguntas embutidas num questionário podem estar cifradas.
 - A execução de um questionário pode exigir credenciais de arranque.
- Deve permitir que se responda a K perguntas num grupo de N ($K < N$).
- Podem ser considerados mecanismos dependência entre perguntas:
 - permitindo, por exemplo, apenas avançar para uma pergunta se se acertar noutra;
 - permitindo repetir até acertar, registando as iterações.

Desafios

Consegue escrever um programa na linguagem *quiz* que criou que defina um questionário que comece por pedir ao respondedor para escolher 1 tema entre 3 e a seguir faça 4 perguntas sobre esse tema, 2 de escolha múltipla e 2 de correspondência?