

Compiladores

Trabalho Prático

Departamento de Electrónica, Telecomunicações e Informática
Universidade de Aveiro

2021–2022, 2^o semestre

1 Introdução

Um compilador pode ser encarado como sendo um *tradutor* da linguagem fonte (i.e. da linguagem a compilar) para uma linguagem destino. A linguagem destino pode ser próxima da linguagem fonte, ou muito distante (por exemplo, *assembly* ou linguagem máquina). Neste processo de *tradução* o compilador deve não só garantir a validade sintáctica do programa, como também a sua correcção semântica (i.e. tanto quanto possível, uma utilização com significado das instruções da linguagem).

2 Objectivos

O trabalho a desenvolver deve envolver, pelo menos, duas linguagens: uma para um compilador (que será a principal linguagem do trabalho) e outra para ler informação estruturada (por exemplo, um ficheiro de configuração, ou uma linguagem de especificação complementar à linguagem principal).

O desenvolvimento do compilador deve envolver, tanto quanto possível, todas as fases de construção de linguagens de programação:

1. Concepção e definição de uma linguagem de programação (sintaxe e semântica).
2. Implementação em ANTLR4 da análise léxica e da análise sintáctica de um compilador para a linguagem;
3. Definição das regras semânticas a aplicar à linguagem, e sua implementação no contexto do ponto anterior.
4. Escrita de documento que descreva a linguagem (instruções existentes e o seu significado; exemplos de programas; etc.).

5. Escolha criteriosa de uma linguagem destino, onde se possa implementar a síntese (*backend*) do compilador.
6. Definição dos padrões de geração de código para as instruções da linguagem.
7. Concretização completa do compilador.

3 Temas

Este ano, foram definidos dois possíveis temas para o trabalho: *pdraw* e *ciberRato*. A descrição detalhada destes temas pode ser encontrada (respectivamente) nos documentos `pdraw.pdf` e `mus.pdf`.

As escolhas a tomar no desenvolvimento das linguagens e respectivas gramáticas são relativamente livres, embora condicionadas ao que está escrito nesses documentos. Sugere-se a implementação de algumas das seguintes operações:

- Definição de variáveis;
- Definição de expressões que definam uma *álgebra* (à semelhança do que é possível fazer-se com expressões aritméticas) sobre elementos da linguagem (números, figuras, pontos, velocidades, ...);
- Operações interactivas com o utilizador;
- Expressões booleanas (predicados) e instruções condicionais;
- Instruções iterativas;
- Funções.

Para além do documento que descreve as linguagens desenvolvidas, tem de fazer parte da entrega do trabalho um conjunto adequado de programas (que funcionem!) de exemplo das linguagens.

A autoria de cada parte do código desenvolvido deve estar devidamente documentada no código.

4 Grupos

O trabalho deve ser realizado por grupos de 5/6 elementos. Os grupos devem ser formados por elementos de turmas que partilhem o mesmo horário.

Aplicam-se os seguintes prazos:

Aula prática 8: Constituição dos grupos e seriação dos temas.

Na seriação dos temas, cada grupo deve indicar uma ordem de preferência dos dois temas (por ordem decrescente de interesse). A escolha final caberá aos docentes que terão em conta as preferências apresentadas e uma distribuição tão equitativa quanto possível dos temas.

5 Avaliação

Os projectos serão desenvolvidos na plataforma `github`. **Os projectos serão criados pelos docentes**, e atribuídos a cada grupo. Alerta-se desde já que as actualizações feitas ao repositório devem ser executadas por quem desenvolve o código, usando mensagens adequadas. Serão mal toleradas situações do tipo “tive que pedir ao meu colega para o fazer”.

A entrega do trabalho será feita recorrendo a estes repositórios.

O trabalho será constituído pelo código fonte desenvolvido (gramáticas e código Java); vários exemplos de utilização das linguagens (que devem funcionar); relatório que sirva de manual de programação da linguagem desenvolvida e onde é obrigatório incluir uma secção “Contribuições dos autores” onde se descrevem resumidamente as contribuições de cada elemento do grupo. Essa secção deve conter uma distribuição da percentagem de trabalho de cada um. Esta auto-avaliação poderá afetar a ponderação da nota a atribuir a cada elemento.

Na avaliação do trabalho, 10% da classificação será retirada da percepção por parte do docente da prática sobre o empenho e interesse no desenvolvimento do mesmo **durante as aulas práticas**.

Na avaliação dos restantes 90% vão pesar a qualidade da solução desenvolvida e os objectivos parcelares por ela cobertos. Fazem parte destes objectivos os seguintes pontos:

1. Concepção das linguagens. A simplicidade e expressividade de cada linguagem definida serão aspectos a valorizar.
2. Gramáticas desenvolvidas.
3. Análise semântica.
4. Gestão de erros.
5. Legibilidade do código e documentação.
6. Geração de código (será valorizado o uso de uma linguagem destino mais “baixo nível”).

O grau de ambição do trabalho desenvolvido, confrontado com os resultados obtidos, será também tido em conta.

6 Execução do trabalho e prazos de entrega

Estão previstas quatro semanas para o desenvolvimento do trabalho (atenção que o tempo passa depressa!). Assim, as aulas práticas até 13 de Junho serão dedicadas ao desenvolvimento do trabalho (sendo que este requererá também tempo de trabalho fora do horário das aulas).

A entrega do trabalho para a época normal terá o prazo limite a **13 de Junho**, sendo obrigatório o envio de um email ao Prof. Artur, com CC a todos os elementos do grupo, indicando essa entrega.

Como indicado nas “Notas Gerais sobre o funcionamento” da disciplina, pode haver recurso neste elemento de avaliação. Nesta situação, o prazo limite de entrega será uma semana antes da data dos testes de recurso, sendo que o aluno terá de realizar um novo trabalho com um tema a definir pelos docentes (estando excluídos os dois temas propostos para a época normal).