UNIVERSIDADE DE ÉVORA

Compiladores

Relatório do Trabalho Prático

Autores:
João Marques, 39996
Tiago Martinho, 35735

Docente:
Pedro Patinho

Junho de 2020

Índice

1 Introdução

No âmbito da unidade curricular de compiladores, pretende-se desenhar e implementar um **compilador de *ya!***. Este trabalho foi desenvolvido ao longo do semestre e divide-se em 4 fases (**análise lexical**, **análise sintáctica**, **análise semântica** e **geração de código**).

2 Desenvolvimento

2.1 Análise Lexical

Para fazer a análise lexical foi utilizado o ***Flex***, programa abordado na disciplina. O *Flex* transforma palavras reservadas em **tokens** que são depois usados pelo compilador. Na linguagem *ya!* exitem *strings* que tiveram que ser tratadas como **literais**.

2.2 Análise Sintática

O ***Bison*** foi o programa usado para **definir a gramática** e irá percorrer o ficheiro a compilar criando a **APT** (*abstract parse tree*). Caso encontre algum erro sintático termina a compilação com uma mensagem de erro. Este analisador sintático foi construido com base no que se aprendeu nas aulas práticas sobre a linguagem *Ya!*. Sendo assim o analisador garante que todas as **regras de sintaxes** sejam feitas corretamente.

2.3 Análise Semântica

A nossa análise semântica baseia-se na implementação de ***buckets*** do livro *(Appel, Andrew W. Modern compiler implementation in C. Cambridge university press.)*. Que consiste numa *hashtable* de **enderençamento encadeado** usando *linked lists*. Deste modo garante que a variável no *scope* corrente está sempre no inicio da lista.
Criou-se também uma função que verifica se **dois tipos são compatíveis** e ainda uma função que faz ***castings* implicitos**. No entanto, a ultima não foi utilizada visto que seria necessária uma segunda passagem na árvore para estes *castings* serem realizados, algo que será deixado para uma fase de melhoramento do trabalho.

2.4 Geração de Código

Decidiu-se seguir a abordagem de máquina de pilha que utiliza a pilha de memória como zona de *buffer*, seguindo o diagrama da figura 1. A arquitetura escolhida foi ***Mips***.

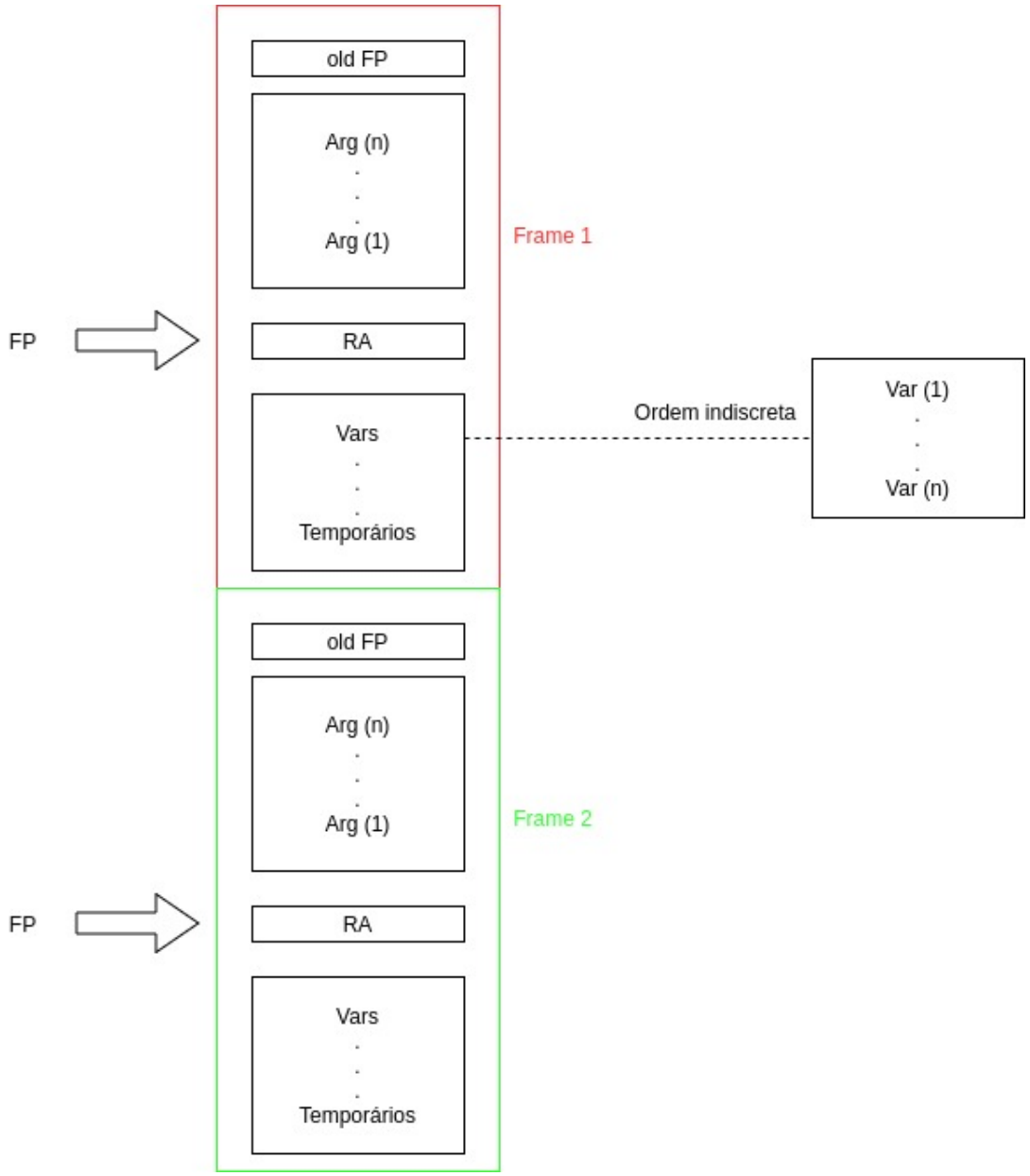

Figura 1: Registo de Ativação

3 Conclusão

Até ao presente o trabalho encontra-se a cumprir 3 dos seus 4 objectivos, sendo que a fase final da **geração de código** não se encontra completamente funcional. Existem muitos aspectos a melhorar nomeadamente a gestão de memória dos *arrays*, tratamento de floats, algumas operações e funções como o *print* e *input*. Ainda assim o trabalho está com uma boa **fundação** para uma melhoria futura, nomeadamente o casting implicito e o acesso a variáveis declaradas em contextos anteriores.
Ainda assim o grupo sente que conseguiu aprender os mecanismos internos que fazem um compilador, criando assim uma maior relação de **respeito** com as ferramentas que usamos todos os dias.

4 Referências

[1] https://www.cambridge.org/core/books/modern-compiler-implementation-in-c/0F85704413FC010C1D1C691C4D2A0865