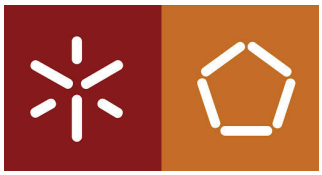

12 de maio de 2024

Relatório do Trabalho Prático de Processamento de Linguagens

Afonso Oliveira e Silva, a100604

Lara Regina da Silva Pereira, a100556

Martim de Oliveira e Melo Ferreira, a100653

Conteúdo

1. Introdução

O presente relatório refere-se à descrição do processo de desenvolvimento do Trabalho Prático da Unidade Curricular de Processamento de Linguagens, do segundo semestre do 3º ano da Licenciatura em Engenharia Informática.

O projeto consiste na implementação de um compilador de Forth, uma linguagem de programação de baixo nível, baseada numa *stack* criada por Charles H. Moore na década de 1960. O compilador deve gerar código para uma máquina virtual (VM) disponibilizada online.

O compilador deverá suportar: todas as expressões aritméticas (soma, adição, subtração, divisão, resto da divisão inteira), criação de funções, *print* de carateres e strings, condicionais, ciclos e variáveis.

2. Gramática

Definimos a nossa gramática e os nossos tokens de forma a não haver conflitos, fazer recursividade à esquerda e permitir-nos reconhecer toda a sintaxe necessária em Forth para fazermos uma boa compilação do código.

```
Grammar

Rule 0     S' -> Operacoes
Rule 1     Operacoes -> Operacoes Operacao
Rule 2     Operacoes -> Operacao
Rule 3     Operacao -> : ID Operacoes ;
Rule 4     Operacao -> PARENCOMMENTS
Rule 5     Operacao -> DO
Rule 6     Operacao -> LoopEnd
Rule 7     LoopEnd -> LOOP
Rule 8     LoopEnd -> PLUSLOOP
Rule 9     Operacao -> ArithmeticOperation
Rule 10    Operacao -> Condicional
Rule 11    Operacao -> PRINTSTRING
Rule 12    Operacao -> Number
Rule 13    Number -> FLOAT
Rule 14    Number -> INT
Rule 15    Operacao -> PRINTINT
Rule 16    Operacao -> PRINTFLOAT
Rule 17    Operacao -> CHAR
Rule 18    Operacao -> COMPARISON
Rule 19    Operacao -> LINECOMMENT
Rule 20    Operacao -> ID
Rule 21    Condicional -> IF Operacoes ELSE Operacoes THEN
Rule 22    Condicional -> IF Operacoes THEN
Rule 23    ArithmeticOperation -> ADD
Rule 24    ArithmeticOperation -> FADD
Rule 25    ArithmeticOperation -> SUB
Rule 26    ArithmeticOperation -> FSUB
Rule 27    ArithmeticOperation -> MUL
Rule 28    ArithmeticOperation -> FMUL
Rule 29    ArithmeticOperation -> DIV
Rule 30    ArithmeticOperation -> FDIV
```

Figura 1: Gramática definida

forth_lexer.py > ...

```python
import ply.lex as lex

tokens = [
    'FLOAT',
    'INT',
    'FADD',
    'ADD',
    'FSUB',
    'SUB',
    'FMUL',
    'MUL',
    'FDIV',
    'DIV',
    'CHAR',
    'IF',
    'ELSE',
    'THEN',
    'COMPARISON',
    'LINECOMMENT',
    'DO',
    'PLUSLOOP',
    'LOOP',
    'ID',
    'PRINTSTRING',
    'PRINTFLOAT',
    'PRINTINT'
]

literals = [':', ';', '"']

reserved = {
    'CHAR': 'CHAR'
}

```

Figura 2: Tokens definidos

3. Compilador

O propósito do compilador é o reconhecimento dos *tokens* e a sua tradução para linguagem da VM.

Para tal, utilizamos vários contadores para armazenar o número de elementos na *stack* e o número de condicionais e ciclos (para podermos controlar os pontos para onde realizar o `JUMP`).

Quando o compilador se depara com uma função, ele injeta o código da função no resultado em linguagem máquina. Esta não é a forma ideal como gostaríamos de resolver este problema, porém, devido a problemas de gestão de tempo, foi a decisão tomada e verificou-se funcional.

O compilador é capaz ainda de identificar condicionais do tipo `if else then` e `if then`, bem como ciclos do tipo `do ... loop` e `do ... +loop`.

4. Conclusão

Terminada a nossa implementação do projeto, concluímos que nem todos os objetivos foram cumpridos com sucesso.

Primeiramente, as funções, em vez de serem definidas, armazenadas, e posteriormente compiladas, estão a ser injetadas diretamente no código a executar. Este aspeto, apesar de garantir a execução das funções não é a forma ideal de resolver o problema.

Adicionalmente, os ciclos não funcionam quando há elementos anteriores na *stack*. Em qualquer outra situação, funcionam corretamente.

Devido a problemas de gestão de tempo, não conseguimos concluir o trabalho da forma que pretendíamos, e estes problemas ficaram por resolver. Com o devido tempo e dedicação a este projeto, consideramos que seriamos capazes de resolver estas questões e efetuar algumas melhorias.