

# CES-41: Compiladores

## Laboratório 6: Interpretador para uma linguagem de programação utilizando a ferramenta Yacc

Cássio dos Santos Sousa, Renan Pablo Rodrigues da Cruz

Professor: Fábio Carneiro Mokarzel

28 de dezembro de 2015

### 1. Introdução

Até o momento, foi possível criar código intermediário para programas válidos escritos na linguagem COMP-ITA 2015, além da validação propriamente dita por meio das ferramentas *Lex* e *Yacc*.

Para este laboratório, o objetivo foi interpretar o código intermediário de um programa em análise. Após interpretado, foi necessário também imprimir o resultado da execução, por meio do interpretador, do programa analisado. Os testes feitos aqui, portanto, precisaram ser capazes de demonstrar esta propriedade.

Este laboratório é complexo o suficiente para que seu desenvolvimento seja feito por duplas de alunos sem prejuízo de aprendizado, e o último dos laboratórios de CES-41.

## 2. Resultados

### 2.1. Atividades realizadas

A base mais essencial do interpretador já tinha sido discutida em aula. Com a exceção de esclarecimentos completos do interpretador, que ficaram de fora da aula prática, a base de leitura das quádruplas e suas interpretações nos casos mais comuns tinham sido exploradas. O foco das atividades foi o de trazer o código feito no laboratório para o código em *Yacc* que já possuíamos, tendo certeza de que todas as interpretações tinham sido feitas e explicadas no código de saída.

### 2.2. Formato do código interpretado

A escrita do código interpretado foi feita logo após a escrita da tabela de símbolos e das quádruplas dos laboratórios anteriores.

O trecho a seguir foi extraído do arquivo `Tests/1_atribuicao/lab06results1`, para ilustrar o formato do código intermediário:

INTERPRETADOR:

1) OPENMOD

Alocando as variaveis:

i: 1 elemento(s) alocado(s)

2) ATRIB

Foi atribuido o valor INTEIRO 1 para a variavel i

3) OPEXIT

Programa encerrado

## 2.3. Detalhes notados

Um detalhe percebido foi o de que erros semânticos não impediam a escrita de código intermediário. Dependendo do erro, a interpretação das quádruplas ainda pode ser realizada, mas a saída obtida fica completamente sem sentido. Por questões de tempo, impedimentos na criação de quádruplas não foram implementados.

## 2.4. Casos de teste

Aproveitou-se a estrutura dos testes já realizados no laboratório 5, que já passavam por uma análise esperada em suas quádruplas, para a criação dos casos de teste do laboratório 6:

- **1\_atribuicao:** código para verificar a correta interpretação das quádruplas de atribuição de variáveis;
- **2\_if\_sem\_else:** código para verificar a correta interpretação das quádruplas do if statement sem else;
- **3\_if\_com\_else:** código para verificar a correta interpretação das quádruplas do if statement seguindo de else;
- **4\_while:** código para verificar a correta interpretação das quádruplas do comando while;
- **5\_for:** código para verificar a correta interpretação das quádruplas do comando for;
- **6\_read:** código para verificar a correta interpretação das quádruplas do comando read;
- **7\_write:** código para verificar a correta interpretação das quádruplas do comando write.

## 3. Conclusões

Com o código sendo interpretado, termina aqui a criação de um interpretador para a Linguagem COMP-ITA 2015. Foi possível verificar inúmeras implementações (ou a falta das mesmas) que poderiam ter sido feitas ou melhoradas, mas frente à complexidade que enfrentaríamos e às adversidades que ocorreram no decorrer do desenvolvimento, acabaram ficando de lado.

Esperamos com este laboratório sedimentar os conhecimentos principais que envolvem a criação de compiladores.