Relatório CP1

Athus Assunção Cavalini
Murilo Ferri Schirmer
Rebeca Cecco de Oliveira

16 de outubro de 2020

Resumo

Este é o relatório referente ao checkpoint 1 do trabalho para a disciplina de Compiladores.

1 Status

O trabalho até então consistiu em desenvolver os analisadores léxico (scanner) e sintático (parser) de um compilador para Python. Primeiramente, utilizamos da documentação da linguagem para definir os principais tokens (trabalho que foi complementado à medida que o parser era desenvolvido).

Já no scanner nos deparamos com a primeira complicação da linguagem: a indentação como indicador de bloco de código. Para contornar este problema foi necessário definir os tokens INDENT e DEDENT (representando espaço e a ausência de espaço, respectivamente) e utilizar uma pilha para controlar a leitura destes tokens de forma a detectar um erro na indentação do programa. Esta questão nos leva a primeira simplificação: o caracter do tab ("\t") não é reconhecido pelo scanner, apenas os espaços em branco unitários.

Isso nos gerou dois novos problemsa. Como o Bison aceita apenas um token por vez do Flex, a "dedentação"múltipla não era possível apenas a partir de um loop. Além disso, a indentação, que deve ser conferida sempre que uma nova linha se inicia, prejudicava o funcionamento do token "NEWLINE", já que o caracter "\n"era consumido pela função de indentação.

Estes problemas podem ser solucionados de duas formas: permitir ao Bison receber múltiplos tokens; ou criar uma estrutura que armazene os tokens no Flex e envie apenas um por vez, como uma fila circular. Essas soluções estão sendo implementadas e testadas pelo grupo.

Quanto ao parser, utilizamos a gramática disponibilizada na documentação da linguagem (versão 3.8.6) e, com isso, tivemos que transformar a linguagem que estava em EBNF para o formato BNF (suportado pelo Bison). Obtivemos vários conflitos de shift/reduce ao tentar utilizar todas as regras definidas na documentação, o que nos leva as outras simplificações necessárias para o nosso compilador:

- A funcionalidade de uma função ter uma lista de argumentos de tamanho variável não foi implementada neste parser;
- As regras de single_input e eval_input foram descartadas.

Dessa forma, os conflitos foram reduzidos a zero, mas as simplificações podem prejudicar o funcionamento do scanner e pode ser necessário implementar, posteriormente, algumas novas funções.

2 Testes

Infelizmente, por conta dos problemas com indentação, os testes para averiguar o devido funcionamento dos analisadores léxico e sintático foram limitados, já que não podem envolver a "Dedentação".

No teste piloto (test0.py) notamos que todos os programas precisam encerrar com um "\n". Em nossa lista de prioridades, este problema foi deixado no fim, já que a muitos editores de código inserem uma quebra de linha ao final dos programas de forma automática.

No primeiro teste (test1.py), percebemos que temos mais um problema: Os comentários precisam consumir um "NEWLINE", caso contrário, por permitirem qualquer caractere, consumirão todo o restante do código. A solução que imaginamos para este problema é consumir um "NEWLINE"e, ao emitir o token "TYPE_COMMENT", emitir um "NEWLINE"logo em seguida. Por ora, optamos por ignorar qualquer tipo de comentário.

No segundo teste, averiguamos apenas o problema já conhecido de indentação (comentado no código).

Os demais testes não nos mostraram outros problemas.