

Especificação da implementação do comando Return

Aluno: Lucas Santana Bicalho da Costa

1 - Extensão da gramática de Imp (imp2.ebnf)

Adicionado o comando return na linguagem Imp2

```
ret = "return" e:exp ;
```

que é uma das opções de atom_cmd

```
atom_cmd = print | cond | loop | assign | call | skip |  
ret ;
```

e também a palavra “return” como uma palavra reservada.

```
@@keyword :: 'return'
```

Além disso, para que o valor retornado seja utilizado é necessário adicionar o comando call como uma opção de exp:

```
exp = call | paren_exp | un_exp | bin_exp | @:atom ;
```

2 - Pi denotações de Imp à Pi IR

No arquivo impiler.py o método ret

```
def ret(self, ast):  
    return pi.Return(ast.e)
```

foi adicionado à classe Impiler. Este método retorna um objeto do tipo Return, classe que foi adicionada ao arquivo pi.py. Esta classe segue o formato:

```
<Cmd> ::= 'Return'(<Exp>)
```

3 - Novas equações adicionadas ao Pi framework que dão semântica à return em Pi IR

$X \in \langle \text{Exp} \rangle$.

$$\delta(\text{Return}(X) :: C, V, E, S, L) = \delta(X :: \#RETURN :: C, V, E, S, L)$$

Quando a keyword “#RETURN” é lida do topo da pilha, para cada keyword “#BLKCMD” lido posteriormente é desempilhado da pilha de valores um Env e um Loc. Quando o objeto lido da pilha de controle for diferente “#BLKCMD” o valor a ser retornado será empilhado na pilha de valores e todas as “#BLKCMD”, e Env e Locs, serão retornados às pilhas de controle e de valor, respectivamente.

Além disso, as classes Print, Assign e Bind foram modificadas para aceitar objetos do tipo Call também.