UNIVERSIDADE FEDERAL DO PARANÁ

DEPARTAMENTO DE INFORMÁTICA

BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

CI211 - CONSTRUÇÃO DE COMPILADORES

Trabalho 3 - Verificador de Tipos

Nome: Henrique Colodetti Escanferla - GRR20135427

COMENTÁRIOS SOBRE A IMPLEMENTAÇÃO

1. Modificações no Analisador Léxico foram feitas para popular estruturas que não existiam antes. Agora existem tabelas das declarações mapeando os identificadores e seus tipos. Há tabela para constantes, funções e variáveis. Existe uma pilha para salvar todos os identificadores lidos que não são declarações. Aqui uma parte das verificações que envolve declaração inválida ou uso sem declarar são executadas.
2. Modificações no Analisador Sintático permitiram que os valores fossem lidos e seus tipos identificados, guardando-os em uma pilha de valores. É aqui que a constante recebe seu tipo e é mapeada na tabela com seu tipo.
3. No Analisador Sintático, existe um método chamado typeChecker. Este faz todo o resto da verificação: desempilha valores e identificadores de acordo com o estado atual do autômato do Analisador Sintático e executa as verificações de acordo com o tópico abaixo.

REGRAS, ESTADOS E AÇÕES PARA O VERIFICADOR DE TIPOS

Obs.: Cada regra abaixo demonstra o que deve ser desempilhado e verificado. A ordem do desempilhamento é importante, é o inverso da leitura. Existem pilhas separadas para constantes, funções, variáveis e valores. Só declarações de funções são empilhadas para verificar seus retornos.

EXPR\_REL => num op\_rel num e95: pop 2 values.

FUNCAO => function type id ( ) { DECL\_VAR CMDS return id ; }

e19: pop 1 variável e 1 função do mesmo tipo.

ATR => id = CH\_FUNC e74: pop 1 função e 1 variável do mesmo tipo.

ATR => id = id ; e76: pop 2 variáveis do mesmo tipo.

ATR => id = num ; e67: pop 1 variável e 1 valor. Se a variável é int o valor não pode ser float.

ATR\_FOR => id = num e83: pop 1 variável e 1 valor, ambos int.

ATR => id = ch ; e71: pop 1 variável e 1 valor do mesmo tipo.

ATR => id = str ; e73: pop 1 variável e 1 valor do mesmo tipo.

EXPR\_REL => num op\_rel id e96: pop 1 variável e 1 valor do mesmo tipo.

EXPR\_REL => id op\_rel num e99: pop 1 variável e 1 valor do mesmo tipo.

CMD => CH\_FUNC e30: pop 1 função int.

INCR => id inc e80: pop 1 variavel int.

EXPR\_MAT => num op\_art num e85: pop 1 variável e 2 valores. Não podem haver valores float se a variável for int.

EXPR\_MAT => num op\_art id e86: pop 2 variáveis e 1 valor. Não pode haver valor ou 1ª variável desempilhada float se a 2ª variável desempilhada for int.

EXPR\_MAT => id op\_art num e88: pop 2 variáveis e 1 valor. Não pode haver valor ou 1ª variável desempilhada float se a 2ª variável desempilhada for int.

EXPR\_MAT => id op\_art id e89: pop 3 variáveis. A 1ª ou 2ª variável desempilhada não pode ser float se a 3ª variável desempilhada for int.

EXPR\_REL => id op\_rel id e100: pop 2 variáveis int ou float.