**RELATORIO TRABALHO PRATICO**

**COMPILADORES I**

1. Ester Amorim

2. Lucas Paraiso

3. Luis Gustavo Oliveira

4. Matheus Henrique

5. Renan Cruz

Exigências propostas:

* A criação de um compilador sintático que criasse uma tabela de lexemas, especificando tipo e alguns parâmetros.
* Linguagem usada: Python

Planejamento inicial:

O nosso grupo primeiro estudou um pouco sobre mini pascal, principalmente a parte léxica, que era a que iriamos utilizar e, a partir de um arquivo disponibilizado no moodle, fizemos nossa própria tabela de tokens e importamos para o programa por meio de dicionário python.

Para contexto geral, após pesquisarmos sobre estruturas de dados em python, viemos a descobrir que dicionário funcionava de maneira praticamente idêntica a uma hashmap não balanceada e a usamos. No final do programa importamos pandas para deixar a tabela mais organizada e de maneira fácil.

Para armazenar as variáveis para a gente já ter ideia de quando já foram utilizadas, usamos duas listas, uma para armazenar de fato o conteúdo e outra para sabermos o tipo dela para que futuramente em uma possível chamada possamos identificar qual seu tipo.

Desenvolvimento:

Começamos nosso projeto com um programa ao qual importamos um documento de texto que o próprio usuário pode colocar o nome do arquivo ao executa o bloco de código, armazenamos todas as linhas em uma lista, sendo cada index a sua linha.

O algoritmo ignora linhas em branco ou linhas de comentário (definidas com inicial //) e começa tratamentos especiais na linha, como os exemplos a seguir:

Texto

Descrição gerada automaticamente

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

Tratamento para Arrays, analisa se foi corretamente declarado e se retira espaços dentro dos conchetes, para melhor manipulacao

Tratamento para atribuições de strings, se for corretamente colocada, ele separa a linha corretamente, para que os espaços entre parênteses não sumam e caso

Seguindo na mesma logica, também foi tratado atribuições de int, para evitar uma string ser alocada sem querer ou ate mesmo um operador terminar a mensage (exemplo a =9 +)

Texto

Descrição gerada automaticamente

Texto

Descrição gerada automaticamente

Lidar com atribuição de string funciona de forma parecida

Texto

Descrição gerada automaticamente

Após esses tratamentos de variáveis, se todos forem aceitos partimos para a próxima parte do algoritmo onde

Caso o programe chame uma variável array, há um tratamento também nessa parte:

Alem disso, o nosso grupo resolveu fazer que so tipos da array sejam imprimidos em apenas 1 instancia da tabela, contendo como parâmetro seu tamanho, tendo sua capacidade como default se não especificar:

Texto

Descrição gerada automaticamente

Ttype verifica se ele esta em algum lugar do token, e se sim, ele terá mais caractere no seu string que o normal e utilizamos disso. Na imagem não mostra mas se não for programa ou não for um atribuidor de string, você não é detectado

Texto

Descrição gerada automaticamente

No print anterior foi feito também uma função de program, para ser reconhecido posteriormente para sua adição na tabela, há também um tratamento para não ser chamado 2 vezes.

Além disso, incrementamos o write e o read ao nosso programa que diferente de outras funções ele pega a frase como um todo e divide em pedaços, se ocorrer algum erro como ; não existir, o programa para. Quando bem sucedido o write e o read são adicionados a nossa tabela de tokens

If e while foram resolvidos em apenas tres linha:

elif v == 'if' or v == 'while':

tabela[str(i+1)+'°'] = [str(v),Ttype, line.replace('do','').replace(v,'')]

i = i + 1

Por ultimo e não menos importante nosso programa avalia se o argumento passado é uma variável existente. Quando ela é, faz uma varredura pela frase para ver se esta tudo ocorrendo. Algumas exceções ocorrem, como por exemplo quando é uma function ou procedure, que passa como argumento todos os parâmetros da função



Após todos esses loopings o nosso programa imprime uma tabela visualmente entendível feita pela framework PANDAS

EXEMPLOS

Nos criamos essa mini linguagem pascal, claro com possíveis erros sintáticos, mas queríamos apenas testar nosso analisador léxico (arquivo consta junto com o projeto) :

Texto

Descrição gerada automaticamente

Após a leitura do algoritmo a seguinte tabela foi criada: