**Comparador de Programas Monolíticos**

**Êndril Castilho da Silveira, Leonardo Pellegrini Silva**

Disciplina de Teoria da Computação

Departamento de Informática – UNISC

Campus de Santa Cruz do Sul

96.810-206 - Santa Cruz do Sul - RS - Brasil

[endrilcs@mx2.unisc.br](mailto:endrilcs@mx2.unisc.br), leonardo@mx2.unisc.com

**1. Sobre o projeto**

Desenvolvido em Python 3, o aplicativo conta com uma interface gráfica construída com base na ferramenta Tkinter.

O aplicativo é capaz de traduzir programas monolíticos simples em programas monolíticos compostos. Após a etapa de tradução, o resultado é exibido na tela para que o usuário possa visualizar a conversão. Os programas, são então comparados exibindo o passo a passo de sua comparação até que o veredito seja dado pelo programa com a exibição dos resultados da comparação.

**2. Modo de usar**

Há dois campos em brancos na primeira tela exibição. No campo da esquerda você pode digitar o primeiro programa monolítico simples e, no campo da esquerda, você pode digitar o segundo programa monolítico simples, ambos precisam estar digitados para que o programa possa continuar. Quando os dois estiverem digitados você poderá pressionar o botão de Prosseguir. Ainda, se for de sua preferência, você pode carregar os programas por meio do Arquivo>Abrir Programas. Uma tela irá se abrir para que você possa selecionar onde está os arquivos .txt, um para cada programa.

Imagem do Passo 0

**3. Padrão de entrada de dados**

Em relação as instruções do programa há um certo padrão que precisa ser seguido. Sempre comece a linha com uma instrução “Se” ou “Faça” dê espaço uma e apenas uma vez, logo em seguida defina uma palavra para nomear instrução e encerre com espaço. Depois escreva “para” e dê espaço uma e somente uma vez seguido de um número, sinalizando a linha do programa monolítico simples em que a instrução irá apontar. Se for um teste (ou seja, “se”), você precisa dar espaço e escrever novamente um “para” seguido de um e somente um espaço. Após isso você pode sinalizar o número para qual a instrução irá apontar em caso de o teste retornar falso.

Ainda sobre o padrão de entrada dos dados é importante salientar que:

* Um teste não pode apontar para ele mesmo, caso contrário, o aplicativo acusará erro.
* Uma instrução de teste (ou seja, do tipo “se”) não pode apontar para o mesmo lugar em ambos os casos de verdadeiro de falso, caso contrário o aplicativo acusará erro.
* Nenhum dos campo de digitação de programas pode estar vazio, caso contrário, o aplicativo acusará erro.
* Uma instrução de execução de função (ou seja., do tipo “para”) não pode apontar para uma para outra função que aponte para a primeira.