Universidade Eduardo Mondlane
Faculdade de Engenharia
Departamento de Engenharia Electrotécnica

1º Teste de Compiladores	Curso de Licenciatura em Engenharia Informática	(100 min)

1. Liste as fases e componentes de um compilador e descreva brevemente suas funcionalidades e actividades. (**3V**)

R: As fases de um compilador são:

- **Analise Léxica**: É a primeira fase do processo de compilação, também é conhecida como leitura ou scanning. O objetivo nessa fase é identificar unidades léxicas ou lexemas que compõem o programa. O analisador léxico lê todos os caracteres do programa fonte e verifica se eles pertencem ao alfabeto da linguagem. Caso um caractere não pertença ao alfabeto da linguagem deve ser gerado um erro léxico.
- **Analise Sintática**: A analise sintática tem como objeto validar a gramática do programa, nessa etapa o objetivo é reconhecer se a estrutura gramatical do código fonte esta de acordo com as regras sintáticas da linguagem. Nessa etapa é feita uma varredura na sequência de tokens recebidas do analisador léxico e produzida uma estrutura de dados em formato de árvore conhecida como árvore sintática. A árvore sintática representa a hierarquia do programa fonte. Caso uma construção seja reconhecida com inválida um erro sintético deve ser gerado.
- **Analise Semântica:** O objetivo dessa etapa é verificar se a semântica do programa fonte tem consistência. Para isso é utiliza a árvore sintática e as informações contidas na tabela de símbolos. Ou seja ver se as sentenças bem estruturadas tem significado valido.
- **Geração de código intermediário:** Nesse fase é gerado uma sequência de código denominada código intermediário, que posteriormente em outras fases irá gerar o código objeto. Por ventura essa fase pode não existir e a compilação pode ser feita diretamente para o código objeto, isso é comum em compiladores auto residentes. E importante destacar que o código intermediário não especifica detalhes da construção do código objeto final.
- **Optimização de Código:** Nessa fase o objetivo é otimizar o código em termos de velocidade de execução e consumo de memória. Essa etapa não depende da arquitetura de máquina e tem como objetivo fazer transformações no código intermediário afim obter um código objeto mais optimizado.
- **Geração de código objeto**: A geração de código objeto é a última etapa do processo de compilação e recebe como entrada uma representação intermediaria que mapeia a linguagem objeto. Desse momento deve ser feito a seleção de registradores e reserva de memória para contantes e variáveis. Essa é uma etapa muito importante pois a produção de código objeto eficiente deve ter uma cuidadosa seleção de registros.

Bom Trabalho!

2. Comente a afirmação: “O Analisador léxico é uma sub-rotina do analisador sintactico”. (**2V**)

R: O analisador léxico é considerado como sendo uma subrotina do analisador sintactico pois ele faz a varredura do programa fonte caractere por caractere e, traduz em uma sequência de símbolos léxicos ou tokens que são no geral enviados a ele pelo analisador sintactico para que estes possam ser validados e por sua vez este terá a função de ver se eles estão bem posicionados. É nessa fase que são reconhecidas as palavras reservadas, constantes, identificadores e outras palavras que pertencem a linguagem de programação. O analisador léxico executas outras tarefas como por exemplo o tratamento de espaços, eliminação de comentários, contagem do número de linhas que o programa possui e etc.

3. Qual é a relação entre o compilador e o montador? Quais as diferenças entre estas 2 ferramentas e pre-processadores? (**2V**)

R: Compilador traduz um programa em alto nível para um programa de baixo nível (linguagem simbólica). O montador é um tradutor que vem logo a seguir ao termino da compilação e tem a função de traduzir uma linguagem simbólica em linguagem de maquina.
A diferença com pre-processadores reside no facto dos pre-processadores não traduzirem de uma linguagem para outra, somente fazendo tendo como linguagem alvo da tradução a própria linguagem em que foi escrito o código de entrada

4. Em que fase da compilação podemos detectar cada um dos erros seguintes: (**6V**)

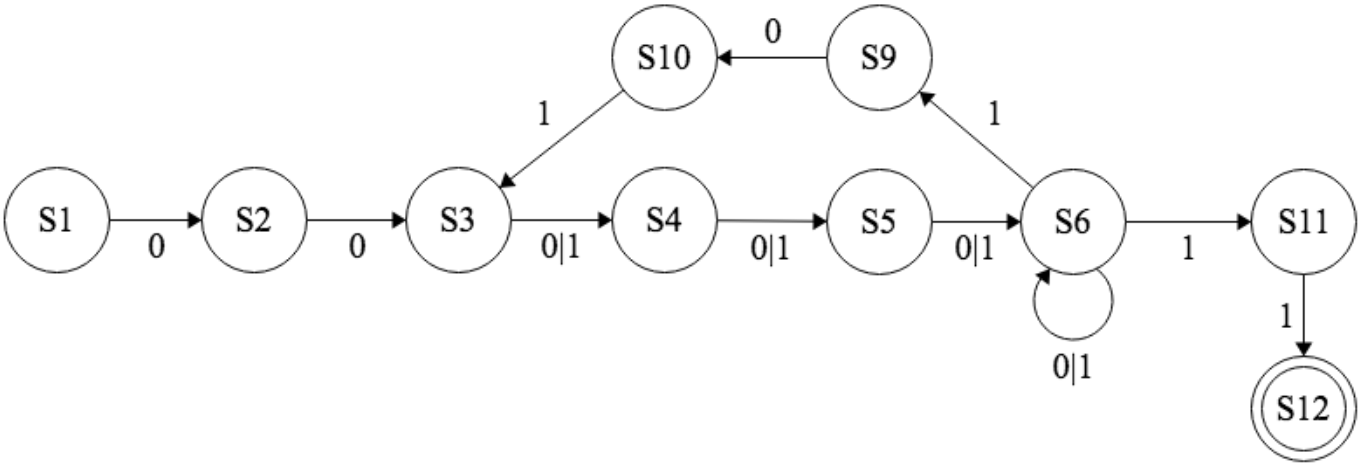
   a) Identificador mal formado: 12K3 em C; - **Analise Léxica**
   b) Conflito de tipo na função sin(“a”). – **Analise Semântica**
   c) Instrução nunca executada. – **Analise Semântica**
   d) Variável não declarada. - **Analise Semântica**
   e) Comentário aberto, mas não fechada. – **Analise Sintáctica**
   f) Parêntesis não fechado. – Analise Sintáctica
   g) BEGIN não fechado. – **Analise Sintáctica**
   h) Palavra chave utilizada como um identificador. – **Analise Sintáctica**
   i) Incoerência entre o número de parâmetros duma definição e o numero de parâmetro na chamada dum procedimento. – **Analise Semântica**
   j) Tentativa de modificar uma constante. – **Analise Semântica**
   k) Constante demasiada grande. – **Analise Léxica**
   l) Passagem do limite dum vector. – **Analise Semântica**

5. Considere o seguinte protocolo de comunicação de mensagens binárias entre dois computadores. O início e o fim de cada transmissão é marcada pelos caracteres, respectivamente, 00 e 11. As mensagens são obrigatoriamente iniciadas por três bits, indicativos do seu comprimento. Entre duas

mensagens há um separador constituído pelo padrão 101. Em todas as transmissões há, no mínimo, uma mensagem por transmitir. Um exemplo de transmissão válida de mensagem é dada por:

init	**size**	**msg**	**sep**	**size**	**msg**	**end**
00	010	11	101	001	1	11

a) Construa e Defina um AFND para o Problema dado; (**4V**)



Conjunto de Estados: {S1, S2, S3, S4, S5, S6, S9, S10, S11, S12}

Estado Inicial: S1

Estados Finais: {S12}

Alfabeto: {0,1}

Função de Transição:

	ENTRADA	
ESTADO	0	1
S1	{S2}	---
S2	{S3}	---
S3	{S4}	{S4}
S4	{S5}	{S5}
S5	{S6}	{S6}
S6	{S6}	{S6,S9,S11}
S9	{S10}	---
S10	---	{S3}
S11	---	{S12}
S12	---	---

b) Converta o AFND do exercício anterior em AFD (**3V**)

Bom Trabalho!

Função de Transição:

	ENTRADA	
ESTADO	0	1
S1	{S2}	---
S2	{S3}	---
S3	{S4}	{S4}
S4	{S5}	{S5}
S5	{S6}	{S6}
S6	{S6}	{S6,S9,S11}
S9	{S10}	---
S10	---	{S3}
S11	---	{S12}
S12	---	---
{S6,S9,S11}	{S6,S10}	{S6,S9,S11,S12}
{S6,S10}	{S6}	{S3,S6,S9,S11}
{S6,S9,S11,S12}	{S6,S10}	{S6,S9,S11,S12}
{S3,S6,S9,S11}	{S4,S6,S10}	{S4,S6,S9,S11,S12}
{S4,S6,S10}	{S5,S6}	{S3,S5,S6,S9,S11}
{S5,S6}	{S6}	{S6,S9,S11}
{S4,S6,S9,S11,S12}	{S5,S6,S10}	{S5,S6,S9,S11,S12}
{S3,S5,S6,S9,S11}	{S4,S6,S10}	{S4,S6,S9,S11,S12}
{S5,S6,S10}	{S6}	{S3,S6,S9,S11}
{S5,S6,S9,S11,S12}	{S6,S10}	{S6,S9,S11,S12}