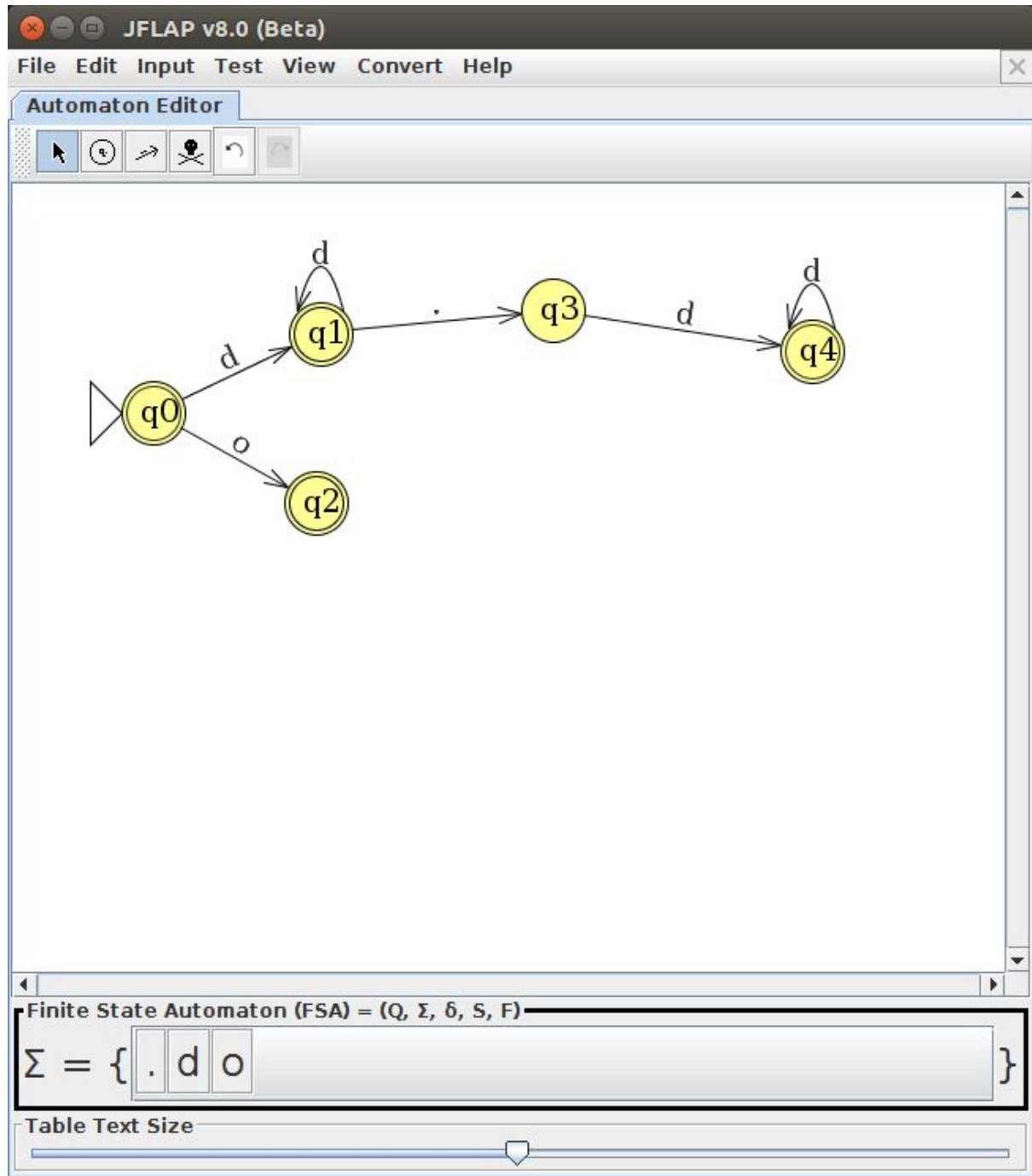


Alunos: Guilherme Utiana, Peter Brendel

1.



2.

As linguagens aceitas são: A linguagem vazia, Números inteiros, números decimais, e um e apenas um operando.

3. questao3.cpp, todas as outras questões de implementação.

4. a) Regra de formação de uma linguagem ao aplicar operadores sobre o alfabeto, definindo um padrão.

b) Descrição do conjunto de identificadores válidos em alguma linguagem ou algoritmos de plagio.

c) 1ª extensão: Operador + “Uma ou mais instâncias”. Representa o fechamento positivo de uma expressão regular, é igual ao fecho de Kleene, removendo o vazio.

2ª extensão: Operador ? “Zero ou uma instância”. Representa que a instância da palavra deve aparecer no máximo uma vez.

3ª extensão: “Classes de caracteres”: é a abreviação de um alfabeto, por exemplo: o alfabeto [abcde] pode ser escrito como a|b|c|d|e que pode ser escrito como [a-e].

 Questão 9. Considere a gramática livre de contexto: $S \rightarrow S S + \mid S S * \mid a$

a) A gramática é ambígua ou não ambígua? Justifique.

Não é ambígua pois na derivação da palavra $aa+a^*a+$ não se consegue árvores de diferentes derivações (esq. e dir.)

b) Descreva a linguagem gerada por essa gramática.

Se a palavra não for ‘a’, então ela terminará em * ou + e iniciará com ‘aa’.

- a
- $aa+a^*a+$

 10- $C1 \rightarrow \text{if } E \text{ then } C1 \mid \text{if } E \text{ then } C2 \text{ else } C1 \mid s$
 $C2 \rightarrow \text{if } E \text{ then } C2 \text{ else } C2 \mid s$
 $E \rightarrow e$

 13 -

FIRST	FOLLOW	NTERM	[]	a	,	\$
{[; a}	{ \$,] ; , }	S	$S \rightarrow [L]$		$S \rightarrow a$		
{[; a}	{] ; , }	L	$L \rightarrow L, S$ $L \rightarrow S$		$L \rightarrow L, S$ $L \rightarrow S$		

Pilha	Entrada	Ação
0	id+id*id\$	push 5 and a = “+”
05	+id*id\$	reduce F-> id
03	+id*id\$	reduce T-> F
02	+id*id\$	reduce E-> T
01	+id*id\$	push 6 and a = “id”
016	id*id\$	push 5 and a = “*”
0165	*id\$	reduce F -> id
0163	*id\$	reduce T -> F
0169	*id\$	push 7 and a = “id”
01697	id\$	push 5 and a = “\$”
016975	\$	reduce F -> id
0169710	\$	reduce T -> T*F
0169	\$	reduce E -> E + T
01	\$	acc

Questão 15. Explique o problema do conflito empilhar/reduzir (shift/reduce) :

O conflito acontece quando o analisador chega em um ponto onde tanto um shift quanto um reduce são ações válidas para continuar.

a) dê um exemplo de gramática que cause esse problema;

A \rightarrow B | C

C \rightarrow a B c C | a B c C d C

B \rightarrow b

b) Como a ferramenta Bison trata esse problema?

Escolhendo fazer o shift ao invés do reduce, sempre que encontra situações como esta.

c) Toda gramática LR(1) apresentará esse problema se tentarmos construir a tabela de análise através do método SLR?

Apenas gramáticas que contenham alguma ambiguidade.

Questão 16.

Indique Verdadeiro ou Falso

- a) [F] O método LALR é o método mais geral de reconhecimento sintático;
- b) [F] Gramáticas ambíguas não podem ser reconhecidas por métodos LL. Portanto, deve se utilizar métodos de reconhecimento sintático mais poderosos como o LR(1);
- c) [V] Os métodos descendentes recursivos reconhecem uma entrada através de tentativas de derivações a partir do símbolo inicial da gramática;
- d) [V] São métodos de reconhecimento top down: Descendente Recursivo, Descendente Preditivo e LL(1);
- e) [V] São métodos de reconhecimento bottom up: SLR, LR e LALR;
- f) [V] Se a tabela ACTION criada para uma gramática através do método SLR contiver conflitos, então pode-se dizer que a gramática não é uma gramática SLR;
- g) [V] Uma gramática LALR não pode ser reconhecida por um método de reconhecimento descendente recursivo;