Teoría de Lenguajes 2do. Parcial – Curso 2016

Consideraciones generales

- i) Escriba nombre y C.I. en todas las hojas.
- ii) Numere todas las hojas.
- iii) En la primera hoja, indique el total de hojas.
- iv) Comience cada ejercicio en una hoja nueva.
- v) Utilice las hojas de un solo lado.
- vi) Entregue los ejercicios en orden.

Ejercicio 1 [Evaluación individual del obligatorio]

a) Dada la siguiente EBNF que define un lista de parámetros (similar a la lista de parámetros de la declaración de funciones del laboratorio):

```
params ::= ((pTypeList ';' ) * pTypeList) ?
pTypeList ::= type (paramId ',') * paramId
paramId ::= 'ID' | 'ID' '[' ']'
type ::= 'int' | 'bool' | 'char'
```

Indicar cuáles de los siguientes conjuntos de reglas para JCUP, sirven para generar un parser que reconozca el lenguaje definido por la EBNF anterior. Si no sirven, justificar por qué no.

- i)

```
params ::= paramList | ;
paramList ::= paramList SEMICOLON pTypeList | pTypeList ;
pTypeList ::= type pIdList ;
pIdList ::= pIdList COMMA paramId | paramId ;
```
- ii)

```
params ::= params SEMICOLON pTypeList | pTypeList | ;
pTypeList ::= type pIdList ;
pIdList ::= pIdList COMMA paramId | paramId ;
```
- iii)

```
params ::= pTypeList | pAux pTypeList | ;
pAux ::= pTypeList SEMICOLON | pAux pTypeList SEMICOLON;
pTypeList ::= type pIdAux paramId;
pIdAux ::= pIdAux paramId COMMA | ;
```
- iv)

```
params ::= params SEMICOLON pTypeList | pTypeList;
pTypeList ::= type pIdAux paramId;
pIdAux ::= pIdAux paramId COMMA | ;
```

para todos casos se definen paramId y type de la siguiente forma:

```
paramId ::= ID | ID LEFTBRACKET RIGHTBRACKET;
type ::= INT | BOOL | CHAR ;
```

b) Indicar si las siguientes afirmaciones son verdaderas o falsas:

i) No se pueden representar las producciones épsilon en JCUP, excepto que se derive de la variable inicial, por eso se debe aplicar el algoritmo de eliminación de producciones épsilon necesariamente, y luego agregar una regla que derive épsilon de la variable inicial, si corresponde.

ii) El orden en que se debían ejecutar los scripts en el laboratorio era: compilar, generar-parser, ejecutar y comparar.

iii) Los terminales de la EBNF de la parte a) son: '(', ')', '*', '?', 'ID', '[', ']', 'int', 'bool', 'char', ';' y ','.

Ejercicio 2 [20 puntos]

Dados los siguientes lenguajes:

$$L_a = \{ x / x \in \{a,b,c\}^* \text{ donde } 2 \leq |x|_a \leq 3, |x|_b = |x|_c \}$$

$$L_b = \{ x / x \in \{a,b,c\}^* \text{ donde } 2 \leq |x|_a \leq 3, |x|_b \geq 0, |x|_c > 0 \}$$

- Construya gramáticas simplificadas G_a y $G_b / L_a = L(G_a)$ y $L_b = L(G_b)$. Deberá justificar porque las gramáticas propuestas están simplificadas.
- Construya un autómata $M_a / L_a = L(M_a)$. ¿Es determinista? Justifique.

Ejercicio 3 [16 puntos]

Sean:

L_1 finito

L_2 libre de contexto no regular

L_3 libre de contexto

L_4 recursivamente enumerable no libre de contexto

$$L_5 = \{ x / x \in \{a,b,c\}^* \text{ donde } x \text{ es de la forma } a^n b^m c^{\max\{n,m\}} \}$$

Decir si las siguientes afirmaciones son Verdaderas o Falsas. Justifique.

- $L_2 - L_1$ es recursivamente enumerable
- $L_3 \cap L_2$ es libre de contexto no regular
- $L_4 \cap L_1$ es libre de contexto
- L_5 no es libre de contexto

Ejercicio 4 [20 puntos]

Dado el siguiente lenguaje:

$$L_4 = \{ x / x \in \{a,b\}^* \text{ donde } x \text{ es de la forma } a^n b^m \text{ donde } m = \sum_{i=0}^{i=n} i \}$$

Ejemplos:

n=2	m=3	aabbb
n=3	m=6	aaabbbbb
n=4	m=10	aaaabbbbbbbbbb

- Construya una gramática $G_4 / L_4 = L(G_4)$.
- Construya un autómata $M_4 / L_4 = L(M_4)$.