

## Teoría de Lenguajes 2do. Parcial – Curso 2014

### 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 una "condición", similar al obligatorio:

```
condition ::= 'not' ? logic_term ( 'or' logic_term ) *
logic_term ::= logic_fact ( 'and' logic_fact ) *
logic_fact ::= simple_exp ( operator simple_exp ) *
simple_exp ::= 'id' | constant
operator ::= '=' | '<' | '>' | '<>' | '<=' | '>='
constant ::= 'string' | 'number'
```

Asumiendo que las reglas de JCUP para reconocer: *logic\_fact*, *simple\_exp*, *operator* y *constant* fueron correctamente definidas.

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

i.

```
condition ::= NOT condition2 | condition2;
condition2 ::= logic_term | condition2 OR logic_term;
logic_term ::= logic_fact | logic_term AND logic_fact;
```

ii.

```
condition ::= NOT condition2 | condition2;
condition2 ::= logic_term | logic_term OR condition2;
logic_term ::= logic_fact | logic_fact AND logic_term;
```

iii.

```
condition ::= NOT logic_term | logic_term |
              NOT logic_term OR condition |
              logic_term OR condition;
logic_term ::= logic_fact | logic_fact AND logic_term;
```

iv.

```
condition ::= NOT logic_term listT | logic_term listT;
listT ::= OR logic_term listT | ;
logic_term ::= logic_fact listF;
listF ::= AND logic_fact listF | ;
```

v.

```
condition ::= NOT logic_term listT | logic_term listT;
listT ::= OR logic_term listT;
logic_term ::= logic_fact | logic_fact listF;
listF ::= AND logic_fact listF;
```

b) Sea la siguiente condición:  $z = 1$  or  $x < 9$  and  $x <> 5$  and  $3 \leq x$

Suponiendo que a las reglas de JCUP de la parte anterior, se le agregan instrucciones para imprimir "AND", "OR", "CONSTANTE NUMERO" y "CONSTANTE CADENA DE

CARACTERES", como se pidió en el laboratorio, indicar cuáles de las reglas de la parte anterior (que sirven para reconocer la EBNF) producen las siguientes salidas:

| Salida i)        | Salida ii)       |
|------------------|------------------|
| CONSTANTE NUMERO | CONSTANTE NUMERO |
| CONSTANTE NUMERO | CONSTANTE NUMERO |
| CONSTANTE NUMERO | CONSTANTE NUMERO |
| AND              | CONSTANTE NUMERO |
| CONSTANTE NUMERO | CONSTANTE NUMERO |
| AND              | AND              |
| AND              | AND              |
| OR               | OR               |

- c) Para la EBNF de la parte a) indique cuáles son *terminales*, *no terminales* y *símbolos del meta-lenguaje de las EBNF*.
- d) Los parsers que genera JCUP toman una supuesta tira del lenguaje, e intentan armar la derivación que generaría la tira. Si esta derivación la vemos como un árbol de derivación, ¿el árbol se arma "de la raíz hacia las hojas" o "de las hojas hacia la raíz"?

### Ejercicio 2 [ 35 puntos ]

Dado el siguiente lenguaje:

$$L_2 = \{ x / x \in \{a,b,0,1\}^* \text{ donde } |x|_a = |x|_b > 0, |x|_0 \geq |x|_1 \geq 0 \}$$

- Clasifique según la Jerarquía de Chomsky. Justifique.
- Construya una gramática  $G_2 / L(G_2) = L_2$ .
- Construya un autómata que reconozca a  $L_2$ .

### Ejercicio 3 [ 20 puntos ]

Dado el siguiente lenguaje:

$$L_3 = \{ x / x \in \{a,b\}^* \text{ es de la forma } a^{2k+t}b^{k+t}a^p ; k,t,p > 0 \}$$

- Construya una gramática simplificada  $G_3 / L(G_3) = L_3$ . Explique porqué la gramática construida está simplificada.
- Construya un autómata que reconozca a  $L_3$ . ¿Es determinista? Justifique.