

Procesadores de Lenguaje. Curso 2003-2004. Junio de 2004. Cuestiones.

No se permiten libros ni apuntes. Tiempo: 1 hora.

FINAL: Contestad a las preguntas 1, 2, 4

SEGUNDO PARCIAL: Contestad a las preguntas 3, 4, 5

Todas las preguntas valen igual.

- Indica si son verdaderas o falsas las siguientes afirmaciones. Justifica la respuesta.
 - Para portar un compilador escrito en su propio lenguaje en una máquina H a otra máquina K, tenemos que implementar un compilador escrito en K que tome como entrada H y la traduzca al lenguaje a K.
 - Una gramática con ciclos es LL(1) si contiene la producción ϵ .
 - ¿Qué finalidad tienen las reglas *catch-all* (trágalos-todo) en el método de recuperación de errores mediante producciones de error?. Indica su forma y pon un ejemplo.
 - En el procedimiento *EvaluatePreOrder* para la obtención de los atributos heredados en una gramática de atributos, indica por qué importa el orden de evaluación de los hijos.
- Dada la siguiente gramática, construye la tabla de análisis sintáctico. ¿Es una gramática LL(1)?

$$E \rightarrow - E \mid (E) \mid V E'$$

$$E' \rightarrow - E \mid \epsilon$$

$$V \rightarrow id V'$$

$$V' \rightarrow (E) \mid \epsilon$$

- Indica si la siguiente gramática es LR(0).

$$S \rightarrow L$$

$$L \rightarrow L ; A \mid A$$

$$A \rightarrow a$$

- Diseña una gramática de atributos para evaluar el valor numérico de un número real:

$$N \rightarrow NUM . NUM$$

$$NUM \rightarrow NUM DIGIT$$

$$NUM \rightarrow \epsilon$$

$$DIGIT \rightarrow 0 \mid 1 \mid \dots \mid 9$$

Ayuda: puedes necesitar tres atributos.

- Dada la siguiente gramática de precedencia de operadores que permite generar paréntesis anidados:

$$S \rightarrow (S) \mid id$$

Se pide: (i) Obtén la tabla de precedencia de operadores. (ii) Describe el contenido de la pila y de la entrada para el análisis de la cadena $((id))$.

Procesadores de Lenguaje. Curso 2003-2004. Junio de 2004. Problemas.

Se permiten libros y apuntes. Tiempo: 1 hora 15 min.

FINAL: Contestad a las preguntas 1, 2.

SEGUNDO PARCIAL: Contestad a las preguntas 2, 3.

1. Decimos que una gramtica es $LL(k)$, si dadas dos derivaciones ms a la izquierda, tales que

$$S \xRightarrow{*} uY\alpha \Rightarrow u\beta\alpha \xRightarrow{*} ux$$

y

$$S \xRightarrow{*} uY\alpha \Rightarrow u\gamma\alpha \xRightarrow{*} uy$$

entonces si los $k : x = k : y$ implica $\beta = \gamma$.

Indica si la siguiente gramtica es $LL(k)$, para algn k .

$S \rightarrow$ if id then S else S fi |
 while id do S od |
 begin S end |
 VAR := VAR |
 id (IDLIST) (llamada a procedimiento)
 $VAR \rightarrow$ id | id(IDLIST) (acceso a elemento de un vector)
 $IDLIST \rightarrow$ id | id , IDLIST

2. Indica si la siguiente gramtica es:

i) SLR(1)

$S' \rightarrow S$
 $S \rightarrow L = R \mid R$
 $L \rightarrow * R \mid id$
 $R \rightarrow L$

ii) LALR(1)?

Para ambos casos construye el automta.

3. Supongamos la construccin de los lenguajes de programacin que ya conocis que permite ejecutar un nmero determinado de veces un bloque de sentencias. La sintaxis de esta construccin es:

$$S \rightarrow \mathbf{for\ id = E\ to\ id = E\ step\ E\ \{S*\}}$$

donde se ha usado el operador $*$ para indicar cero o ms repeticiones. Se pide:

- Dibujar el diagrama de flujo de esta construccin.
- Dibujar la forma del rbol abstracto de anlisis sintctico que usaras para su traduccin a cdigo de 3-direcciones.
- Escribir el pseudocdigo de la funcin `generar_cdigo` para traducir este tipo de sentencias a una lista de cudruplos.