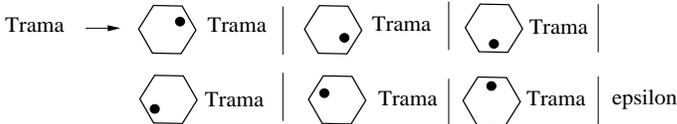
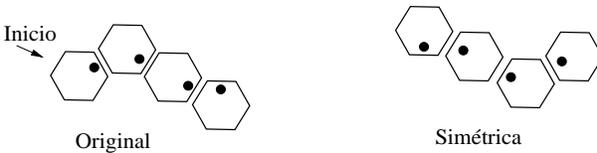


No se permiten libros ni apuntes. Tiempo: 1 hora 15 min.

- (1 ptos) ¿Cuál es la diferencia más significativa entre los métodos de análisis sintáctico LL(k) y LR(k) que hace que los métodos LR sean más poderosos y permitan reconocer un mayor número de lenguajes?.
- (1 ptos) Si tenemos atributos heredados y sintetizados. ¿Se podría realizar su cálculo usando el método de análisis sintáctico LR recorriendo el árbol de análisis sintáctico en una única pasada?
- (3 ptos) Supongamos una gramática que permite generar una trama de hexágonos. El punto significa el lado de unión con el siguiente hexágono.



- ¿ Es LL(1)? Diseña un esquema de traducción dirigido por la sintaxis (ETDS) que obtenga el número de hexágonos utilizados y genere la figura simétrica a la original. Por ejemplo:



- El análisis sintáctico es uno de los métodos de reconocimiento de patrones que permite decidir si un patrón ha podido ser generado por una cierta gramática. Construye una trama hexagonal que no pueda ser generada por la gramática.
- (3 ptos) Supongamos la construcción de los lenguajes de programación **do-while**.

$$S \rightarrow \mathbf{do} S * \mathbf{while} E \mid \epsilon$$

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

- Dibujar el diagrama de flujo de esta construcción.
  - Dibujar la forma del árbol abstracto de análisis sintáctico que usarías para su traducción a código de 3-direcciones.
  - Escribir el pseudocódigo de la función `generar_código` para traducir este tipo de sentencias a una lista de cuádruplos.
- (2 ptos) ¿ Qué se entiende por la propgación de copias y la reducción de la intensidad en la optimización de código?. Optimiza el siguiente fragmento de código aplicando el número mayor de transformaciones que se pueda:

```
t0=6
t1=2*t0
t2=t1/3
t3=t2+b
t4=a*t3
```

**Procesadores de Lenguaje.** Curso 2001-2002. Sept. de 2002. Problemas.

Se permiten libros y apuntes. Tiempo: 1 hora 30 min. Todas las preguntas valen igual.

1. (5 pts.) Supongamos la siguiente gramática que genera divisiones de números enteros y reales. La idea de esta gramática es que la operación de división se ha de interpretar de forma diferente dependiendo de si se trata de números enteros o reales. Por ejemplo:  $5/4$  puede tener el valor 1.2 ó 1, si se trata de una división en punto flotante o entera.

$$exp \rightarrow exp / exp \mid \mathbf{num\_int} \mid \mathbf{num\_real}$$

Supongamos un lenguaje de programación que requiere que las operaciones en las que aparecen números enteros combinados con reales tienen que ser promocionadas a expresiones en punto flotante. Por ejemplo: la expresión  $5/4$  es 1, mientras que el significado de  $5/4.0$  es 1.2. Se pide:

- Insertar las reglas semánticas necesarias para que se puedan distinguir entre estos dos tipos de operaciones. ¿ Cuántos atributos y de qué tipo son necesarios?
  - Dibuja el grafo de dependencias para la entrada  $5/4.0$ .
  - ¿ Es necesario dos recorridos sobre el árbol de análisis sintáctico para calcular el valor de los atributos?
  - Escribe el pseudocódigo de los procedimientos necesarios para el cálculo de estos atributos.
2. (5 pts) Dada la siguiente gramática

$$\begin{aligned} S &\rightarrow I \mid \mathbf{other} \\ I &\rightarrow \mathbf{if} E S \mid \mathbf{if} E S \mathbf{else} S \\ E &\rightarrow \mathbf{id} \end{aligned}$$

Se pide:

- Construye el autómata de elementos LR(0). Justifica si se trata de una gramática SLR(1).
- En caso de que existan conflictos, indica la acción a realizar teniendo en cuenta que la parte del **else** está asociada al **if** más cercano.
- Construye la tabla de análisis sintáctico.
- Indica un sencillo mecanismo de recuperación de errores.