

Universidad Rey Juan Carlos – Escuela Técnica Superior de Ingeniería Informática
 Grado en Ingeniería Informática – Procesadores de Lenguajes
 Prueba tema 4 – Convocatoria extraordinaria Junio 2014

Duración del examen: 1 hora.

El siguiente fragmento de gramática corresponde a un lenguaje de programación de cálculos matemáticos. Este fragmento se encarga de las operaciones aritméticas básicas (suma, resta, multiplicación y división) con números quebrados (o fracciones). Estos cálculos también darán como resultado números quebrados. A continuación se facilita la gramática, cuyo axioma es el símbolo **X**, y las fórmulas de cálculo de las operaciones con quebrados.

X ::= X MM Y | Y
MM ::= "+" | "-"
Y ::= Y PD Z | Z
PD ::= "*" | ":"
Z ::= "(" X ")" | cte "/" cte

$$\frac{a}{b} + \frac{c}{d} = \frac{(a*d+c*b)}{(b*d)} \quad \frac{a}{b} * \frac{c}{d} = \frac{(a*c)}{(b*d)}$$

$$\frac{a}{b} - \frac{c}{d} = \frac{(a*d-c*b)}{(b*d)} \quad \frac{a}{b} : \frac{c}{d} = \frac{(a*d)}{(b*c)}$$

Se desea diseñar un traductor que **calcule el valor resultante de las expresiones aritméticas** asegurándose que el símbolo **X** almacena el resultado de la expresión como número quebrado. Nótese que se debe respetar la precedencia de operadores y la asociatividad a izquierdas de los mismos.

Para los cálculos **sólo** se pueden usar (aunque no es obligatorio usar todo):

- sentencias de asignación de valores.
- sentencias de control de flujo tipo “if-then”, “if-then-else” o “switch-case”.
- los operadores lógicos and, or y not.
- los operadores de comparación menos, mayor e igual.
- Los operadores aritméticos de valores enteros de suma, resta, multiplicación y división.

Se pide:

1. **[4 ptos]** Especificar el traductor dirigido por la sintaxis descrito anteriormente.
2. **[1 pto]** Explicar de forma justificada si se ha usado una definición dirigida por la sintaxis o un esquema de traducción.
3. **[2 pto]** Identificar explícitamente los atributos utilizados y su tipo (heredados o sintetizados), ofreciendo al menos un ejemplo de cada uno extraído del traductor especificado.
4. **[3 ptos]** Explicar de forma justificada si el traductor especificado podría implementarse mediante un traductor descendente recursivo:
 - Si se pudiera, hacer el traductor descendente recursivo del **símbolo X** de la gramática.
 - Si no se pudiera, hacer las modificaciones pertinentes en el traductor para poder hacer el traductor descendente recursivo **sin implementarlo**.

Solución

A continuación se ofrece una posible solución al examen, indicando las respuestas a cada pregunta planteada:

1. Especificar el traductor dirigido por la sintaxis descrito anteriormente.

Los símbolos que deban transmitir algún valor numérico(**X**, **Y** y **Z**) deberán mantener dos atributos que llamaremos **n** y **d** para el numerador y denominador del numero quebrado respectivamente.

```
X ::= X1 MM Y {
    switch(MM.op){
        case 's': X.n = X1.n * Y.d + X1.d * Y.n;
        case 'r': X.n = X1.n * Y.d - X1.d * Y.n;
    }
    X.d = X1.d * Y.d;
}
| Y {X.n = Y.n; X.d = Y.d;}
MM ::= "+" {MM.op = "s";} | "-" {MM.op = "r";}
Y ::= Y1 PD Z {
    switch(PD.op){
        case 'p':
            Y.n = Y1.n * Z.n;
            Y.d = Y1.d * Z.d;
        case 'd':
            Y.n = Y1.n * Z.d;
            Y.d = Y1.d * Z.n;
    }
}
| Z {Y.n = Z.n; Y.d = Z.d;}
PD ::= "*" {PD.op = "p";} | ":" {PD.op = "d";}
Z ::= "(" X ")" {Z.n = X.n; Z.d = X.d;}
| cte1 "/" cte2 {Z.n = cte1.valex; Z.d = cte2.valex;}
```

2. Explicar de forma justificada si se ha usado una definición dirigida por la sintaxis o un esquema de traducción.

Se ha usado una definición dirigida por la sintaxis puesto que las acciones se asignan a consecuentes completos. También se podría decir que se ha usado un esquema de traducción donde todas las acciones semánticas se ejecutan al final de cada producción.

3. Identificar explícitamente los atributos utilizados y su tipo (heredados o sintetizados), ofreciendo al menos un ejemplo de cada uno extraído del traductor especificado.

Todos los atributos utilizados son sintetizados, puesto que son atributos del antecedente que se calculan en función de valores del consecuente. Por ejemplo el atributo **X.d** de la siguiente producción:

```
X ::= X1 MM Y { ...
    X.d = X1.d * Y.d;
}
```

4. Explicar de forma justificada si el traductor especificado podría implementarse mediante un traductor descendente recursivo.

El traductor implementado no podría implementarse mediante un traductor descendente recursivo puesto que este se basa en analizadores descendentes y la gramática del enunciado no es LL(1) ya que es recursiva por la izquierda.

“(...) hacer las modificaciones pertinentes en el **traductor** para poder hacer el traductor descendente recursivo **sin implementarlo**.”

Las producciones:

```
X ::= X MM Y | Y
Y ::= Y PD Z | Z
```

La sustituiríamos por la siguiente gramática que no es recursiva por la izquierda:

```
X ::= Y Xp
Xp ::= MM Y Xp | λ

Y ::= Z Yp
Yp ::= PD Z Yp | λ
```

Quedando el traductor de la siguiente forma:

```
X ::= Y {Xp.nh = Y.n; Xp.dh = Y.d;} Xp {X.n = Xp.ns; X.d = Xp.ds;}
Xp ::= MM Y {
    switch(MM.op){
        case 's': Xp1.nh = Xp.nh * Y.d + Xp.dh * Y.n;
        case 'r': Xp1.nh = Xp.nh * Y.d - Xp.dh * Y.n;
    }
    Xp1.dh = Xp.dh * Y.d;
} Xp1 {Xp.ns = Xp1.ns; Xp.ds = Xp1.ds;}
| λ {Xp.ns = Xp.nh; Xp.ds = Xp.dh;}

Y ::= Z {Yp.nh = Z.n; Yp.dh = Z.d;} Yp {Y.n = Yp.ns; Y.d = Yp.ds;}

Yp ::= PD Z {
    switch(PD.op){
        case 'p':
            Yp1.n = Yp.n * Z.n;
            Yp1.d = Yp.d * Z.d;
        case 'd':
            Yp1.n = Yp.n * Z.d;
            Yp1.d = Yp.d * Z.n;
    }
} Yp1 {Yp.ns = Yp1.ns; Yp.ds = Yp1.ds;}
| λ {Yp.ns = Yp.nh; Yp.ds = Yp.dh;}
```

Criterios de corrección

El traductor

Fallos muy graves:

- Usar atributos sin valor asignado, sin que falte ninguna producción.

Fallos graves:

- Usar atributos sin valor asignado, por falta de producción correspondiente.
- Asignar valores a atributos con posteriormente no se usan o no se pueden usar.

Fallos leves:

- Cálculo erróneo de las operaciones.

Cuestiones

Fallos muy graves:

- No justificar las respuestas.
- Justificar lo contrario de lo que se responde.