

Ejercicios sobre aspectos sintácticos y gramáticas incontextuales.

[Febrero 2007] Considera los siguientes operadores: un operador binario no asociativo **&**, un operador unario prefijo **@** y un operador binario asociativo a derechas *****. Además, **&** y **@** tienen igual prioridad, y menor que *****. Formaliza la sintaxis para las expresiones formadas mediante estos operadores. Las expresiones básicas son únicamente las constantes numéricas, y, como es habitual, pueden utilizarse paréntesis para alterar las prioridades y asociatividades. La sintaxis debe reflejar las prioridades y asociatividades de los operadores.

```
Exp->Fact&Term
Exp->@Fact
Exp->@Term
Fact->Val*Fact
Fact->Val
Term->Val*Term
Term->Val
Val->(Exp)
Val->Num
```

[Septiembre de 2007] Considera la siguiente gramática para las expresiones booleanas:

```
ExpB --> TermB or ExpB
ExpB --> TermB
TermB --> TermB and FactB
TermB --> not FactB
TermB --> FactB
FactB --> true
FactB --> false
FactB --> ( ExpB )
```

¿Cuáles son las prioridades y las asociatividades *naturales* de los operadores **or**, **and** y **not** caracterizadas por esta sintaxis? Razona tu respuesta.

Or -> asocia a derechas y es un operador binario
And -> asocia a izquierdas y es un operador binario
Not-> no asocia, es un prefijo y es un operador unario

Las prioridades son: not es la mas prioritaria , luego va la and y luego la or.

[Junio 2006] Considera los dos siguientes operadores binarios: ***** y **+**. ***** tiene mayor prioridad que **+** y asocia a izquierdas. Por su parte, **+** asocia a derechas. Formaliza la sintaxis para las expresiones formadas mediante los operadores ***** y **+**. Las expresiones básicas pueden ser números y variables, y, como es habitual, pueden utilizarse paréntesis para alterar las prioridades y asociatividades. La sintaxis debe reflejar las prioridades y asociatividad de los operadores.

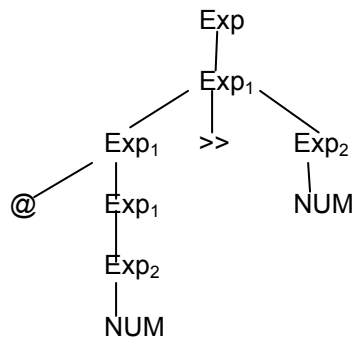
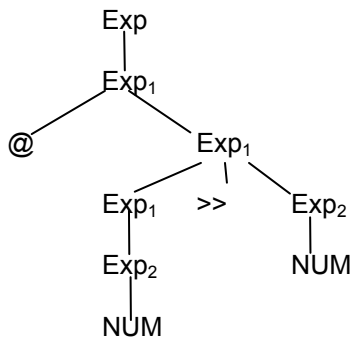
```
Exp-> Fact + Exp
Exp->Fact
Fact->Fact*Term
Fact->Term
Term->(Exp)
Term->Num
Term->Var
```

[Junio 2007] Se desea diseñar un lenguaje que conste de expresiones construidas a partir de números, los operadores binarios = y >>, y el operador unario prefijo @. Se desea que = asocie a derechas, >> asocie a izquierdas, y @ también asocie. Se desea, además, que @ y >> tengan la misma prioridad, y que ésta sea mayor que la de =. Como es habitual, pueden utilizarse paréntesis para alterar las prioridades y asociatividades.

- Formaliza la sintaxis para las expresiones formadas mediante estos operadores, de tal forma que dicha sintaxis refleje las prioridades y asociatividades de los mismos.
- La especificación exigida tiene un problema. Identifícalo y propón modificaciones que lo solucionen.

```
Exp->Fact = Exp
Exp->Fact
Fact->Fact>>Term
Fact->Fact@Term
Fact->Term@Fact
Fact->Term
Term->(Exp)
Term->Num
```

@NUM >> NUM



El árbol que queda al construir la gramática es un árbol que da lo mismo leas a derechas que a izquierdas que se puede construir el mismo lenguaje porque la gramática es ambigua.

Y por tanto el problema lo causa que @ y >> tienen el mismo valor y @ da igual a que lado asocie.

[Junio 2007] Escribe una gramática no ambigua que genere el lenguaje de las expresiones regulares, enunciando explícitamente cuáles son las precedencias y asociatividades que la gramática induce en los operadores.

```
Exp0 ::= Exp1 Op Exp0
Exp0 ::= Exp1
Op0 ::= =
Exp1 ::= Exp1 Op1 Exp2
Exp1 ::= Op2 Exp1
Exp1 ::= Exp2
Op1 ::= >>
Op2 ::= @
Exp2 ::= num
Exp2 ::= id
Exp2 ::= (Exp0)
```