

Parte II: Analizador semántico

Objetivo:

Incorporar acciones semánticas en Bison para realizar comprobaciones semánticas.

Fecha límite entrega Parte II:

26/11/2017

Modo de entrega:

- Grupo
- A través de una tarea PoliformaT

Bison II: Atributos y acciones semánticas

Lenguajes de Programación y
Procesadores de Lenguajes

Objetivos

1. Incorporar acciones semánticas.
2. Definir los tipos de los atributos
3. Asociar atributos sintetizados a los símbolos

1. Definir tipos de atributos

Se definen con `%union`

```
%union {  
    tipoC1 nombre_tipo1 ;  
    tipoC2 nombre_tipo2 ;  
    ...  
}
```

¿ Y si un símbolo necesita más de un atributo ?

Definir un tipo estructura con tantos campos como atributos necesite

Ejemplo:

```
%union {  
    char    *ident ;  
    int     cent ;  
    float   creal ;  
}
```

2. Asociar tipos a símbolos

- Cada símbolo solo tiene un tipo de atributo
- Atributos de terminales:

`%token <nombre_tipo> <símbolo terminal>`

Devolver desde Flex en la variable `yylval`

- Atributos de no-terminales:

`%type <nombre_tipo> <símbolo no-terminal>`

Ejemplo:

```
%token    WHILE_  
%token    <cent>    CTE_  
%left     OPSUMA_  
%type     <cent>    expresion
```

3. Acciones semánticas

- Sentencias C entre llaves
- Aparecen entre símbolos del lado derecho de regla
- $\$n$ Atributo asociado al símbolo n-ésimo del lado derecho
- $$$$ en la última acción: Atributo del no-terminal del lado izquierdo
- Acción por defecto: $\{ \$\$ = \$1 ; \}$

Ejemplo:

```
expresion : expresion OPSUMA_ termino { $$ = $1 + $3 }
```

Acciones semánticas a mitad de regla

- La acción semántica cuenta como símbolo
- \$\$ a mitad de regla asigna valor a la **propia acción** semántica
- Tipo de la acción semántica indicado explícitamente al usarlo

Ejemplo:

```
f : expresion      { $<cent>$ = $1 ; }  
OPSUMA_ expresion { $$ = $<cent>2 + $4 ; }
```