
Bison II: Resolución de conflictos y errores

Lenguajes de Programación y
Procesadores de Lenguajes

Objetivos

- Aprender a analizar y resolver conflictos en el analizador sintáctico.
- Conocer el mecanismo de recuperación de errores sintácticos empleado por Bison.

Resolución de conflictos

Depuración en Bison

Opción -v

da información sobre el analizador: reglas, estados, conflictos, autómata LALR(1), ...

Ejemplo:

```
bison -d -v -o ejemplo ejemplo.y
```

Generará:

```
ejemplo.c  
ejemplo.h  
ejemplo.output
```

Conflictos reducción/reducción

Se elige la regla que **aparece primero** en la gramática

Conflictos reducción/desplazamiento

Por defecto desplaza

Declaración de asociatividad:

```
%left    <tokens asociativos a izquierda>
%right    <tokens asociativos a derecha >
%nonassoc <tokens no-asociativos>
```

Declaración de precedencia:

Depende del orden en el que se declaran los símbolos con %left o %right (de menor a mayor).

Ejemplo:

```
%token PARA_ PARC_
%token CTE_
%left MAS_ MENOS_
%left POR_ DIV_
```

Resolución conflicto r/d

- Cada **regla** toma la precedencia del **último terminal** que aparece en su lado derecho.
- Resolver conflicto comparando precedencia de la regla considerada y la precedencia del token de anticipación:

if precedencia del token > precedencia de la regla
then desplaza

if precedencia del token < precedencia de la regla **then** reduce

if precedencia del token = precedencia de la regla
then considera la asociatividad de ese nivel de precedencia

Recuperación de errores

Recuperación de errores

1. Ante un error se genera el token "error"
2. Desapilará estados y símbolos hasta encontrar un estado en el que se acepte el token "error".
3. Se desplaza el token "error".
4. Saltará todos los token hasta encontrar uno con el que se pueda desplazar o una reducir.
5. Para evitar bucles infinitos, a veces se consume un símbolo de la entrada.

Recuperación de errores

Ejemplo:

```
expMat : exp
      ;
exp    : exp MAS_ term
      | exp MENOS_ term
      | term
      ;
term   : term POR_ fac
      | term DIV_ fac
      | fac
      ;
fac    : PARA_ exp PARC_
      | PARA_ error PARC_
      | CTE_
      ;
```

Tras la recuperación, el A.S. no mostrará mensajes de error hasta **procesar con éxito 3** tokens consecutivos.