

Compiladors

(Examen final, 16 de juny de 2022)

1 Parsing (1/3)

Donada la gramàtica següent :

- (1) $S \rightarrow E$
- (2) $E \rightarrow -E$
- (3) $E \rightarrow id [E]$
- (4) $E \rightarrow E !$
- (5) $E \rightarrow id$

- Calcula les taules *Nullable*, *First* i *Follow* per a tots els símbols no terminals.

| <i>Nullable</i> | | |
|-----------------|------|-------|
| | true | false |
| <i>S</i> | | |
| <i>E</i> | | |

| <i>First</i> | | | | | |
|--------------|---|----|---|---|---|
| | - | id | [|] | ! |
| <i>S</i> | | | | | |
| <i>E</i> | | | | | |

| <i>Follow</i> | | | | | | |
|---------------|---|----|---|---|---|----|
| | - | id | [|] | ! | \$ |
| <i>S</i> | | | | | | × |
| <i>E</i> | | | | | | |

- Genera l'autòmat i la taula SLR(1).

0)

$S \rightarrow \cdot E$

1)

4)

2)

5)

3)

6)

7)

8)

| state | action | | | | | | goto |
|-------|--------|----|---|---|---|----|------|
| | - | id | [|] | ! | \$ | E |
| 0 | | | | | | | |
| 1 | | | | | | | |
| 2 | | | | | | | |
| 3 | | | | | | | |
| 4 | | | | | | | |
| 5 | | | | | | | |
| 6 | | | | | | | |
| 7 | | | | | | | |
| 8 | | | | | | | |

- Indica breument la raó del conflicte en la taula SLR(1). Posa un senzill exemple d'entrada que el provoqui, i descriu el contingut de la pila i de l'entrada quan el conflicte es produeix.

2 Notació polonesa inversa (1/3)

Volem fer un analitzador d'expressions aritmètiques en notació postfix (*RPN* en anglès). A continuació tenim alguns exemples d'expressions en notació algebraica, en *RPN*, i el seu resultat:

| notació algebraica | notació <i>RPN</i> | valor |
|-----------------------------------|-----------------------|-------|
| $(17 - 4)/(11 - 10)$ | 17 4 - 11 10 - / | 13 |
| $5 + ((1 + 2) * 4) - 3$ | 5 1 2 + 4 * + 3 - | 14 |
| $8/2 - (6 \wedge (1 \wedge 2))/3$ | 8 2 / 6 1 2 ^ ^ 3 / - | 2 |
| $7 - (3! * 2)$ | 7 3 ! 2 * - | -5 |

L'avaluació d'expressions en *RPN* fa servir una **pila** de dades on es es pugen els operands, que després són operats i substituïts per resultat. Per exemple en l'avaluació de l'expressió del primer exemple, el contingut de la pila serà (*top* a la dreta):

- (1) *empty*
- (2) 17
- (3) 17 4
- (4) 13
- (5) 13 11
- (6) 13 11 10
- (7) 13 1
- (8) 13

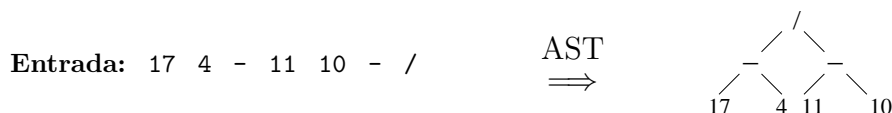
- Dissenya una gramàtica BNF que accepti expressions en *RPN*. L'alfabet d'entrada és $\Sigma = \{\text{num}, +, -, *, /, \wedge, !\}$.

Volem calcular tres atributs d'aquestes expressions, donant a l'operador / la semàntica de quocient de la *divisió entera*:

- a) el valor resultant de l'avaluació (atribut *eval*)
- b) el mínim valor que hi ha a la pila al llarg del procés d'avaluar-la (atribut *minim*)
- c) la màxima altura de la pila al llarg de l'avaluació (atribut *height*)

En l'exemple anterior els valors dels atributs de l'expressió sencera són *eval*=13, *minim*=1 i *height*=3.

Suposem que la gramàtica genera un AST amb 6+1 tipus de nodes corresponents als 6 operadors existents, més el node **num**. A continuació es mostra un exemple de l'AST que s'obtingria a partir d'una entrada. Cal observar que l'estructura de l'AST no correspon a l'estructura sintàctica de l'entrada.



- Dissenya les funcions $\text{RPN}(N)$ que visita l'AST i defineix els atributs *N.eval*, *N.minim* i *N.height* de cada node *N*. Considerar les 6+1 possibles versions de la funció. Podeu fer servir aquests mateixos operadors pels càlculs a dins de les funcions, i/o també funcions com *min*, *max*, *pot* o *fact*.

```

RPN( $N \rightarrow E_1 + E_2$ )  {1. suma}
RPN( $N \rightarrow E_1 - E_2$ )  {2. resta}
...
RPN( $N \rightarrow \text{num}$ )      {num és un enter}

```

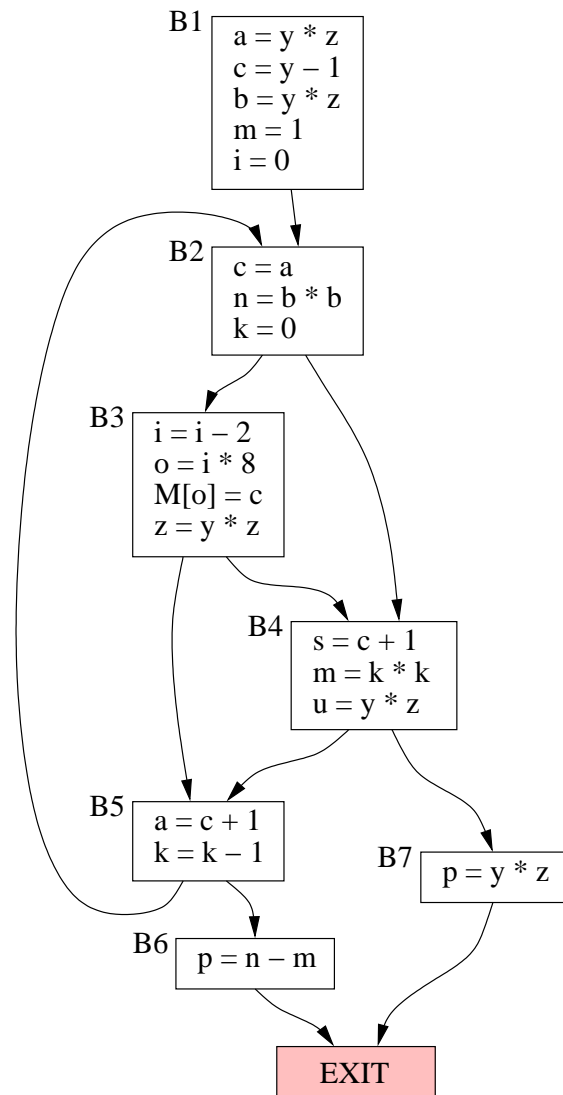
```

RPN( $N \rightarrow \text{num}$ ) {
  N.eval  = num.intval;
  N.minim = ... ;
  N.height = ... ;
}

```

3 Optimització (1/3)

Considerem el següent diagrama de blocs bàsics:



Suposem que totes les variables han estat inicialitzades abans del bloc d'entrada i que les variables vives en el bloc de sortida són M , a , k , p , u , z .

Es demana:

- quines són les variables vives a l'inici del bloc B4
- quins blocs dominen el bloc B5
- aplicar totes les optimitzacions possibles explicant el raonament utilitzat per a cada optimització. Dibuixar el graf amb el codi final simplificat.