

Compiladors

(Examen final, 18 de juny de 2021)

1 Parsing (1/3)

Donada la gramàtica següent :

- (1) $S \rightarrow T\$$
- (2) $T \rightarrow R$
- (3) $T \rightarrow aTc$
- (4) $R \rightarrow$
- (5) $R \rightarrow bR$

- Descriu el llenguatge que genera.
- Calcula *First* i *Follow* per a tots els símbols no terminals. Afegeix ϵ al *First* dels símbols anul·lables.
- Generar l'autòmat i la taula LR(1).
- Indica en quin estat es trobarà l'autòmat quan detecti un error en llegir l'entrada *abcc\$*. Indica el símbols que hi haurà a la pila en aquell estat.

2 Identificadors amb subíndexos (1/3)

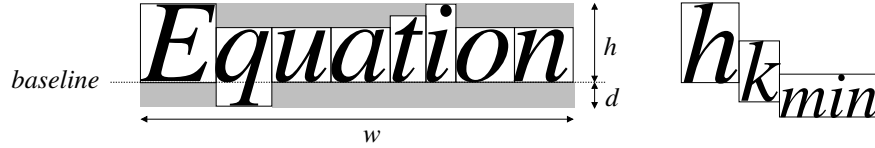
Volem fer un analitzador d'identificadors amb subíndexos per a un llenguatge que s'utilitza per a la descripció d'equacions. Aquí hi ha alguns exemples del llenguatge d'identificadors amb subíndexos.

Entrada	Visualització	Entrada	Visualització
<code>lambda</code>	λ	<code>Vector[i[13][min]]</code>	$Vector_{i_{13,min}}$
<code>a[i]</code>	a_i	<code>C[1][2][arg[3]]</code>	$C_{1,2,arg_3}$
<code>Matriu[arg[i]]</code>	$Matriu_{arg_i}$	<code>M[fab[3]][2]</code>	$M_{fab_3,2}$
<code>a[i][j[min]]</code>	$a_{i,j_{min}}$	<code>Vector[1[i]]</code>	<code>String il·legal</code>
<code>Vector[i[min]]</code>	$Vector_{i_{min}}$		

En aquest llenguatge, els identificadors només poden contenir lletres. Els subíndexos poden ser identificadors o números enters sense signe. Els subíndexos que siguin identificadors poden tenir més subíndexos. En canvi, els números no poden tenir més subíndexos (veure el darrer exemple).

- Dissenya una gramàtica EBNF que accepti un identificador amb subíndexos. L'alfabet d'entrada és $\Sigma = \{a \dots z, A \dots Z, 0 \dots 9, [,]\}$.

Volem calcular els atributs de la *bounding box* d'un string del llenguatge. Donat un tamany de font (point size (p)): 8, 10, 12, ...), el tamany de cada string es caracteritza amb tres atributs: *height* (h), *depth* (d) i *width* (w). Els atributs h i d representen l'alçada i la profunditat del string respecte a una línia base on descansa el text (veure la figura). Els valors w , h i d sempre són positius.



Cada vegada que especifiquem un subíndex, el tamany de font (p) és reduït un 30% i la línia base del subíndex es desplaça $0.25p$ cap avall, on p és el tamany de font del pare. El tamany de font pot tenir qualsevol valor real (per exemple, 11.83).

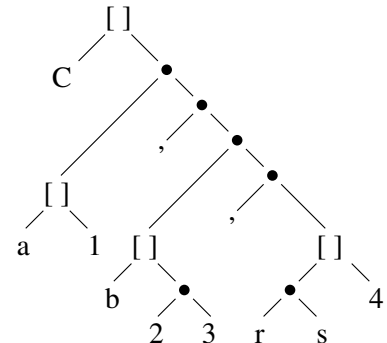
Suposem que la gramàtica genera un AST amb tres tipus de nodes: s (símbol), \bullet (concatenació) i $[]$ (subíndex), on s són els nodes que hi ha a les fulles. Cada node N de l'AST conté tres atributs, $N.h$, $N.d$ i $N.w$, que representen les dimensions del string associat al node. Cada node N de tipus s conté un atribut $N.val$ que enmagatzema el símbol que representa (per exemple, 'a', '3', ',', ...). Donat un tamany de font p i un símbol s , podem obtenir el valor dels seus atributs amb les funcions $Getw(p,s)$, $Geth(p,s)$ i $Getd(p,s)$.

A continuació es mostra un exemple de l'AST que s'obtindria a partir d'una entrada. Cal observar que l'estructura de l'AST no correspon a l'estructura sintàctica de l'entrada.

Entrada: `C[a[1]][b[23]][rs[4]]`

Visualització: C_{a_1,b_{23},rs_4}

AST
 \Rightarrow

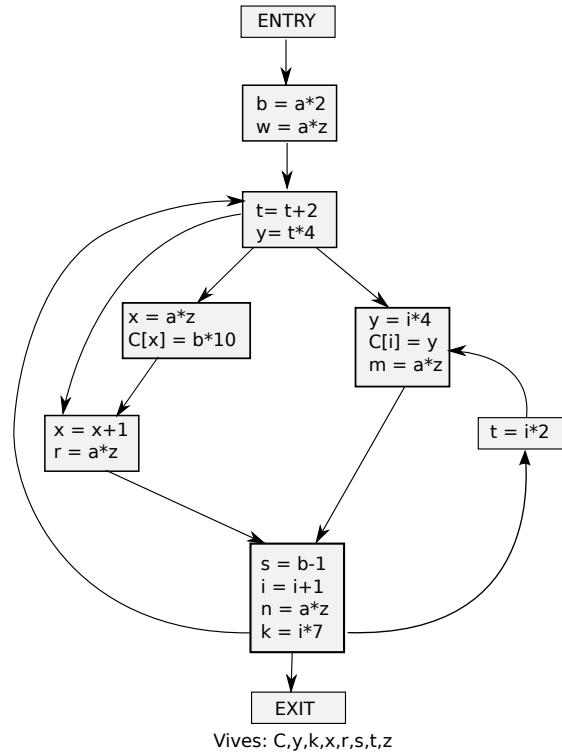


- Dissenya la funció $BBox(p, N)$ que visita l'AST i defineix els atributs $N.h$, $N.d$ i $N.w$ de cada node N per a un tamany de font p donat. Considerar les tres possibles versions de la funció:

$BBox(p, N \rightarrow s)$ $\{s \text{ és un símbol}\}$
 $BBox(p, N \rightarrow B_1 \bullet B_2)$ $\{\text{concatenació de strings}\}$
 $BBox(p, N \rightarrow B_1[B_2])$ $\{\text{subíndex}\}$

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 $\{C, y, k, x, r, s, t, z\}$. Es demana aplicar totes les optimitzacions possibles explicant el raonament utilitzat per a cada optimització.