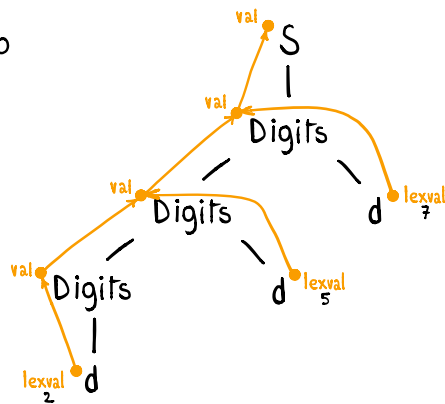


Esercizio 1

Da una stringa di caratteri che rappresentano cifre, ottenere il valore del numero.

```
S -> Digits { S.val = Digits.val }
Digits -> Digits1 d { Digits.val = Digits1.val * 10 + d.lexval }
Digits -> d { Digits.val = d.lexval }
```

ESEMPIO

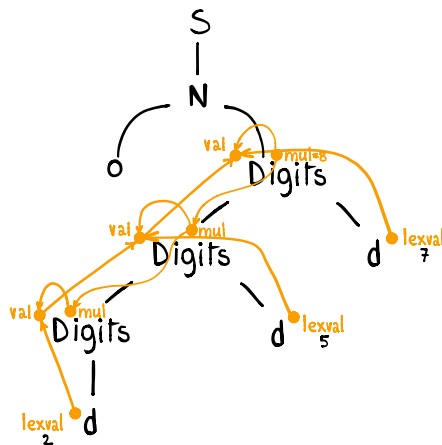


L'SDD è S-attribuito: solo attributi sintetizzati:
Quindi con un post-ordine si può valutare.

Esercizio 2

Come prima, solo che la stringa può (o no) cominciare con un prefisso 'o' che indica che i caratteri rappresentano cifre ottali anziché decimali.

```
S -> N { S.val = N.val }
N -> o Digits { N.val = Digits.val, Digits.mul = 8 }
N -> Digits { N.val = Digits.val, Digits.mul = 10 }
Digits -> Digits1 d { Digits.val = Digits.mul * Digits1.val + d.lexval, Digits1.mul = Digits.mul }
Digits -> d { Digits.val = d.lexval }
```



Esercizio 3

Un robottino, rappresentato da un punto, che parte dall'incrocio degli assi cartesiani, si può muovere in su, in giù, a sinistra e a destra. Data una stringa di caratteri che rappresentano gli spostamenti del robottino (WASD), restituire la posizione finale.

```
S -> Moves
Moves -> w Moves1 { Moves.y = Moves1.y + 1, Moves.x = Moves1.x }
Moves -> s Moves1 { Moves.y = Moves1.y - 1, Moves.x = Moves1.x }
Moves -> a Moves1 { Moves.y = Moves1.y, Moves.x = Moves1.x - 1 }
Moves -> d Moves1 { Moves.y = Moves1.y, Moves.x = Moves1.x + 1 }
Moves -> esilon { Moves.x = 0, Moves.y = 0 }
```