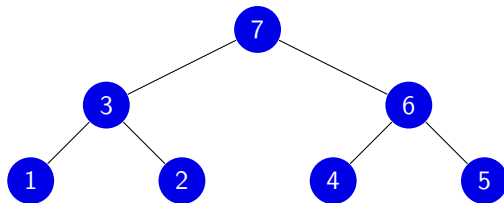


# CM 4 de Compilation : Parseurs remontants (LR)

Flavien BREUVART

February 18, 2022

# Leftmost-innermost / LR



lecture/décodage	écriture/encodage	modification
profondeur gauche remontant	suffix	leftmost-innermost

# Parseur LR en action

## Construction de l'arbre profondeur gauche remontant

### Grammaire :

$$\begin{aligned} \langle \text{expr} \rangle &::= \langle \text{expr} \rangle - \langle \text{expr} \rangle \\ &| \langle \text{expr} \rangle * \langle \text{expr} \rangle \\ &| ( \langle \text{expr} \rangle ) \\ &| - \langle \text{expr} \rangle \\ &| \langle \text{NB} \rangle \end{aligned}$$

### Reconstruction de l'arbre :

$$\begin{array}{ccccccc} \langle \text{NB} \rangle & * & \langle \text{NB} \rangle & - & \langle \text{NB} \rangle & * & \langle \text{NB} \rangle \\ | & & | & & | & & | \\ 3 & * & 7 & - & 42 & * & 0 \end{array}$$

# Parseur LR en action

## Construction de l'arbre profondeur gauche remontant

### Grammaire :

$\langle \text{expr} \rangle ::= \langle \text{expr} \rangle - \langle \text{expr} \rangle$   
|  $\langle \text{expr} \rangle * \langle \text{expr} \rangle$   
|  $( \langle \text{expr} \rangle )$   
|  $- \langle \text{expr} \rangle$   
|  $\langle \text{NB} \rangle$

### Reconstruction de l'arbre :

$\langle \text{expr} \rangle$   
|  
 $\langle \text{NB} \rangle * \langle \text{NB} \rangle - \langle \text{NB} \rangle * \langle \text{NB} \rangle$   
| | | | | |  
3 \* 7 - 42 \* 0

# Parseur LR en action

## Construction de l'arbre profondeur gauche remontant

### Grammaire :

$\langle \text{expr} \rangle ::= \langle \text{expr} \rangle - \langle \text{expr} \rangle$   
|  $\langle \text{expr} \rangle * \langle \text{expr} \rangle$   
|  $( \langle \text{expr} \rangle )$   
|  $- \langle \text{expr} \rangle$   
|  $\langle \text{NB} \rangle$

### Reconstruction de l'arbre :

$\langle \text{expr} \rangle \quad \langle \text{expr} \rangle$   
|                    |  
 $\langle \text{NB} \rangle * \langle \text{NB} \rangle - \langle \text{NB} \rangle * \langle \text{NB} \rangle$   
|                    |                    |                    |  
3                    \*                    7                    -                    42                    \*                    0

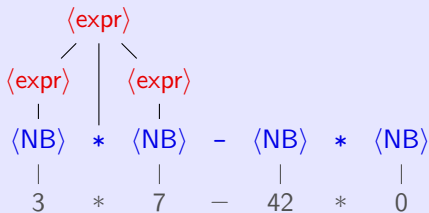
# Parseur LR en action

## Construction de l'arbre profondeur gauche remontant

### Grammaire :

$\langle \text{expr} \rangle ::= \langle \text{expr} \rangle - \langle \text{expr} \rangle$   
 $\quad \quad \quad | \langle \text{expr} \rangle * \langle \text{expr} \rangle$   
 $\quad \quad \quad | ( \langle \text{expr} \rangle )$   
 $\quad \quad \quad | - \langle \text{expr} \rangle$   
 $\quad \quad \quad | \langle \text{NB} \rangle$

### Reconstruction de l'arbre :



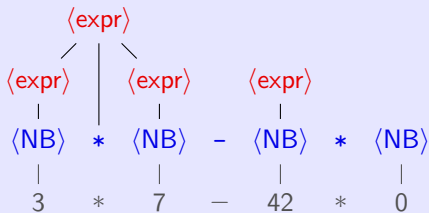
# Parseur LR en action

## Construction de l'arbre profondeur gauche remontant

### Grammaire :

$\langle \text{expr} \rangle ::= \langle \text{expr} \rangle - \langle \text{expr} \rangle$   
 $\quad | \langle \text{expr} \rangle * \langle \text{expr} \rangle$   
 $\quad | ( \langle \text{expr} \rangle )$   
 $\quad | - \langle \text{expr} \rangle$   
 $\quad | \langle \text{NB} \rangle$

### Reconstruction de l'arbre :



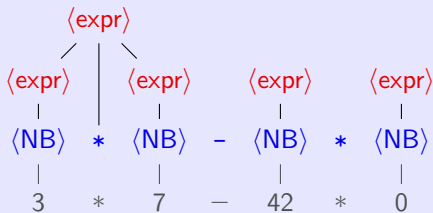
# Parseur LR en action

## Construction de l'arbre profondeur gauche remontant

### Grammaire :

$\langle \text{expr} \rangle ::=$   
|  $\langle \text{expr} \rangle - \langle \text{expr} \rangle$   
|  $\langle \text{expr} \rangle * \langle \text{expr} \rangle$   
|  $( \langle \text{expr} \rangle )$   
|  $- \langle \text{expr} \rangle$   
|  $\langle \text{NB} \rangle$

### Reconstruction de l'arbre :





# Parseur LR en action

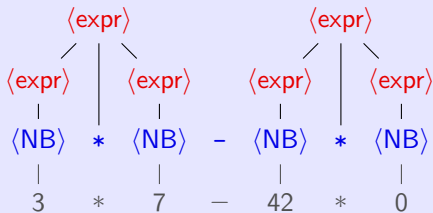
## Construction de l'arbre profondeur gauche remontant

### Grammaire :

$\langle \text{expr} \rangle ::=$ 

- $\langle \text{expr} \rangle - \langle \text{expr} \rangle$
- $\langle \text{expr} \rangle * \langle \text{expr} \rangle$
- $( \langle \text{expr} \rangle )$
- $- \langle \text{expr} \rangle$
- $\langle \text{NB} \rangle$

### Reconstruction de l'arbre :



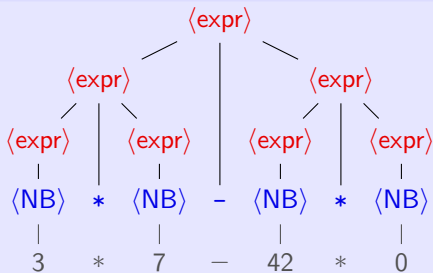
# Parseur LR en action

## Construction de l'arbre profondeur gauche remontant

### Grammaire :

$\langle \text{expr} \rangle ::=$   
|  $\langle \text{expr} \rangle - \langle \text{expr} \rangle$   
|  $\langle \text{expr} \rangle * \langle \text{expr} \rangle$   
|  $( \langle \text{expr} \rangle )$   
|  $- \langle \text{expr} \rangle$   
|  $\langle \text{NB} \rangle$

### Reconstruction de l'arbre :



# Parseur LR en action

Le parseur ne lit que les noms des tokens construits

Sur un arbre on ne regarde que la racine.

## Grammaire :

$\langle \text{exp} \rangle ::= \langle \text{exp} \rangle - \langle \text{terme} \rangle$   
|  $\langle \text{terme} \rangle$

$\langle \text{terme} \rangle ::= \langle \text{terme} \rangle * \langle \text{fac} \rangle$   
|  $\langle \text{fac} \rangle$

$\langle \text{fac} \rangle ::= (\langle \text{exp} \rangle)$   
|  $-\langle \text{fac} \rangle$   
|  $\langle \text{NB} \rangle$

## Reconstruction de l'arbre :

$\langle \text{NB} \rangle$	*	$\langle \text{NB} \rangle$	-	$\langle \text{NB} \rangle$	*	$\langle \text{NB} \rangle$
3		7		42		0

# Parseur LR en action

Le parseur ne lit que les noms des tokens construits

Sur un arbre on ne regarde que la racine.

## Grammaire :

```

<exp> ::= <exp>-<terme>
        | <terme>
<terme> ::= <terme>*<fac>
          | <fac>
<fac> ::= (<exp>)
        | -<fac>
        | <NB>
  
```

## Reconstruction de l'arbre :

$\langle \text{fac} \rangle$	*	$\langle \text{NB} \rangle$	-	$\langle \text{NB} \rangle$	*	$\langle \text{NB} \rangle$
$\langle \text{NB} \rangle$		7		42		0
3						

# Parseur LR en action

Le parseur ne lit que les noms des tokens construits

Sur un arbre on ne regarde que la racine.

## Grammaire :

$$\begin{aligned} \langle \text{exp} \rangle &::= \langle \text{exp} \rangle - \langle \text{terme} \rangle \\ &| \langle \text{terme} \rangle \\ \langle \text{terme} \rangle &::= \langle \text{terme} \rangle * \langle \text{fac} \rangle \\ &| \langle \text{fac} \rangle \\ \langle \text{fac} \rangle &::= (\langle \text{exp} \rangle) \\ &| - \langle \text{fac} \rangle \\ &| \langle \text{NB} \rangle \end{aligned}$$

## Reconstruction de l'arbre :

$\langle \text{terme} \rangle$	*	$\langle \text{NB} \rangle$	-	$\langle \text{NB} \rangle$	*	$\langle \text{NB} \rangle$
$\langle \text{fac} \rangle$		7		42		0
$\langle \text{NB} \rangle$						
3						

# Parseur LR en action

Le parseur ne lit que les noms des tokens construits

Sur un arbre on ne regarde que la racine.

## Grammaire :

$$\begin{aligned} \langle \text{exp} \rangle &::= \langle \text{exp} \rangle - \langle \text{terme} \rangle \\ &\quad | \langle \text{terme} \rangle \\ \langle \text{terme} \rangle &::= \langle \text{terme} \rangle * \langle \text{fac} \rangle \\ &\quad | \langle \text{fac} \rangle \\ \langle \text{fac} \rangle &::= (\langle \text{exp} \rangle) \\ &\quad | - \langle \text{fac} \rangle \\ &\quad | \langle \text{NB} \rangle \end{aligned}$$

## Reconstruction de l'arbre :

$\langle \text{terme} \rangle$	*	$\langle \text{fac} \rangle$	-	$\langle \text{NB} \rangle$	*	$\langle \text{NB} \rangle$
$\langle \text{fac} \rangle$		$\langle \text{NB} \rangle$		42		0
$\langle \text{NB} \rangle$		7				
3						

# Parseur LR en action

Le parseur ne lit que les noms des tokens construits

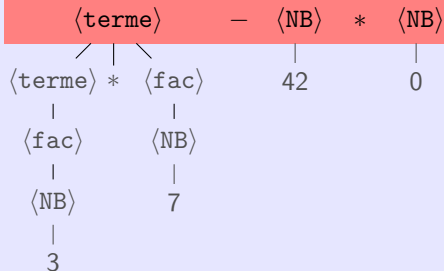
Sur un arbre on ne regarde que la racine.

## Grammaire :

```

<exp> ::= <exp>-<terme>
        | <terme>
<terme> ::= <terme>*<fac>
          | <fac>
<fac> ::= (<exp>)
        | -<fac>
        | <NB>
  
```

## Reconstruction de l'arbre :



# Parseur LR en action

Le parseur ne lit que les noms des tokens construits

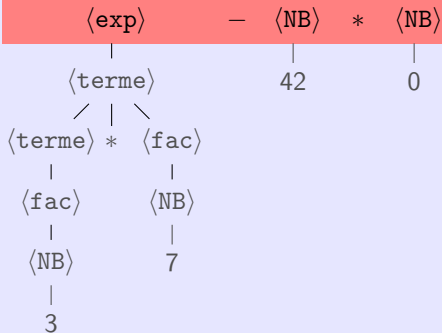
Sur un arbre on ne regarde que la racine.

## Grammaire :

```

<exp> ::= <exp>-<terme>
        | <terme>
<terme> ::= <terme>*<fac>
          | <fac>
<fac> ::= (<exp>)
        | -<fac>
        | <NB>
  
```

## Reconstruction de l'arbre :





# Parseur LR en action

Le parseur ne lit que les noms des tokens construits

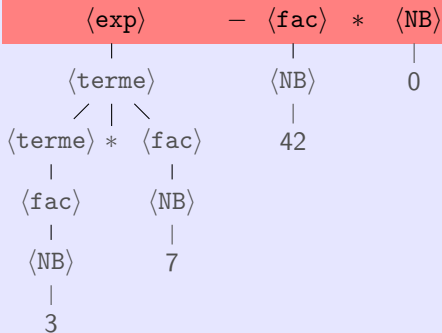
Sur un arbre on ne regarde que la racine.

## Grammaire :

```

<exp> ::= <exp>-<terme>
        | <terme>
<terme> ::= <terme>*<fac>
          | <fac>
<fac> ::= (<exp>)
        | -<fac>
        | <NB>
  
```

## Reconstruction de l'arbre :



# Parseur LR en action

Le parseur ne lit que les noms des tokens construits

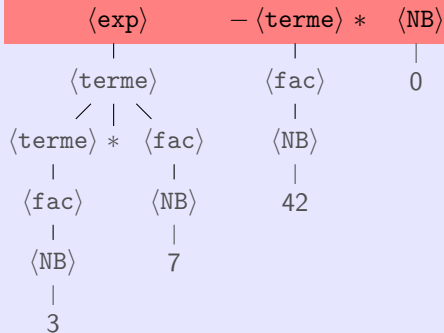
Sur un arbre on ne regarde que la racine.

## Grammaire :

```

<exp> ::= <exp>-<terme>
        | <terme>
<terme> ::= <terme>*<fac>
          | <fac>
<fac> ::= (<exp>)
        | -<fac>
        | <NB>
  
```

## Reconstruction de l'arbre :



# Parseur LR en action

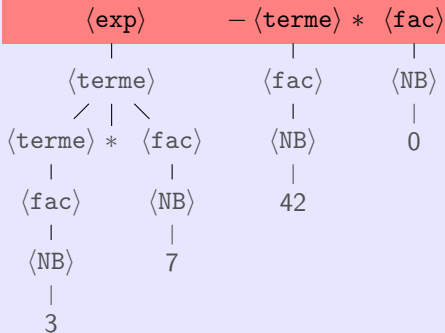
Le parseur ne lit que les noms des tokens construits

Sur un arbre on ne regarde que la racine.

## Grammaire :

$$\begin{aligned} \langle \text{exp} \rangle &::= \langle \text{exp} \rangle - \langle \text{terme} \rangle \\ &\quad | \langle \text{terme} \rangle \\ \langle \text{terme} \rangle &::= \langle \text{terme} \rangle * \langle \text{fac} \rangle \\ &\quad | \langle \text{fac} \rangle \\ \langle \text{fac} \rangle &::= (\langle \text{exp} \rangle) \\ &\quad | - \langle \text{fac} \rangle \\ &\quad | \langle \text{NB} \rangle \end{aligned}$$

## Reconstruction de l'arbre :



# Parseur LR en action

Le parseur ne lit que les noms des tokens construits

Sur un arbre on ne regarde que la racine.

## Grammaire :

$\langle \text{exp} \rangle ::= \langle \text{exp} \rangle - \langle \text{terme} \rangle$

|  $\langle \text{terme} \rangle$

$\langle \text{terme} \rangle ::= \langle \text{terme} \rangle * \langle \text{fac} \rangle$

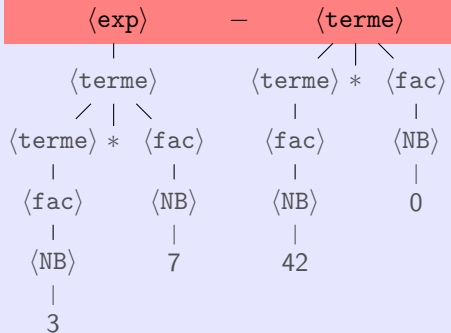
|  $\langle \text{fac} \rangle$

$\langle \text{fac} \rangle ::= (\langle \text{exp} \rangle)$

|  $-\langle \text{fac} \rangle$

|  $\langle \text{NB} \rangle$

## Reconstruction de l'arbre :



# Parseur LR en action

Le parseur ne lit que les noms des tokens construits

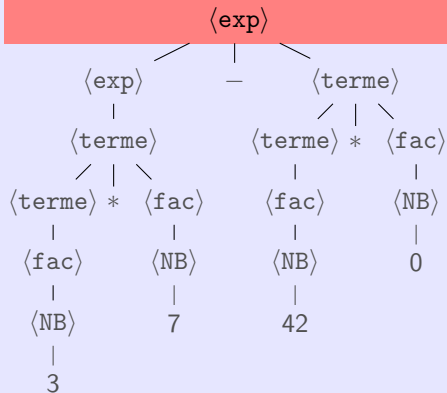
Sur un arbre on ne regarde que la racine.

## Grammaire :

```

<exp> ::= <exp>-<terme>
        | <terme>
<terme> ::= <terme>*<fac>
          | <fac>
<fac> ::= (<exp>)
        | -<fac>
        | <NB>
  
```

## Reconstruction de l'arbre :



# Principe du parseur non déterministe

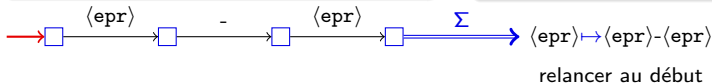
Idée : écrire un automate shift-action non déterministe

## Shifts :

- lisent un mode de terminaux et NT
- y cherchent le pattern d'une règle.

## Action :

- applique la règle trouvée,
- forme un noeud,
- relance l'automate au début.



Remarque :  $\Sigma$  remplace Char

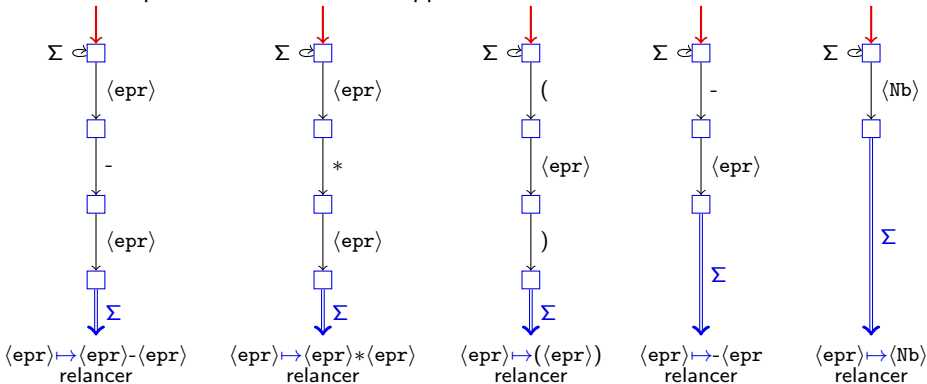
$\Sigma = \mathcal{T} \cup \mathcal{N}$  est l'ensemble des terminaux + non-terminaux

# Première tentative

## Idee : Faire comme le lexeur

- un petit automate pour chaque règle qui “test” le pattern,
- précédé d'une boucle “retardatrice” infinie,
- puis on leur met une action et on fait la somme.

Remarque : une telle action s'appelle une action de **réduction**

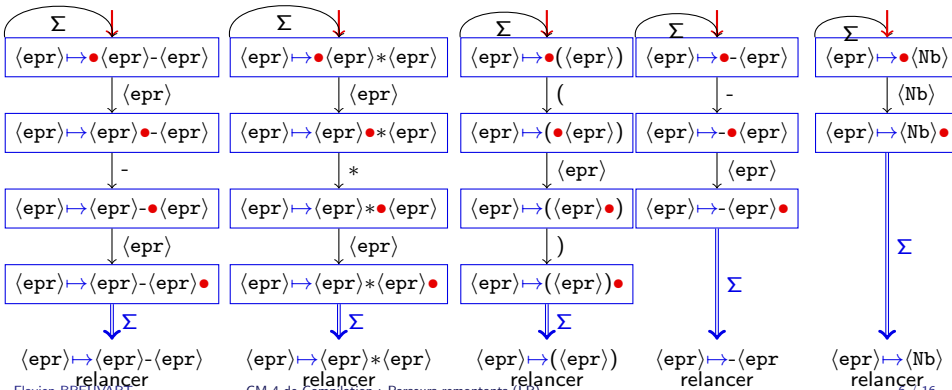


# Première tentative

## Idee : Faire comme le lexeur

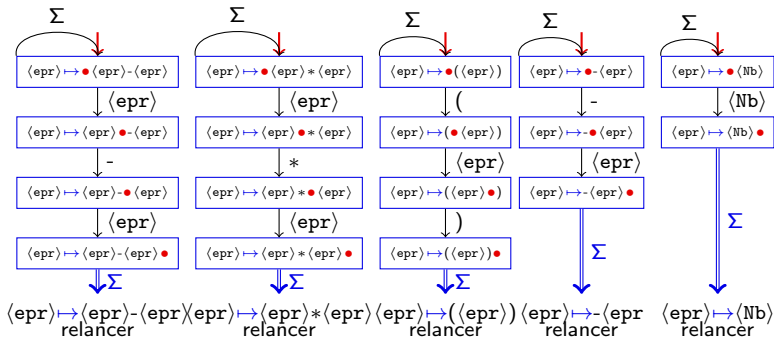
- un petit automate pour chaque règle qui "test" le pattern,
- précédé d'une boucle "retardatrice" infinie,
- puis on leur met une action et on fait la somme.

Remarque : une telle action s'appel une action de **réduction**



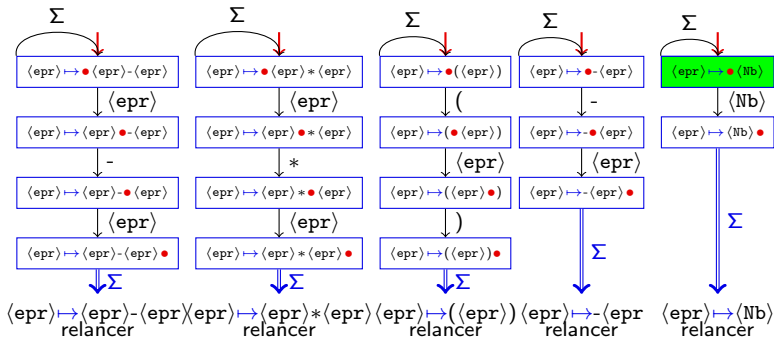


# Automate du parseur LR en action



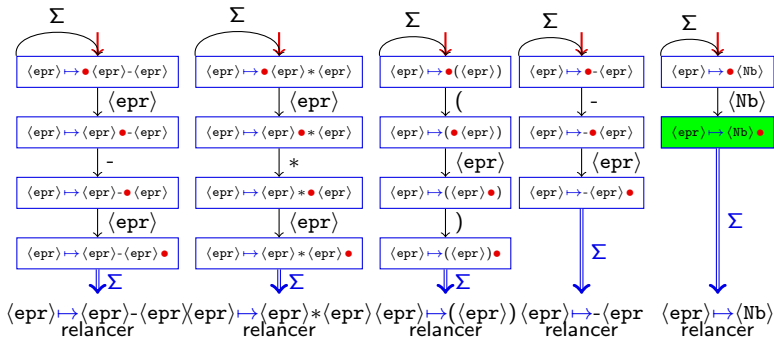
$\langle \text{NB} \rangle \quad * \quad \langle \text{NB} \rangle \quad - \quad \langle \text{NB} \rangle \quad * \quad \langle \text{NB} \rangle \quad \$$   
 |                    |                    |                    |  
 3                    7                    42                    0

# Automate du parseur LR en action



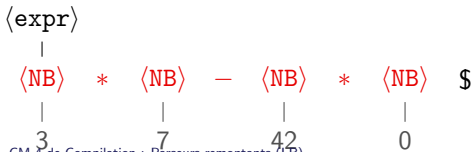
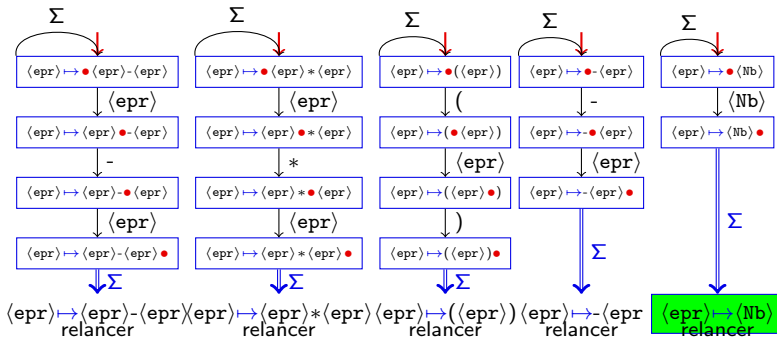
(NB) \* (NB) - (NB) \* (NB) \$  
 |                    |                    |                    |  
 3                    7                    42                    0

# Automate du parseur LR en action

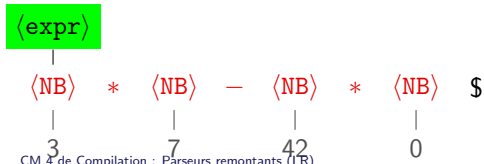
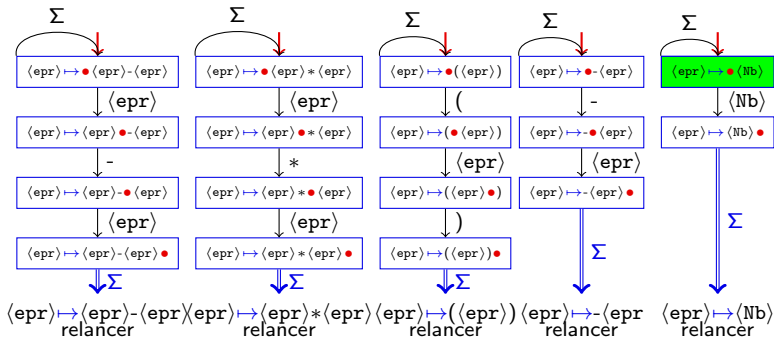


$\langle \text{NB} \rangle$   $*$   $\langle \text{NB} \rangle$   $-$   $\langle \text{NB} \rangle$   $*$   $\langle \text{NB} \rangle$   $\$$   
 | | | | |  
 3 7 42 0

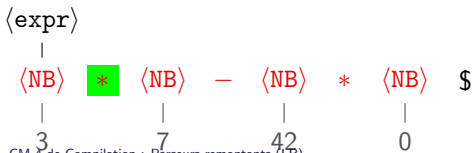
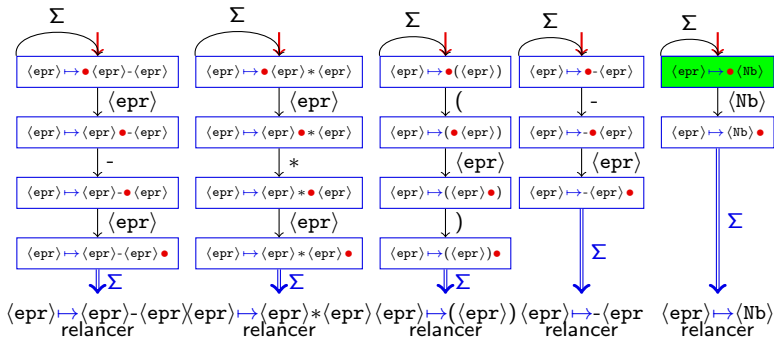
# Automate du parseur LR en action



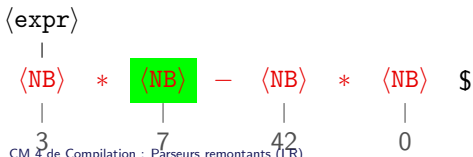
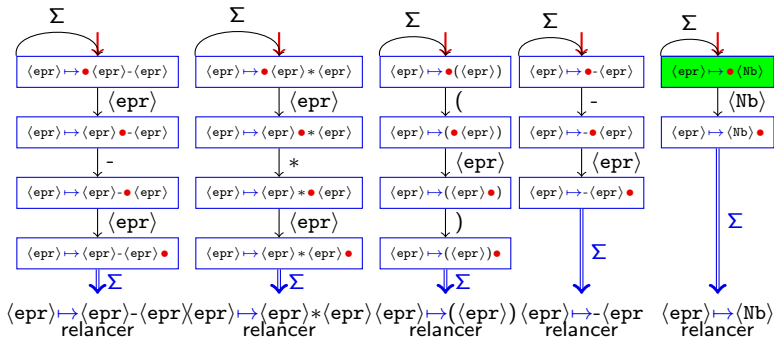
# Automate du parseur LR en action



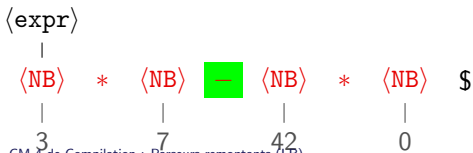
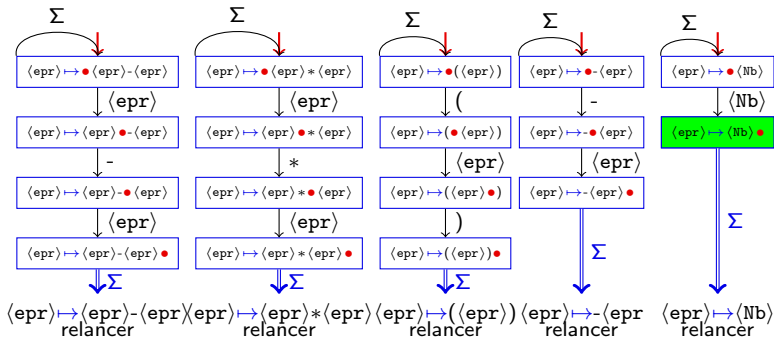
# Automate du parseur LR en action



# Automate du parseur LR en action

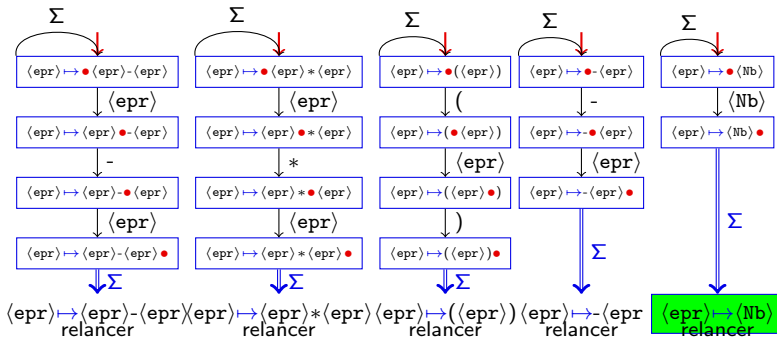


# Automate du parseur LR en action



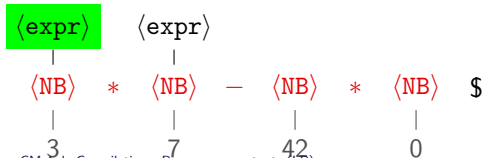
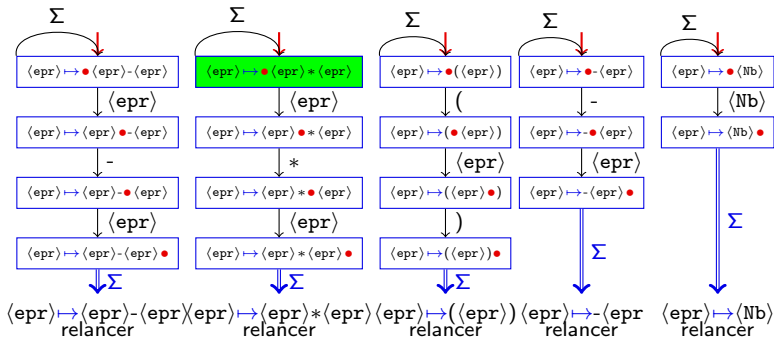


# Automate du parseur LR en action

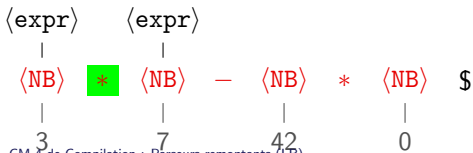
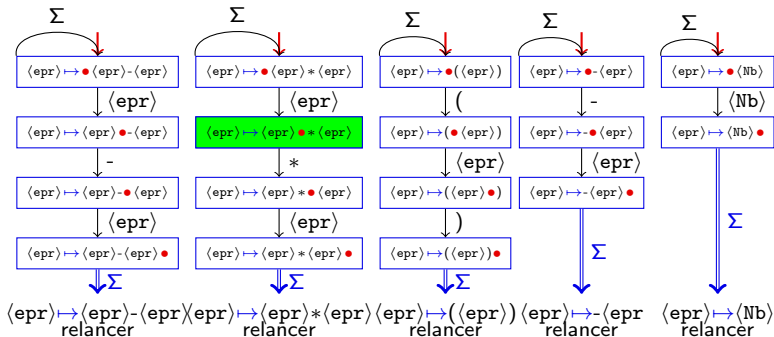


$\langle \text{expr} \rangle$      $\langle \text{expr} \rangle$   
 |            |  
 $\langle \text{NB} \rangle$  \*  $\langle \text{NB} \rangle$  -  $\langle \text{NB} \rangle$  \*  $\langle \text{NB} \rangle$  \$  
 |            |            |            |  
 3            7            42            0

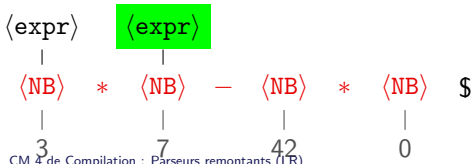
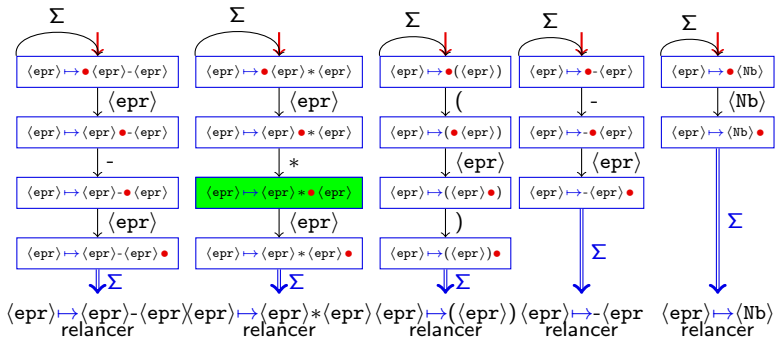
# Automate du parseur LR en action



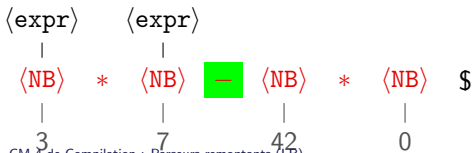
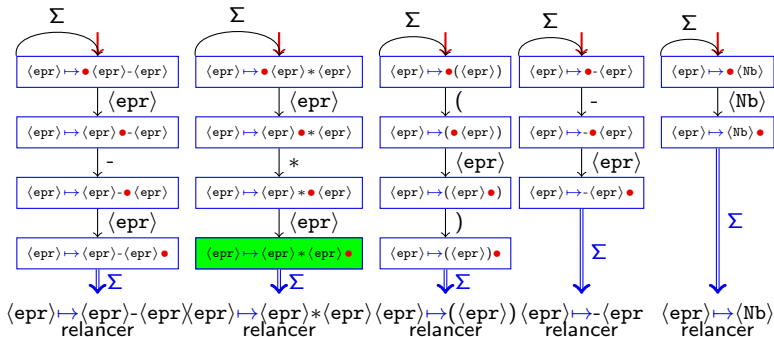
# Automate du parseur LR en action



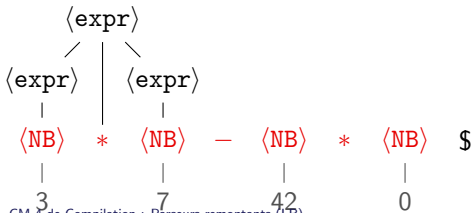
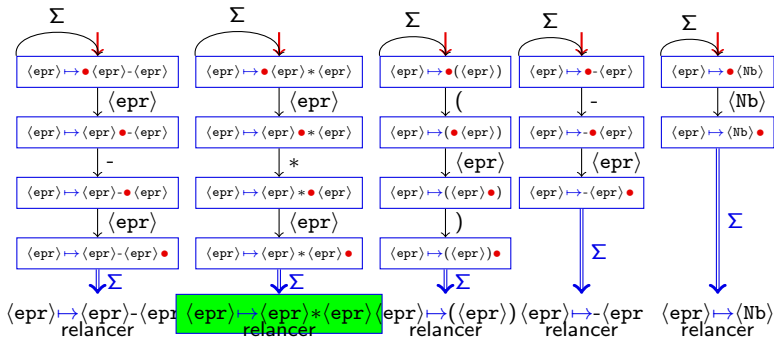
# Automate du parseur LR en action



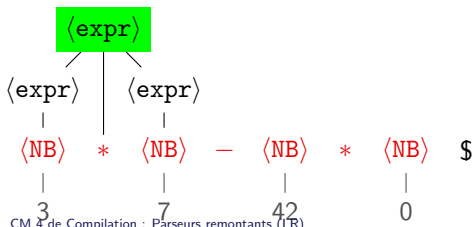
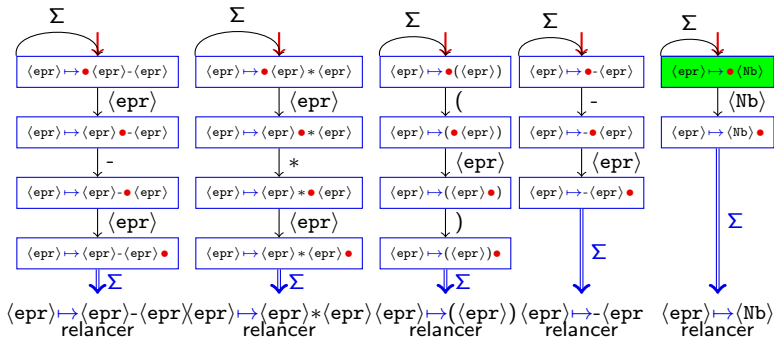
# Automate du parseur LR en action



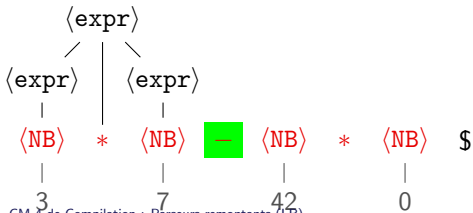
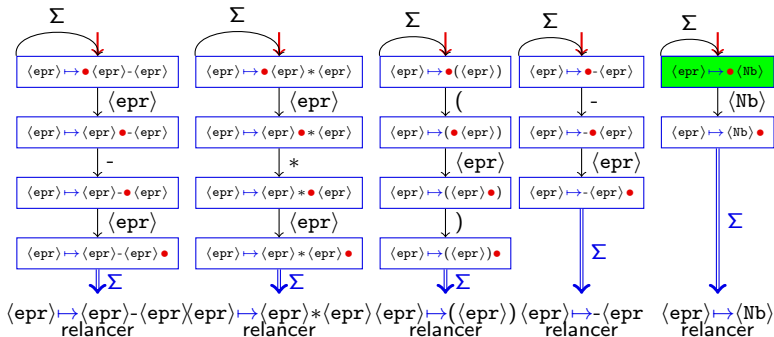
# Automate du parseur LR en action



# Automate du parseur LR en action

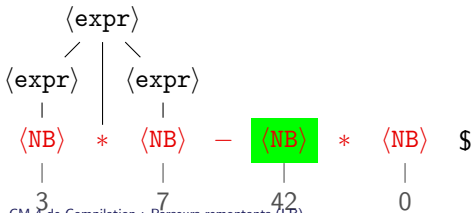
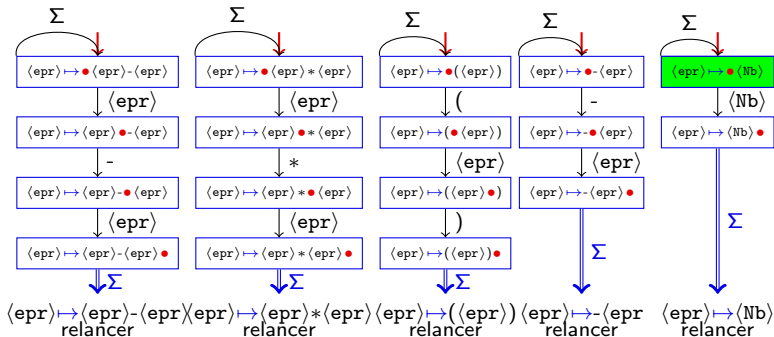


# Automate du parseur LR en action

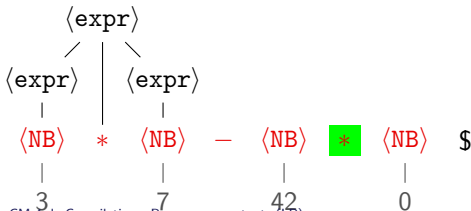
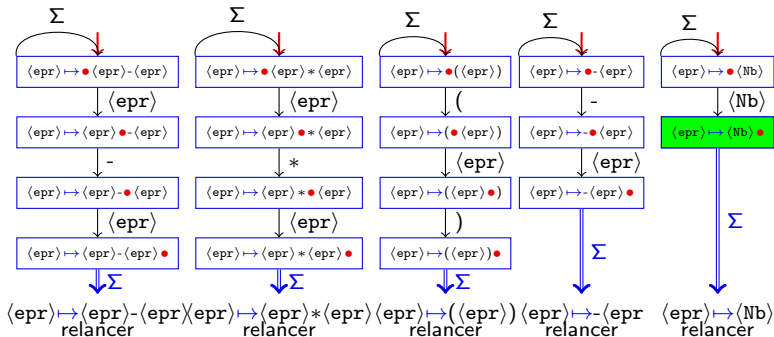




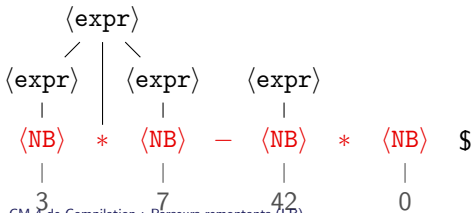
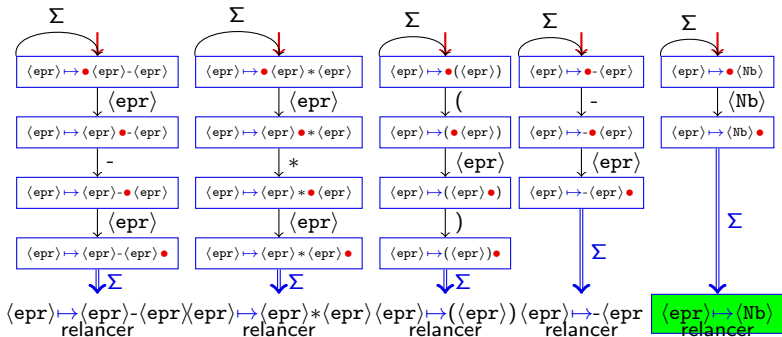
# Automate du parseur LR en action



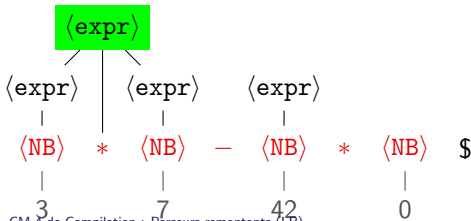
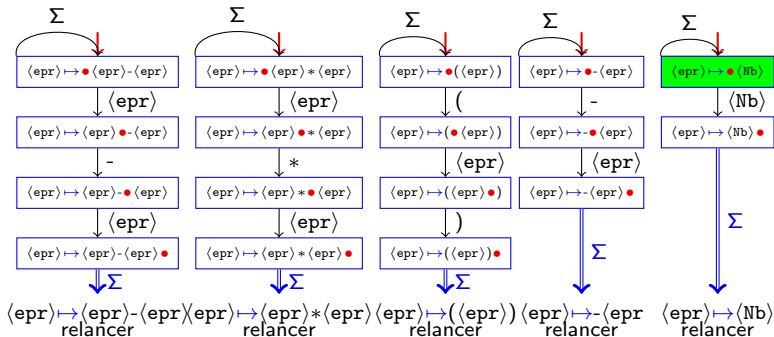
# Automate du parseur LR en action



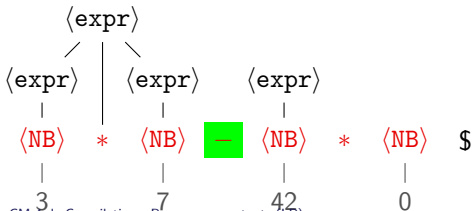
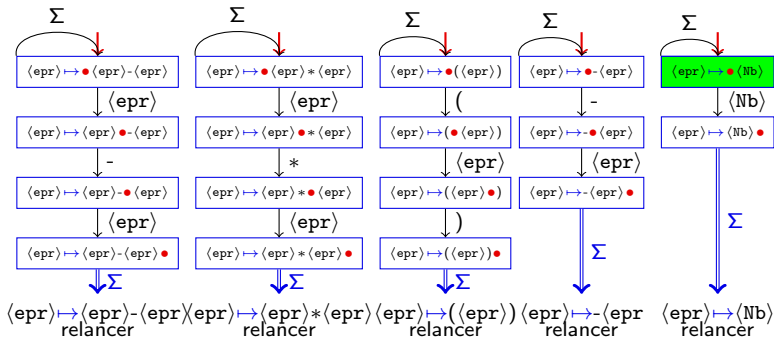
# Automate du parseur LR en action



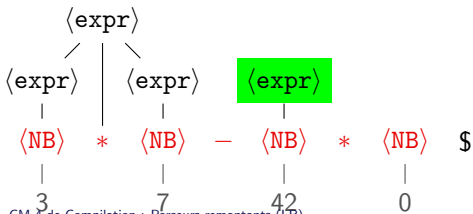
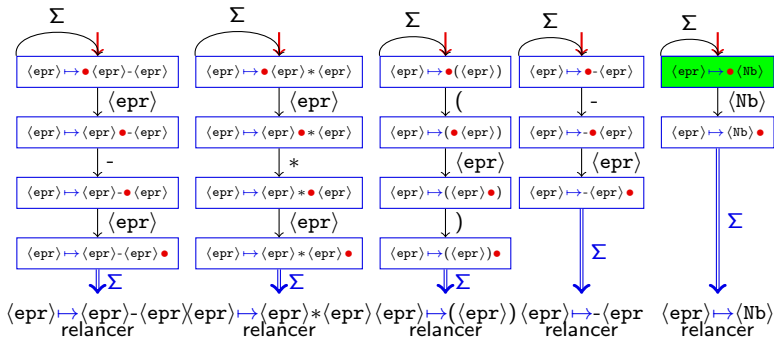
# Automate du parseur LR en action



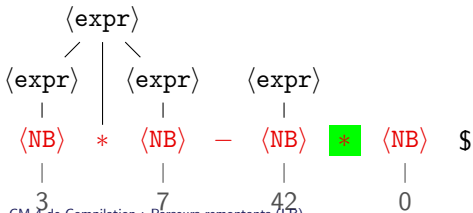
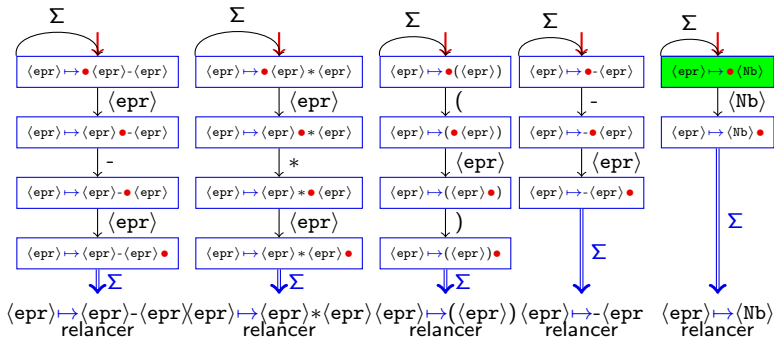
# Automate du parseur LR en action



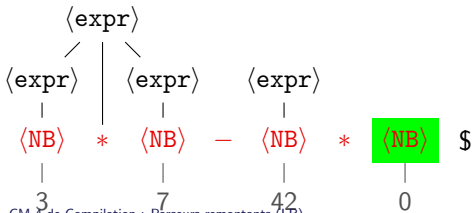
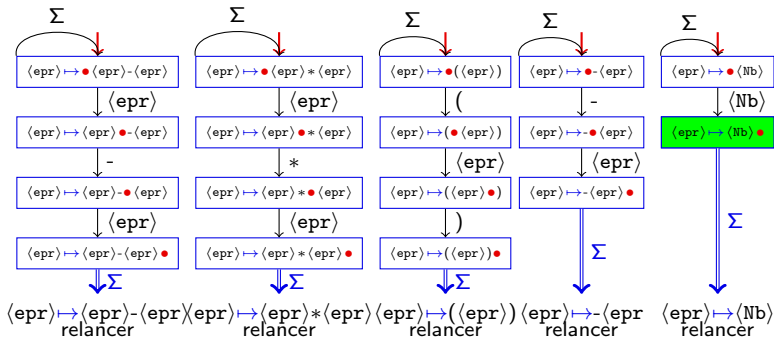
# Automate du parseur LR en action



# Automate du parseur LR en action

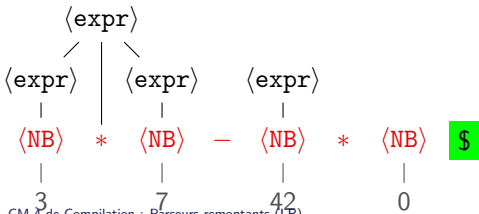
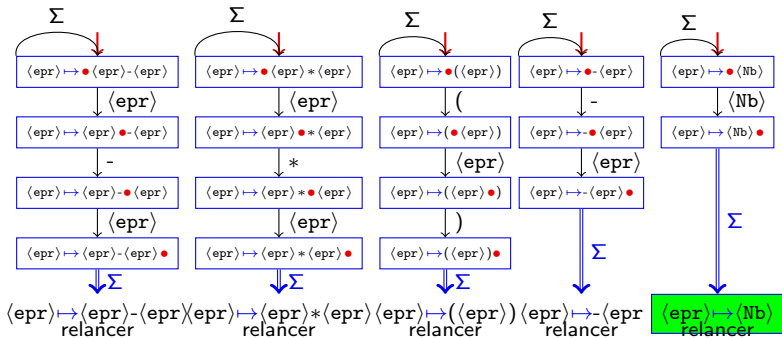


# Automate du parseur LR en action

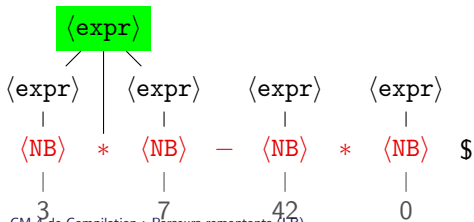
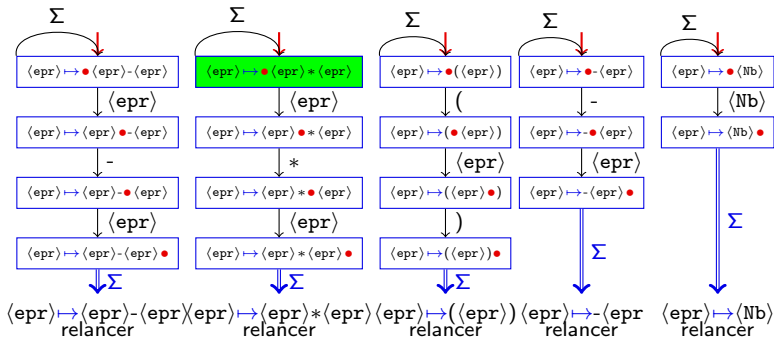




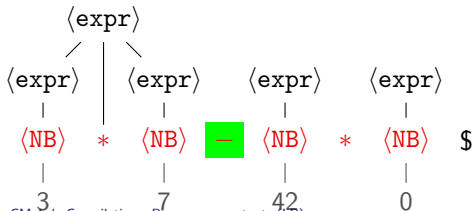
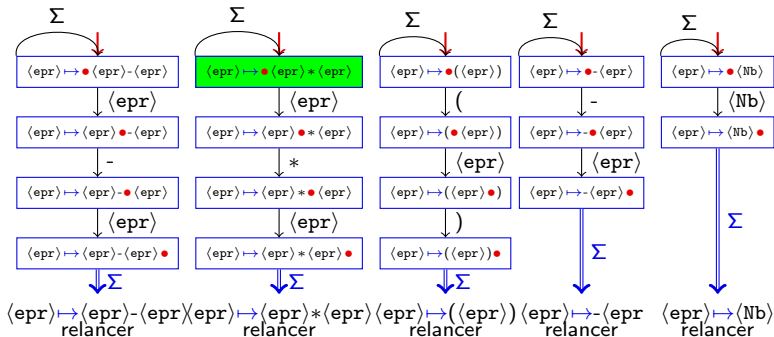
# Automate du parseur LR en action



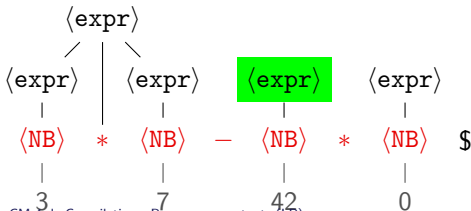
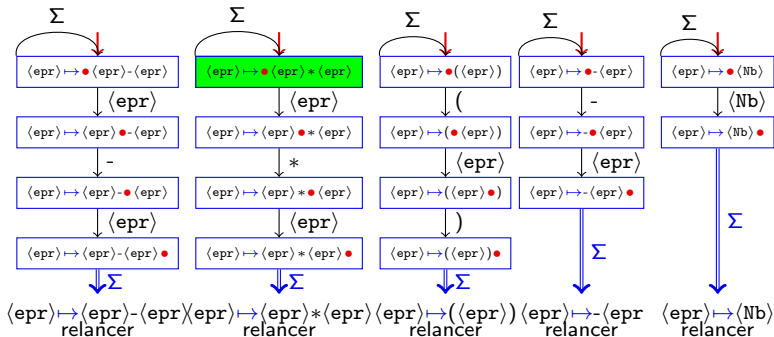
# Automate du parseur LR en action



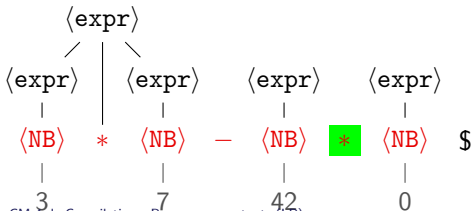
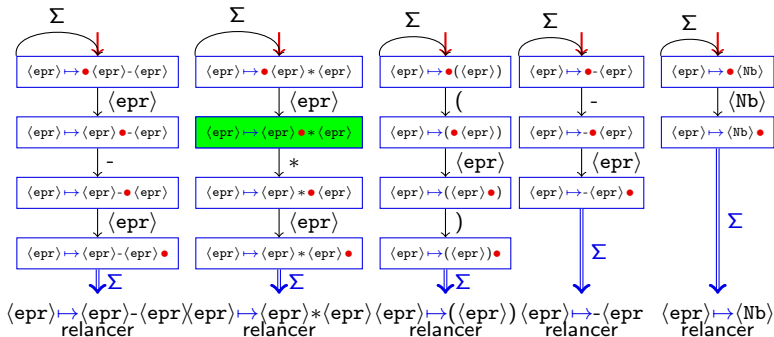
# Automate du parseur LR en action



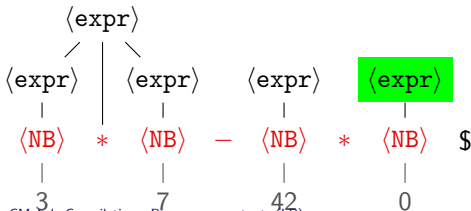
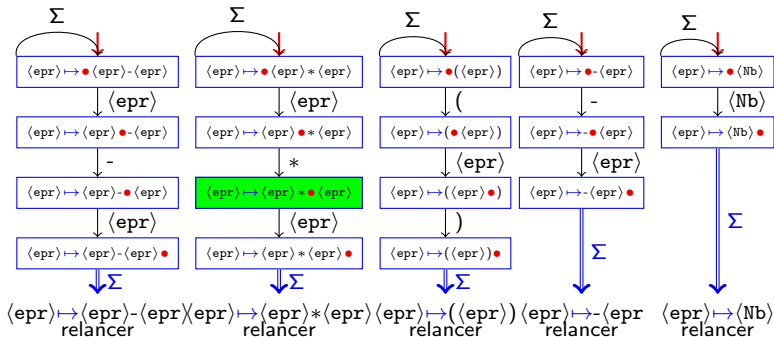
# Automate du parseur LR en action



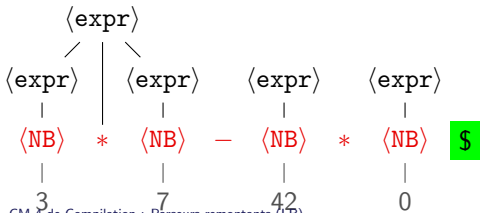
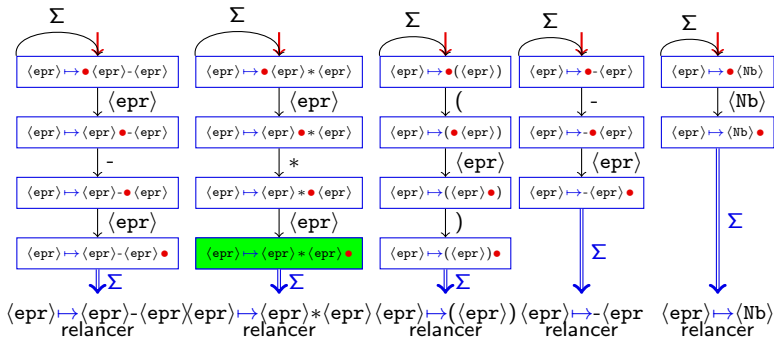
# Automate du parseur LR en action



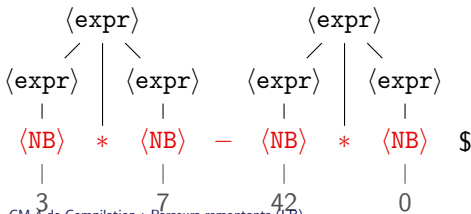
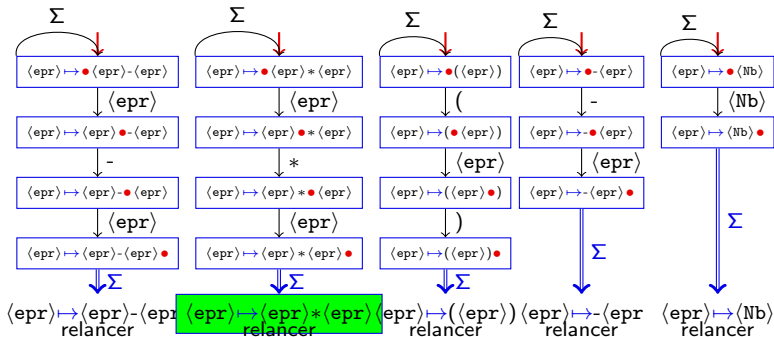
# Automate du parseur LR en action



# Automate du parseur LR en action

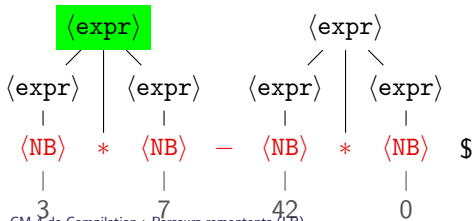
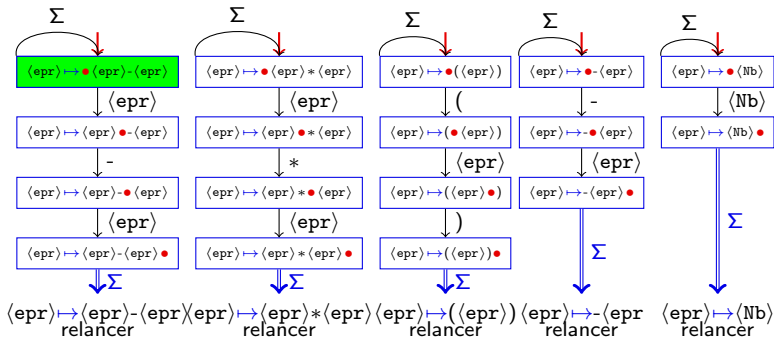


# Automate du parseur LR en action

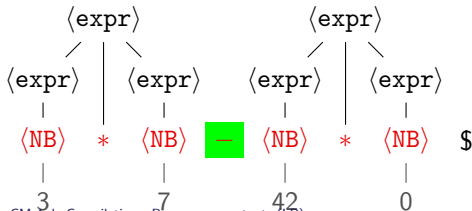
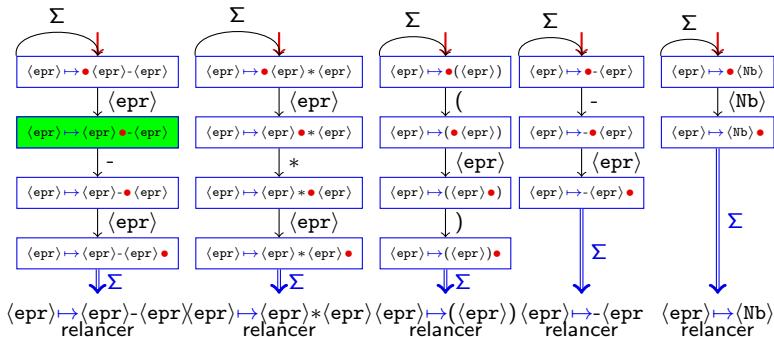




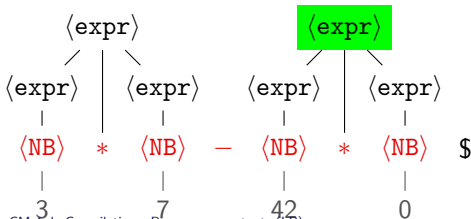
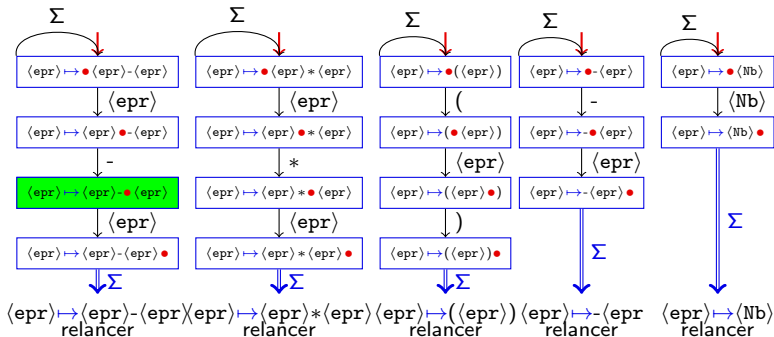
# Automate du parseur LR en action



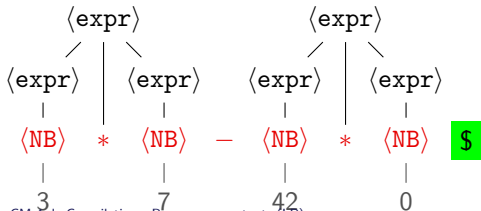
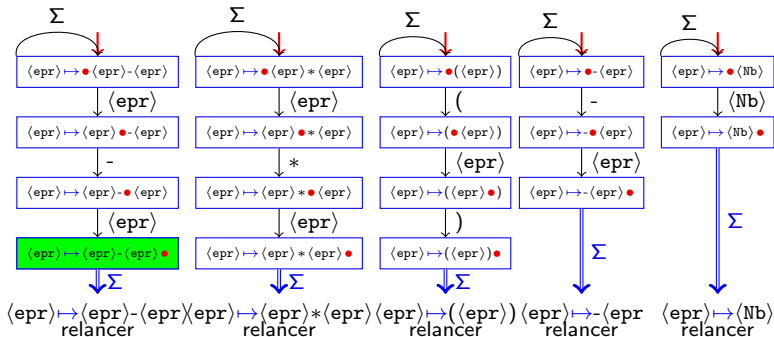
# Automate du parseur LR en action



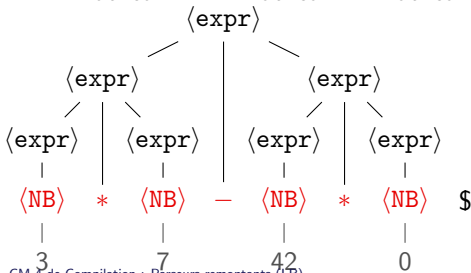
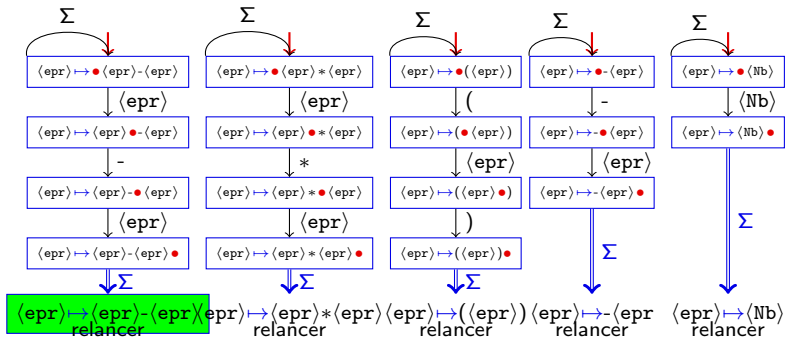
# Automate du parseur LR en action



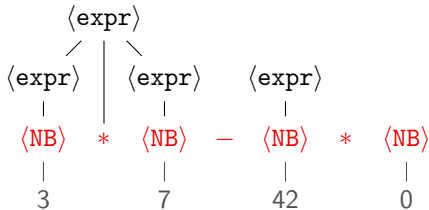
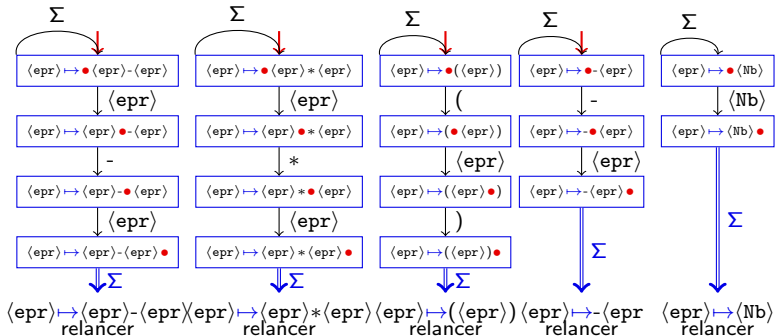
# Automate du parseur LR en action



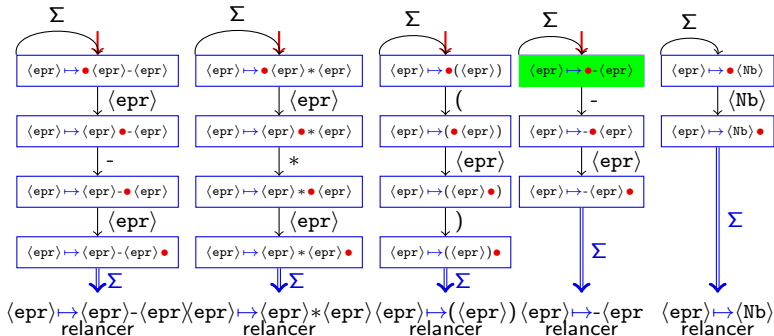
# Automate du parseur LR en action



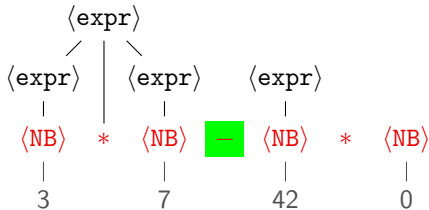
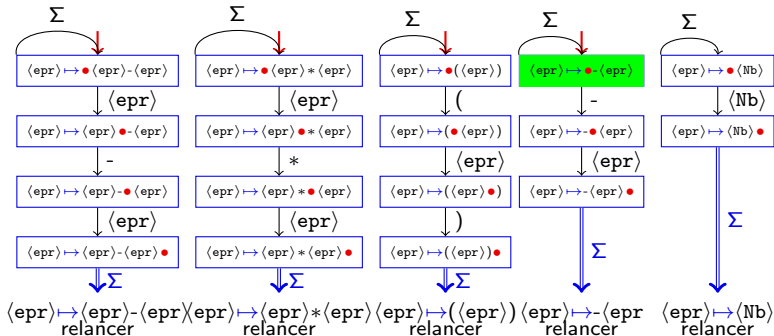
# Échec de la tentative



# Échec de la tentative

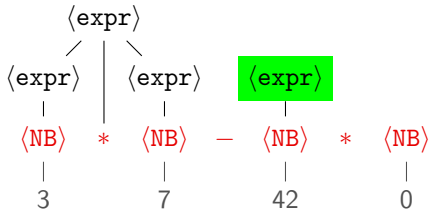
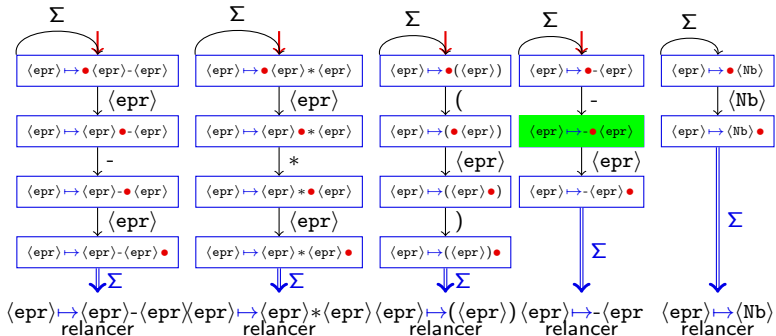


# Échec de la tentative

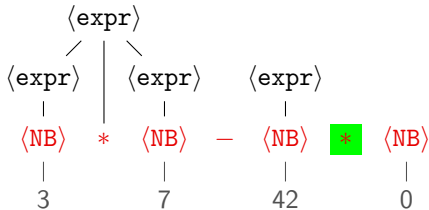
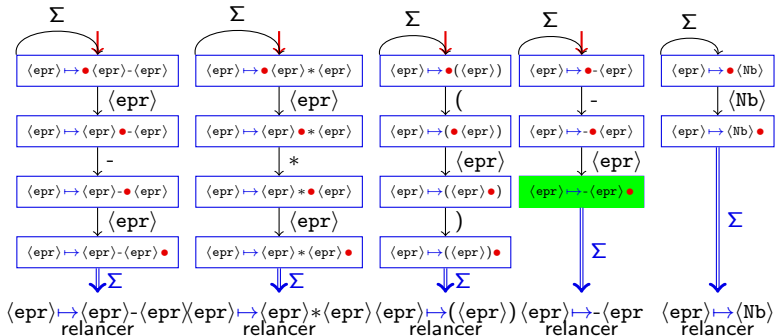




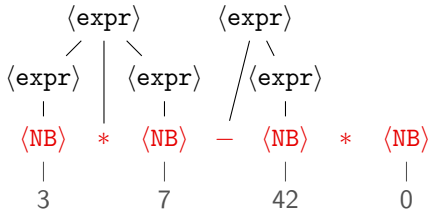
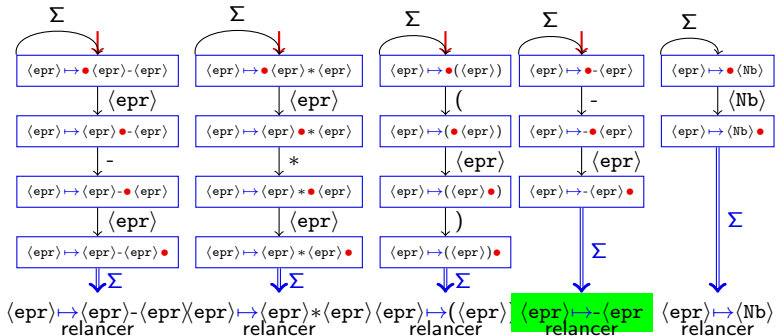
# Échec de la tentative



# Échec de la tentative



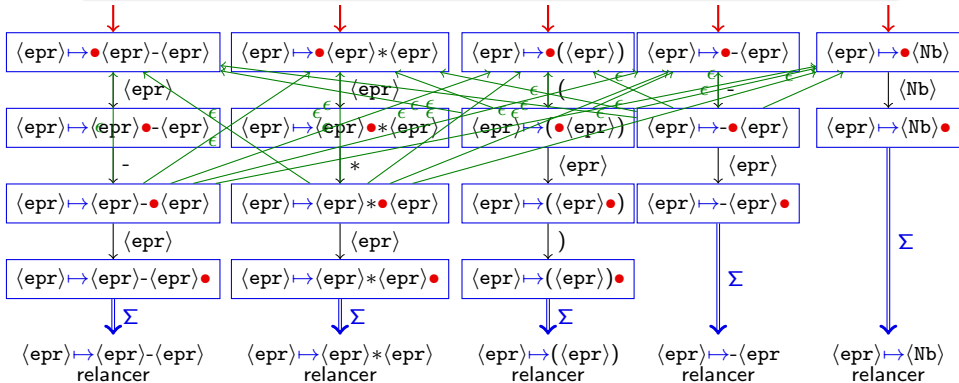
# Échec de la tentative



# Seconde tentative

## Idee : Introduire de la contextualité

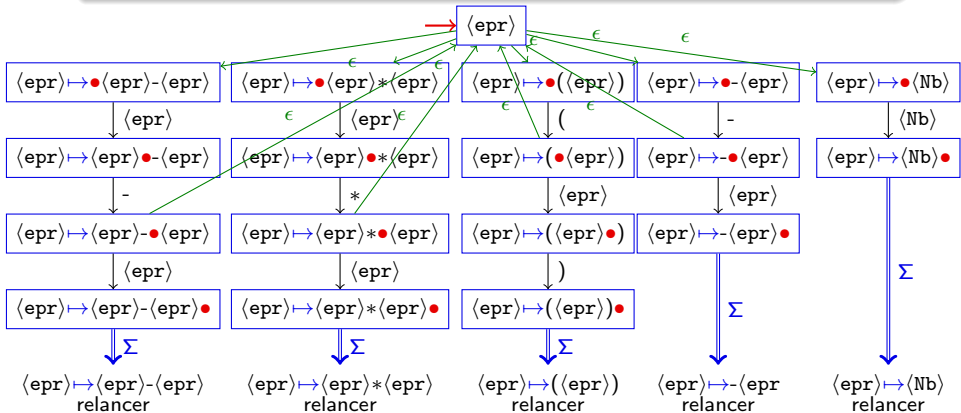
- On ne "teste" pas toutes les règles à tout moment,
- uniquement quand une autre règle peut avoir besoin du NT,
- en utilisant des  $\epsilon$ -transitions



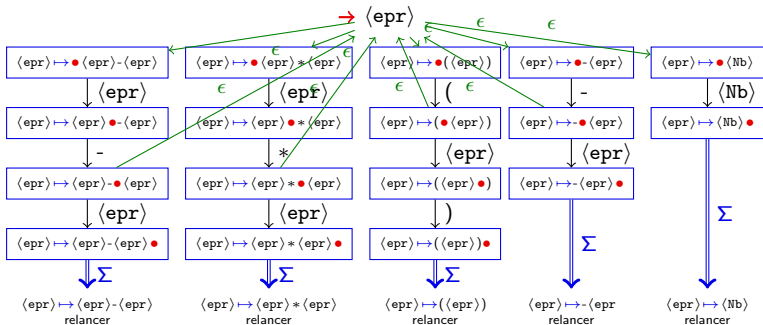
# Seconde tentative

## Idee : Introduire de la contextualité

- On ne "teste" pas toutes les règles à tout moment,
- uniquement quand une autre règle peut avoir besoin du NT,
- en utilisant des  $\epsilon$ -transitions

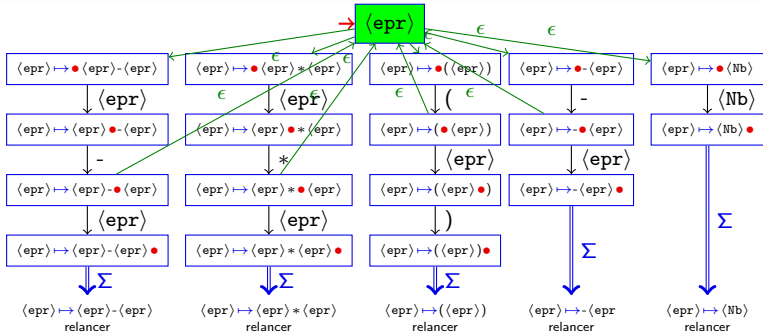


# Seconde tentative en action



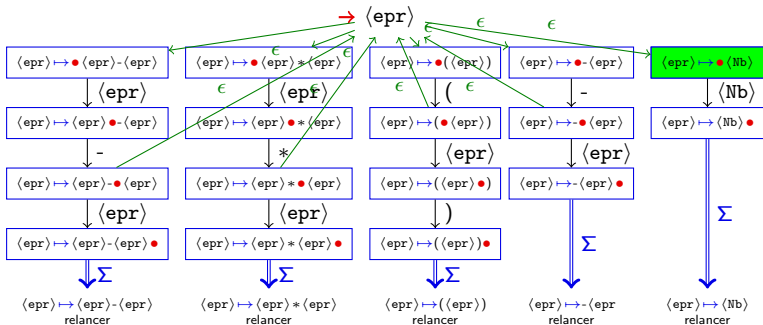
$\langle \text{NB} \rangle$  \*  $\langle \text{NB} \rangle$  -  $\langle \text{NB} \rangle$  \*  $\langle \text{NB} \rangle$  \$  
 |            |            |            |  
 3            7            42            0

## Seconde tentative en action



(NB) \* (NB) - (NB) \* (NB) \$  
 |            |            |            |  
 3            7            42          0

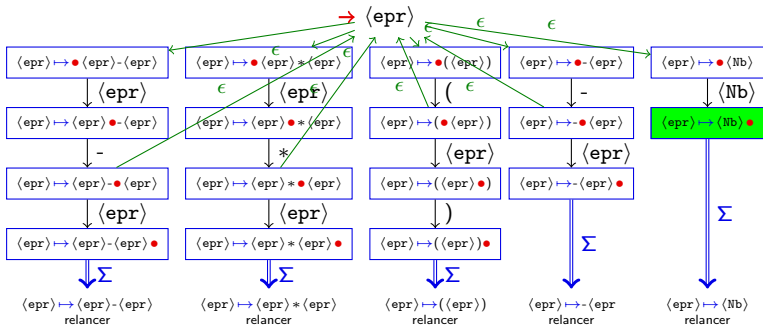
# Seconde tentative en action



(NB) \* (NB) - (NB) \* (NB) \$  
 |            |            |            |  
 3            7            42          0

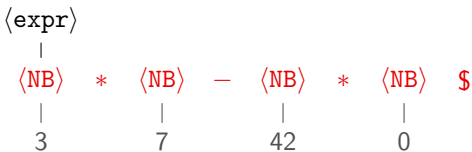
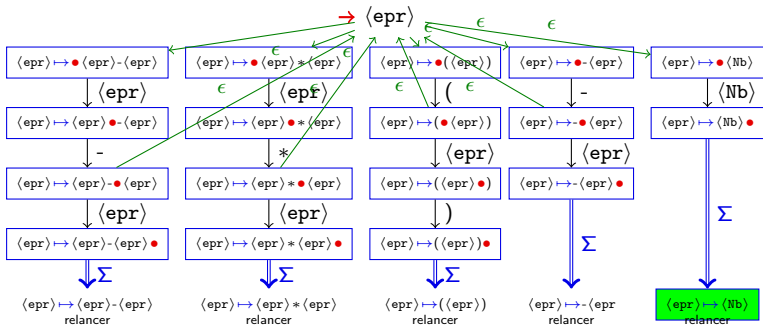


# Seconde tentative en action

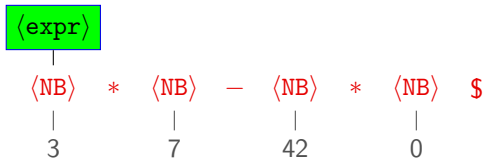
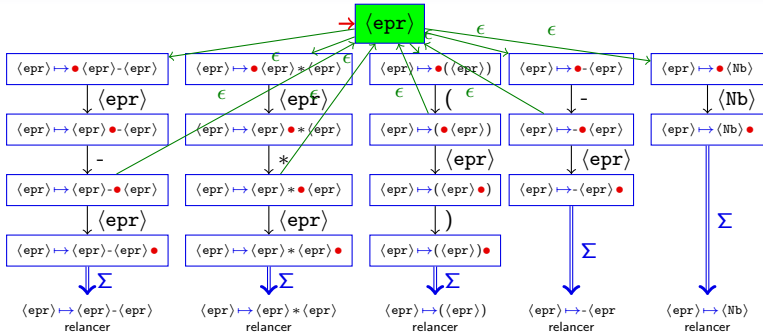


$\langle \text{NB} \rangle$  \*  $\langle \text{NB} \rangle$  -  $\langle \text{NB} \rangle$  \*  $\langle \text{NB} \rangle$  \$  
 |            |            |            |  
 3            7            42            0

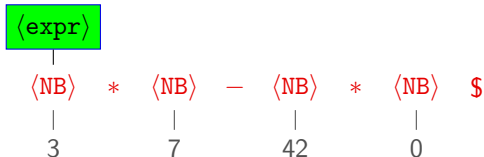
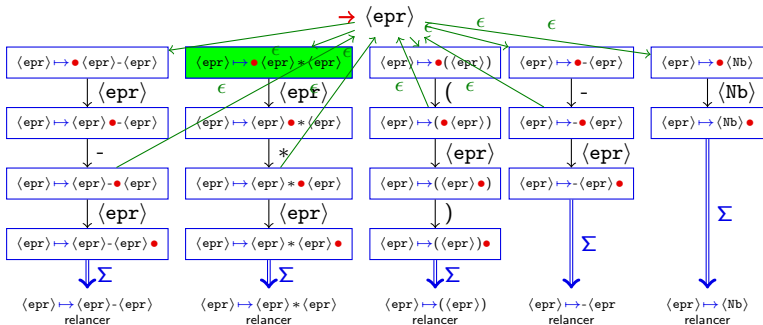
# Seconde tentative en action



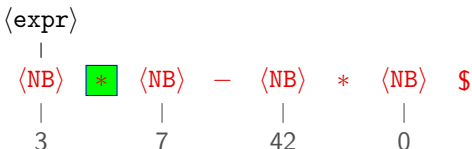
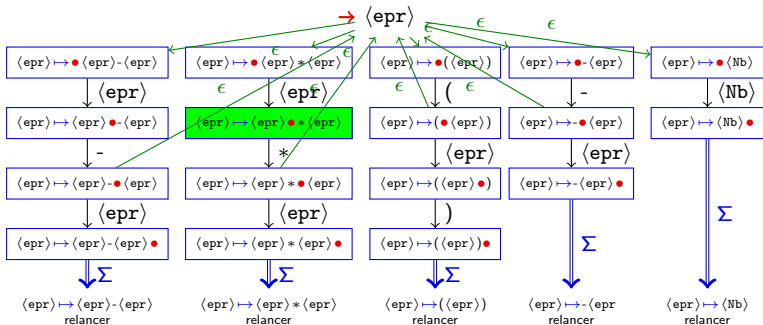
# Seconde tentative en action



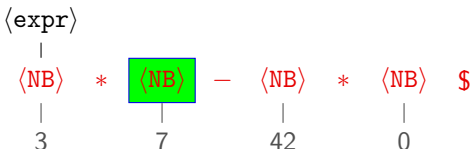
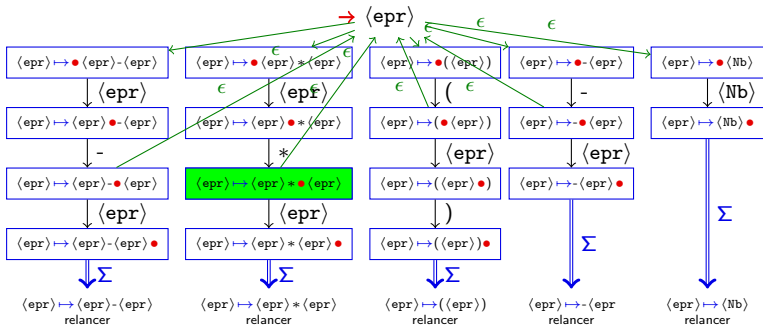
# Seconde tentative en action



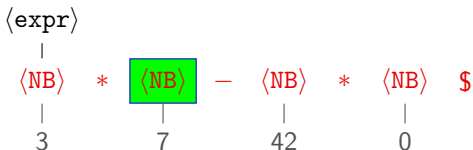
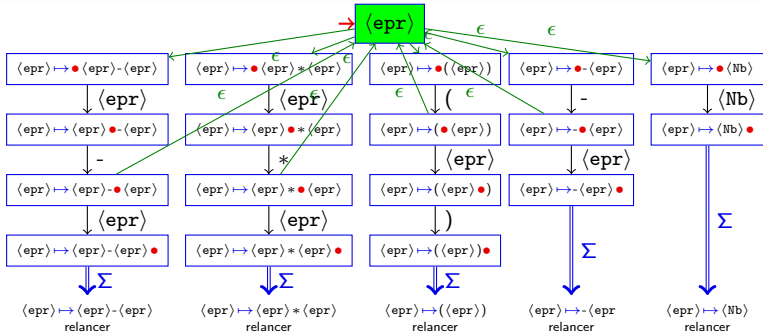
# Seconde tentative en action



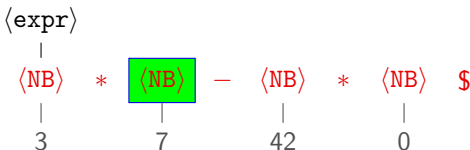
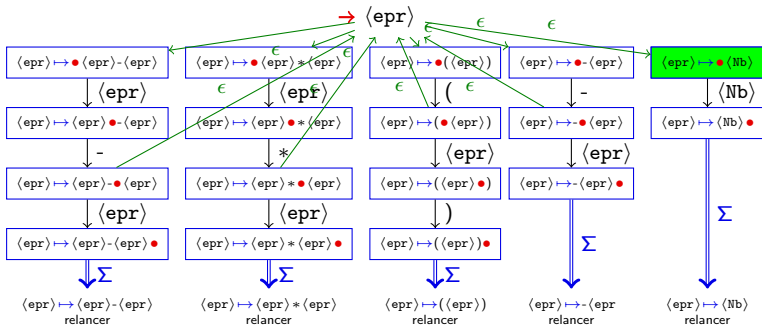
# Seconde tentative en action



# Seconde tentative en action

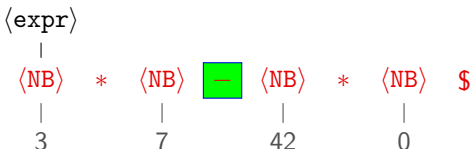
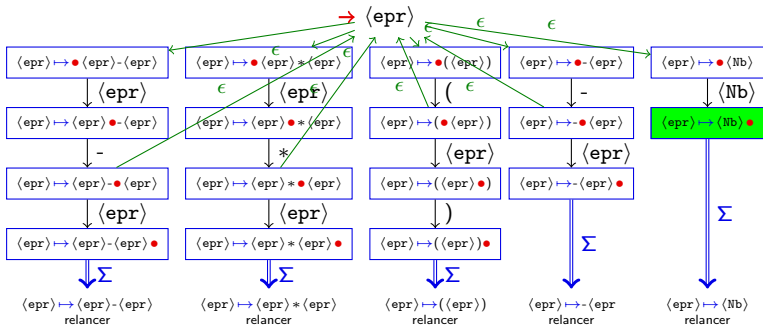


# Seconde tentative en action

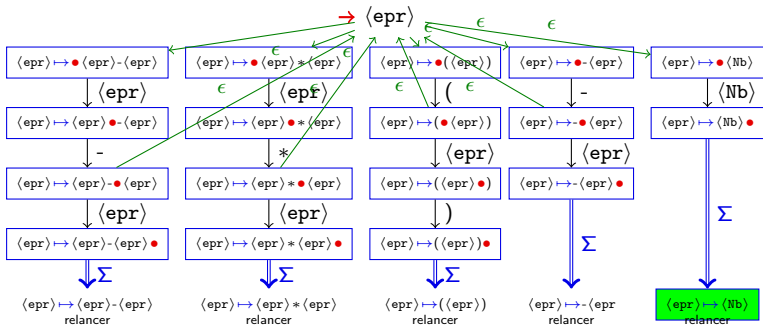




# Seconde tentative en action

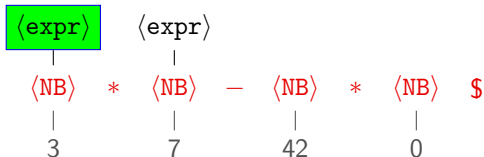
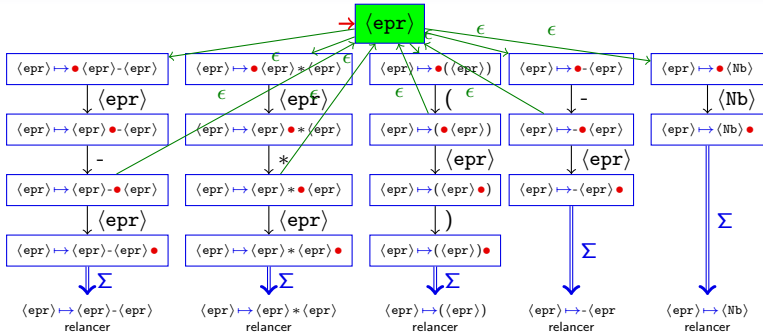


## Seconde tentative en action

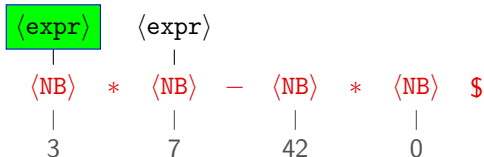
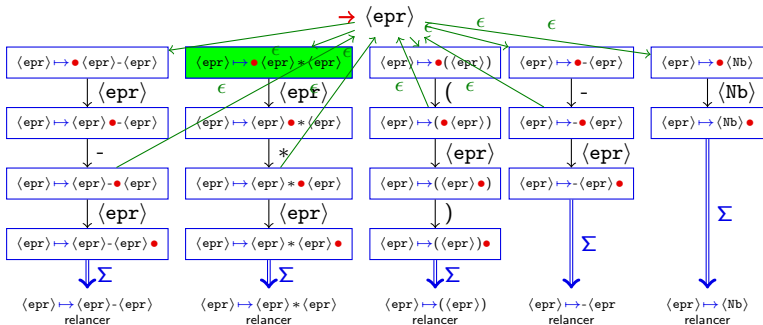


$\langle \text{expr} \rangle$       $\langle \text{expr} \rangle$   
 |            |  
 $\langle \text{NB} \rangle$  \*  $\langle \text{NB} \rangle$  -  $\langle \text{NB} \rangle$  \*  $\langle \text{NB} \rangle$  \$  
 |            |            |            |  
 3            7            42            0

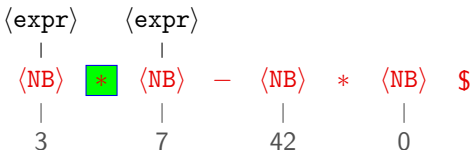
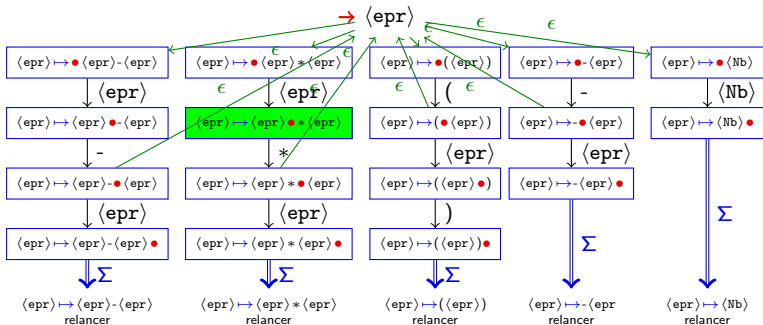
## Seconde tentative en action



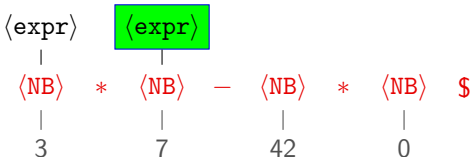
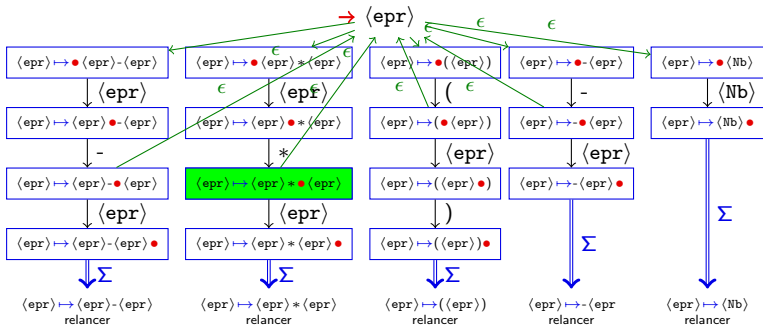
## Seconde tentative en action



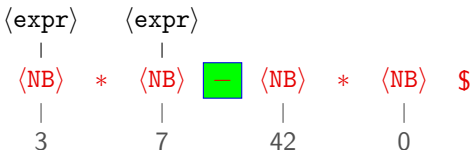
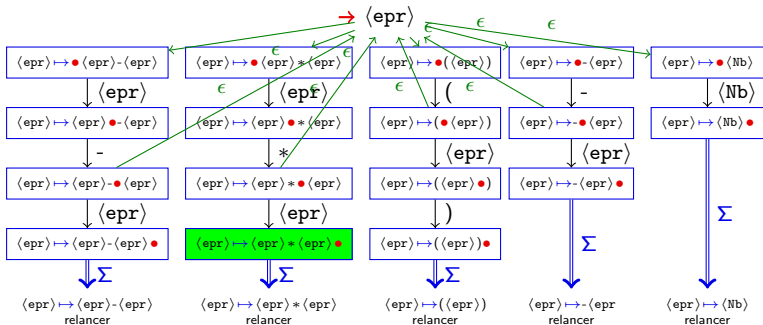
# Seconde tentative en action



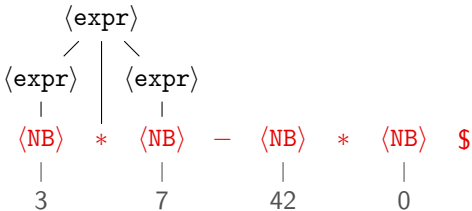
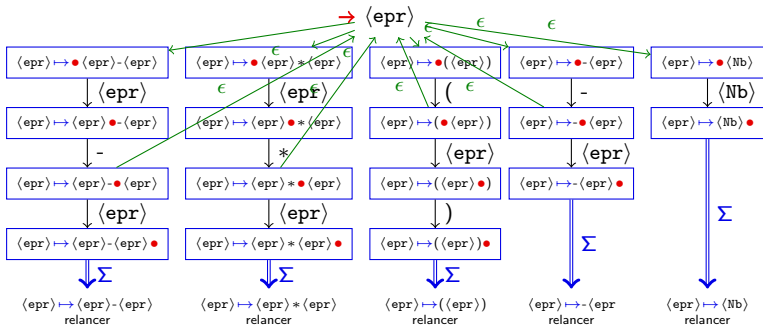
# Seconde tentative en action



# Seconde tentative en action

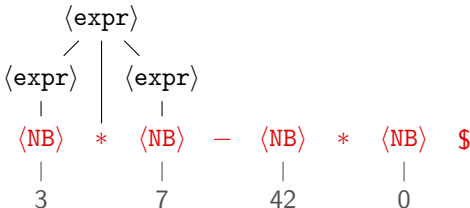
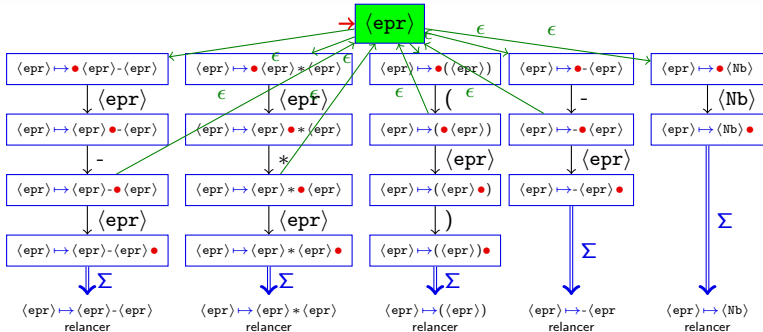


## Seconde tentative en action

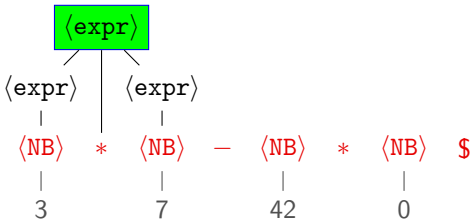
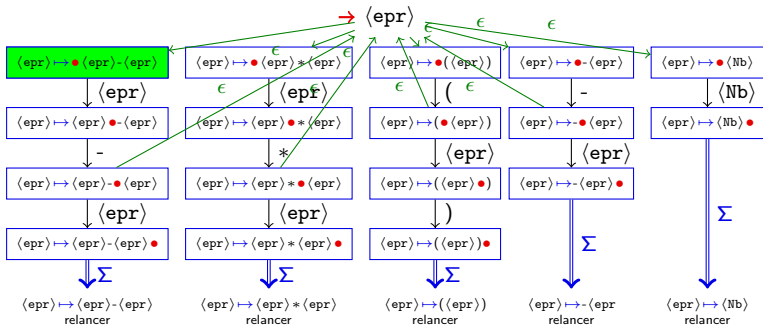




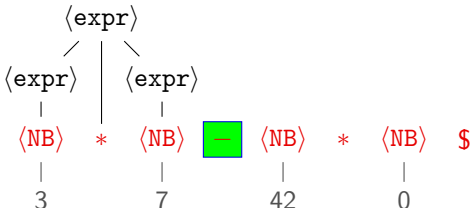
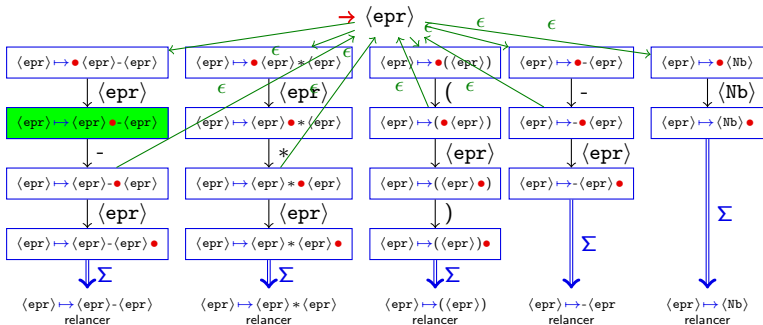
# Seconde tentative en action



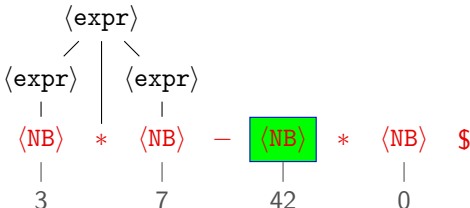
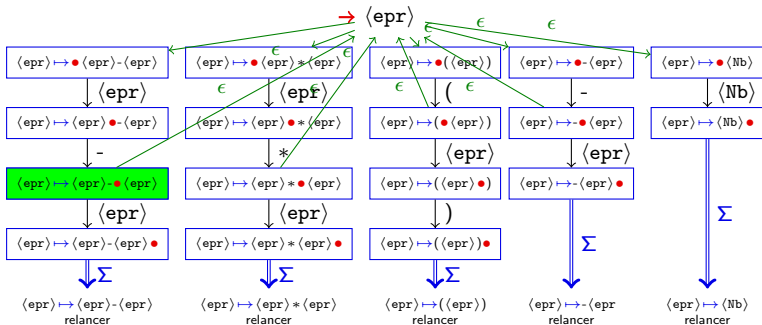
## Seconde tentative en action



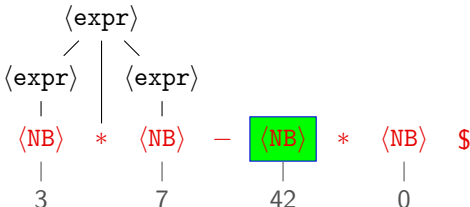
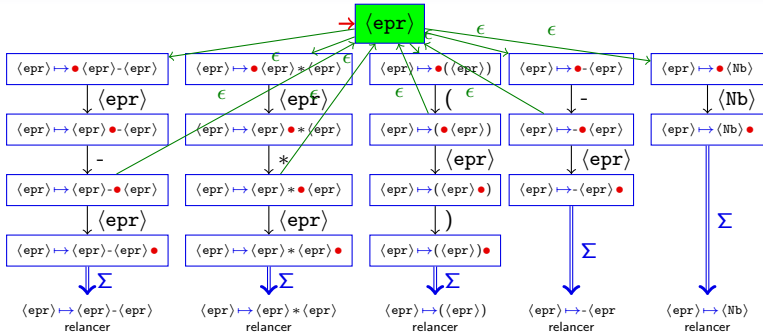
# Seconde tentative en action



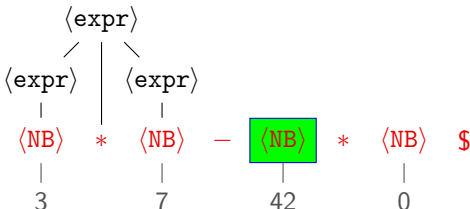
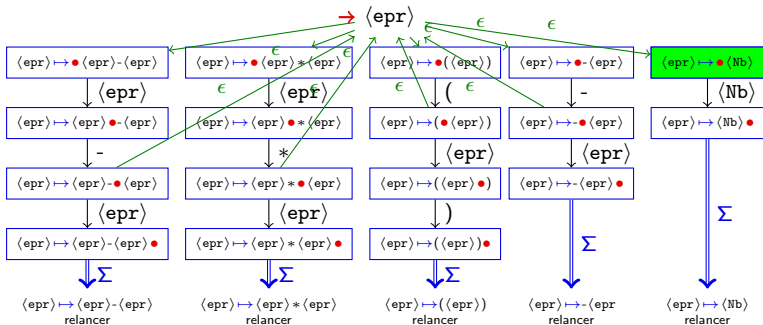
# Seconde tentative en action



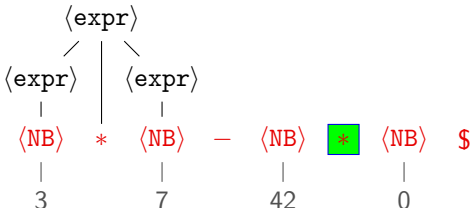
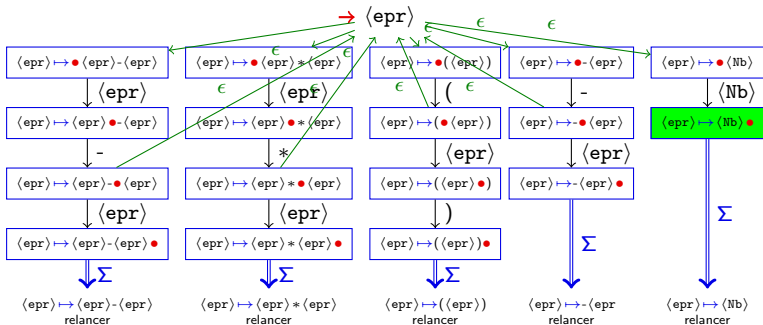
# Seconde tentative en action



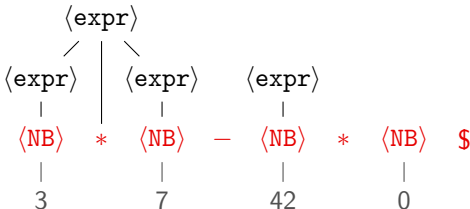
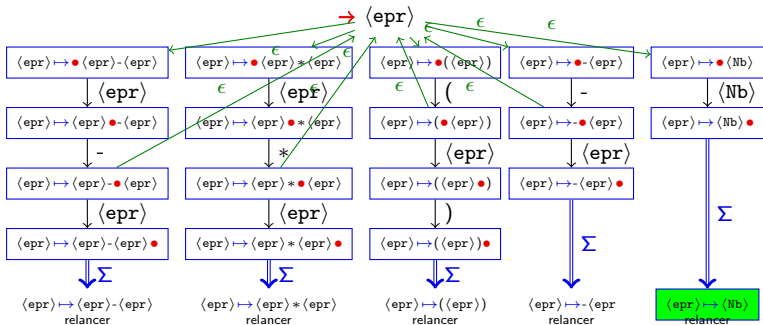
# Seconde tentative en action



# Seconde tentative en action

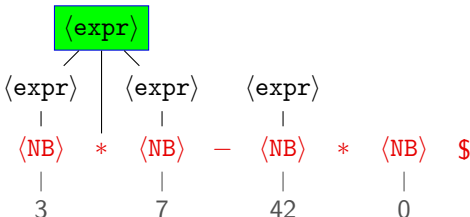
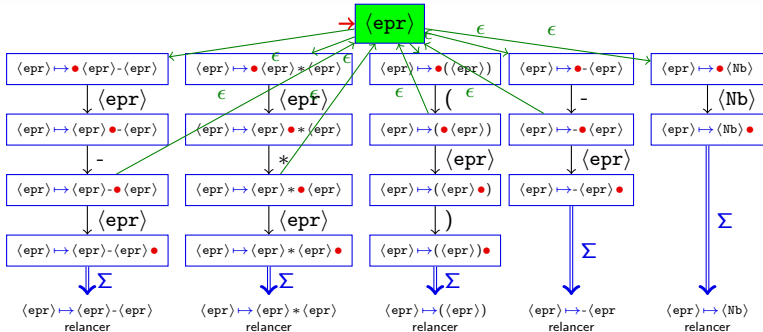


# Seconde tentative en action

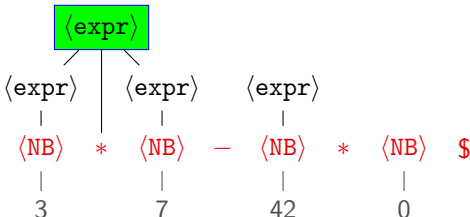
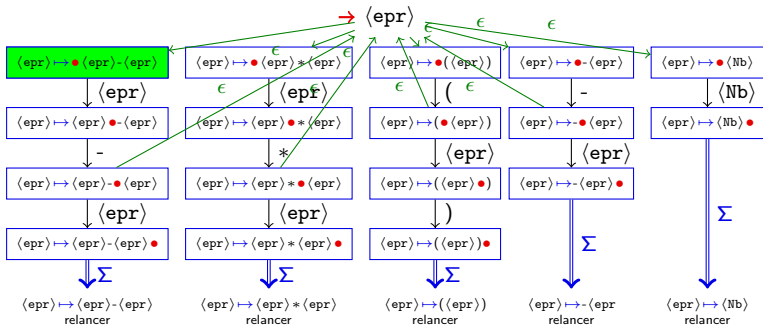




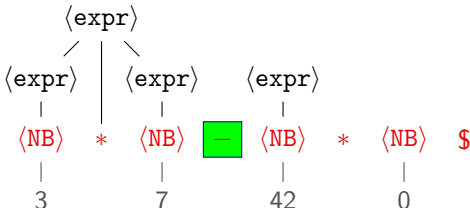
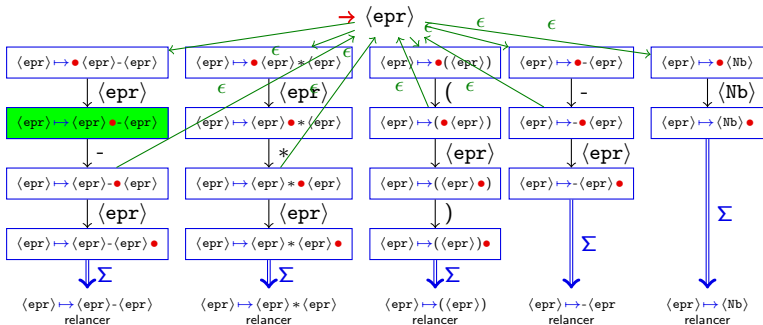
## Seconde tentative en action



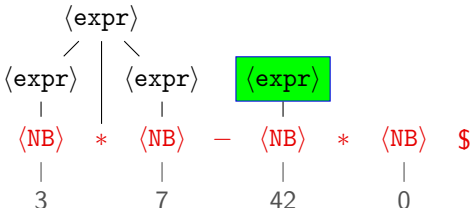
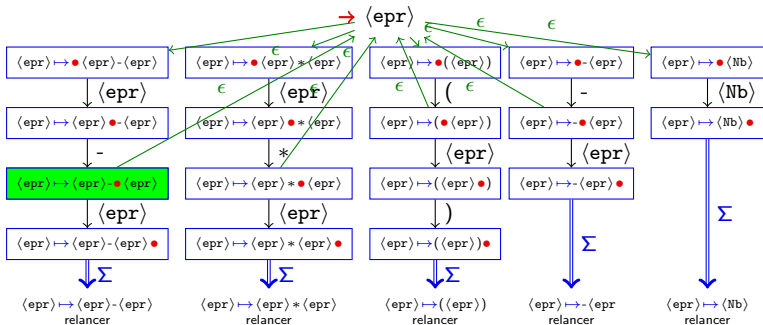
# Seconde tentative en action



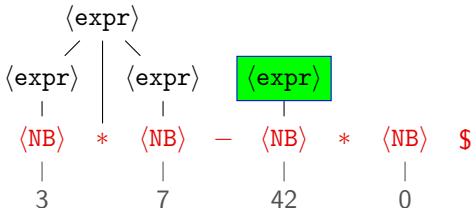
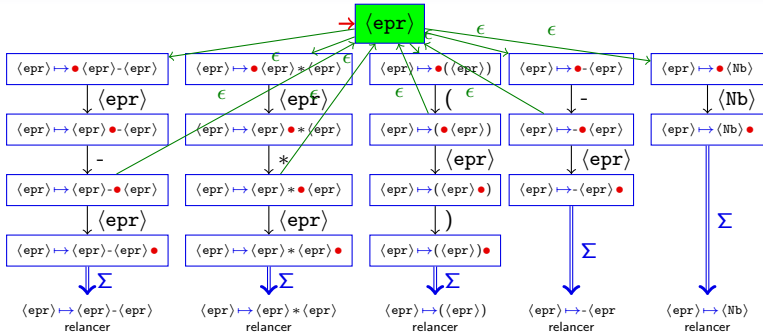
# Seconde tentative en action



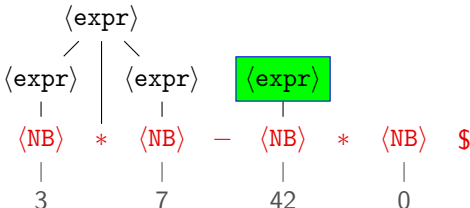
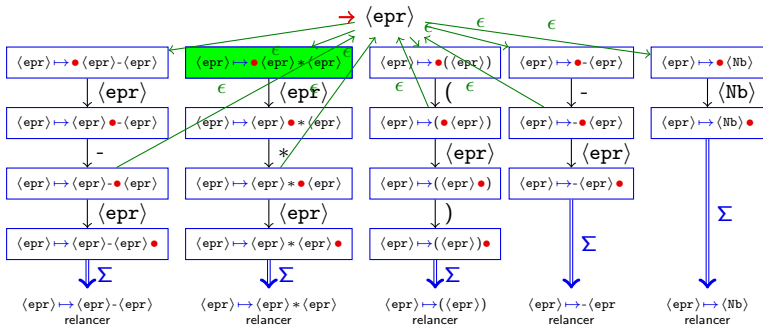
# Seconde tentative en action



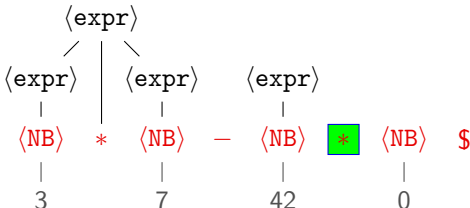
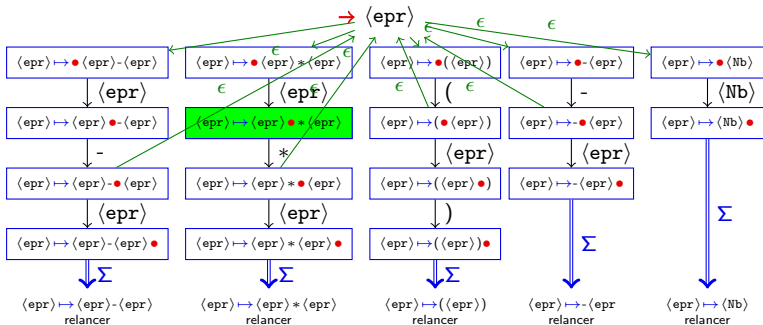
## Seconde tentative en action



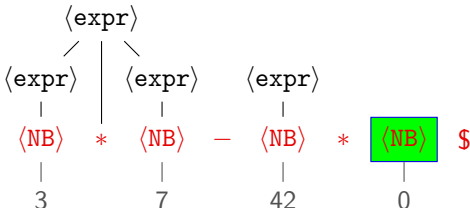
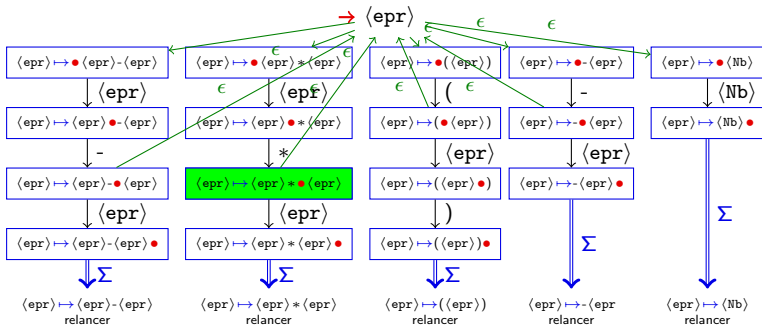
# Seconde tentative en action



# Seconde tentative en action

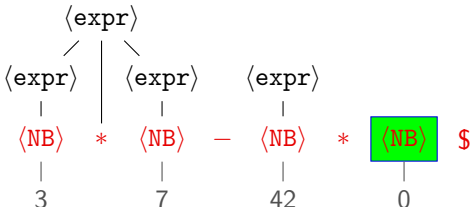
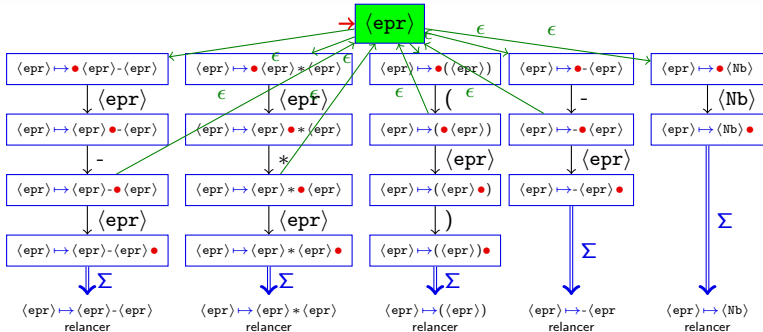


# Seconde tentative en action

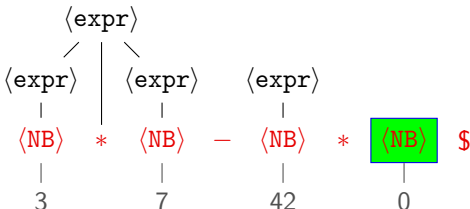
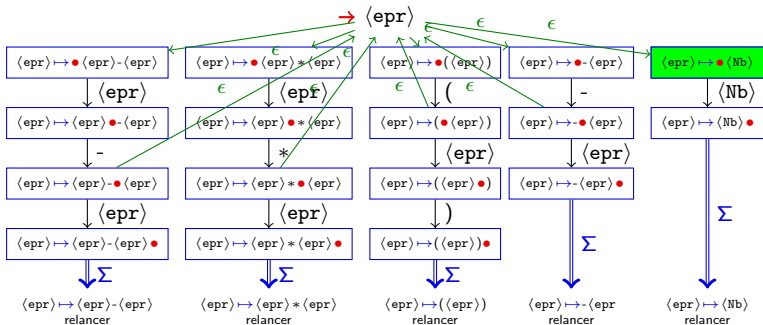




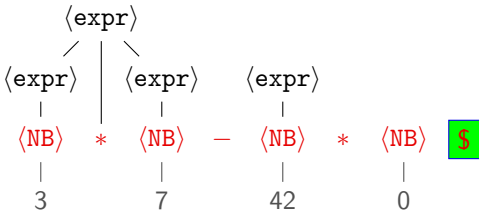
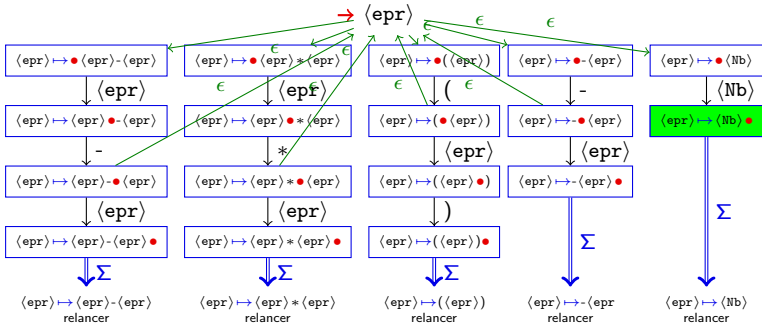
# Seconde tentative en action



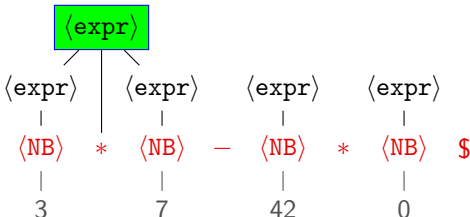
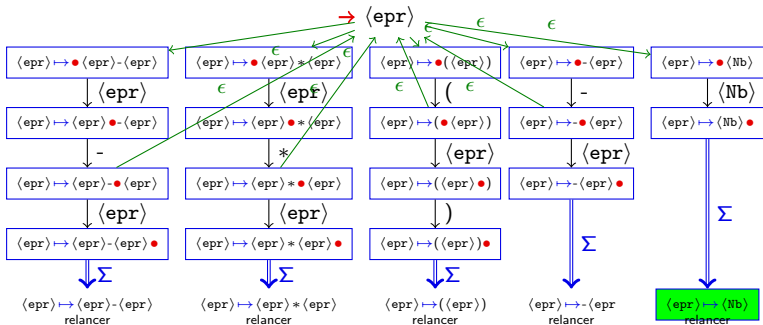
# Seconde tentative en action



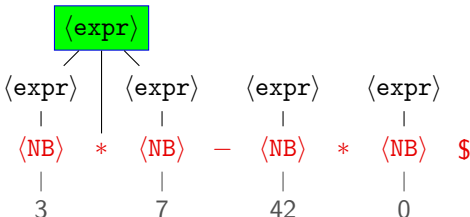
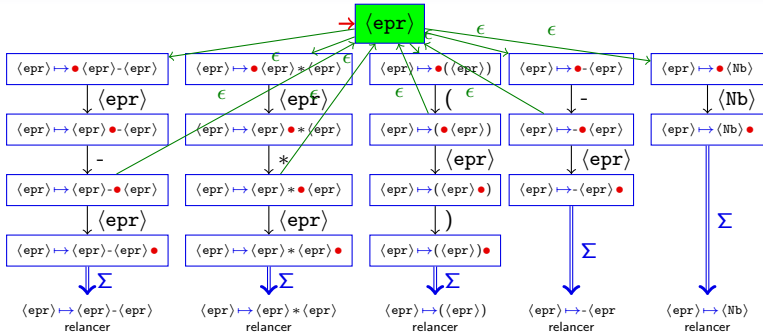
## Seconde tentative en action



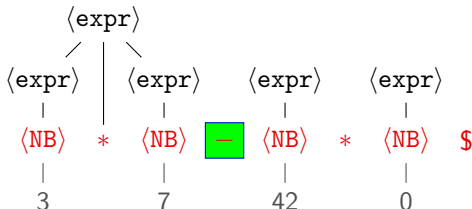
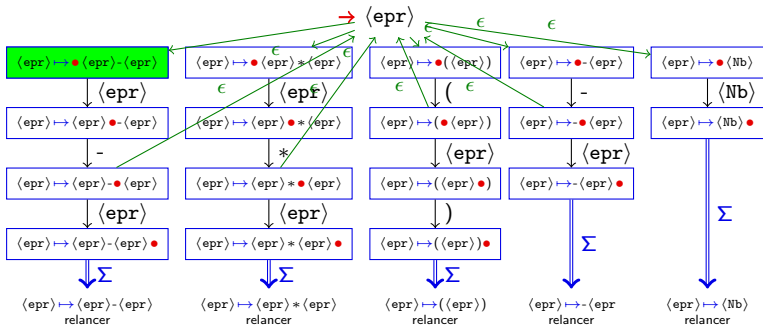
# Seconde tentative en action



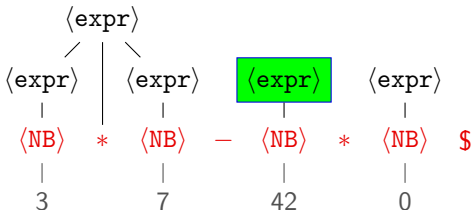
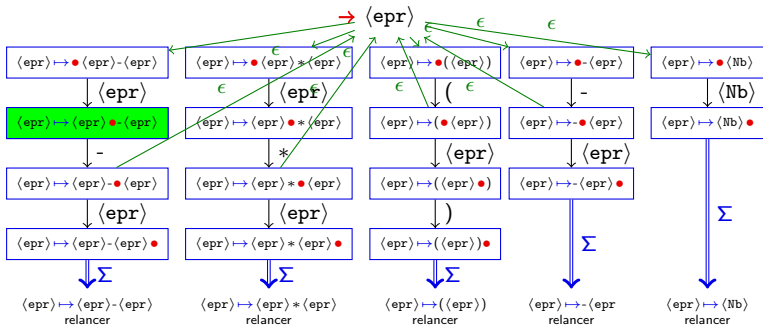
## Seconde tentative en action



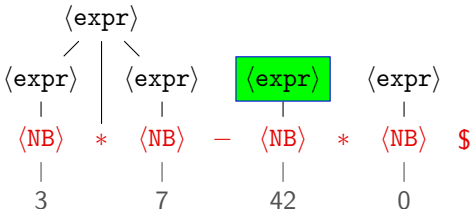
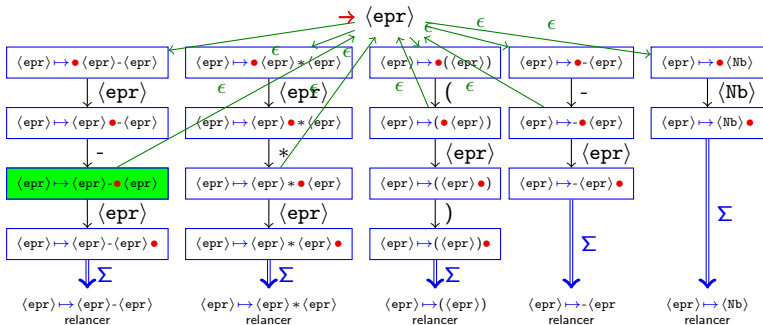
# Seconde tentative en action



## Seconde tentative en action

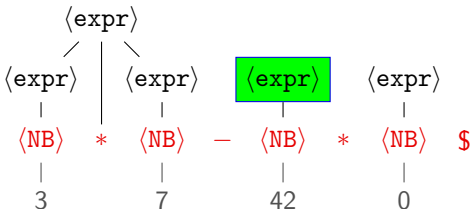
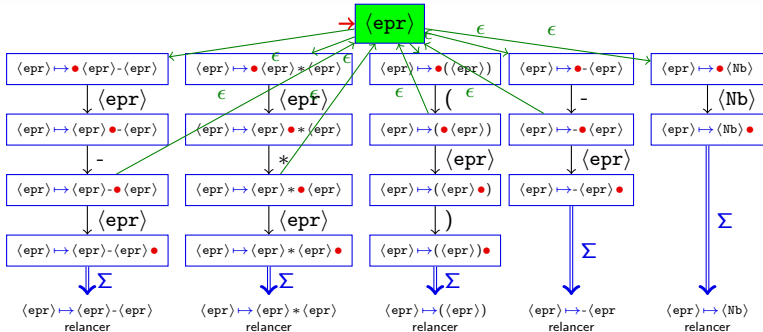


# Seconde tentative en action

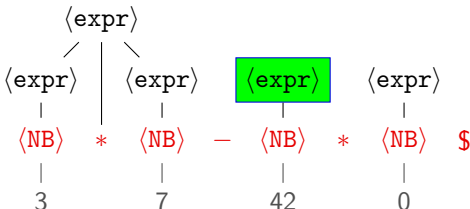
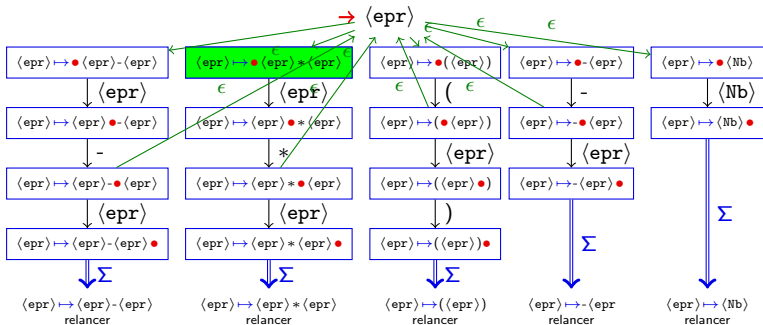




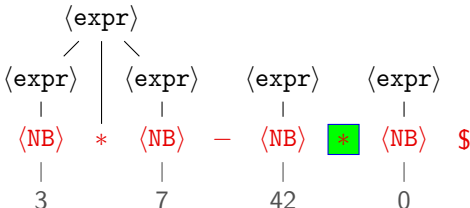
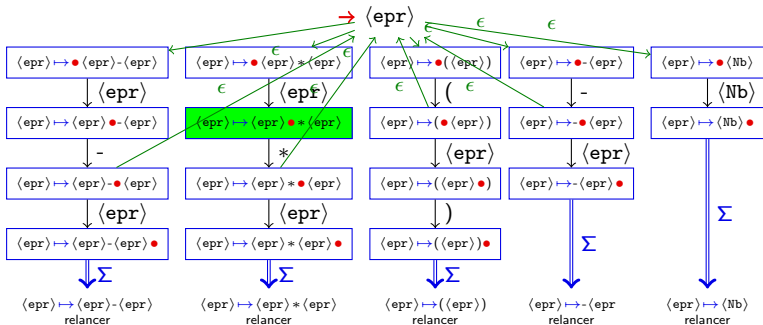
# Seconde tentative en action



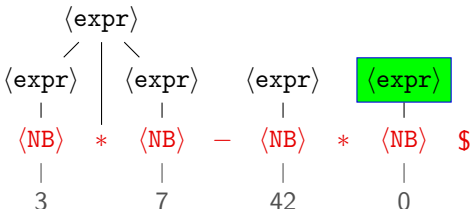
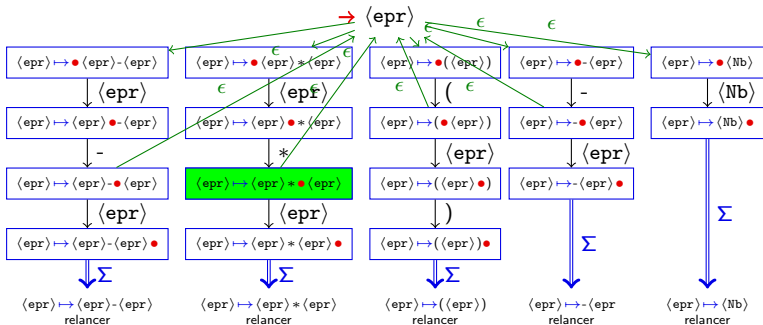
## Seconde tentative en action



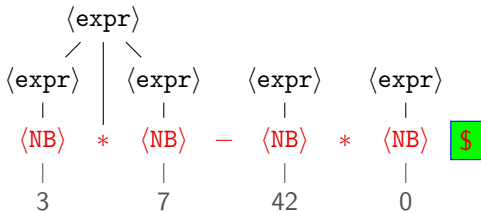
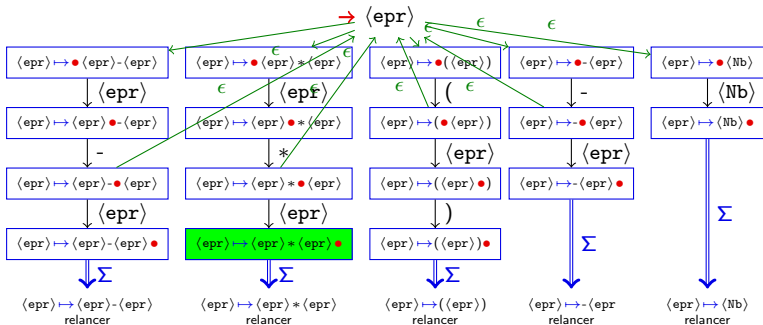
# Seconde tentative en action



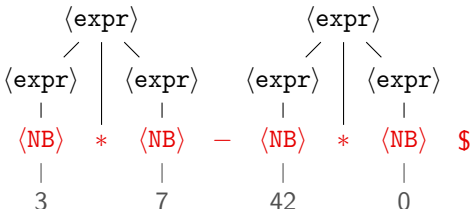
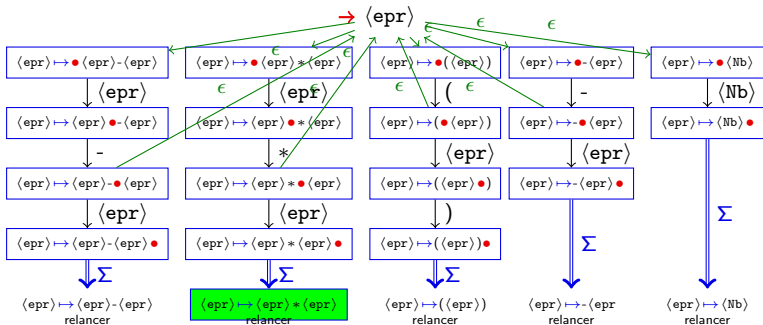
## Seconde tentative en action



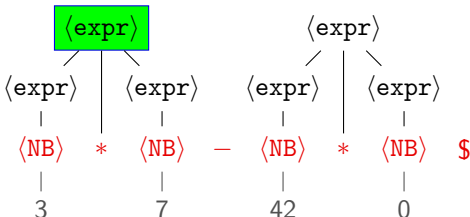
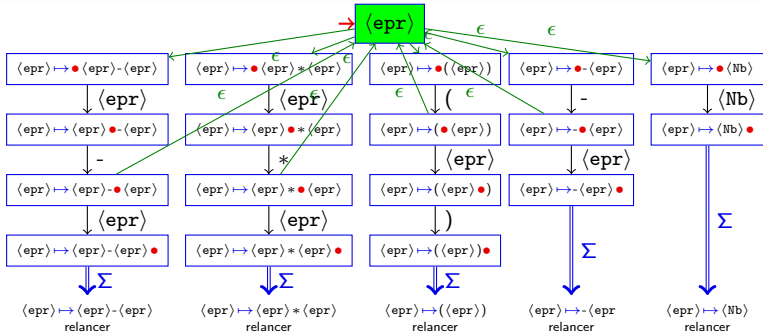
# Seconde tentative en action



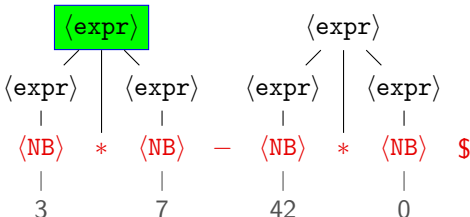
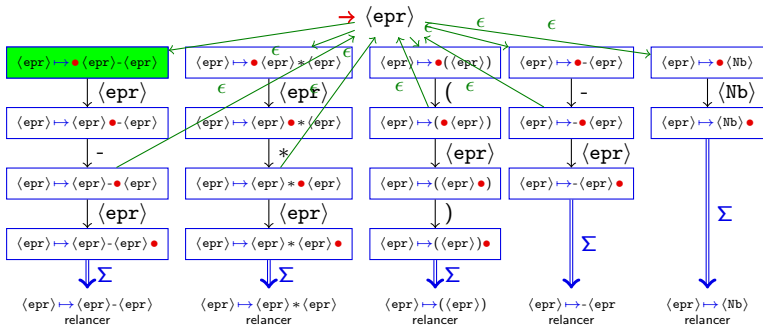
# Seconde tentative en action



## Seconde tentative en action

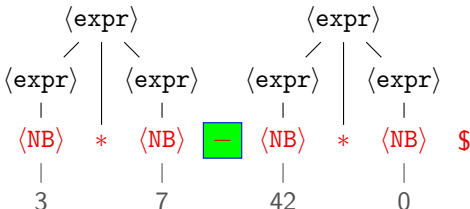
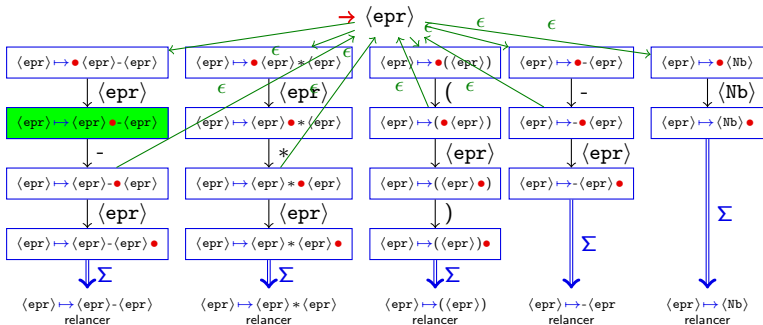


## Seconde tentative en action

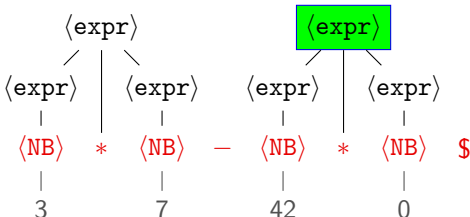
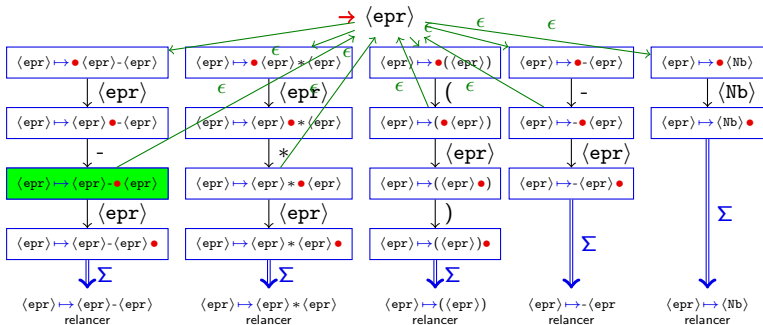




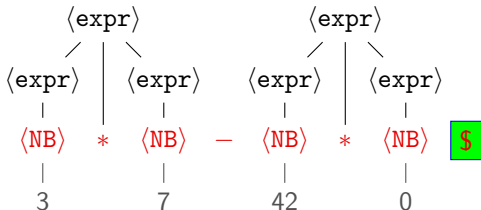
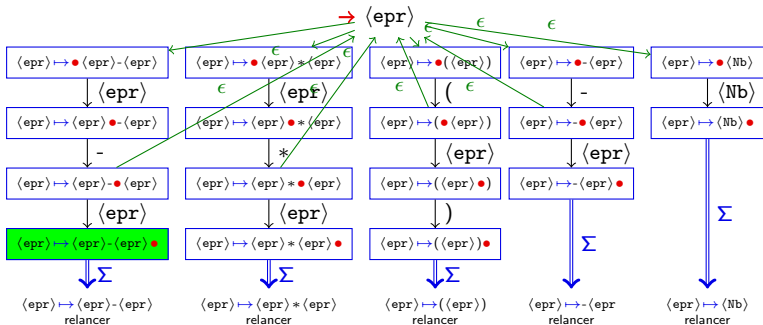
# Seconde tentative en action



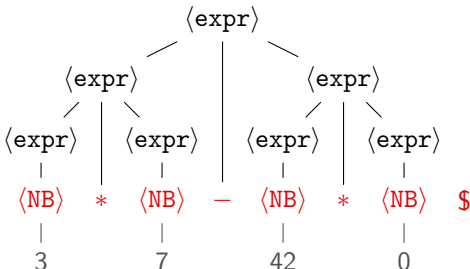
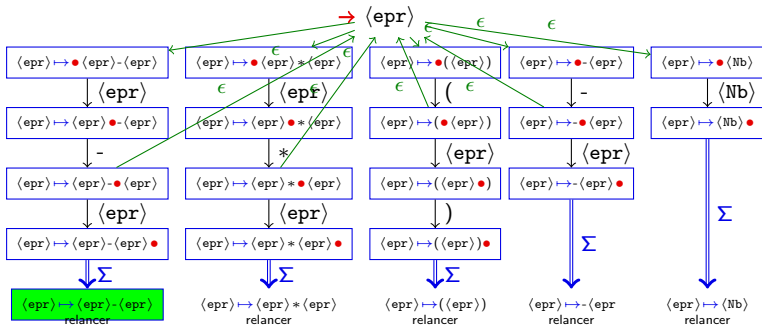
# Seconde tentative en action



# Seconde tentative en action

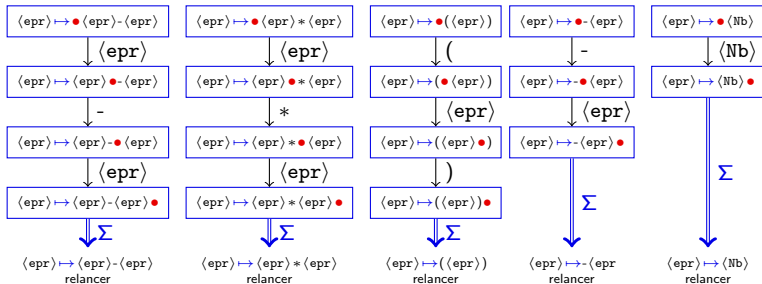


# Seconde tentative en action



# Quels $\epsilon$ -transitions ?

## Concretement

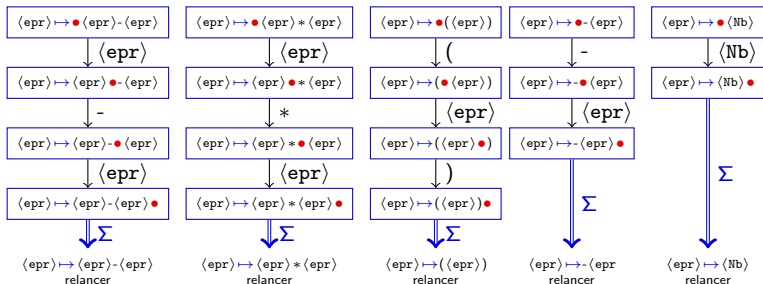


# Quels $\epsilon$ -transitions ?

## Concrètement

- ajouter un états pour chaque non terminal  $N$ ,

$\langle \text{epr} \rangle$

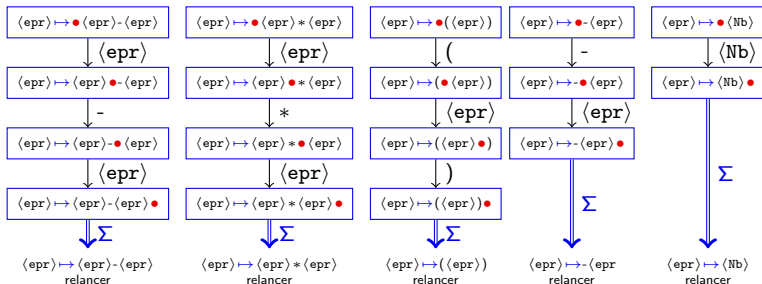


# Quels $\epsilon$ -transitions ?

## Concrètement

- ajouter un états pour chaque non terminal  $N$ ,
- seul le principal est un état initial,

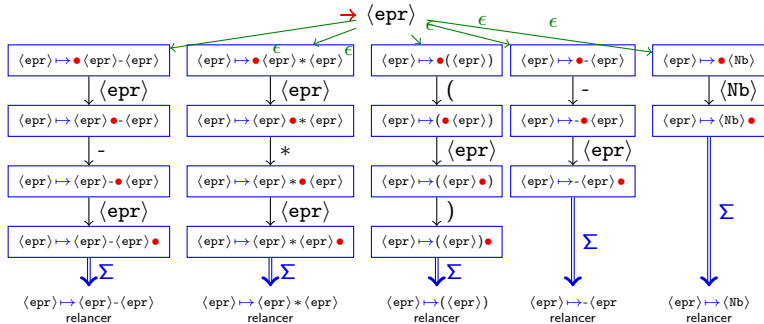
$\rightarrow \langle \text{epr} \rangle$



# Quels $\epsilon$ -transitions ?

## Concrètement

- ajouter un états pour chaque non terminal  $N$ ,
- seul le principal est un état initial,
- ajouter des  $\epsilon$ -transitions vers les règles,

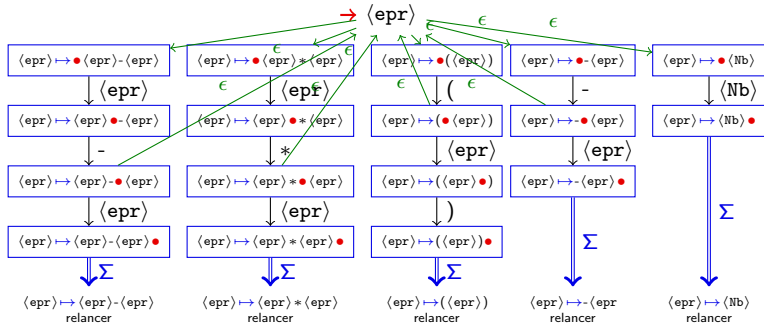




# Quels $\epsilon$ -transitions ?

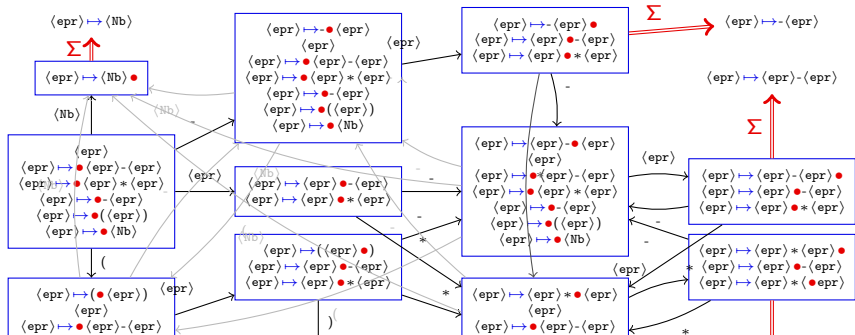
## Concretement

- ajouter un états pour chaque non terminal  $N$ ,
- seul le principal est un état initial,
- ajouter des  $\epsilon$ -transitions vers les règles,
- pour chaque règle de la forme  $N \rightarrow w_1 \bullet N' w_2$ , ajouter une  $\epsilon$ -transition vers  $N'$





# Pseudo-détermination : Parseur LR<sub>0</sub>



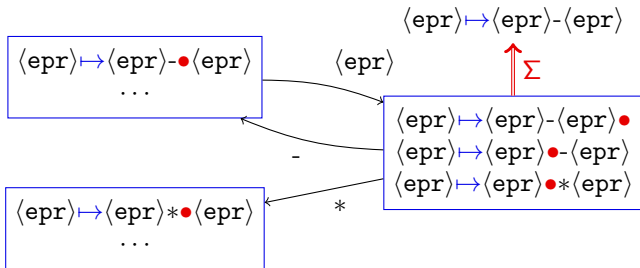
## Que faire des conflits ?

Remarque : on parle de conflit shift/reduce et reduce/reduce

- Priorité : règle certains conflits shift-action
- Associativité : règle certains conflits shift-action
- Mais ne règle pas tous les conflits...

Une grammaire dont le pseudo-déterminisé LR<sub>0</sub> est déterministe est dite LR<sub>0</sub>

# Conflit de priorité et associativité



## Shift/reduce de priorité

$\langle \text{epr} \rangle \mapsto \langle \text{epr} \rangle \bullet * \langle \text{epr} \rangle$

vs

$\langle \text{epr} \rangle \mapsto \langle \text{epr} \rangle - \langle \text{epr} \rangle \bullet$

Choisir la règle prioritaire

## Shift/reduce d'associativité

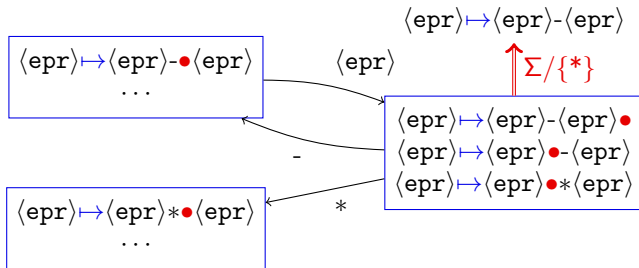
$\langle \text{epr} \rangle \mapsto \langle \text{epr} \rangle \bullet - \langle \text{epr} \rangle$

vs

$\langle \text{epr} \rangle \mapsto \langle \text{epr} \rangle - \langle \text{epr} \rangle \bullet$

Choisir le shift si asso. droit

# Conflit de priorité et associativité



## Shift/reduce de priorité

$\langle \text{epr} \rangle \mapsto \langle \text{epr} \rangle \bullet * \langle \text{epr} \rangle$

vs

$\langle \text{epr} \rangle \mapsto \langle \text{epr} \rangle - \langle \text{epr} \rangle \bullet$

Choisir la règle prioritaire

## Shift/reduce d'associativité

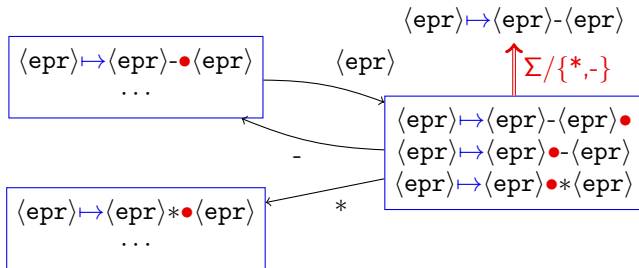
$\langle \text{epr} \rangle \mapsto \langle \text{epr} \rangle \bullet - \langle \text{epr} \rangle$

vs

$\langle \text{epr} \rangle \mapsto \langle \text{epr} \rangle - \langle \text{epr} \rangle \bullet$

Choisir le shift si asso. droit

# Conflit de priorité et associativité



## Shift/reduce de priorité

$\langle \text{epr} \rangle \mapsto \langle \text{epr} \rangle \bullet * \langle \text{epr} \rangle$

vs

$\langle \text{epr} \rangle \mapsto \langle \text{epr} \rangle - \langle \text{epr} \rangle \bullet$

Choisir la règle prioritaire

## Shift/reduce d'associativité

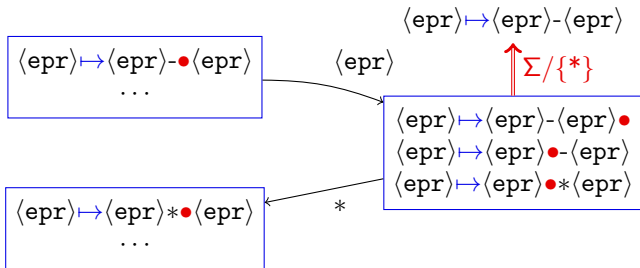
$\langle \text{epr} \rangle \mapsto \langle \text{epr} \rangle \bullet - \langle \text{epr} \rangle$

vs

$\langle \text{epr} \rangle \mapsto \langle \text{epr} \rangle - \langle \text{epr} \rangle \bullet$

Choisir le shift si asso. droit

# Conflit de priorité et associativité



## Shift/reduce de priorité

$\langle \text{epr} \rangle \mapsto \langle \text{epr} \rangle \bullet * \langle \text{epr} \rangle$

vs

$\langle \text{epr} \rangle \mapsto \langle \text{epr} \rangle - \langle \text{epr} \rangle \bullet$

Choisir la règle prioritaire

## Shift/reduce d'associativité

$\langle \text{epr} \rangle \mapsto \langle \text{epr} \rangle \bullet - \langle \text{epr} \rangle$

vs

$\langle \text{epr} \rangle \mapsto \langle \text{epr} \rangle - \langle \text{epr} \rangle \bullet$

Choisir le shift si asso. droit

# Pour ne pas perdre de temps en exam : Faire la construction déterministe directement

Expliqué en TD



# Attention, il reste des conflits "non-ambigus"

On verra ça la prochaine fois

# Pour ne pas perdre de temps en exam : Faire la construction déterministe directement

Fait en TD