

Grammar Engineering Support for Precedence Rule Recovery and Compatibility Checking

Eric Bouwers

Martin Bravenboer

Eelco Visser

Software Engineering Research Group
Department of Software Technology
Delft University of Technology
The Netherlands

March 25, 2007



Delft University of Technology



PHP

C

Java

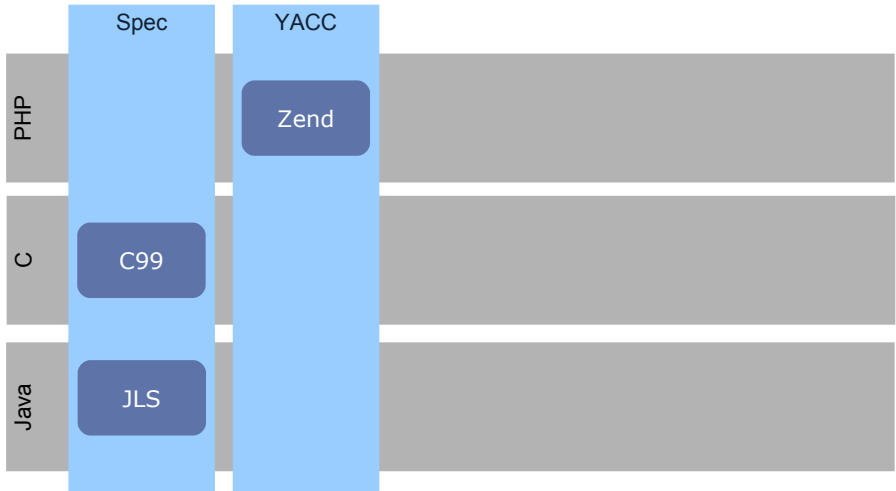
Goal: grammar engineering support

1



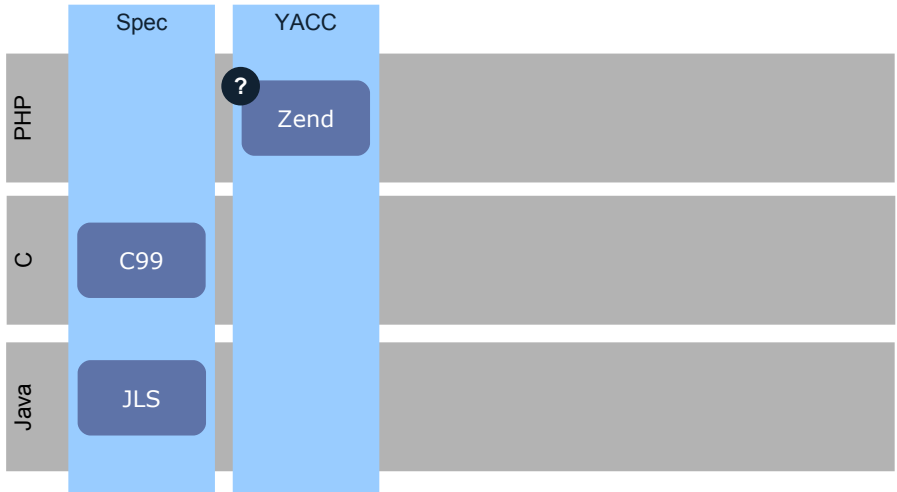
Goal: grammar engineering support

1



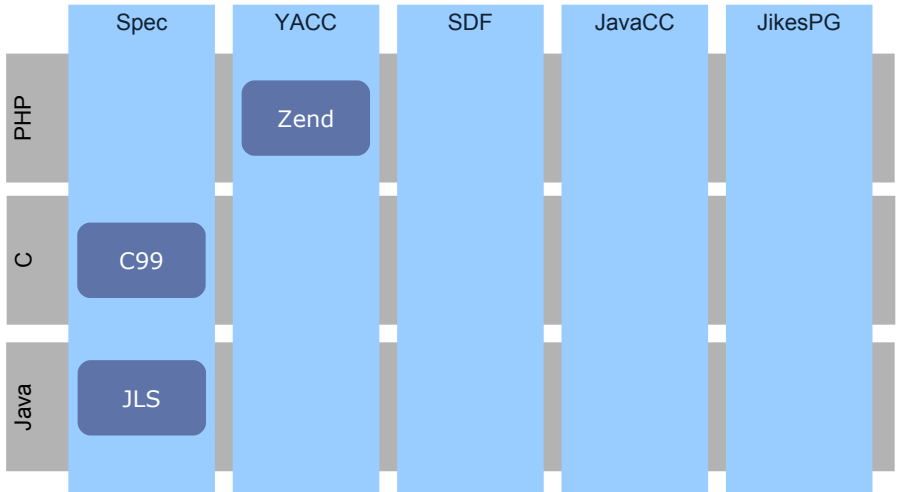
Goal: grammar engineering support

1



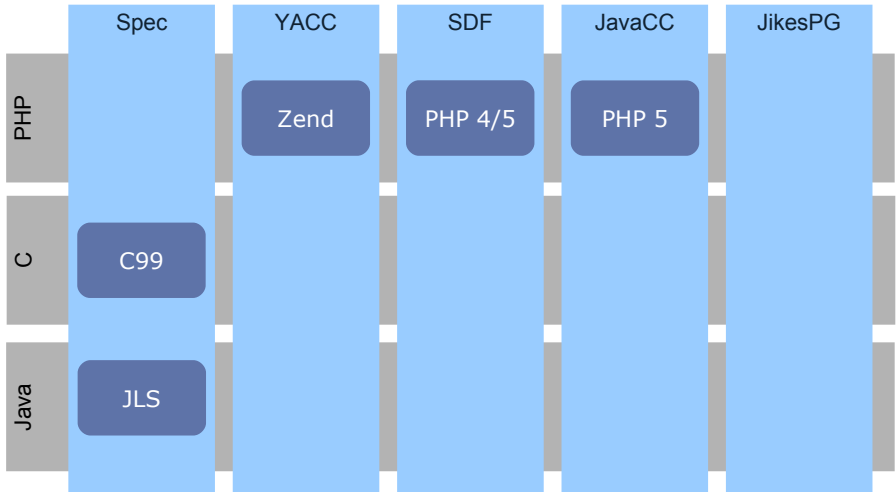
Goal: grammar engineering support

1



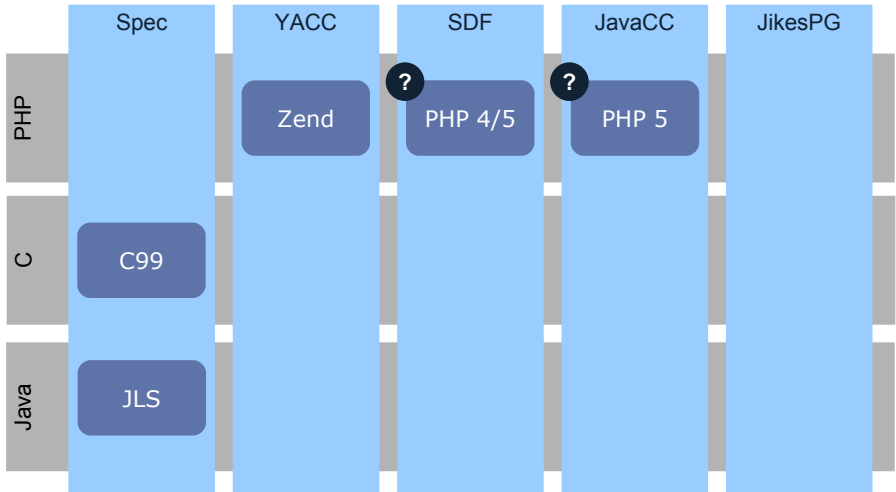
Goal: grammar engineering support

1



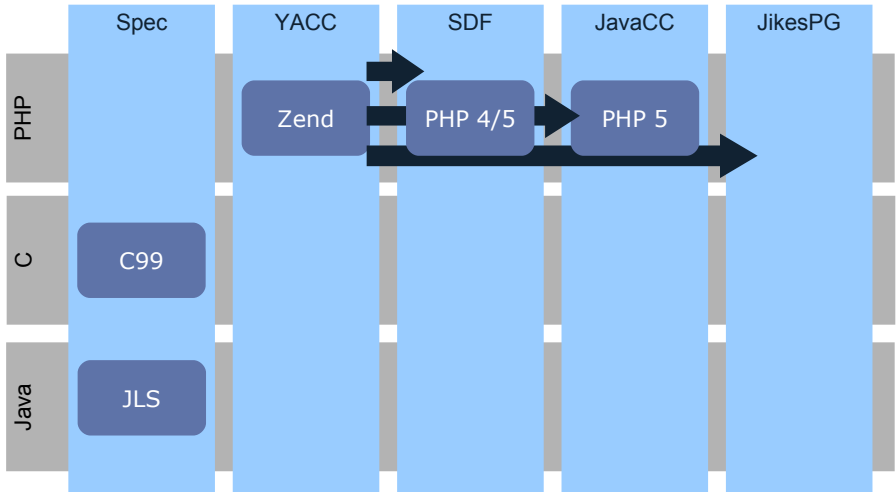
Goal: grammar engineering support

1



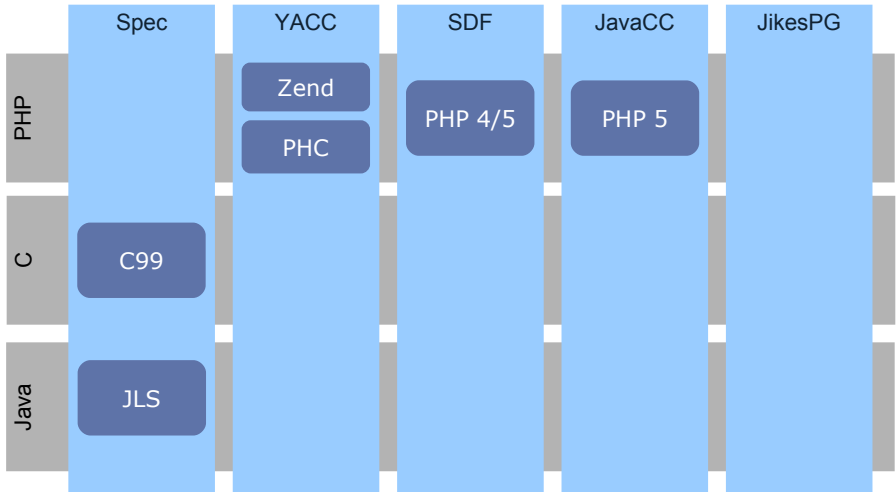
Goal: grammar engineering support

1



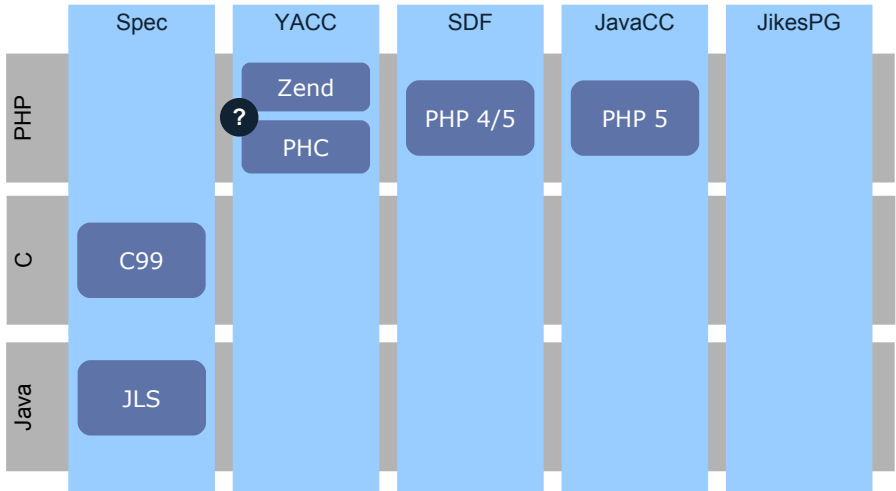
Goal: grammar engineering support

1



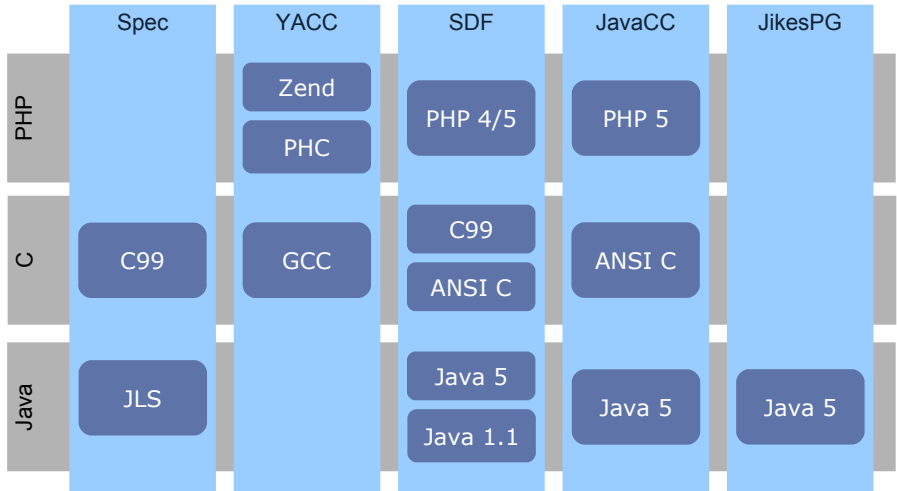
Goal: grammar engineering support

1



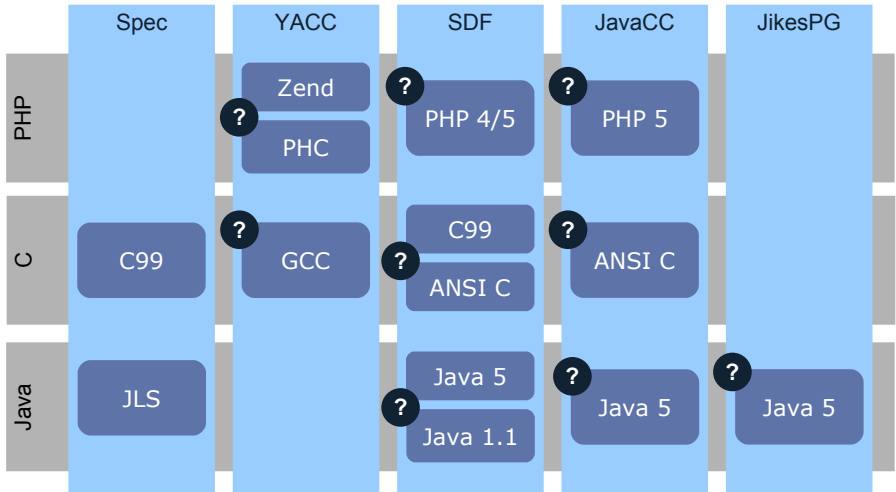
Goal: grammar engineering support

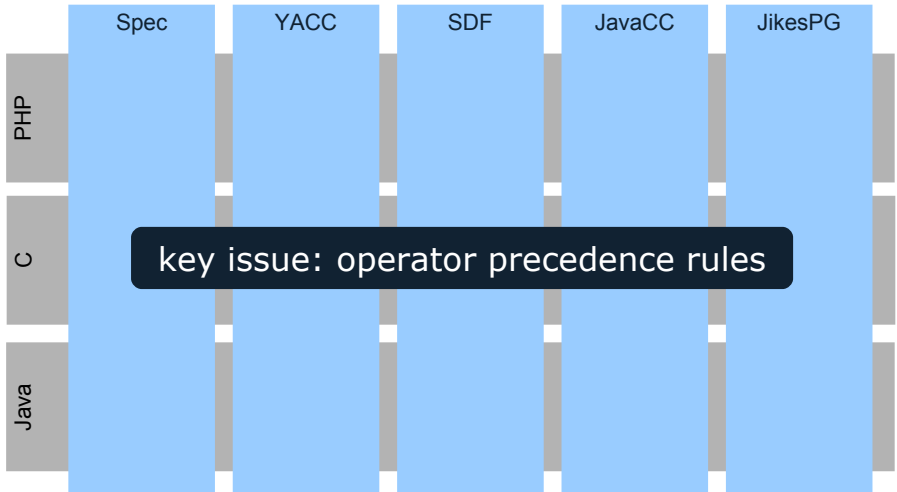
1



Goal: grammar engineering support

1



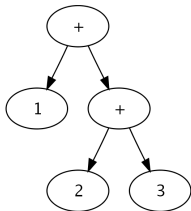
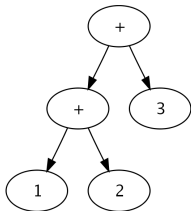


What are precedence rules?

2

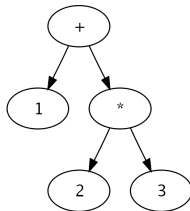
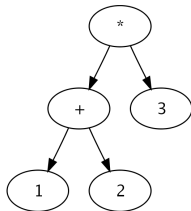
associativity

$1 + 2 + 3$



precedence (priority)

$1 + 2 * 3$



Liberate precedence rules from idiosyncrasies of parser generators

Formalism

Parsing technology independent definition of precedence

Recovery

Recover precedence rules from existing grammars

Compatibility

Compare precedence rules of grammars

Migration

Migrate a grammar from one grammar formalism to another, including its precedence rules

yacc without precedence

```
%token NUM
```

```
E: E '+' T
```

```
    E '-' T
```

```
    | T
```

```
T: T '*' F
```

```
    | T '/' F
```

```
    | F
```

```
F: NUM
```

yacc without precedence

```
%token NUM

E: E '+' T
   E '-' T
   | T
T: T '*' F
   | T '/' F
   | F
F: NUM
```

yacc with precedence

```
%token NUM
%left '+' '-'
%left '*' '/'

E: NUM
   | E '+' E
   | E '-' E
   | E '*' E
   | E '/' E
```

sdf without precedence

context-free syntax

$E \rightarrow E + T$

$E \rightarrow E - T$

$T \rightarrow E$

$T \rightarrow T * F$

$T \rightarrow T / F$

$F \rightarrow T$

$NUM \rightarrow F$

sdf without precedence

context-free syntax

 $E \rightarrow E + T$ $E \rightarrow E - T$ $T \rightarrow E$ $T \rightarrow T * F$ $T \rightarrow T / F$ $F \rightarrow T$ $NUM \rightarrow F$

sdf with precedence

context-free priorities

{ left:

 $E \rightarrow E * E$ {left} $E \rightarrow E / E$ {left}

}

> { left:

 $E \rightarrow E + E$ {left} $E \rightarrow E - E$ {left}

}

context-free syntax

 $NUM \rightarrow E$

yacc without precedence

%token NUM

E: E '+' T

 E '-' T

 | T

T: T '*' F

 | T '/' F

 | F

F: NUM

sdf with precedence

context-free priorities

{ left:

 E "*" E -> E {left}

 E "/" E -> E {left}

}

> { left:

 E "+" E -> E {left}

 E "-" E -> E {left}

}

context-free syntax

NUM -> E

yacc with precedence

```
%token NUM
%left '+' '-'
%left '*' '/'
```

```
E: NUM
  | E '+' E
  | E '-' E
  | E '*' E
  | E '/' E
```

sdf with precedence

```
context-free priorities
{ left:
  E "*" E -> E {left}
  E "/" E -> E {left}
}
```

```
> { left:
  E "+" E -> E {left}
  E "-" E -> E {left}
}
```

context-free syntax

```
NUM -> E
```

yacc with precedence

```
%token V  
%left '='  
%left '+'  
%left '!'
```

```
E: V  
  | V '=' E  
  | E '+' E  
  | '!' E
```

sdf with precedence

```
context-free priorities  
    "!" E -> E  
> E "+" E -> E {left}  
> V "=" E -> E
```

```
context-free syntax  
    V -> E
```

yacc with precedence

```
%token V
%left '='
%left '+'
%left '!'
```

```
E: V
  | V '=' E
  | E '+' E
  | '!' E
```

```
echo '!a=b' | ./expr
Not(Assign("a",Var("b")))
```

sdf with precedence

```
context-free priorities
    "!" E -> E
> E "+" E -> E {left}
> V "=" E -> E
```

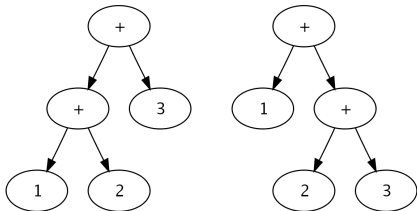
```
context-free syntax
    V -> E
```

```
echo '!a=b' | sglr -p expr.tbl
character '=' unexpected
```

How to define precedence rules in a parsing technology independent way?

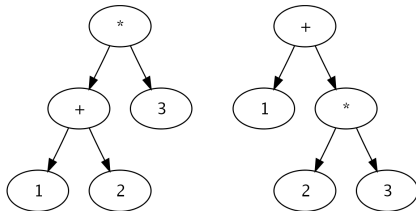
associativity

$1 + 2 + 3$



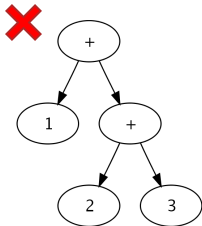
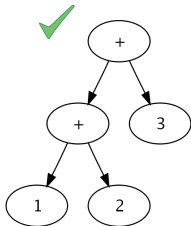
precedence (priority)

$1 + 2 * 3$



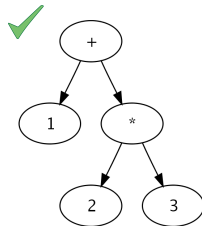
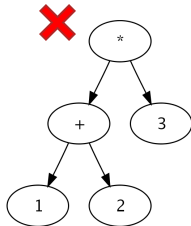
associativity

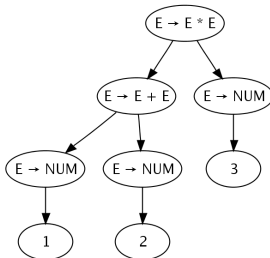
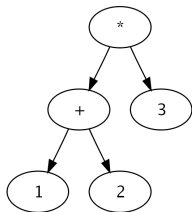
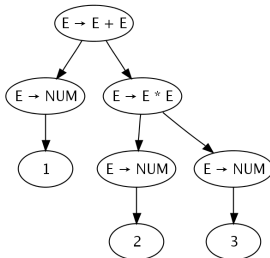
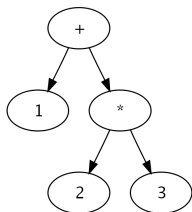
$1 + 2 + 3$



precedence (priority)

$1 + 2 * 3$



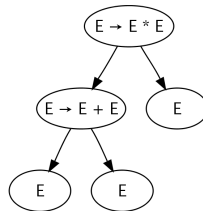
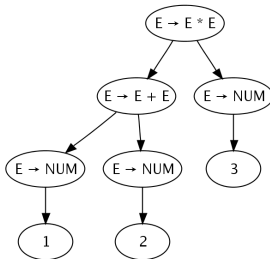
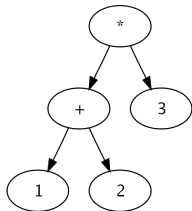
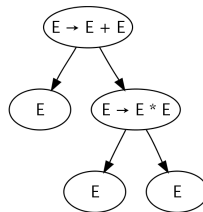
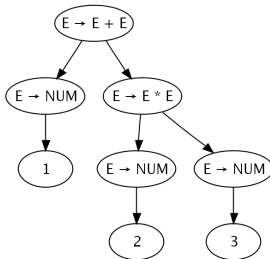
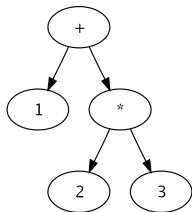


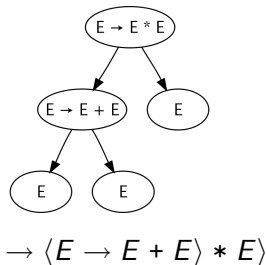
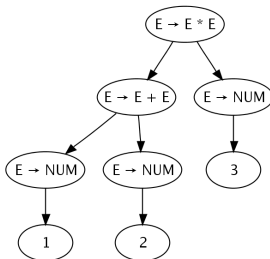
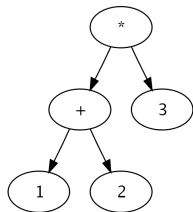
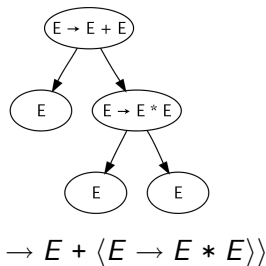
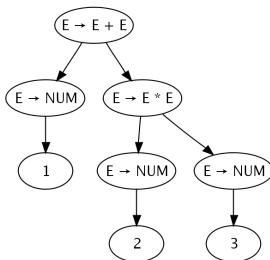
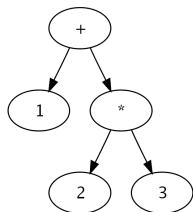
```

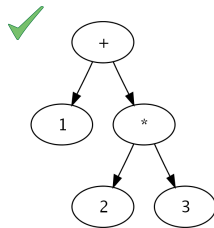
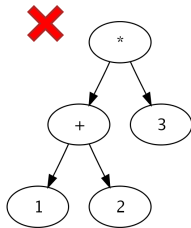
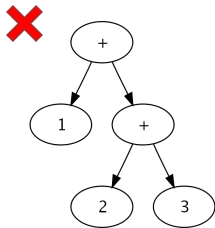
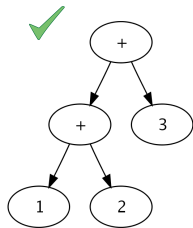
%token NUM
%left '+'
%left '*'
    
```

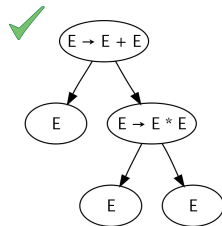
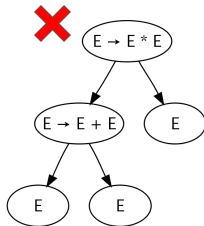
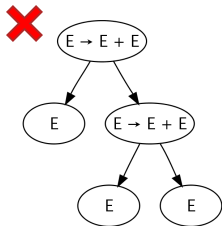
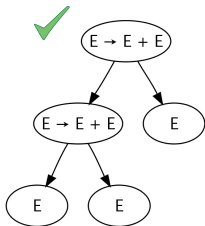
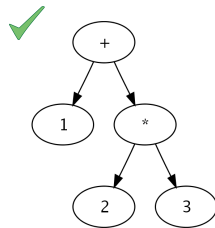
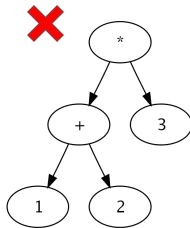
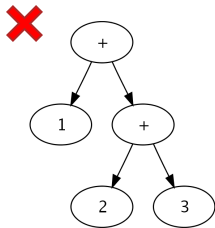
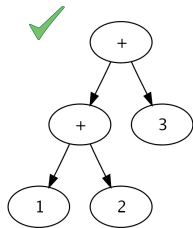
```

E: NUM
  | E '+' E
  | E '*' E
    
```

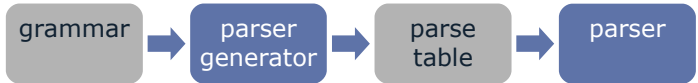


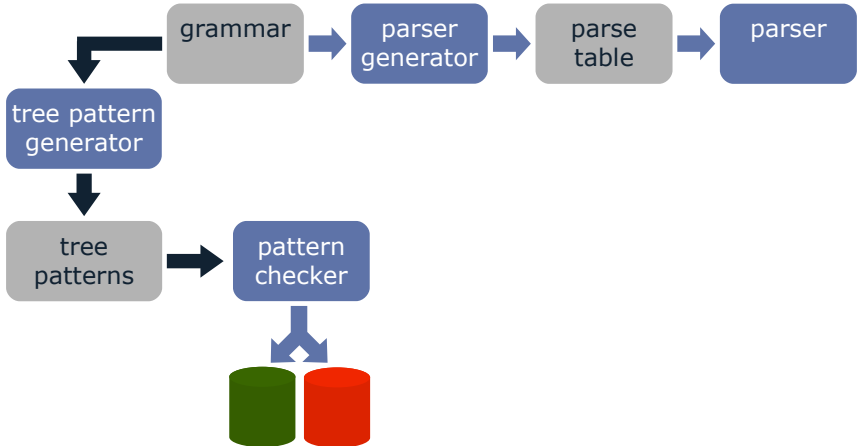






Precedence rules: set of invalid tree patterns





```
%token NUM
```

```
%left '+'
```

```
%left '*'
```

```
E: NUM
```

```
  | E '+' E
```

```
  | E '*' E
```

%token NUM

%left '+'

%left '*'

E: NUM

| E '+' E

| E '*' E

$\langle E \rightarrow E * \langle E \rightarrow E * E \rangle \rangle$

$\langle E \rightarrow E + \langle E \rightarrow E + E \rangle \rangle$

$\langle E \rightarrow E * \langle E \rightarrow E + E \rangle \rangle$

$\langle E \rightarrow E + \langle E \rightarrow E * E \rangle \rangle$

$\langle E \rightarrow E * \langle E \rightarrow \text{NUM} \rangle \rangle$

$\langle E \rightarrow E + \langle E \rightarrow \text{NUM} \rangle \rangle$

$\langle E \rightarrow \langle E \rightarrow E * E \rangle * E \rangle$

$\langle E \rightarrow \langle E \rightarrow E + E \rangle + E \rangle$

$\langle E \rightarrow \langle E \rightarrow E + E \rangle * E \rangle$

$\langle E \rightarrow \langle E \rightarrow E * E \rangle + E \rangle$

$\langle E \rightarrow \langle E \rightarrow \text{NUM} \rangle * E \rangle$

$\langle E \rightarrow \langle E \rightarrow \text{NUM} \rangle + E \rangle$

```
%token NUM
```

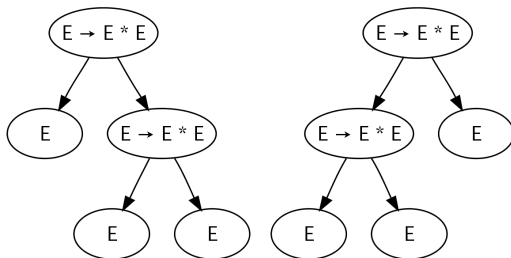
```
%left '+'
```

```
%left '*'
```

```
E: NUM
```

```
| E '+' E
```

```
| E '*' E
```



- $\langle E \rightarrow E * \langle E \rightarrow E * E \rangle \rangle$
 $\langle E \rightarrow E + \langle E \rightarrow E + E \rangle \rangle$
 $\langle E \rightarrow E * \langle E \rightarrow E + E \rangle \rangle$
 $\langle E \rightarrow E + \langle E \rightarrow E * E \rangle \rangle$
 $\langle E \rightarrow E * \langle E \rightarrow \text{NUM} \rangle \rangle$
 $\langle E \rightarrow E + \langle E \rightarrow \text{NUM} \rangle \rangle$

- $\langle E \rightarrow \langle E \rightarrow E * E \rangle * E \rangle$
 $\langle E \rightarrow \langle E \rightarrow E + E \rangle + E \rangle$
 $\langle E \rightarrow \langle E \rightarrow E + E \rangle * E \rangle$
 $\langle E \rightarrow \langle E \rightarrow E * E \rangle + E \rangle$
 $\langle E \rightarrow \langle E \rightarrow \text{NUM} \rangle * E \rangle$
 $\langle E \rightarrow \langle E \rightarrow \text{NUM} \rangle + E \rangle$

%token NUM

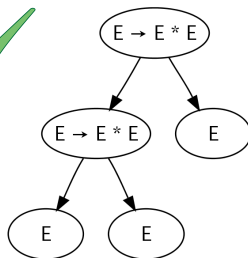
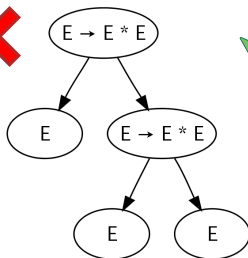
%left '+'

%left '*'

E: NUM

| E '+' E

| E '*' E



● $\langle E \rightarrow E * \langle E \rightarrow E * E \rangle \rangle$
 $\langle E \rightarrow E + \langle E \rightarrow E + E \rangle \rangle$
 $\langle E \rightarrow E * \langle E \rightarrow E + E \rangle \rangle$
 $\langle E \rightarrow E + \langle E \rightarrow E * E \rangle \rangle$
 $\langle E \rightarrow E * \langle E \rightarrow \text{NUM} \rangle \rangle$
 $\langle E \rightarrow E + \langle E \rightarrow \text{NUM} \rangle \rangle$

● $\langle E \rightarrow \langle E \rightarrow E * E \rangle * E \rangle$
 $\langle E \rightarrow \langle E \rightarrow E + E \rangle + E \rangle$
 $\langle E \rightarrow \langle E \rightarrow E + E \rangle * E \rangle$
 $\langle E \rightarrow \langle E \rightarrow E * E \rangle + E \rangle$
 $\langle E \rightarrow \langle E \rightarrow \text{NUM} \rangle * E \rangle$
 $\langle E \rightarrow \langle E \rightarrow \text{NUM} \rangle + E \rangle$

%token NUM

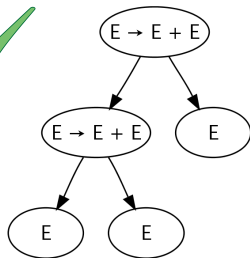
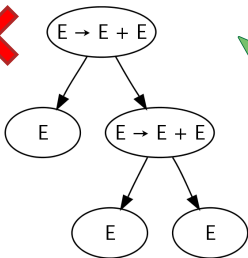
%left '+'

%left '*'

E: NUM

| E '+' E

| E '*' E



● $\langle E \rightarrow E * \langle E \rightarrow E * E \rangle \rangle$

● $\langle E \rightarrow E + \langle E \rightarrow E + E \rangle \rangle$

$\langle E \rightarrow E * \langle E \rightarrow E + E \rangle \rangle$

$\langle E \rightarrow E + \langle E \rightarrow E * E \rangle \rangle$

$\langle E \rightarrow E * \langle E \rightarrow \text{NUM} \rangle \rangle$

$\langle E \rightarrow E + \langle E \rightarrow \text{NUM} \rangle \rangle$

● $\langle E \rightarrow \langle E \rightarrow E * E \rangle * E \rangle$

● $\langle E \rightarrow \langle E \rightarrow E + E \rangle + E \rangle$

$\langle E \rightarrow \langle E \rightarrow E + E \rangle * E \rangle$

$\langle E \rightarrow \langle E \rightarrow E * E \rangle + E \rangle$

$\langle E \rightarrow \langle E \rightarrow \text{NUM} \rangle * E \rangle$

$\langle E \rightarrow \langle E \rightarrow \text{NUM} \rangle + E \rangle$

%token NUM

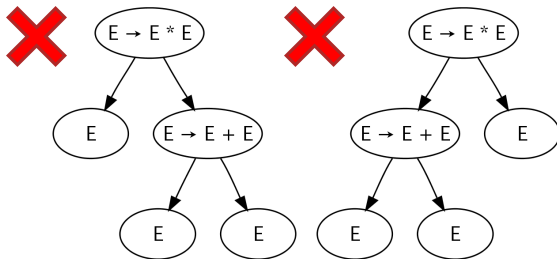
%left '+'

%left '*'

E: NUM

| E '+' E

| E '*' E



● $\langle E \rightarrow E * \langle E \rightarrow E * E \rangle \rangle$

● $\langle E \rightarrow E + \langle E \rightarrow E + E \rangle \rangle$

● $\langle E \rightarrow E * \langle E \rightarrow E + E \rangle \rangle$

$\langle E \rightarrow E + \langle E \rightarrow E * E \rangle \rangle$

$\langle E \rightarrow E * \langle E \rightarrow \text{NUM} \rangle \rangle$

$\langle E \rightarrow E + \langle E \rightarrow \text{NUM} \rangle \rangle$

● $\langle E \rightarrow \langle E \rightarrow E * E \rangle * E \rangle$

● $\langle E \rightarrow \langle E \rightarrow E + E \rangle + E \rangle$

● $\langle E \rightarrow \langle E \rightarrow E + E \rangle * E \rangle$

$\langle E \rightarrow \langle E \rightarrow E * E \rangle + E \rangle$

$\langle E \rightarrow \langle E \rightarrow \text{NUM} \rangle * E \rangle$

$\langle E \rightarrow \langle E \rightarrow \text{NUM} \rangle + E \rangle$

%token NUM

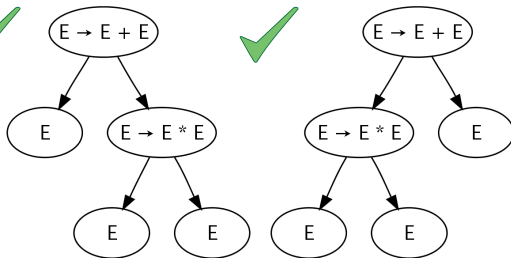
%left '+'

%left '*'

E: NUM

| E '+' E

| E '*' E



● $\langle E \rightarrow E * \langle E \rightarrow E * E \rangle \rangle$

● $\langle E \rightarrow E + \langle E \rightarrow E + E \rangle \rangle$

● $\langle E \rightarrow E * \langle E \rightarrow E + E \rangle \rangle$

● $\langle E \rightarrow E + \langle E \rightarrow E * E \rangle \rangle$

$\langle E \rightarrow E * \langle E \rightarrow \text{NUM} \rangle \rangle$

$\langle E \rightarrow E + \langle E \rightarrow \text{NUM} \rangle \rangle$

● $\langle E \rightarrow \langle E \rightarrow E * E \rangle * E \rangle$

● $\langle E \rightarrow \langle E \rightarrow E + E \rangle + E \rangle$

● $\langle E \rightarrow \langle E \rightarrow E + E \rangle * E \rangle$

● $\langle E \rightarrow \langle E \rightarrow E * E \rangle + E \rangle$

$\langle E \rightarrow \langle E \rightarrow \text{NUM} \rangle * E \rangle$

$\langle E \rightarrow \langle E \rightarrow \text{NUM} \rangle + E \rangle$

%token NUM

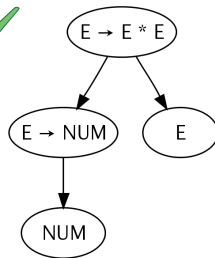
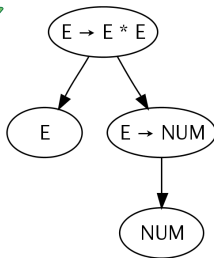
%left '+'

%left '*'

E: NUM

| E '+' E

| E '*' E



● $\langle E \rightarrow E * \langle E \rightarrow E * E \rangle \rangle$

● $\langle E \rightarrow E + \langle E \rightarrow E + E \rangle \rangle$

● $\langle E \rightarrow E * \langle E \rightarrow E + E \rangle \rangle$

● $\langle E \rightarrow E + \langle E \rightarrow E * E \rangle \rangle$

● $\langle E \rightarrow E * \langle E \rightarrow \text{NUM} \rangle \rangle$

$\langle E \rightarrow E + \langle E \rightarrow \text{NUM} \rangle \rangle$

● $\langle E \rightarrow \langle E \rightarrow E * E \rangle * E \rangle$

● $\langle E \rightarrow \langle E \rightarrow E + E \rangle + E \rangle$

● $\langle E \rightarrow \langle E \rightarrow E + E \rangle * E \rangle$

● $\langle E \rightarrow \langle E \rightarrow E * E \rangle + E \rangle$

● $\langle E \rightarrow \langle E \rightarrow \text{NUM} \rangle * E \rangle$

$\langle E \rightarrow \langle E \rightarrow \text{NUM} \rangle + E \rangle$

```
%token NUM
```

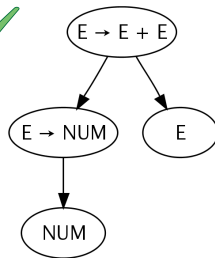
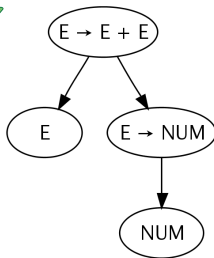
```
%left '+'
```

```
%left '*'
```

```
E: NUM
```

```
| E '+' E
```

```
| E '*' E
```



● $\langle E \rightarrow E * \langle E \rightarrow E * E \rangle \rangle$

● $\langle E \rightarrow E + \langle E \rightarrow E + E \rangle \rangle$

● $\langle E \rightarrow E * \langle E \rightarrow E + E \rangle \rangle$

● $\langle E \rightarrow E + \langle E \rightarrow E * E \rangle \rangle$

● $\langle E \rightarrow E * \langle E \rightarrow \text{NUM} \rangle \rangle$

● $\langle E \rightarrow E + \langle E \rightarrow \text{NUM} \rangle \rangle$

● $\langle E \rightarrow \langle E \rightarrow E * E \rangle * E \rangle$

● $\langle E \rightarrow \langle E \rightarrow E + E \rangle + E \rangle$

● $\langle E \rightarrow \langle E \rightarrow E + E \rangle * E \rangle$

● $\langle E \rightarrow \langle E \rightarrow E * E \rangle + E \rangle$

● $\langle E \rightarrow \langle E \rightarrow \text{NUM} \rangle * E \rangle$

● $\langle E \rightarrow \langle E \rightarrow \text{NUM} \rangle + E \rangle$

context-free priorities

$E \text{ "*" } E \rightarrow E$

$> E \text{ "+" } E \rightarrow E$

context-free syntax

$\text{NUM} \rightarrow E$

● $\langle E \rightarrow E * \langle E \rightarrow E * E \rangle \rangle$

● $\langle E \rightarrow E + \langle E \rightarrow E + E \rangle \rangle$

● $\langle E \rightarrow E * \langle E \rightarrow E + E \rangle \rangle$

● $\langle E \rightarrow E + \langle E \rightarrow E * E \rangle \rangle$

● $\langle E \rightarrow E * \langle E \rightarrow \text{NUM} \rangle \rangle$

● $\langle E \rightarrow E + \langle E \rightarrow \text{NUM} \rangle \rangle$

● $\langle E \rightarrow \langle E \rightarrow E * E \rangle * E \rangle$

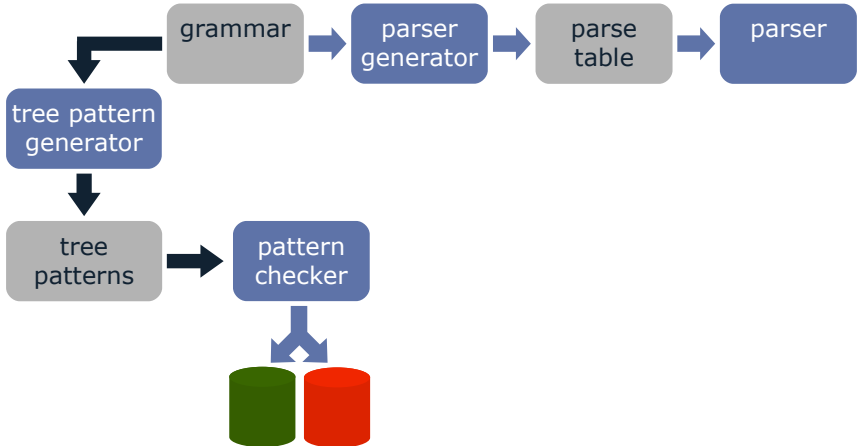
● $\langle E \rightarrow \langle E \rightarrow E + E \rangle + E \rangle$

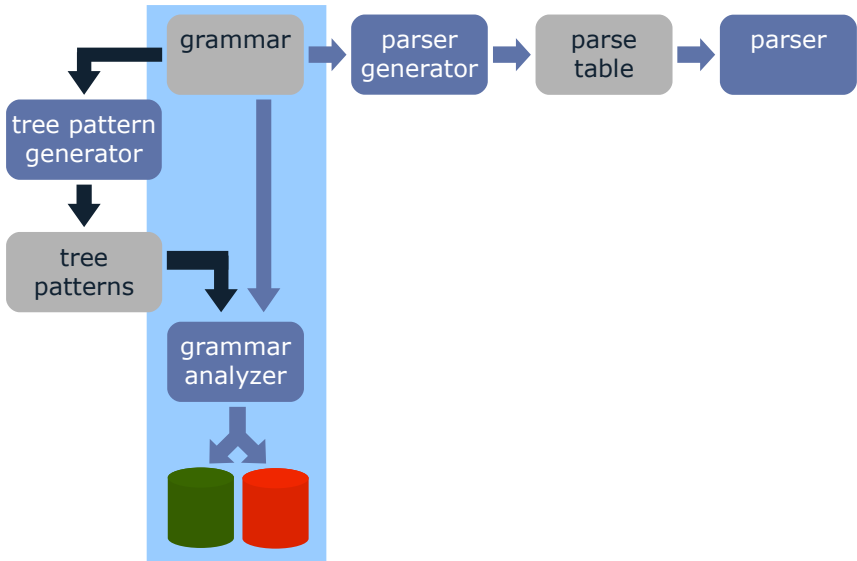
● $\langle E \rightarrow \langle E \rightarrow E + E \rangle * E \rangle$

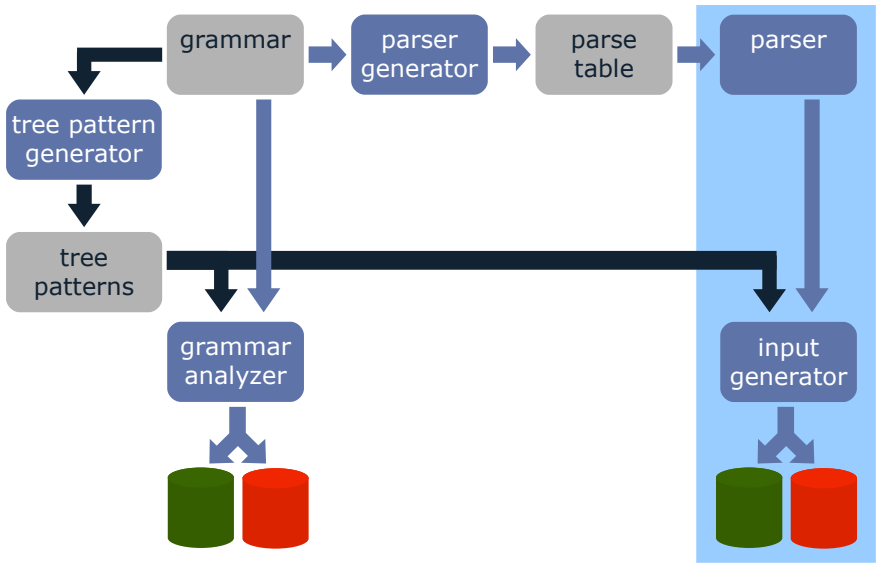
● $\langle E \rightarrow \langle E \rightarrow E * E \rangle + E \rangle$

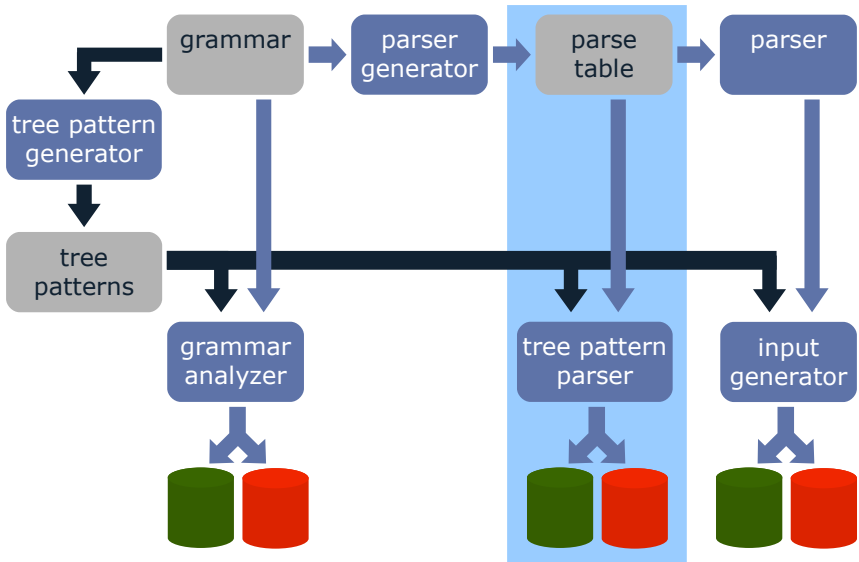
● $\langle E \rightarrow \langle E \rightarrow \text{NUM} \rangle * E \rangle$

● $\langle E \rightarrow \langle E \rightarrow \text{NUM} \rangle + E \rangle$

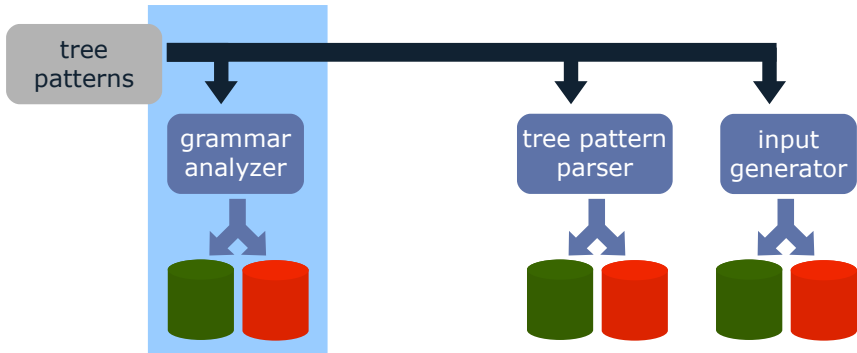




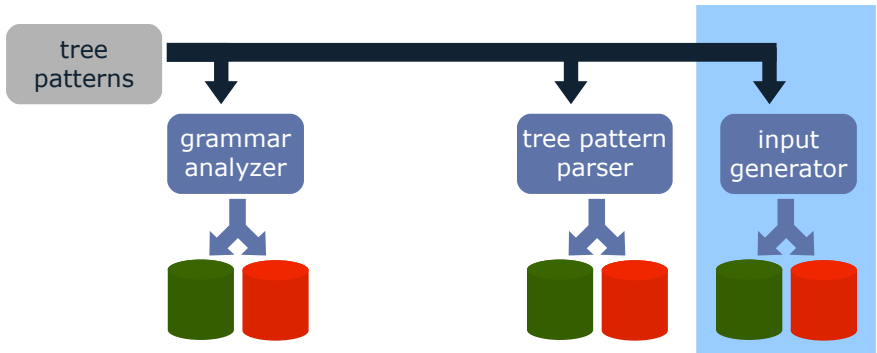




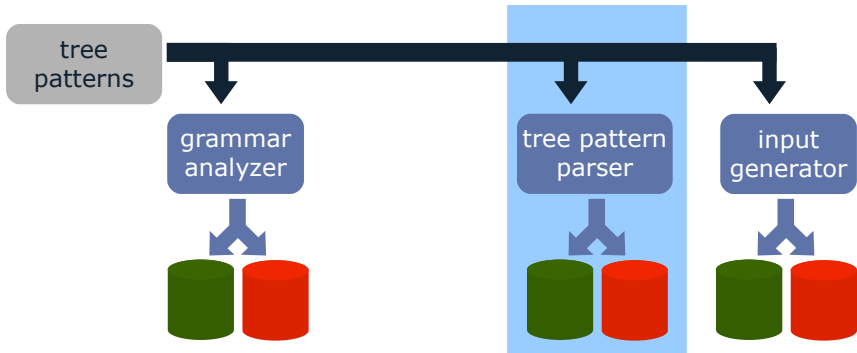
- unreliable reimplementation of parser *generator*
- often consider only precedence declarations

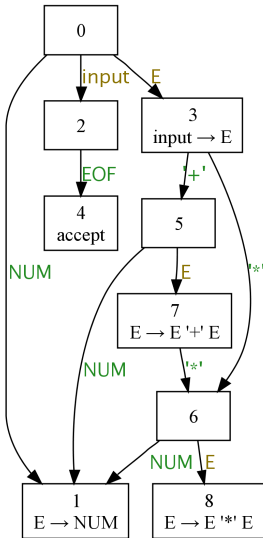


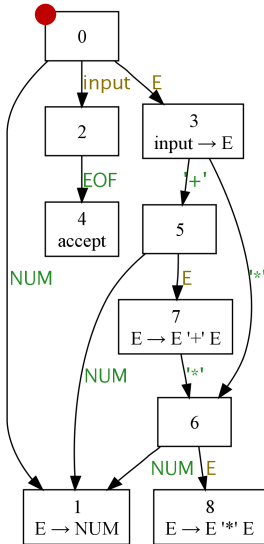
- unreliable reimplementation of parser *generator*
- often consider only precedence declarations
- semantic actions
- check resulting parse tree
- reliable parser implementation



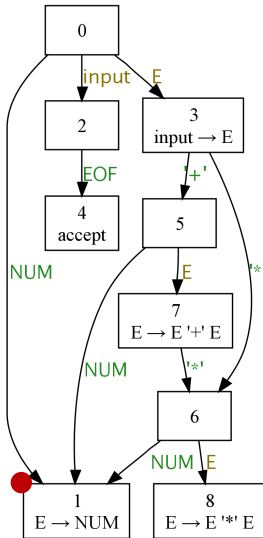
- unreliable reimplementation of parser *generator*
- often consider only precedence declarations
- no output checking necessary
- reimplement the parser
- semantic actions
- check resulting parse tree
- reliable parser implementation



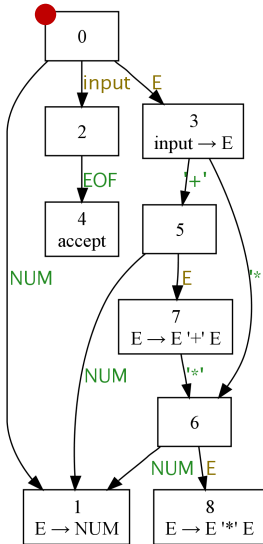




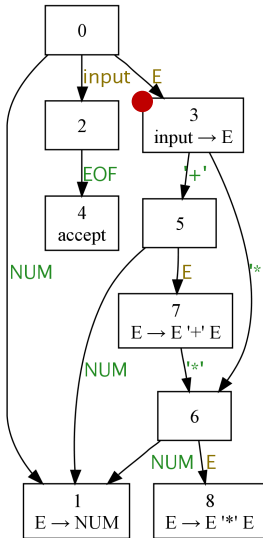
| 0 | NUM + NUM |



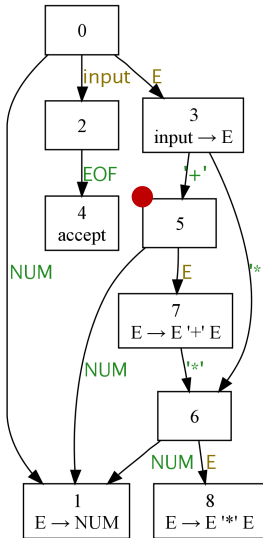
	0	NUM + NUM
shift	0, 1	+ NUM



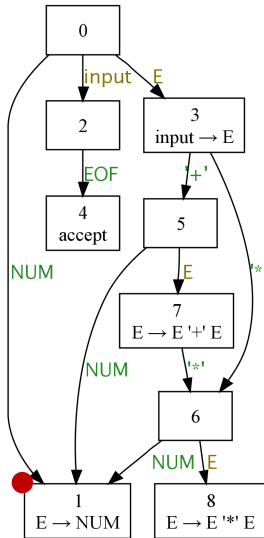
	0	NUM + NUM
shift	0, 1	+ NUM
reduce	0	$E + \text{NUM}$



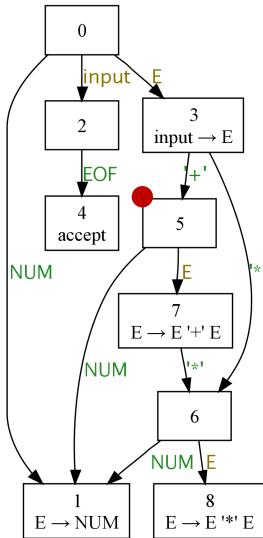
	0	NUM + NUM
shift	0, 1	+ NUM
reduce	0	$E + \text{NUM}$
goto	0, 3	+ NUM



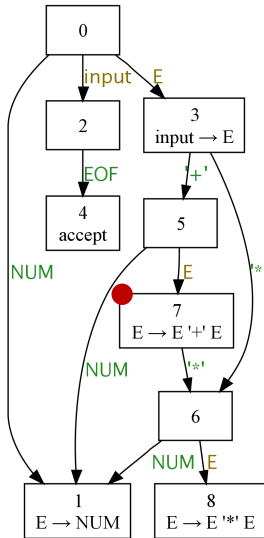
	0	NUM + NUM
shift	0, 1	+ NUM
reduce	0	$E + \text{NUM}$
goto	0, 3	+ NUM
shift	0, 3, 5	NUM



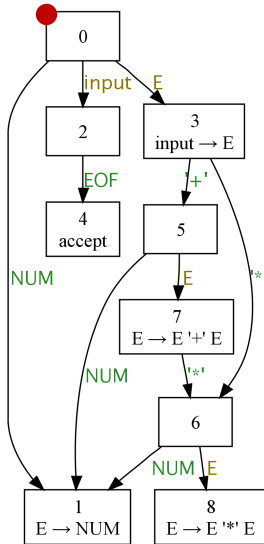
	0	NUM + NUM
shift	0, 1	+ NUM
reduce	0	$E + \text{NUM}$
goto	0, 3	+ NUM
shift	0, 3, 5	NUM
shift	0, 3, 5, 1	



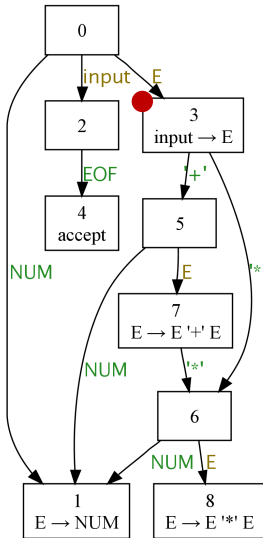
	0	NUM + NUM
shift	0, 1	+ NUM
reduce	0	E + NUM
goto	0, 3	+ NUM
shift	0, 3, 5	NUM
shift	0, 3, 5, 1	
reduce	0, 3, 5	E



	0	NUM + NUM	
shift	0, 1	+ NUM	
reduce	0	$E + \text{NUM}$	
goto	0, 3	+ NUM	
shift	0, 3, 5	NUM	
shift	0, 3, 5, 1		
reduce	0, 3, 5	E	
goto	0, 3, 5, 7		



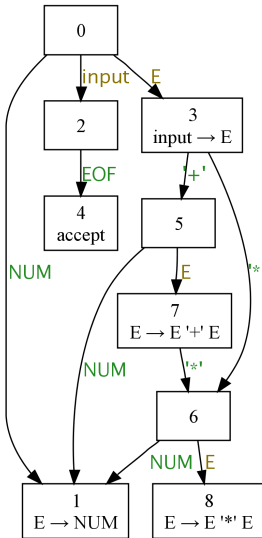
	0	NUM + NUM	
shift	0, 1	+ NUM	
reduce	0	$E + \text{NUM}$	
goto	0, 3	+ NUM	
shift	0, 3, 5	NUM	
shift	0, 3, 5, 1		
reduce	0, 3, 5	E	
goto	0, 3, 5, 7		
reduce	0	E	

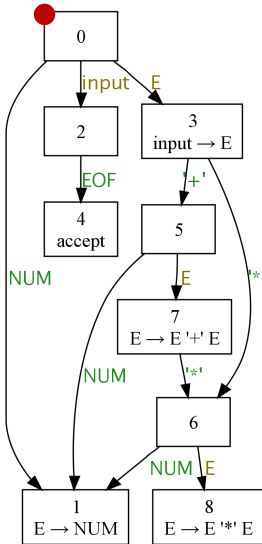


	0	NUM + NUM	
shift	0, 1	+ NUM	
reduce	0	$E + \text{NUM}$	
goto	0, 3	+ NUM	
shift	0, 3, 5	NUM	
shift	0, 3, 5, 1		
reduce	0, 3, 5	E	
goto	0, 3, 5, 7		
reduce	0	E	
goto	0, 3		

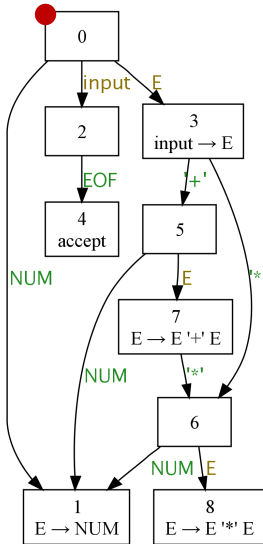
Parse tree pattern parser for yacc

- extended shift/reduce parser
 - input: string of terminals, *non-terminals*, and *tree patterns*
 - stack: plain LR stack
 - new actions for unfolding tree patterns
- enforce structure of tree pattern
 - only checked reductions are allowed
- implementation
 - parser is rewrite system, implemented in Stratego/XT, 55 LOC
 - parsetable obtained by parsing output of Bison

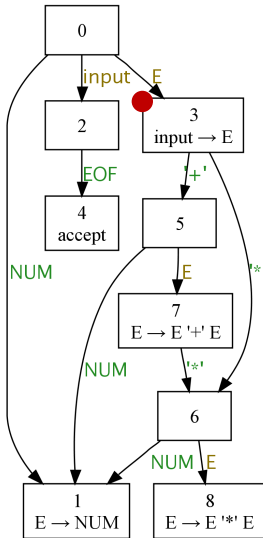




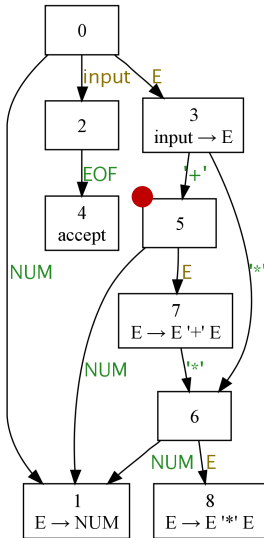
| 0 | $\langle E \rightarrow E + \langle E \rightarrow E * E \rangle \rangle$ |



	0	$\langle E \rightarrow E + \langle E \rightarrow E * E \rangle \rangle$	
unfold	0	$E, +, \langle E \rightarrow E * E \rangle, \mathcal{R}(+)$	

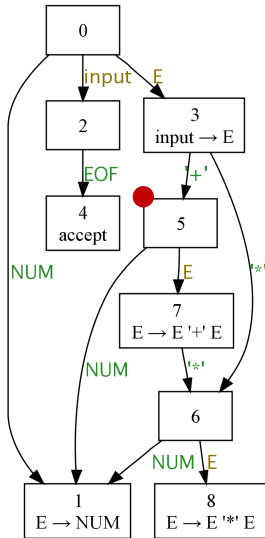


	0	$\langle E \rightarrow E + \langle E \rightarrow E * E \rangle \rangle$	
unfold	0	$E, +, \langle E \rightarrow E * E \rangle, \mathcal{R}(+)$	
goto	0, 3	$+, \langle E \rightarrow E * E \rangle, \mathcal{R}(+)$	

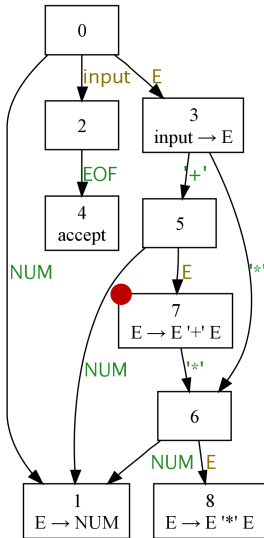


unfold
goto
shift

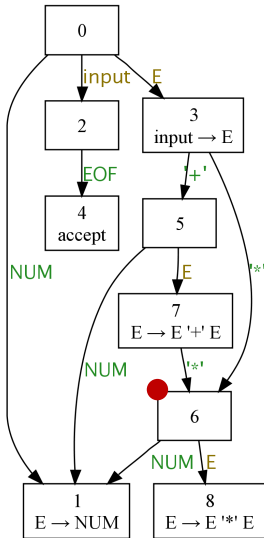
0		$\langle E \rightarrow E + \langle E \rightarrow E * E \rangle \rangle$	
0		$E, +, \langle E \rightarrow E * E \rangle, \mathcal{R}(+)$	
0, 3		$+, \langle E \rightarrow E * E \rangle, \mathcal{R}(+)$	
0, 3, 5		$\langle E \rightarrow E * E \rangle, \mathcal{R}(+)$	



	0	$\langle E \rightarrow E + \langle E \rightarrow E * E \rangle \rangle$	
unfold	0	$E, +, \langle E \rightarrow E * E \rangle, \mathcal{R}(+)$	
goto	0, 3	$+, \langle E \rightarrow E * E \rangle, \mathcal{R}(+)$	
shift	0, 3, 5	$\langle E \rightarrow E * E \rangle, \mathcal{R}(+)$	
unfold	0, 3, 5	$E, *, E, \mathcal{R}(*), \mathcal{R}(+)$	

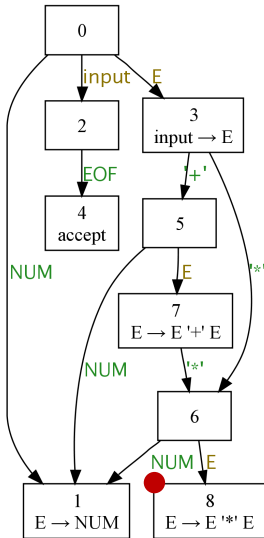


	0		$\langle E \rightarrow E + \langle E \rightarrow E * E \rangle \rangle$	
unfold	0		$E, +, \langle E \rightarrow E * E \rangle, \mathcal{R}(+)$	
goto	0, 3		$+, \langle E \rightarrow E * E \rangle, \mathcal{R}(+)$	
shift	0, 3, 5		$\langle E \rightarrow E * E \rangle, \mathcal{R}(+)$	
unfold	0, 3, 5		$E, *, E, \mathcal{R}(*), \mathcal{R}(+)$	
goto	0, 3, 5, 7		$*, E, \mathcal{R}(*), \mathcal{R}(+)$	



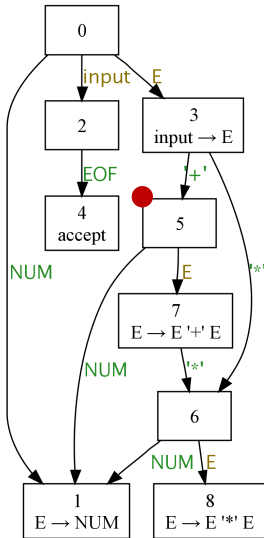
unfold
goto
shift
unfold
goto
shift

0		$\langle E \rightarrow E + \langle E \rightarrow E * E \rangle \rangle$	
0		$E, +, \langle E \rightarrow E * E \rangle, \mathcal{R}(+)$	
0, 3		$+, \langle E \rightarrow E * E \rangle, \mathcal{R}(+)$	
0, 3, 5		$\langle E \rightarrow E * E \rangle, \mathcal{R}(+)$	
0, 3, 5		$E, *, E, \mathcal{R}(*), \mathcal{R}(+)$	
0, 3, 5, 7		$*, E, \mathcal{R}(*), \mathcal{R}(+)$	
0, 3, 5, 7, 6		$E, \mathcal{R}(*), \mathcal{R}(+)$	

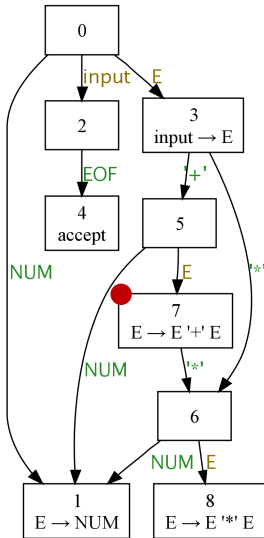


unfold
goto
shift
unfold
goto
shift
goto

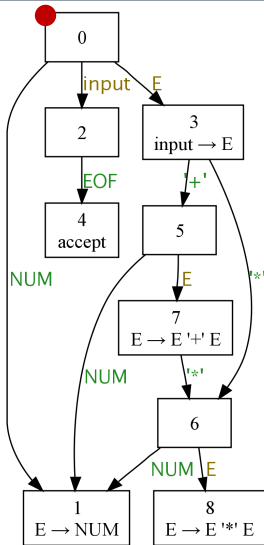
0		$\langle E \rightarrow E + \langle E \rightarrow E * E \rangle \rangle$	
0		$E, +, \langle E \rightarrow E * E \rangle, \mathcal{R}(+)$	
0, 3		$+, \langle E \rightarrow E * E \rangle, \mathcal{R}(+)$	
0, 3, 5		$\langle E \rightarrow E * E \rangle, \mathcal{R}(+)$	
0, 3, 5		$E, *, E, \mathcal{R}(*), \mathcal{R}(+)$	
0, 3, 5, 7		$*, E, \mathcal{R}(*), \mathcal{R}(+)$	
0, 3, 5, 7, 6		$E, \mathcal{R}(*), \mathcal{R}(+)$	
0, 3, 5, 7, 6, 8		$\mathcal{R}(*), \mathcal{R}(+)$	



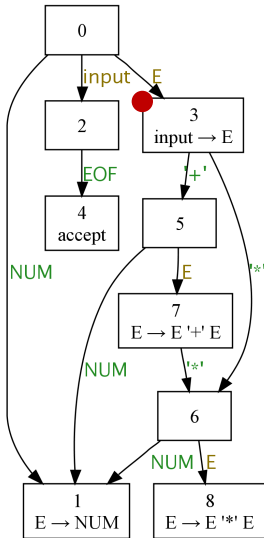
	0		$\langle E \rightarrow E + \langle E \rightarrow E * E \rangle \rangle$	
unfold	0		$E, +, \langle E \rightarrow E * E \rangle, \mathcal{R}(+)$	
goto	0, 3		$+, \langle E \rightarrow E * E \rangle, \mathcal{R}(+)$	
shift	0, 3, 5		$\langle E \rightarrow E * E \rangle, \mathcal{R}(+)$	
unfold	0, 3, 5		$E, *, E, \mathcal{R}(*), \mathcal{R}(+)$	
goto	0, 3, 5, 7		$*, E, \mathcal{R}(*), \mathcal{R}(+)$	
shift	0, 3, 5, 7, 6		$E, \mathcal{R}(*), \mathcal{R}(+)$	
goto	0, 3, 5, 7, 6, 8		$\mathcal{R}(*), \mathcal{R}(+)$	
reduce	0, 3, 5		$E, \mathcal{R}(+)$	



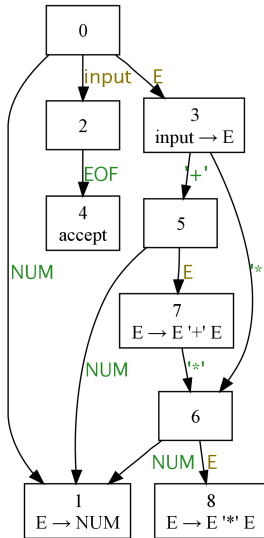
	0		$\langle E \rightarrow E + \langle E \rightarrow E * E \rangle \rangle$	
unfold	0		$E, +, \langle E \rightarrow E * E \rangle, \mathcal{R}(+)$	
goto	0, 3		$+, \langle E \rightarrow E * E \rangle, \mathcal{R}(+)$	
shift	0, 3, 5		$\langle E \rightarrow E * E \rangle, \mathcal{R}(+)$	
unfold	0, 3, 5		$E, *, E, \mathcal{R}(*), \mathcal{R}(+)$	
goto	0, 3, 5, 7		$*, E, \mathcal{R}(*), \mathcal{R}(+)$	
shift	0, 3, 5, 7, 6		$E, \mathcal{R}(*), \mathcal{R}(+)$	
goto	0, 3, 5, 7, 6, 8		$\mathcal{R}(*), \mathcal{R}(+)$	
reduce	0, 3, 5		$E, \mathcal{R}(+)$	
goto	0, 3, 5, 7		$\mathcal{R}(+)$	



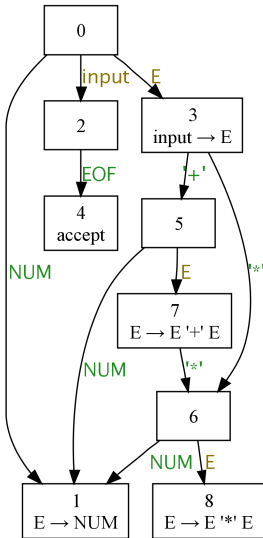
	0		$\langle E \rightarrow E + \langle E \rightarrow E * E \rangle \rangle$	
unfold	0		$E, +, \langle E \rightarrow E * E \rangle, \mathcal{R}(+)$	
goto	0, 3		$+, \langle E \rightarrow E * E \rangle, \mathcal{R}(+)$	
shift	0, 3, 5		$\langle E \rightarrow E * E \rangle, \mathcal{R}(+)$	
unfold	0, 3, 5		$E, *, E, \mathcal{R}(*), \mathcal{R}(+)$	
goto	0, 3, 5, 7		$*, E, \mathcal{R}(*), \mathcal{R}(+)$	
shift	0, 3, 5, 7, 6		$E, \mathcal{R}(*), \mathcal{R}(+)$	
goto	0, 3, 5, 7, 6, 8		$\mathcal{R}(*), \mathcal{R}(+)$	
reduce	0, 3, 5		$E, \mathcal{R}(+)$	
goto	0, 3, 5, 7		$\mathcal{R}(+)$	
reduce	0		E	

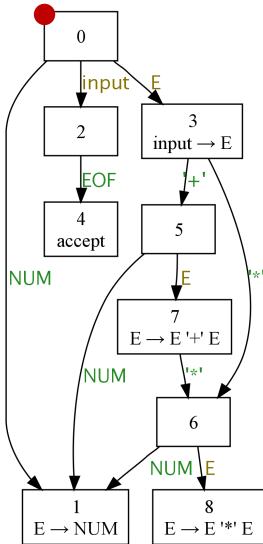


	0		$\langle E \rightarrow E + \langle E \rightarrow E * E \rangle \rangle$	
unfold	0		$E, +, \langle E \rightarrow E * E \rangle, \mathcal{R}(+)$	
goto	0, 3		$+, \langle E \rightarrow E * E \rangle, \mathcal{R}(+)$	
shift	0, 3, 5		$\langle E \rightarrow E * E \rangle, \mathcal{R}(+)$	
unfold	0, 3, 5		$E, *, E, \mathcal{R}(*), \mathcal{R}(+)$	
goto	0, 3, 5, 7		$*, E, \mathcal{R}(*), \mathcal{R}(+)$	
shift	0, 3, 5, 7, 6		$E, \mathcal{R}(*), \mathcal{R}(+)$	
goto	0, 3, 5, 7, 6, 8		$\mathcal{R}(*), \mathcal{R}(+)$	
reduce	0, 3, 5		$E, \mathcal{R}(+)$	
goto	0, 3, 5, 7		$\mathcal{R}(+)$	
reduce	0		E	
goto	0, 3			

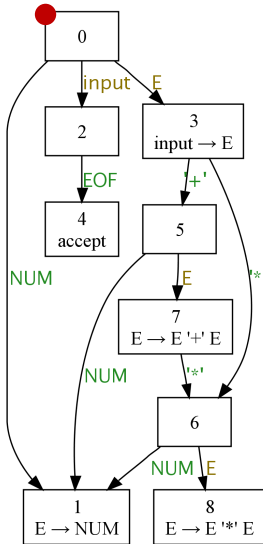


	0		$\langle E \rightarrow E + \langle E \rightarrow E * E \rangle \rangle$	
unfold	0		$E, +, \langle E \rightarrow E * E \rangle, \mathcal{R}(+)$	
goto	0, 3		$+, \langle E \rightarrow E * E \rangle, \mathcal{R}(+)$	
shift	0, 3, 5		$\langle E \rightarrow E * E \rangle, \mathcal{R}(+)$	
unfold	0, 3, 5		$E, *, E, \mathcal{R}(*), \mathcal{R}(+)$	
goto	0, 3, 5, 7		$*, E, \mathcal{R}(*), \mathcal{R}(+)$	
shift	0, 3, 5, 7, 6		$E, \mathcal{R}(*), \mathcal{R}(+)$	
goto	0, 3, 5, 7, 6, 8		$\mathcal{R}(*), \mathcal{R}(+)$	
reduce	0, 3, 5		$E, \mathcal{R}(+)$	
goto	0, 3, 5, 7		$\mathcal{R}(+)$	
reduce	0		E	
goto	0, 3			
accept				

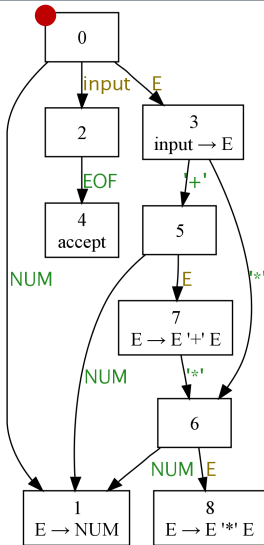




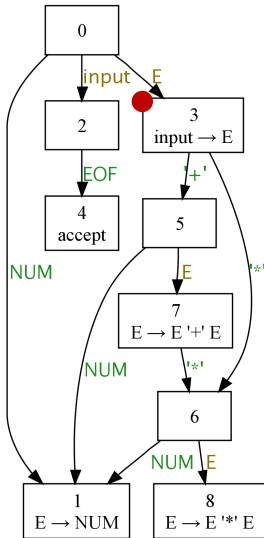
| 0 | $\langle E \rightarrow \langle E \rightarrow E + E \rangle * E \rangle$ |



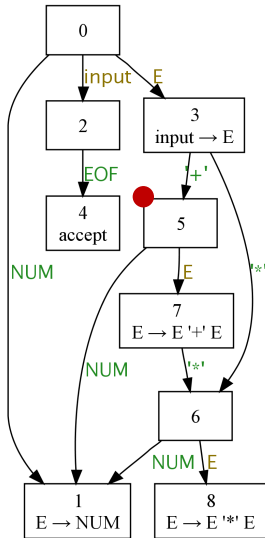
	0	$\langle E \rightarrow \langle E \rightarrow E + E \rangle * E \rangle$	
unfold	0	$\langle E \rightarrow E + E \rangle, *, E, \mathcal{R}(*)$	



	0	$\langle E \rightarrow \langle E \rightarrow E + E \rangle * E \rangle$	
unfold	0	$\langle E \rightarrow E + E \rangle, *, E, \mathcal{R}(*)$	
unfold	0	$E, +, E, \mathcal{R}(+), *, E, \mathcal{R}(*)$	

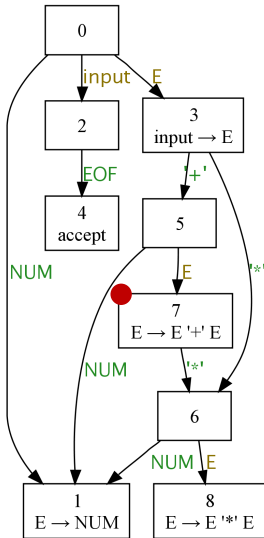


	0	$\langle E \rightarrow \langle E \rightarrow E + E \rangle * E \rangle$	
unfold	0	$\langle E \rightarrow E + E \rangle, *, E, \mathcal{R}(*)$	
unfold	0	$E, +, E, \mathcal{R}(+), *, E, \mathcal{R}(*)$	
goto	0, 3	$+, E, \mathcal{R}(+), *, E, \mathcal{R}(*)$	



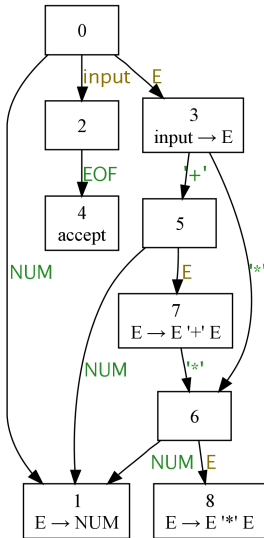
unfold
unfold
goto
shift

0		$\langle E \rightarrow \langle E \rightarrow E + E \rangle * E \rangle$	
0		$\langle E \rightarrow E + E \rangle, *, E, \mathcal{R}(*)$	
0		$E, +, E, \mathcal{R}(+), *, E, \mathcal{R}(*)$	
0, 3		$+, E, \mathcal{R}(+), *, E, \mathcal{R}(*)$	
0, 3, 5		$E, \mathcal{R}(+), *, E, \mathcal{R}(*)$	



unfold
unfold
goto
shift
goto

0		$\langle E \rightarrow \langle E \rightarrow E + E \rangle * E \rangle$	
0		$\langle E \rightarrow E + E \rangle, *, E, \mathcal{R}(*)$	
0		$E, +, E, \mathcal{R}(+), *, E, \mathcal{R}(*)$	
0, 3		$+, E, \mathcal{R}(+), *, E, \mathcal{R}(*)$	
0, 3, 5		$E, \mathcal{R}(+), *, E, \mathcal{R}(*)$	
0, 3, 5, 7		$\mathcal{R}(+), *, E, \mathcal{R}(*)$	



unfold
 unfold
 goto
 shift
 goto
 error

0		$\langle E \rightarrow \langle E \rightarrow E + E \rangle * E \rangle$	
0		$\langle E \rightarrow E + E \rangle, *, E, \mathcal{R}(*)$	
0		$E, +, E, \mathcal{R}(+), *, E, \mathcal{R}(*)$	
0, 3		$+, E, \mathcal{R}(+), *, E, \mathcal{R}(*)$	
0, 3, 5		$E, \mathcal{R}(+), *, E, \mathcal{R}(*)$	
0, 3, 5, 7		$\mathcal{R}(+), *, E, \mathcal{R}(*)$	

Parse tree pattern parser for sdf/sglr

- very similar, but extended generalized-LR parser

How to compare precedence rules of different grammars?

- ① recover precedence rule set from two grammars
- ② optionally, normalize the precedence rules
- ③ compare the productions
- ④ compare the sets of precedence rules

Common normalizations

- rename non-terminals and terminals
- inline alternatives of non-terminals and terminals
- remove extensions

Grammars

- YACC C grammar from GCC
 - mixture of precedence declarations and encoding
- SDF C99 grammar from C-Transformers
 - ambiguous, no precedence declarations
- SDF ANSI C grammar from SDF Library
 - ambiguous, precedence declarations

Several incompatibilities and bugs

$\langle E \rightarrow E ? E : \langle E \rightarrow E = E \rangle \rangle$

forbidden in GCC and SDF Library, allowed in C-Transformers

$\langle E \rightarrow \langle E \rightarrow E ? E : E \rangle = E \rangle \quad \langle E \rightarrow \langle E \rightarrow (\textit{TypeName}) E \rangle = E \rangle$

allowed by GCC and SDF Library, forbidden in C-Transformers

Grammars

- YACC PHP5 grammar from official PHP distribution
 - precedence declarations
- YACC PHP5 grammar from PHC
 - fork of the PHP grammar
- SDF PHP5 grammar from PHP-front
 - precedence declarations and various hacks

PHC versus PHP

- `||`, `OR`, and `XOR` precedence over `&&`, `AND`, and `AND`
 - $\langle E \rightarrow \langle E \rightarrow E \mid E \rangle \&\& E \rangle \in \text{PHP precedence rules}$
 - $\langle E \rightarrow E \mid \langle E \rightarrow E \&\& E \rangle \rangle \in \text{PHC precedence rules}$

PHP-front versus PHP

- many differences, switched to generative approach

SDF priorities generated from precedence rule sets

- one-level tree patterns translate to *non-transitive*, *position-specific* SDF priorities

- $\langle E \rightarrow \langle E \rightarrow E + E \rangle * E \rangle$
 $\langle E \rightarrow E * \langle E \rightarrow E + E \rangle \rangle$

E "*" E -> E <0> .> E "+" E -> E

E "*" E -> E <2> .> E "+" E -> E

- PHP: about 1650 generated priority rules
 - solved all precedence issues in PHP-front
- future work
 - analyse precedence rules to infer compact definition
 - encode precedence rules in productions
 - determine `prefer` from analysing n -level tree patterns

Liberated precedence rules from idiosyncrasies of parser generators

Recovery

recover precedence rules from grammars

- strong support for grammar recovery process

Compatibility

compare precedence rules of grammars

- strong support for maintaining consistency

Migration

migrate a grammar from one grammar formalism to another

- all existing migration tools ignore precedence rules

<http://www.strategoxt.org/GrammarEngineeringTools>

YACC PHP

```
internal_functions_in_yacc: T_INCLUDE expr | ...  
expr_without_variable: variable T_PLUS_EQUAL expr | ...
```

SDF PHP-front

```
Variable "+=" Expr -> Expr  
'include' Expr -> InternalFunction
```

Precedence rule normalization

```
cf(sort("T_PLUS_EQUAL")) -> lit("+=")  
cf(sort("T_INCLUDE"))    -> cilit("include")  
cf(sort("expr"))         -> cf(sort("Expr"))  
cf(sort("internal_functions_in_yacc"))  
    -> cf(sort("InternalFunction"))
```

YACC PHP

```
"!=" | "<>" { return T_IS_NOT_EQUAL; }  
expr_without_variable: expr T_IS_NOT_EQUAL expr | ...
```

SDF PHP-front

```
Expr "!=" Expr  -> Expr  
Expr "<>" Expr  -> Expr
```

Precedence rule normalization

```
inline-not-equal =  
  prec-transform-inline(|  
    cf(sort("T_IS_NOT_EQUAL"))  
    , [ lit("!="), lit("<>") ])
```

SDF C-Transformers

```
UnaryOperator CastExpression -> UnaryExpression  
"&" | "*" | "+" | "-" | "~" | "!" -> UnaryOperator
```

C SDF Library

```
"&" Expression -> Expression    "*" Expression -> Expression  
"+" Expression -> Expression    "-" Expression -> Expression  
"~" Expression -> Expression    "!" Expression -> Expression
```

Precedence rule normalization

```
inline-unary-operator =  
  prec-transform-inline(|  
    cf(sort("UnaryOperator"))  
    , [ lit("&"), lit("*"), lit("+"), lit("-"), lit("~")  
      , lit("!") ] ))
```

```
normalize-c-yacc =  
  remove-all(extensions)  
  ; inline-unary-operator  
  ; inline-assignment-operator  
  ; inline-eqcompare-operator  
  ; inline-arithcompare-operator  
  ; topdown(try(NormalizeYacc))  
  
extensions =  
  has-simple-extension(?cf(sort("CHOOSE_EXPR")))  
  <+ has-simple-extension(?cf(sort("VA_ARG")))  
  <+ has-simple-extension(?cf(sort("IMAGPART")))  
  <+ has-simple-extension(?cf(sort("REALPART")))
```

Textbooks

```
%right '='  
%left '|' '  
%left '&' '  
%left '<' '>' '  
%left '-' '+' '  
%left '*' '/' '%' '
```

C

```
%right ASSIGN '='  
%right '?' ':'  
%left OROR  
%left ANDAND  
%left '|'  
%left '^'  
%left '&  
%left EQCOMPARE  
%left ARITHCOMPARE  
%left LSHIFT RSHIFT  
%left '+' '-'  
%left '*' '/' '%'  
%right UNARY PLUSPLUS MINUSMINUS  
%left HYPERUNARY  
%left POINTSAT '.' '(' '['
```

PHP

```
%left T_INCLUDE T_INCLUDE_ONCE T_EVAL T_REQUIRE T_REQUIRE_ONCE
%left ','
%left T_LOGICAL_OR
%left T_LOGICAL_XOR
%left T_LOGICAL_AND
%right T_PRINT
%left '=' T_PLUS_EQUAL T_MINUS_EQUAL T_MUL_EQUAL T_DIV_EQUAL ...
%left '?' ':'
%left T_BOOLEAN_OR
%left T_BOOLEAN_AND
%left '|'
%left '^'
%left '&'
%nonassoc T_IS_EQUAL T_IS_NOT_EQUAL T_IS_IDENTICAL T_IS_NOT_IDENTICAL
%nonassoc '<' T_IS_SMALLER_OR_EQUAL '>' T_IS_GREATER_OR_EQUAL
%left T_SL T_SR
%left '+' '-' '.'
%left '*' '/' '%'
%right '!'
%nonassoc T_INSTANCEOF
%right '~' T_INC T_DEC T_INT_CAST T_DOUBLE_CAST T_STRING_CAST ...
%right '['
%nonassoc T_NEW T_CLONE
```

Parse tree pattern parser for sdf/sglr

- extended generalized-LR parser
 - multiple actions possible in some configuration
 - GLR configuration forked
- SGLR uses production gotos (not symbol gotos)
 - non-terminal gotos translated to forks
- supports ambiguous grammars
- implementation
 - operates on plain sglr parse table
 - parser implemented in Stratego/XT
 - very naive, but performance not an issue

Parse trees

- NUM
- $\langle E \rightarrow \text{NUM} \rangle$
- $\langle E \rightarrow \langle E \rightarrow \text{NUM} \rangle + \langle E \rightarrow \text{NUM} \rangle \rangle$
- $\langle E \rightarrow \langle E \rightarrow \langle T \rightarrow \langle F \rightarrow \text{NUM} \rangle \rangle \rangle + \langle T \rightarrow \langle F \rightarrow \text{NUM} \rangle \rangle \rangle$

1

Parse tree patterns

- extension with non-terminals (derivation of sentential form)
- $\langle E \rightarrow \langle E \rightarrow E * E \rangle + E \rangle$
- $\langle E \rightarrow \langle E \rightarrow E + E \rangle * E \rangle$
- $\langle E \rightarrow \langle E \rightarrow \langle T \rightarrow F \rangle \rangle + T \rangle$

Parse tree patterns with injection chains

- concise notation for injection chains
- $\langle E \rightarrow \langle E \sim F \rightarrow \text{NUM} \rangle + \langle T \sim F \rightarrow \text{NUM} \rangle \rangle$

Parse trees

- NUM
- $\langle E \rightarrow \text{NUM} \rangle$
- $\langle E \rightarrow \langle E \rightarrow \text{NUM} \rangle + \langle E \rightarrow \text{NUM} \rangle \rangle$
- $\langle E \rightarrow \langle E \rightarrow \langle T \rightarrow \langle F \rightarrow \text{NUM} \rangle \rangle \rangle + \langle T \rightarrow \langle F \rightarrow \text{NUM} \rangle \rangle \rangle$

1

Parse tree patterns

- extension with non-terminals (derivation of sentential form)
- $\langle E \rightarrow \langle E \rightarrow E * E \rangle + E \rangle$
- $\langle E \rightarrow \langle E \rightarrow E + E \rangle * E \rangle$
- $\langle E \rightarrow \langle E \rightarrow \langle T \rightarrow F \rangle \rangle + T \rangle$

Parse tree patterns with injection chains

- concise notation for injection chains
- $\langle E \rightarrow \langle E \sim F \rightarrow \text{NUM} \rangle + \langle T \sim F \rightarrow \text{NUM} \rangle \rangle$

Parse trees

- NUM
- $\langle E \rightarrow \text{NUM} \rangle$
- $\langle E \rightarrow \langle E \rightarrow \text{NUM} \rangle + \langle E \rightarrow \text{NUM} \rangle \rangle$
- $\langle E \rightarrow \langle E \rightarrow \langle T \rightarrow \langle F \rightarrow \text{NUM} \rangle \rangle \rangle + \langle T \rightarrow \langle F \rightarrow \text{NUM} \rangle \rangle \rangle$

1 + 2

Parse tree patterns

- extension with non-terminals (derivation of sentential form)
- $\langle E \rightarrow \langle E \rightarrow E * E \rangle + E \rangle$
- $\langle E \rightarrow \langle E \rightarrow E + E \rangle * E \rangle$
- $\langle E \rightarrow \langle E \rightarrow \langle T \rightarrow F \rangle \rangle + T \rangle$

Parse tree patterns with injection chains

- concise notation for injection chains
- $\langle E \rightarrow \langle E \sim F \rightarrow \text{NUM} \rangle + \langle T \sim F \rightarrow \text{NUM} \rangle \rangle$

Parse trees

- NUM
- $\langle E \rightarrow \text{NUM} \rangle$
- $\langle E \rightarrow \langle E \rightarrow \text{NUM} \rangle + \langle E \rightarrow \text{NUM} \rangle \rangle$
- $\langle E \rightarrow \langle E \rightarrow \langle T \rightarrow \langle F \rightarrow \text{NUM} \rangle \rangle \rangle + \langle T \rightarrow \langle F \rightarrow \text{NUM} \rangle \rangle \rangle$

1 + 2

Parse tree patterns

- extension with non-terminals (derivation of sentential form)
- $\langle E \rightarrow \langle E \rightarrow E * E \rangle + E \rangle$
- $\langle E \rightarrow \langle E \rightarrow E + E \rangle * E \rangle$
- $\langle E \rightarrow \langle E \rightarrow \langle T \rightarrow F \rangle \rangle + T \rangle$

Parse tree patterns with injection chains

- concise notation for injection chains
- $\langle E \rightarrow \langle E \sim F \rightarrow \text{NUM} \rangle + \langle T \sim F \rightarrow \text{NUM} \rangle \rangle$

Parse trees

- NUM
- $\langle E \rightarrow \text{NUM} \rangle$
- $\langle E \rightarrow \langle E \rightarrow \text{NUM} \rangle + \langle E \rightarrow \text{NUM} \rangle \rangle$
- $\langle E \rightarrow \langle E \rightarrow \langle T \rightarrow \langle F \rightarrow \text{NUM} \rangle \rangle \rangle + \langle T \rightarrow \langle F \rightarrow \text{NUM} \rangle \rangle \rangle$

Parse tree patterns

- extension with non-terminals (derivation of sentential form)
- $\langle E \rightarrow \langle E \rightarrow E * E \rangle + E \rangle$
- $\langle E \rightarrow \langle E \rightarrow E + E \rangle * E \rangle$
- $\langle E \rightarrow \langle E \rightarrow \langle T \rightarrow F \rangle \rangle + T \rangle$

Parse tree patterns with injection chains

- concise notation for injection chains
- $\langle E \rightarrow \langle E \sim F \rightarrow \text{NUM} \rangle + \langle T \sim F \rightarrow \text{NUM} \rangle \rangle$

Parse trees

- NUM
- $\langle E \rightarrow \text{NUM} \rangle$
- $\langle E \rightarrow \langle E \rightarrow \text{NUM} \rangle + \langle E \rightarrow \text{NUM} \rangle \rangle$
- $\langle E \rightarrow \langle E \rightarrow \langle T \rightarrow \langle F \rightarrow \text{NUM} \rangle \rangle \rangle + \langle T \rightarrow \langle F \rightarrow \text{NUM} \rangle \rangle \rangle$

Parse tree patterns

- extension with non-terminals (derivation of sentential form)
- $\langle E \rightarrow \langle E \rightarrow E * E \rangle + E \rangle$
- $\langle E \rightarrow \langle E \rightarrow E + E \rangle * E \rangle$
- $\langle E \rightarrow \langle E \rightarrow \langle T \rightarrow F \rangle \rangle + T \rangle$

1 * 2 + 3

Parse tree patterns with injection chains

- concise notation for injection chains
- $\langle E \rightarrow \langle E \sim F \rightarrow \text{NUM} \rangle + \langle T \sim F \rightarrow \text{NUM} \rangle \rangle$

Parse trees

- NUM
- $\langle E \rightarrow \text{NUM} \rangle$
- $\langle E \rightarrow \langle E \rightarrow \text{NUM} \rangle + \langle E \rightarrow \text{NUM} \rangle \rangle$
- $\langle E \rightarrow \langle E \rightarrow \langle T \rightarrow \langle F \rightarrow \text{NUM} \rangle \rangle \rangle + \langle T \rightarrow \langle F \rightarrow \text{NUM} \rangle \rangle \rangle$

Parse tree patterns

- extension with non-terminals (derivation of sentential form)
- $\langle E \rightarrow \langle E \rightarrow E * E \rangle + E \rangle$
- $\langle E \rightarrow \langle E \rightarrow E + E \rangle * E \rangle$
- $\langle E \rightarrow \langle E \rightarrow \langle T \rightarrow F \rangle \rangle + T \rangle$

1 * 2 + (3 * 4)

Parse tree patterns with injection chains

- concise notation for injection chains
- $\langle E \rightarrow \langle E \sim F \rightarrow \text{NUM} \rangle + \langle T \sim F \rightarrow \text{NUM} \rangle \rangle$

Parse trees

- NUM
- $\langle E \rightarrow \text{NUM} \rangle$
- $\langle E \rightarrow \langle E \rightarrow \text{NUM} \rangle + \langle E \rightarrow \text{NUM} \rangle \rangle$
- $\langle E \rightarrow \langle E \rightarrow \langle T \rightarrow \langle F \rightarrow \text{NUM} \rangle \rangle \rangle + \langle T \rightarrow \langle F \rightarrow \text{NUM} \rangle \rangle \rangle$

Parse tree patterns

- extension with non-terminals (derivation of sentential form)
- $\langle E \rightarrow \langle E \rightarrow E * E \rangle + E \rangle$
- $\langle E \rightarrow \langle E \rightarrow E + E \rangle * E \rangle$
- $\langle E \rightarrow \langle E \rightarrow \langle T \rightarrow F \rangle \rangle + T \rangle$

1 * (2 + 3)

Parse tree patterns with injection chains

- concise notation for injection chains
- $\langle E \rightarrow \langle E \sim F \rightarrow \text{NUM} \rangle + \langle T \sim F \rightarrow \text{NUM} \rangle \rangle$

Parse trees

- NUM
- $\langle E \rightarrow \text{NUM} \rangle$
- $\langle E \rightarrow \langle E \rightarrow \text{NUM} \rangle + \langle E \rightarrow \text{NUM} \rangle \rangle$
- $\langle E \rightarrow \langle E \rightarrow \langle T \rightarrow \langle F \rightarrow \text{NUM} \rangle \rangle \rangle + \langle T \rightarrow \langle F \rightarrow \text{NUM} \rangle \rangle \rangle$

Parse tree patterns

- extension with non-terminals (derivation of sentential form)
- $\langle E \rightarrow \langle E \rightarrow E * E \rangle + E \rangle$
- $\langle E \rightarrow \langle E \rightarrow E + E \rangle * E \rangle$
- $\langle E \rightarrow \langle E \rightarrow \langle T \rightarrow F \rangle \rangle + T \rangle$

(1 * 2) + 3

Parse tree patterns with injection chains

- concise notation for injection chains
- $\langle E \rightarrow \langle E \sim F \rightarrow \text{NUM} \rangle + \langle T \sim F \rightarrow \text{NUM} \rangle \rangle$

Parse trees

- NUM
- $\langle E \rightarrow \text{NUM} \rangle$
- $\langle E \rightarrow \langle E \rightarrow \text{NUM} \rangle + \langle E \rightarrow \text{NUM} \rangle \rangle$
- $\langle E \rightarrow \langle E \rightarrow \langle T \rightarrow \langle F \rightarrow \text{NUM} \rangle \rangle \rangle + \langle T \rightarrow \langle F \rightarrow \text{NUM} \rangle \rangle \rangle$

Parse tree patterns

- extension with non-terminals (derivation of sentential form)
- $\langle E \rightarrow \langle E \rightarrow E * E \rangle + E \rangle$
- $\langle E \rightarrow \langle E \rightarrow E + E \rangle * E \rangle$
- $\langle E \rightarrow \langle E \rightarrow \langle T \rightarrow F \rangle \rangle + T \rangle$

1 + 2 * 3

Parse tree patterns with injection chains

- concise notation for injection chains
- $\langle E \rightarrow \langle E \sim F \rightarrow \text{NUM} \rangle + \langle T \sim F \rightarrow \text{NUM} \rangle \rangle$

Parse trees

- NUM
- $\langle E \rightarrow \text{NUM} \rangle$
- $\langle E \rightarrow \langle E \rightarrow \text{NUM} \rangle + \langle E \rightarrow \text{NUM} \rangle \rangle$
- $\langle E \rightarrow \langle E \rightarrow \langle T \rightarrow \langle F \rightarrow \text{NUM} \rangle \rangle \rangle + \langle T \rightarrow \langle F \rightarrow \text{NUM} \rangle \rangle \rangle$

Parse tree patterns

- extension with non-terminals (derivation of sentential form)
- $\langle E \rightarrow \langle E \rightarrow E * E \rangle + E \rangle$
- $\langle E \rightarrow \langle E \rightarrow E + E \rangle * E \rangle$
- $\langle E \rightarrow \langle E \rightarrow \langle T \rightarrow F \rangle \rangle + T \rangle$

(1 + 2) * 3

Parse tree patterns with injection chains

- concise notation for injection chains
- $\langle E \rightarrow \langle E \sim F \rightarrow \text{NUM} \rangle + \langle T \sim F \rightarrow \text{NUM} \rangle \rangle$

Parse trees

- NUM
- $\langle E \rightarrow \text{NUM} \rangle$
- $\langle E \rightarrow \langle E \rightarrow \text{NUM} \rangle + \langle E \rightarrow \text{NUM} \rangle \rangle$
- $\langle E \rightarrow \langle E \rightarrow \langle T \rightarrow \langle F \rightarrow \text{NUM} \rangle \rangle \rangle + \langle T \rightarrow \langle F \rightarrow \text{NUM} \rangle \rangle \rangle$

Parse tree patterns

- extension with non-terminals (derivation of sentential form)
- $\langle E \rightarrow \langle E \rightarrow E * E \rangle + E \rangle$
- $\langle E \rightarrow \langle E \rightarrow E + E \rangle * E \rangle$
- $\langle E \rightarrow \langle E \rightarrow \langle T \rightarrow F \rangle \rangle + T \rangle$

1 + 2

Parse tree patterns with injection chains

- concise notation for injection chains
- $\langle E \rightarrow \langle E \sim F \rightarrow \text{NUM} \rangle + \langle T \sim F \rightarrow \text{NUM} \rangle \rangle$

Parse trees

- NUM
- $\langle E \rightarrow \text{NUM} \rangle$
- $\langle E \rightarrow \langle E \rightarrow \text{NUM} \rangle + \langle E \rightarrow \text{NUM} \rangle \rangle$
- $\langle E \rightarrow \langle E \rightarrow \langle T \rightarrow \langle F \rightarrow \text{NUM} \rangle \rangle \rangle + \langle T \rightarrow \langle F \rightarrow \text{NUM} \rangle \rangle \rangle$

Parse tree patterns

- extension with non-terminals (derivation of sentential form)
- $\langle E \rightarrow \langle E \rightarrow E * E \rangle + E \rangle$
- $\langle E \rightarrow \langle E \rightarrow E + E \rangle * E \rangle$
- $\langle E \rightarrow \langle E \rightarrow \langle T \rightarrow F \rangle \rangle + T \rangle$

1 + (2 * 3)

Parse tree patterns with injection chains

- concise notation for injection chains
- $\langle E \rightarrow \langle E \sim F \rightarrow \text{NUM} \rangle + \langle T \sim F \rightarrow \text{NUM} \rangle \rangle$

Parse trees

- NUM
- $\langle E \rightarrow \text{NUM} \rangle$
- $\langle E \rightarrow \langle E \rightarrow \text{NUM} \rangle + \langle E \rightarrow \text{NUM} \rangle \rangle$
- $\langle E \rightarrow \langle E \rightarrow \langle T \rightarrow \langle F \rightarrow \text{NUM} \rangle \rangle \rangle + \langle T \rightarrow \langle F \rightarrow \text{NUM} \rangle \rangle \rangle$

Parse tree patterns

- extension with non-terminals (derivation of sentential form)
- $\langle E \rightarrow \langle E \rightarrow E * E \rangle + E \rangle$
- $\langle E \rightarrow \langle E \rightarrow E + E \rangle * E \rangle$
- $\langle E \rightarrow \langle E \rightarrow \langle T \rightarrow F \rangle \rangle + T \rangle$

2 * 3 + 1

Parse tree patterns with injection chains

- concise notation for injection chains
- $\langle E \rightarrow \langle E \sim F \rightarrow \text{NUM} \rangle + \langle T \sim F \rightarrow \text{NUM} \rangle \rangle$

Parse trees

- NUM
- $\langle E \rightarrow \text{NUM} \rangle$
- $\langle E \rightarrow \langle E \rightarrow \text{NUM} \rangle + \langle E \rightarrow \text{NUM} \rangle \rangle$
- $\langle E \rightarrow \langle E \rightarrow \langle T \rightarrow \langle F \rightarrow \text{NUM} \rangle \rangle \rangle + \langle T \rightarrow \langle F \rightarrow \text{NUM} \rangle \rangle \rangle$

Parse tree patterns

- extension with non-terminals (derivation of sentential form)
- $\langle E \rightarrow \langle E \rightarrow E * E \rangle + E \rangle$
- $\langle E \rightarrow \langle E \rightarrow E + E \rangle * E \rangle$
- $\langle E \rightarrow \langle E \rightarrow \langle T \rightarrow F \rangle \rangle + T \rangle$

Parse tree patterns with injection chains

- concise notation for injection chains
- $\langle E \rightarrow \langle E \sim F \rightarrow \text{NUM} \rangle + \langle T \sim F \rightarrow \text{NUM} \rangle \rangle$

Parse trees

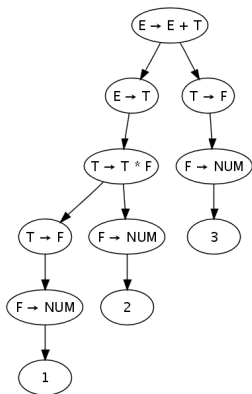
- NUM
- $\langle E \rightarrow \text{NUM} \rangle$
- $\langle E \rightarrow \langle E \rightarrow \text{NUM} \rangle + \langle E \rightarrow \text{NUM} \rangle \rangle$
- $\langle E \rightarrow \langle E \rightarrow \langle T \rightarrow \langle F \rightarrow \text{NUM} \rangle \rangle \rangle + \langle T \rightarrow \langle F \rightarrow \text{NUM} \rangle \rangle \rangle$

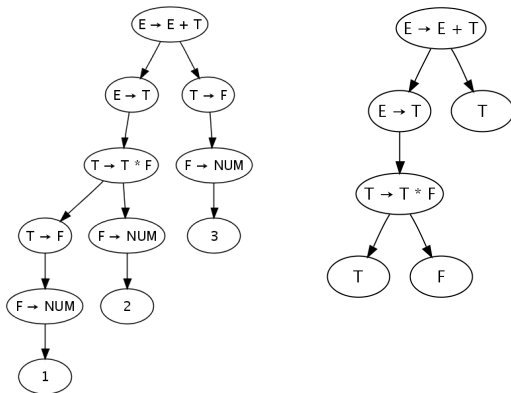
Parse tree patterns

- extension with non-terminals (derivation of sentential form)
- $\langle E \rightarrow \langle E \rightarrow E * E \rangle + E \rangle$
- $\langle E \rightarrow \langle E \rightarrow E + E \rangle * E \rangle$
- $\langle E \rightarrow \langle E \rightarrow \langle T \rightarrow F \rangle \rangle + T \rangle$

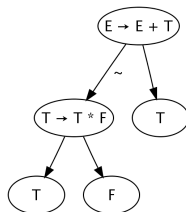
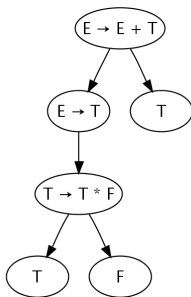
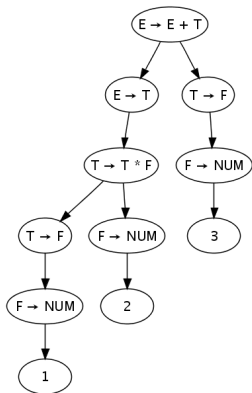
Parse tree patterns with injection chains

- concise notation for injection chains
- $\langle E \rightarrow \langle E \sim F \rightarrow \text{NUM} \rangle + \langle T \sim F \rightarrow \text{NUM} \rangle \rangle$





$$\langle E \rightarrow \langle E \rightarrow \langle T \rightarrow T * \rangle F \rangle + T \rangle$$



$$\langle E \rightarrow \langle E \rightarrow \langle T \rightarrow T * \rangle F \rangle + T \rangle$$

$$\langle E \rightarrow \langle E \sim T \rightarrow T * F \rangle + T \rangle$$

%token NUM

E: E '+' T

| T

T: T '*' F

| F

F: NUM

%token NUM

E: E '+' T

| T

T: T '*' F

| F

F: NUM

$\langle T \rightarrow T * \langle F \sim T \rightarrow T * F \rangle \rangle$

$\langle E \rightarrow E + \langle T \sim E \rightarrow E + T \rangle \rangle$

$\langle E \rightarrow T * \langle F \sim E \rightarrow E + T \rangle \rangle$

$\langle E \rightarrow E + \langle T \rightarrow T * F \rangle \rangle$

$\langle T \rightarrow T * \langle F \rightarrow \text{NUM} \rangle \rangle$

$\langle E \rightarrow E + \langle T \sim F \rightarrow \text{NUM} \rangle \rangle$

$\langle E \rightarrow \langle T \rightarrow T * F \rangle * F \rangle$

$\langle E \rightarrow \langle E \rightarrow E + T \rangle + T \rangle$

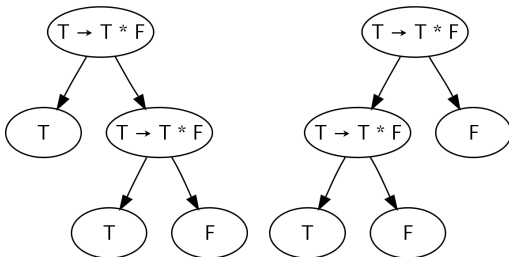
$\langle E \rightarrow \langle T \sim E \rightarrow E + T \rangle * F \rangle$

$\langle E \rightarrow \langle E \sim T \rightarrow T * F \rangle + T \rangle$

$\langle T \rightarrow \langle T \sim F \rightarrow \text{NUM} \rangle * F \rangle$

$\langle E \rightarrow \langle E \sim F \rightarrow \text{NUM} \rangle + T \rangle$

F: NUM



- $\langle E \rightarrow \langle T \rightarrow T * F \rangle * F \rangle$
 $\langle E \rightarrow \langle E \rightarrow E + T \rangle + T \rangle$
 $\langle E \rightarrow \langle T \sim E \rightarrow E + T \rangle * F \rangle$
 $\langle E \rightarrow \langle E \sim T \rightarrow T * F \rangle + T \rangle$
 $\langle T \rightarrow \langle T \sim F \rightarrow \text{NUM} \rangle * F \rangle$
 $\langle E \rightarrow \langle E \sim F \rightarrow \text{NUM} \rangle + T \rangle$

%token NUM

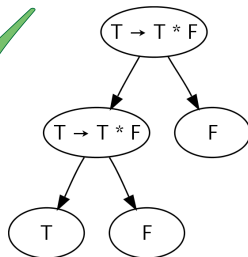
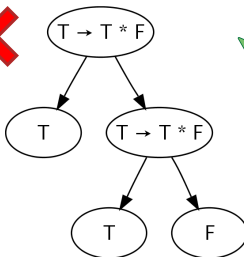
E: E '+' T

| T

T: T '*' F

| F

F: NUM



● $\langle T \rightarrow T * \langle F \sim T \rightarrow T * F \rangle \rangle$
 $\langle E \rightarrow E + \langle T \sim E \rightarrow E + T \rangle \rangle$
 $\langle E \rightarrow T * \langle F \sim E \rightarrow E + T \rangle \rangle$
 $\langle E \rightarrow E + \langle T \rightarrow T * F \rangle \rangle$
 $\langle T \rightarrow T * \langle F \rightarrow \text{NUM} \rangle \rangle$
 $\langle E \rightarrow E + \langle T \sim F \rightarrow \text{NUM} \rangle \rangle$

● $\langle E \rightarrow \langle T \rightarrow T * F \rangle * F \rangle$
 $\langle E \rightarrow \langle E \rightarrow E + T \rangle + T \rangle$
 $\langle E \rightarrow \langle T \sim E \rightarrow E + T \rangle * F \rangle$
 $\langle E \rightarrow \langle E \sim T \rightarrow T * F \rangle + T \rangle$
 $\langle T \rightarrow \langle T \sim F \rightarrow \text{NUM} \rangle * F \rangle$
 $\langle E \rightarrow \langle E \sim F \rightarrow \text{NUM} \rangle + T \rangle$

%token NUM

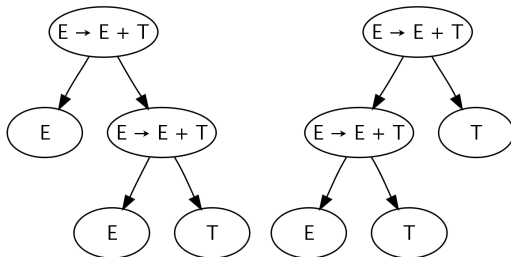
E: E '+' T

| T

T: T '*' F

| F

F: NUM



- | | |
|---|---|
| ● $\langle T \rightarrow T * \langle F \sim T \rightarrow T * F \rangle \rangle$ | ● $\langle E \rightarrow \langle T \rightarrow T * F \rangle * F \rangle$ |
| ● $\langle E \rightarrow E + \langle T \sim E \rightarrow E + T \rangle \rangle$ | ● $\langle E \rightarrow \langle E \rightarrow E + T \rangle + T \rangle$ |
| $\langle E \rightarrow T * \langle F \sim E \rightarrow E + T \rangle \rangle$ | $\langle E \rightarrow \langle T \sim E \rightarrow E + T \rangle * F \rangle$ |
| $\langle E \rightarrow E + \langle T \rightarrow T * F \rangle \rangle$ | $\langle E \rightarrow \langle E \sim T \rightarrow T * F \rangle + T \rangle$ |
| $\langle T \rightarrow T * \langle F \rightarrow \text{NUM} \rangle \rangle$ | $\langle T \rightarrow \langle T \sim F \rightarrow \text{NUM} \rangle * F \rangle$ |
| $\langle E \rightarrow E + \langle T \sim F \rightarrow \text{NUM} \rangle \rangle$ | $\langle E \rightarrow \langle E \sim F \rightarrow \text{NUM} \rangle + T \rangle$ |

%token NUM

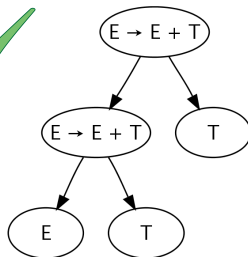
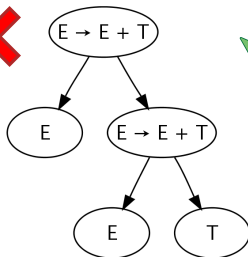
E: E '+' T

| T

T: T '*' F

| F

F: NUM



● $\langle T \rightarrow T * \langle F \sim T \rightarrow T * F \rangle \rangle$

● $\langle E \rightarrow E + \langle T \sim E \rightarrow E + T \rangle \rangle$

$\langle E \rightarrow T * \langle F \sim E \rightarrow E + T \rangle \rangle$

$\langle E \rightarrow E + \langle T \rightarrow T * F \rangle \rangle$

$\langle T \rightarrow T * \langle F \rightarrow \text{NUM} \rangle \rangle$

$\langle E \rightarrow E + \langle T \sim F \rightarrow \text{NUM} \rangle \rangle$

● $\langle E \rightarrow \langle T \rightarrow T * F \rangle * F \rangle$

● $\langle E \rightarrow \langle E \rightarrow E + T \rangle + T \rangle$

$\langle E \rightarrow \langle T \sim E \rightarrow E + T \rangle * F \rangle$

$\langle E \rightarrow \langle E \sim T \rightarrow T * F \rangle + T \rangle$

$\langle T \rightarrow \langle T \sim F \rightarrow \text{NUM} \rangle * F \rangle$

$\langle E \rightarrow \langle E \sim F \rightarrow \text{NUM} \rangle + T \rangle$

%token NUM

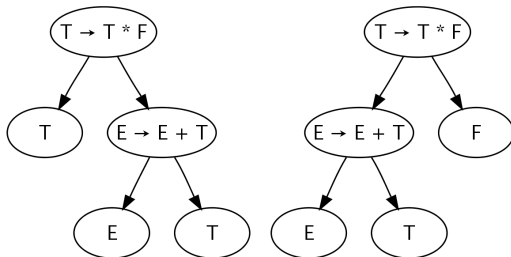
E: E '+' T

| T

T: T '*' F

| F

F: NUM



● $\langle T \rightarrow T * \langle F \sim T \rightarrow T * F \rangle \rangle$

● $\langle E \rightarrow E + \langle T \sim E \rightarrow E + T \rangle \rangle$

● $\langle E \rightarrow T * \langle F \sim E \rightarrow E + T \rangle \rangle$

$\langle E \rightarrow E + \langle T \rightarrow T * F \rangle \rangle$

$\langle T \rightarrow T * \langle F \rightarrow \text{NUM} \rangle \rangle$

$\langle E \rightarrow E + \langle T \sim F \rightarrow \text{NUM} \rangle \rangle$

● $\langle E \rightarrow \langle T \rightarrow T * F \rangle * F \rangle$

● $\langle E \rightarrow \langle E \rightarrow E + T \rangle + T \rangle$

● $\langle E \rightarrow \langle T \sim E \rightarrow E + T \rangle * F \rangle$

$\langle E \rightarrow \langle E \sim T \rightarrow T * F \rangle + T \rangle$

$\langle T \rightarrow \langle T \sim F \rightarrow \text{NUM} \rangle * F \rangle$

$\langle E \rightarrow \langle E \sim F \rightarrow \text{NUM} \rangle + T \rangle$

%token NUM

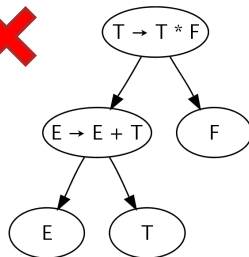
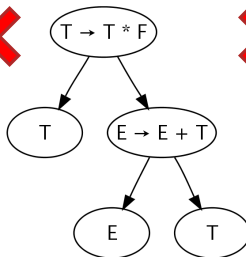
E: E '+' T

| T

T: T '*' F

| F

F: NUM



● $\langle T \rightarrow T * \langle F \sim T \rightarrow T * F \rangle \rangle$

● $\langle E \rightarrow E + \langle T \sim E \rightarrow E + T \rangle \rangle$

● $\langle E \rightarrow T * \langle F \sim E \rightarrow E + T \rangle \rangle$

$\langle E \rightarrow E + \langle T \rightarrow T * F \rangle \rangle$

$\langle T \rightarrow T * \langle F \rightarrow \text{NUM} \rangle \rangle$

$\langle E \rightarrow E + \langle T \sim F \rightarrow \text{NUM} \rangle \rangle$

● $\langle E \rightarrow \langle T \rightarrow T * F \rangle * F \rangle$

● $\langle E \rightarrow \langle E \rightarrow E + T \rangle + T \rangle$

● $\langle E \rightarrow \langle T \sim E \rightarrow E + T \rangle * F \rangle$

$\langle E \rightarrow \langle E \sim T \rightarrow T * F \rangle + T \rangle$

$\langle T \rightarrow \langle T \sim F \rightarrow \text{NUM} \rangle * F \rangle$

$\langle E \rightarrow \langle E \sim F \rightarrow \text{NUM} \rangle + T \rangle$

%token NUM

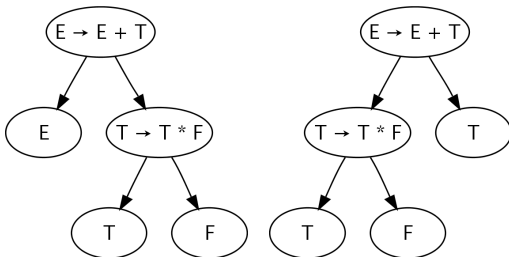
E: E '+' T

| T

T: T '*' F

| F

F: NUM



● $\langle T \rightarrow T * \langle F \sim T \rightarrow T * F \rangle \rangle$

● $\langle E \rightarrow E + \langle T \sim E \rightarrow E + T \rangle \rangle$

● $\langle E \rightarrow T * \langle F \sim E \rightarrow E + T \rangle \rangle$

● $\langle E \rightarrow E + \langle T \rightarrow T * F \rangle \rangle$

$\langle T \rightarrow T * \langle F \rightarrow \text{NUM} \rangle \rangle$

$\langle E \rightarrow E + \langle T \sim F \rightarrow \text{NUM} \rangle \rangle$

● $\langle E \rightarrow \langle T \rightarrow T * F \rangle * F \rangle$

● $\langle E \rightarrow \langle E \rightarrow E + T \rangle + T \rangle$

● $\langle E \rightarrow \langle T \sim E \rightarrow E + T \rangle * F \rangle$

● $\langle E \rightarrow \langle E \sim T \rightarrow T * F \rangle + T \rangle$

$\langle T \rightarrow \langle T \sim F \rightarrow \text{NUM} \rangle * F \rangle$

$\langle E \rightarrow \langle E \sim F \rightarrow \text{NUM} \rangle + T \rangle$

%token NUM

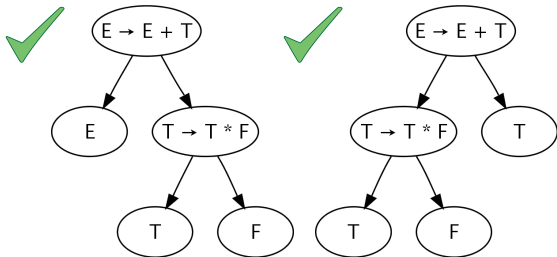
E: E '+' T

| T

T: T '*' F

| F

F: NUM



● $\langle T \rightarrow T * \langle F \sim T \rightarrow T * F \rangle \rangle$

● $\langle E \rightarrow E + \langle T \sim E \rightarrow E + T \rangle \rangle$

● $\langle E \rightarrow T * \langle F \sim E \rightarrow E + T \rangle \rangle$

● $\langle E \rightarrow E + \langle T \rightarrow T * F \rangle \rangle$

$\langle T \rightarrow T * \langle F \rightarrow \text{NUM} \rangle \rangle$

$\langle E \rightarrow E + \langle T \sim F \rightarrow \text{NUM} \rangle \rangle$

● $\langle E \rightarrow \langle T \rightarrow T * F \rangle * F \rangle$

● $\langle E \rightarrow \langle E \rightarrow E + T \rangle + T \rangle$

● $\langle E \rightarrow \langle T \sim E \rightarrow E + T \rangle * F \rangle$

● $\langle E \rightarrow \langle E \sim T \rightarrow T * F \rangle + T \rangle$

$\langle T \rightarrow \langle T \sim F \rightarrow \text{NUM} \rangle * F \rangle$

$\langle E \rightarrow \langle E \sim F \rightarrow \text{NUM} \rangle + T \rangle$

%token NUM

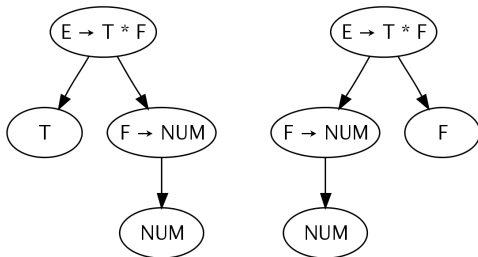
E: E '+' T

| T

T: T '*' F

| F

F: NUM



● $\langle T \rightarrow T * \langle F \sim T \rightarrow T * F \rangle \rangle$

● $\langle E \rightarrow E + \langle T \sim E \rightarrow E + T \rangle \rangle$

● $\langle E \rightarrow T * \langle F \sim E \rightarrow E + T \rangle \rangle$

● $\langle E \rightarrow E + \langle T \rightarrow T * F \rangle \rangle$

● $\langle T \rightarrow T * \langle F \rightarrow \text{NUM} \rangle \rangle$

$\langle E \rightarrow E + \langle T \sim F \rightarrow \text{NUM} \rangle \rangle$

● $\langle E \rightarrow \langle T \rightarrow T * F \rangle * F \rangle$

● $\langle E \rightarrow \langle E \rightarrow E + T \rangle + T \rangle$

● $\langle E \rightarrow \langle T \sim E \rightarrow E + T \rangle * F \rangle$

● $\langle E \rightarrow \langle E \sim T \rightarrow T * F \rangle + T \rangle$

● $\langle T \rightarrow \langle T \sim F \rightarrow \text{NUM} \rangle * F \rangle$

$\langle E \rightarrow \langle E \sim F \rightarrow \text{NUM} \rangle + T \rangle$

%token NUM

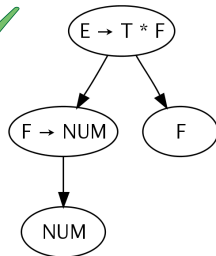
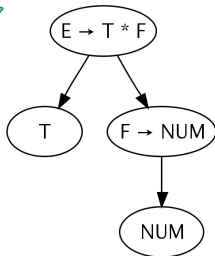
E: E '+' T

| T

T: T '*' F

| F

F: NUM



● $\langle T \rightarrow T * \langle F \sim T \rightarrow T * F \rangle \rangle$

● $\langle E \rightarrow E + \langle T \sim E \rightarrow E + T \rangle \rangle$

● $\langle E \rightarrow T * \langle F \sim E \rightarrow E + T \rangle \rangle$

● $\langle E \rightarrow E + \langle T \rightarrow T * F \rangle \rangle$

● $\langle T \rightarrow T * \langle F \rightarrow \text{NUM} \rangle \rangle$

$\langle E \rightarrow E + \langle T \sim F \rightarrow \text{NUM} \rangle \rangle$

● $\langle E \rightarrow \langle T \rightarrow T * F \rangle * F \rangle$

● $\langle E \rightarrow \langle E \rightarrow E + T \rangle + T \rangle$

● $\langle E \rightarrow \langle T \sim E \rightarrow E + T \rangle * F \rangle$

● $\langle E \rightarrow \langle E \sim T \rightarrow T * F \rangle + T \rangle$

● $\langle T \rightarrow \langle T \sim F \rightarrow \text{NUM} \rangle * F \rangle$

$\langle E \rightarrow \langle E \sim F \rightarrow \text{NUM} \rangle + T \rangle$

%token NUM

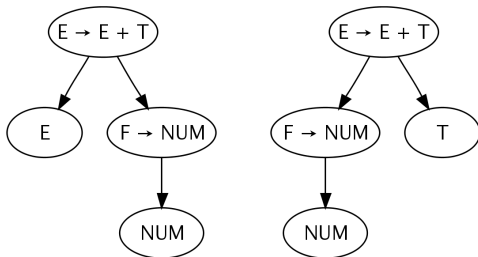
E: E '+' T

| T

T: T '*' F

| F

F: NUM



● $\langle T \rightarrow T * \langle F \sim T \rightarrow T * F \rangle \rangle$

● $\langle E \rightarrow E + \langle T \sim E \rightarrow E + T \rangle \rangle$

● $\langle E \rightarrow T * \langle F \sim E \rightarrow E + T \rangle \rangle$

● $\langle E \rightarrow E + \langle T \rightarrow T * F \rangle \rangle$

● $\langle T \rightarrow T * \langle F \rightarrow \text{NUM} \rangle \rangle$

● $\langle E \rightarrow E + \langle T \sim F \rightarrow \text{NUM} \rangle \rangle$

● $\langle E \rightarrow \langle T \rightarrow T * F \rangle * F \rangle$

● $\langle E \rightarrow \langle E \rightarrow E + T \rangle + T \rangle$

● $\langle E \rightarrow \langle T \sim E \rightarrow E + T \rangle * F \rangle$

● $\langle E \rightarrow \langle E \sim T \rightarrow T * F \rangle + T \rangle$

● $\langle T \rightarrow \langle T \sim F \rightarrow \text{NUM} \rangle * F \rangle$

● $\langle E \rightarrow \langle E \sim F \rightarrow \text{NUM} \rangle + T \rangle$

%token NUM

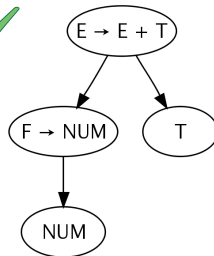
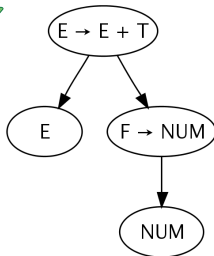
E: E '+' T

| T

T: T '*' F

| F

F: NUM



● $\langle T \rightarrow T * \langle F \sim T \rightarrow T * F \rangle \rangle$

● $\langle E \rightarrow E + \langle T \sim E \rightarrow E + T \rangle \rangle$

● $\langle E \rightarrow T * \langle F \sim E \rightarrow E + T \rangle \rangle$

● $\langle E \rightarrow E + \langle T \rightarrow T * F \rangle \rangle$

● $\langle T \rightarrow T * \langle F \rightarrow \text{NUM} \rangle \rangle$

● $\langle E \rightarrow E + \langle T \sim F \rightarrow \text{NUM} \rangle \rangle$

● $\langle E \rightarrow \langle T \rightarrow T * F \rangle * F \rangle$

● $\langle E \rightarrow \langle E \rightarrow E + T \rangle + T \rangle$

● $\langle E \rightarrow \langle T \sim E \rightarrow E + T \rangle * F \rangle$

● $\langle E \rightarrow \langle E \sim T \rightarrow T * F \rangle + T \rangle$

● $\langle T \rightarrow \langle T \sim F \rightarrow \text{NUM} \rangle * F \rangle$

● $\langle E \rightarrow \langle E \sim F \rightarrow \text{NUM} \rangle + T \rangle$

context-free syntax

$E \rightarrow E + T$

$T \rightarrow E$

$T \rightarrow T * F$

$F \rightarrow T$

$NUM \rightarrow T$

● $\langle T \rightarrow T * \langle F \sim T \rightarrow T * F \rangle \rangle$

● $\langle E \rightarrow E + \langle T \sim E \rightarrow E + T \rangle \rangle$

● $\langle E \rightarrow T * \langle F \sim E \rightarrow E + T \rangle \rangle$

● $\langle E \rightarrow E + \langle T \rightarrow T * F \rangle \rangle$

● $\langle T \rightarrow T * \langle F \rightarrow NUM \rangle \rangle$

● $\langle E \rightarrow E + \langle T \sim F \rightarrow NUM \rangle \rangle$

● $\langle E \rightarrow \langle T \rightarrow T * F \rangle * F \rangle$

● $\langle E \rightarrow \langle E \rightarrow E + T \rangle + T \rangle$

● $\langle E \rightarrow \langle T \sim E \rightarrow E + T \rangle * F \rangle$

● $\langle E \rightarrow \langle E \sim T \rightarrow T * F \rangle + T \rangle$

● $\langle T \rightarrow \langle T \sim F \rightarrow NUM \rangle * F \rangle$

● $\langle E \rightarrow \langle E \sim F \rightarrow NUM \rangle + T \rangle$