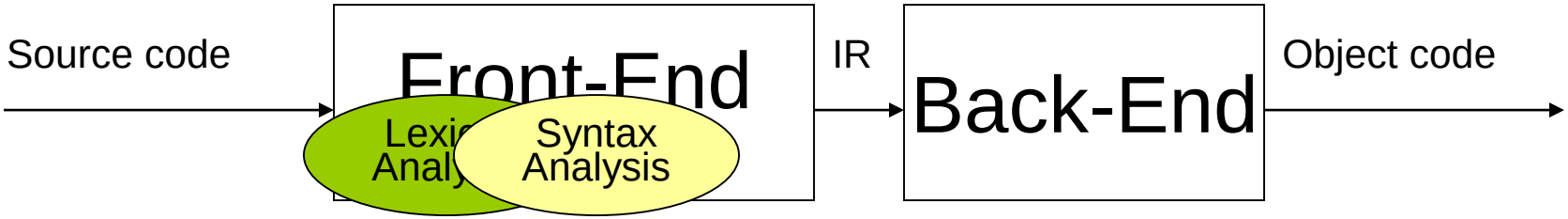
Compiler Design

Lecture 6:
Syntax Analysis
Bottom-Up Parsing (part II)

Dr. Momtazi
momtazi@aut.ac.ir

based on the slides of the course book

Parsing (Syntax Analysis)



- Syntax Analysis:
  - Derivation and parse trees ✓
  - Top-down parsing ✓
  - **Bottom-up parsing**

Outline

- Introduction
- Shift-reduce Parsing
- **LR Parsing**
  - **LR(0)**
  - LR(1)
  - SLR
  - LALR
- Ambiguity
- Error-handling

Our First Algorithm: LR(0)

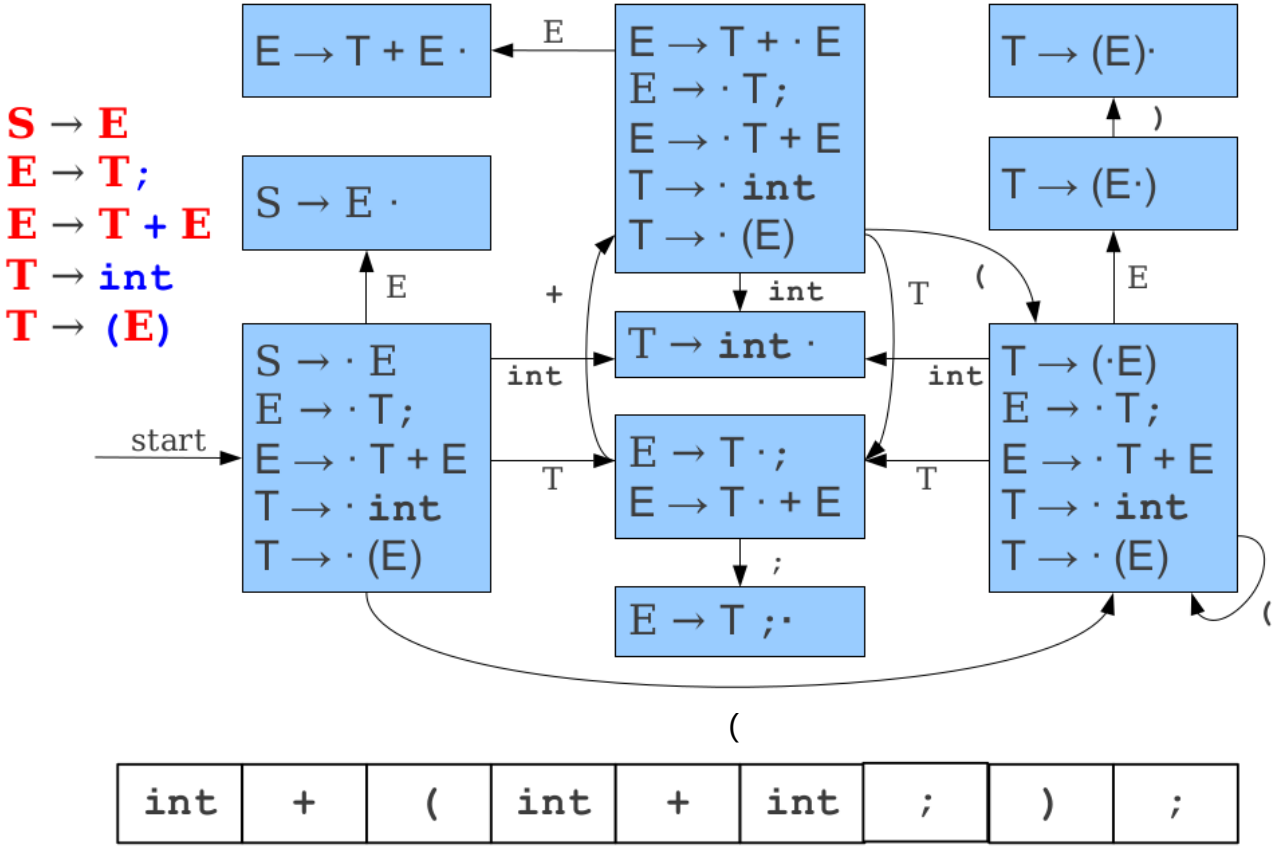
- Bottom-up predictive parsing with:
  - L: Left-to-right scan of the input.
  - R: Rightmost derivation.
  - (0): Zero tokens of lookahead.
- Use the handle-finding automaton, without any lookahead, to predict where handles are.

LR(0) Parsing

$S \rightarrow E$
$E \rightarrow T\,;$
$E \rightarrow T + E$
$T \rightarrow \texttt{int}$
$T \rightarrow (E)$

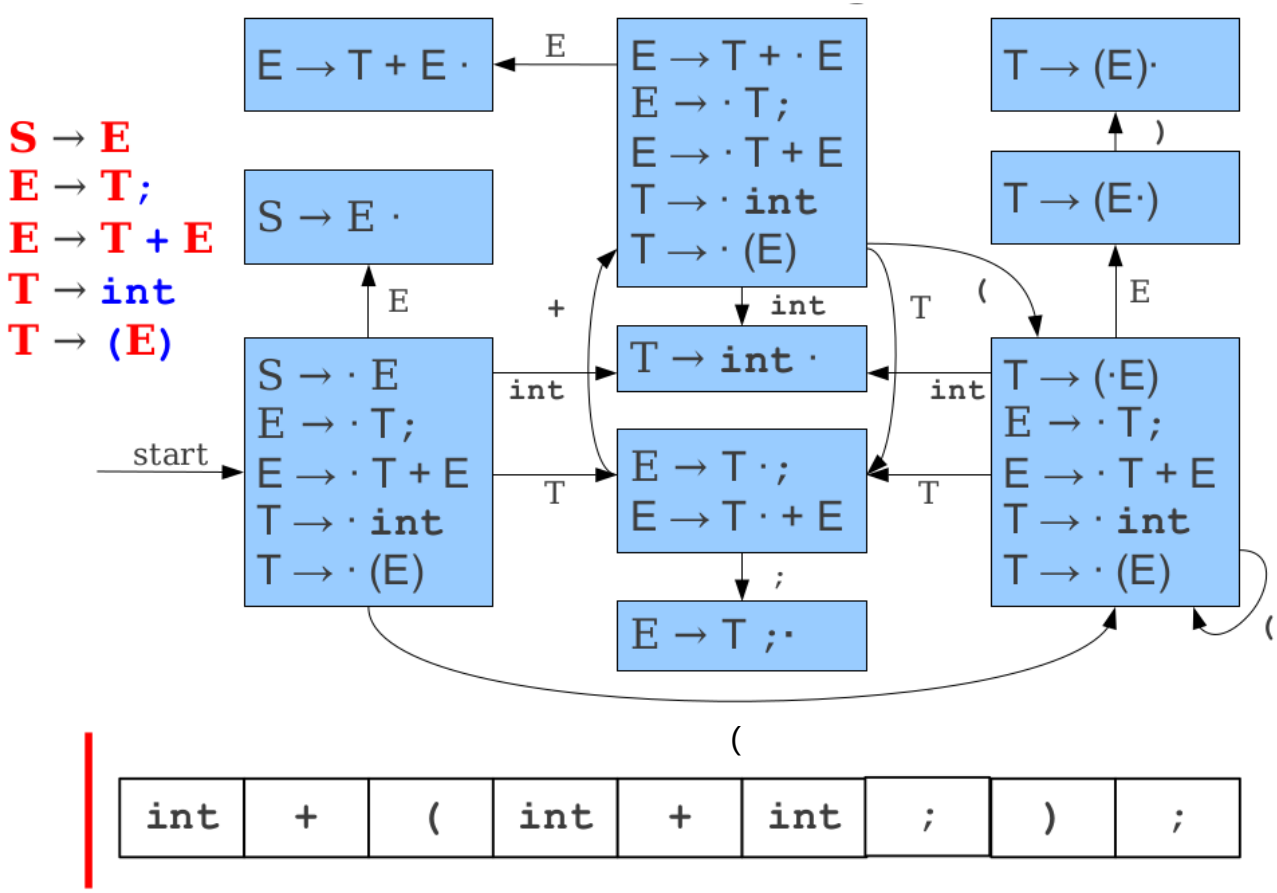
int	+	(	int	+	int	;	)	;

LR(0) Parsing



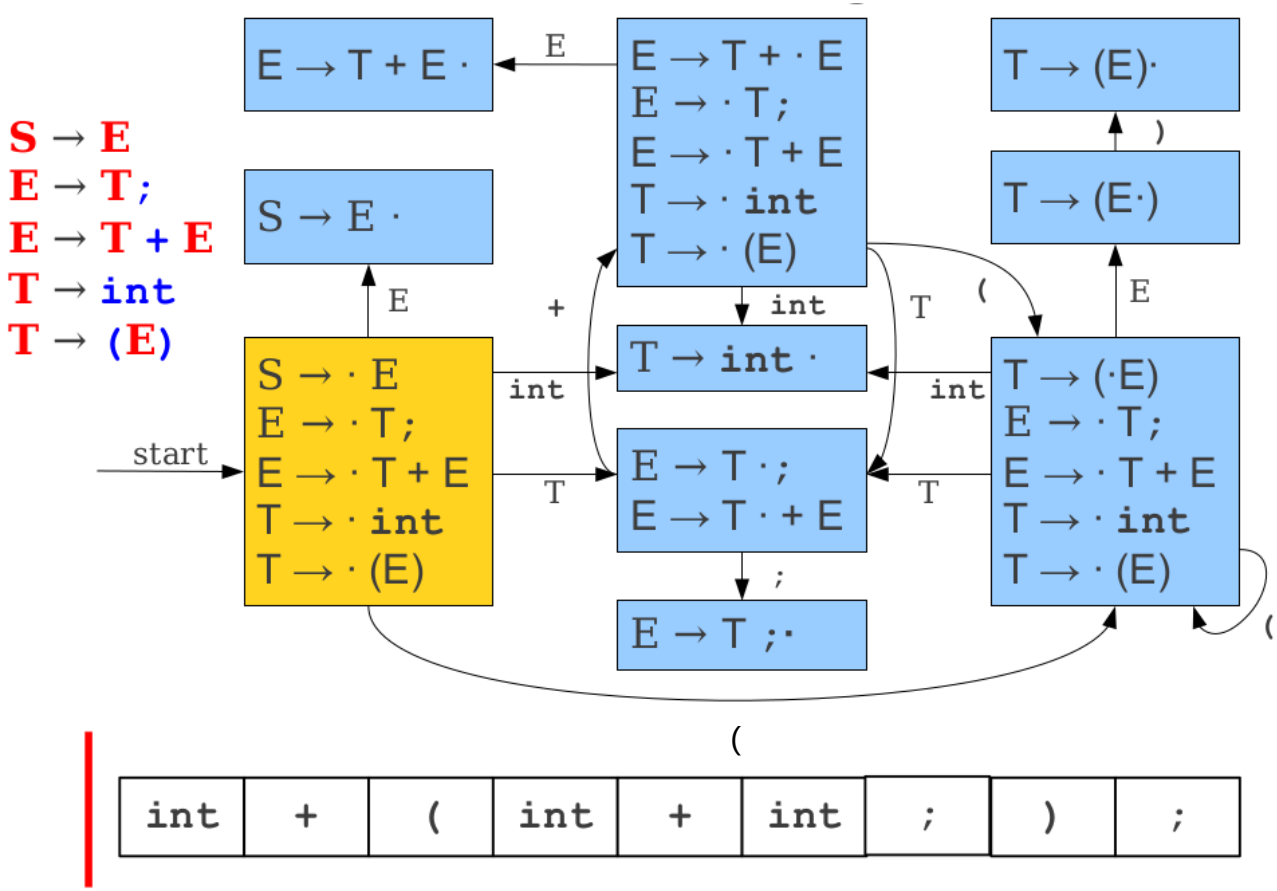
int	+	(	int	+	int	;	)	;

LR(0) Parsing



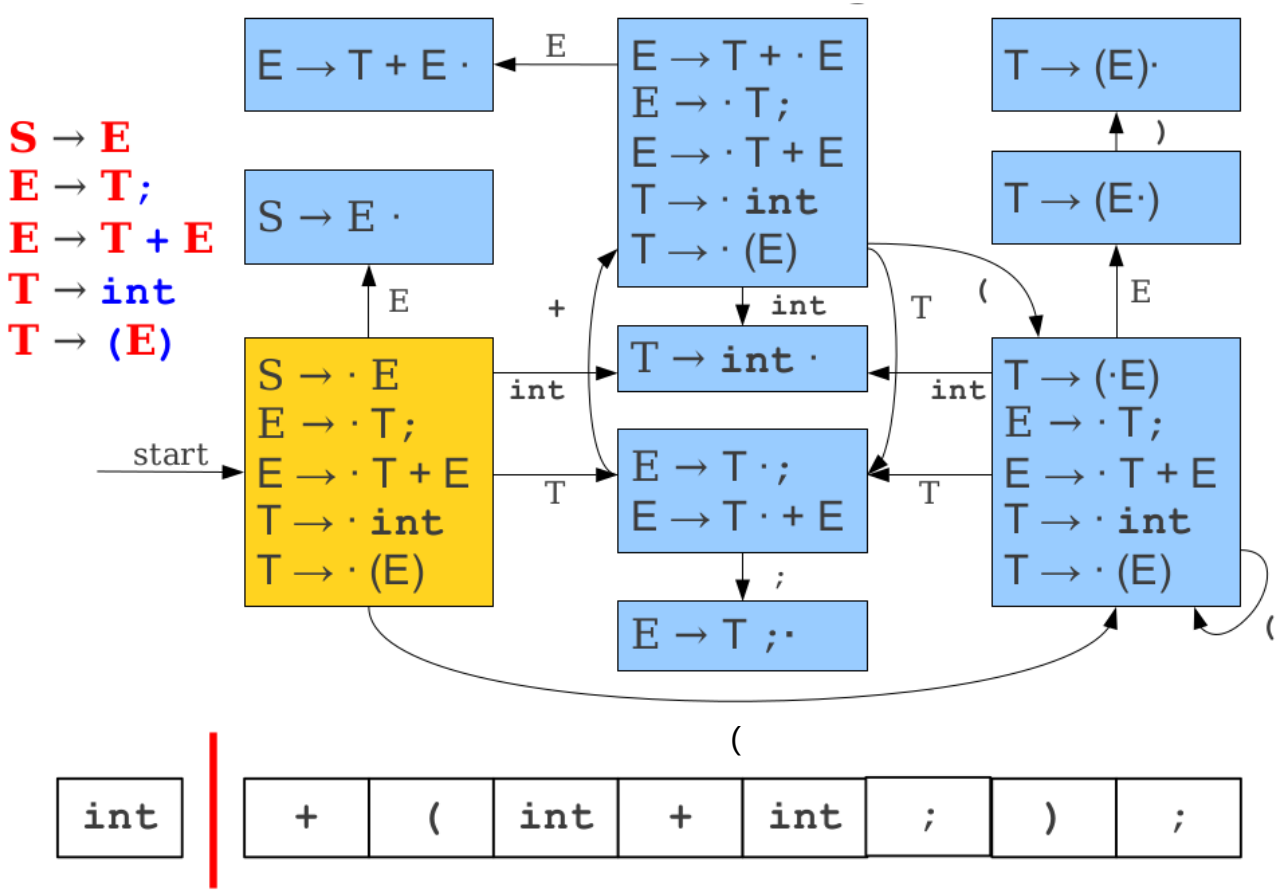
int	+	(	int	+	int	;	)	;

LR(0) Parsing

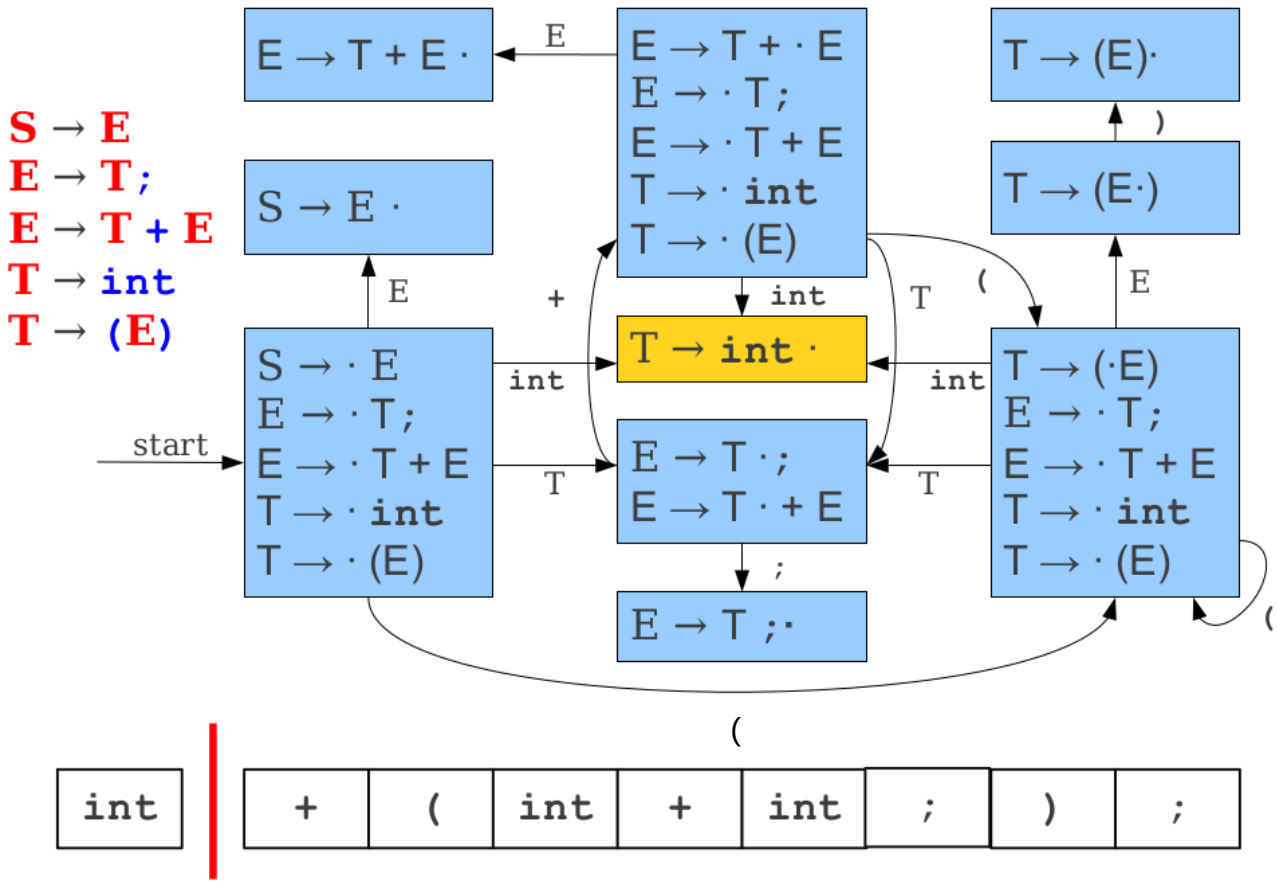


int	+	(	int	+	int	;	)	;

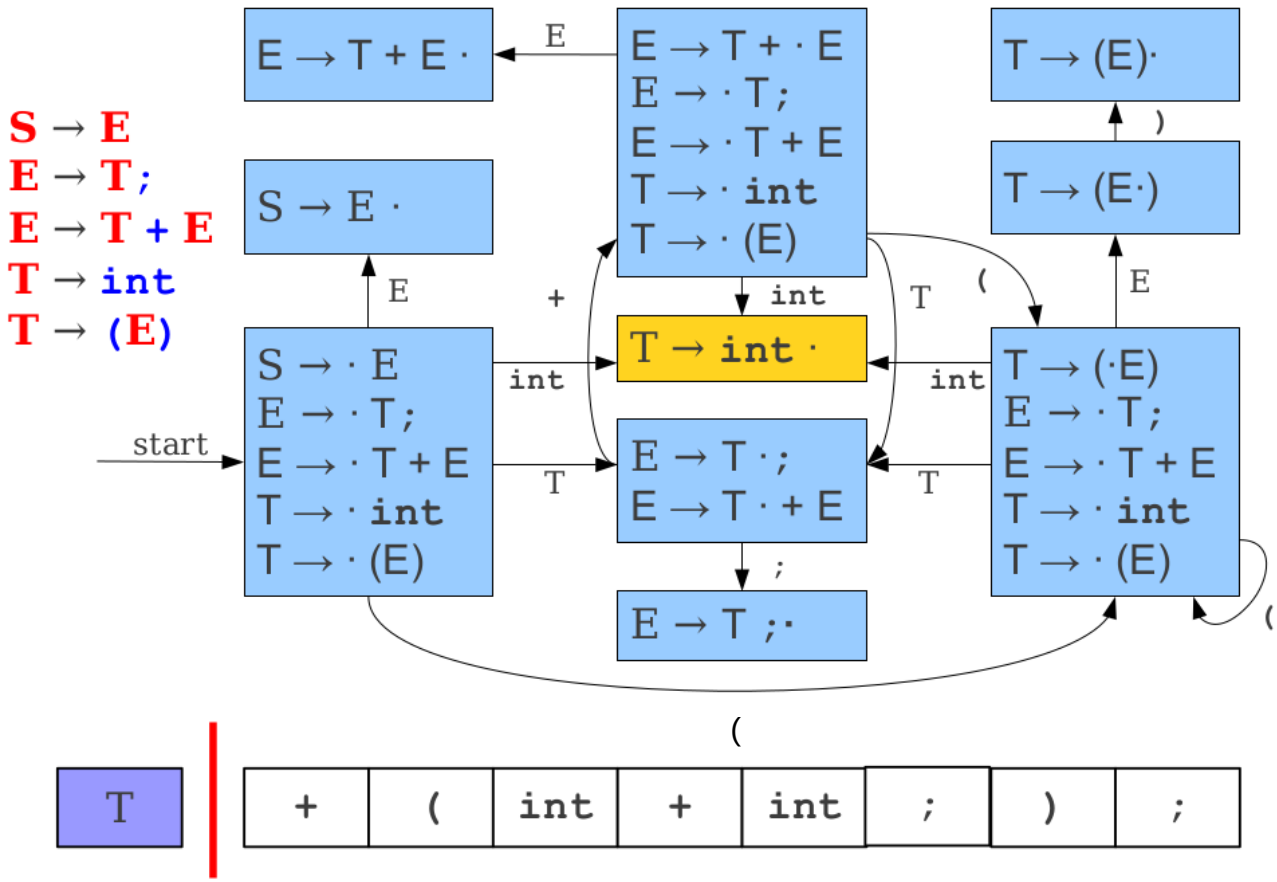
LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
E → T + E ·
S → E ·
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
T → (E)·
T → (E·)
T → int ·
E → T ·;
E → T · + E
E → T ;·
T → (·E)
E → · T;
E → · T + E
T → · int
T → · (E)
S → · E
E → · T;
E → · T + E
T → · int
T → · (E)
start
E
+
int
T
(
)
;
int + (int + int ;) ;

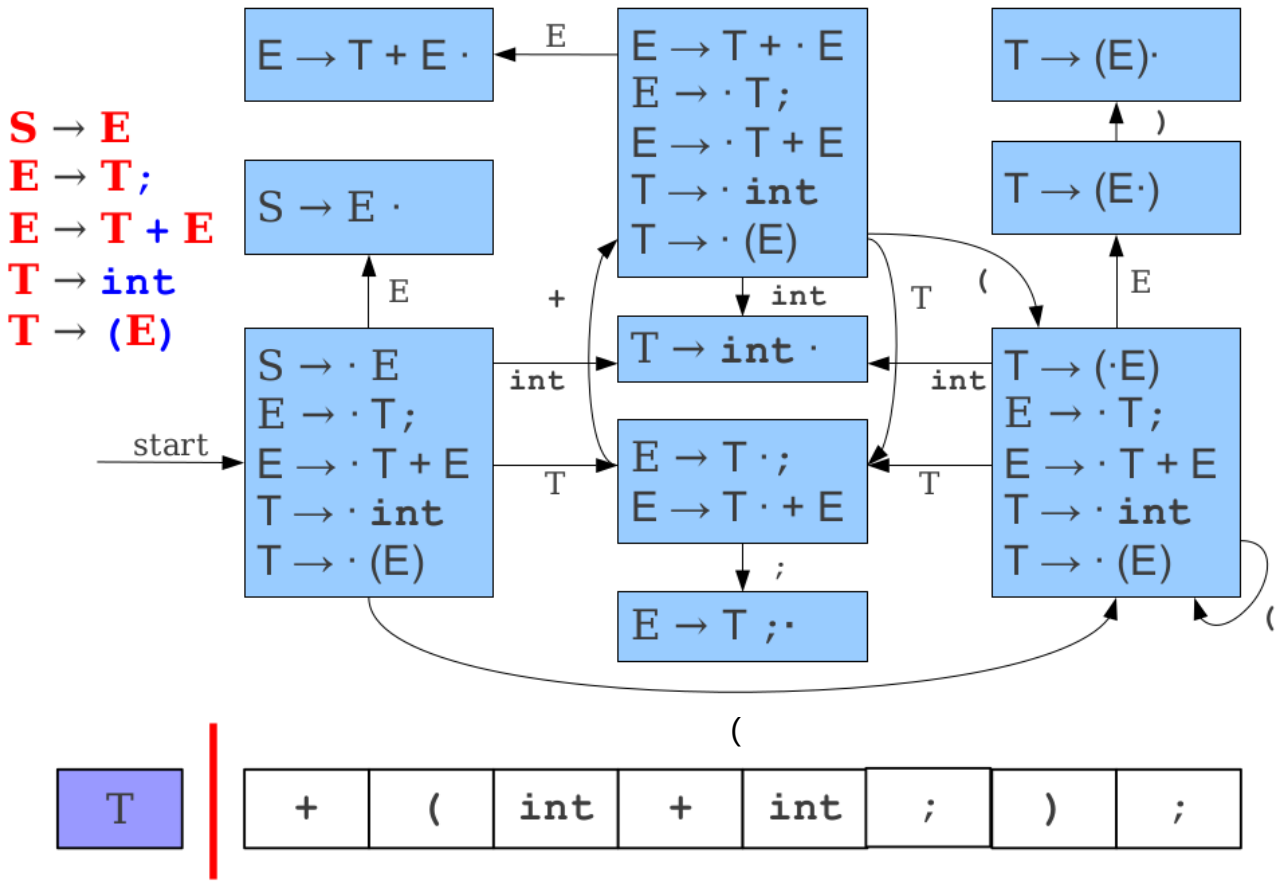
LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
E → T + E ·
S → E ·
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
T → (E)·
T → (E·)
T → int ·
E → T ·;
E → T · + E
E → T ;·
S → · E
E → · T;
E → · T + E
T → · int
T → · (E)
T → (·E)
E → · T;
E → · T + E
T → · int
T → · (E)
start
E
+
int
T
(
)
;
int + (int + int ;) ;

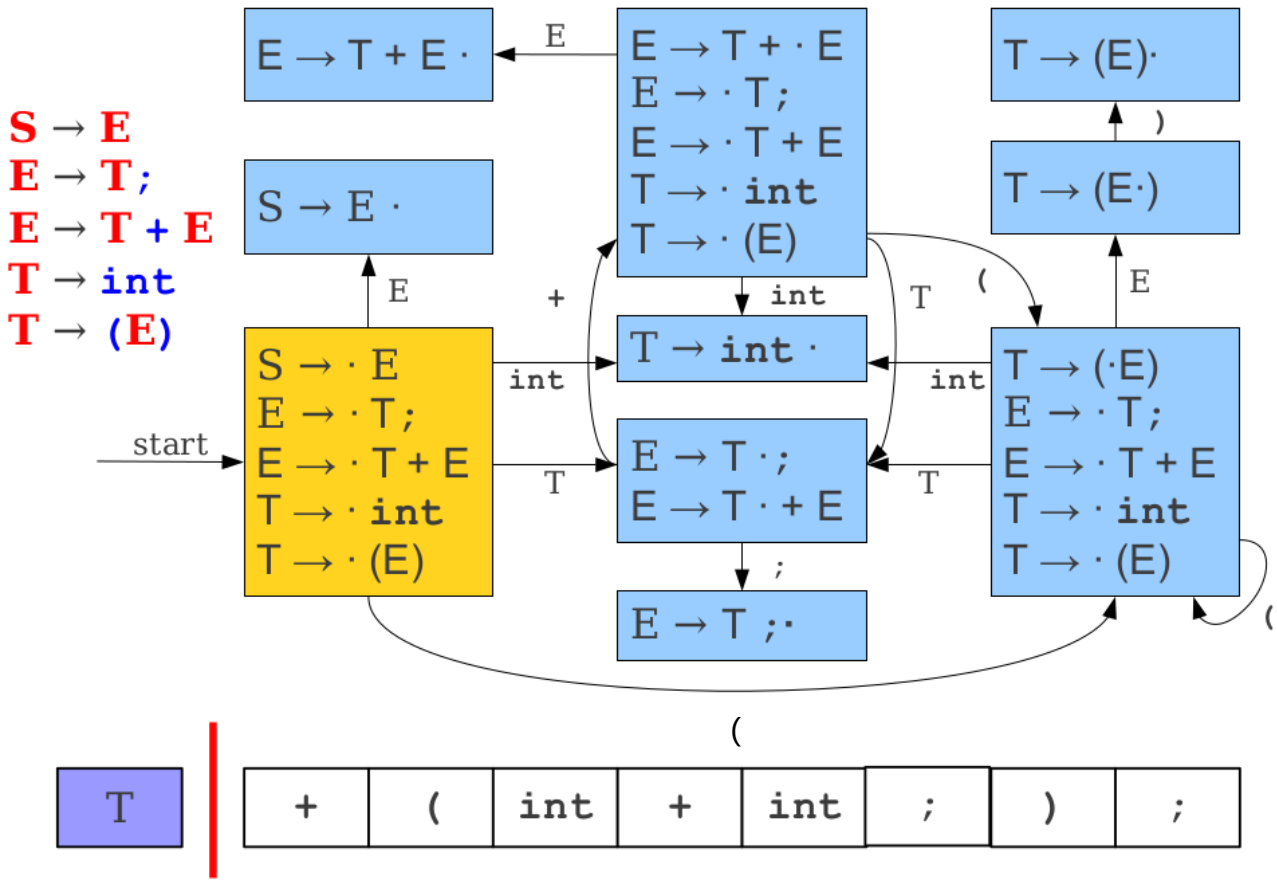
LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
E → T + E ·
S → E ·
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
T → (E)·
T → (E·)
T → int ·
E → T ·;
E → T · + E
E → T ;·
S → · E
E → · T;
E → · T + E
T → · int
T → · (E)
T → (·E)
E → · T;
E → · T + E
T → · int
T → · (E)
start
E
+
int
T
(
;
)
T
+
(
int
+
int
;
)
;

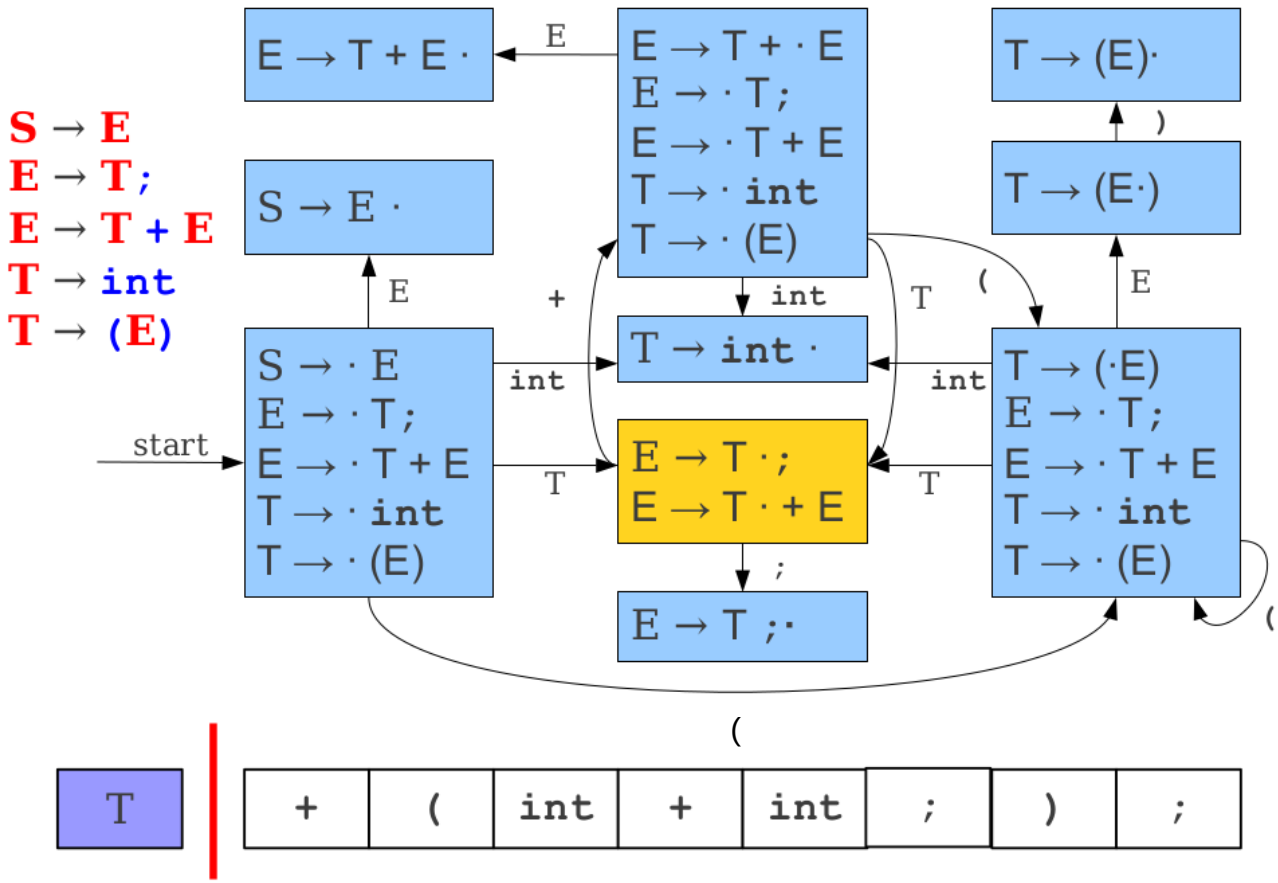
LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
E → T + E ·
S → E ·
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
T → (E)·
T → (E·)
S → · E
E → · T;
E → · T + E
T → · int
T → · (E)
T → int ·
E → T ·;
E → T · + E
E → T ;·
T → (·E)
E → · T;
E → · T + E
T → · int
T → · (E)
start
E
+
int
T
(
)
;
T
+
(
int
+
int
;
)
;

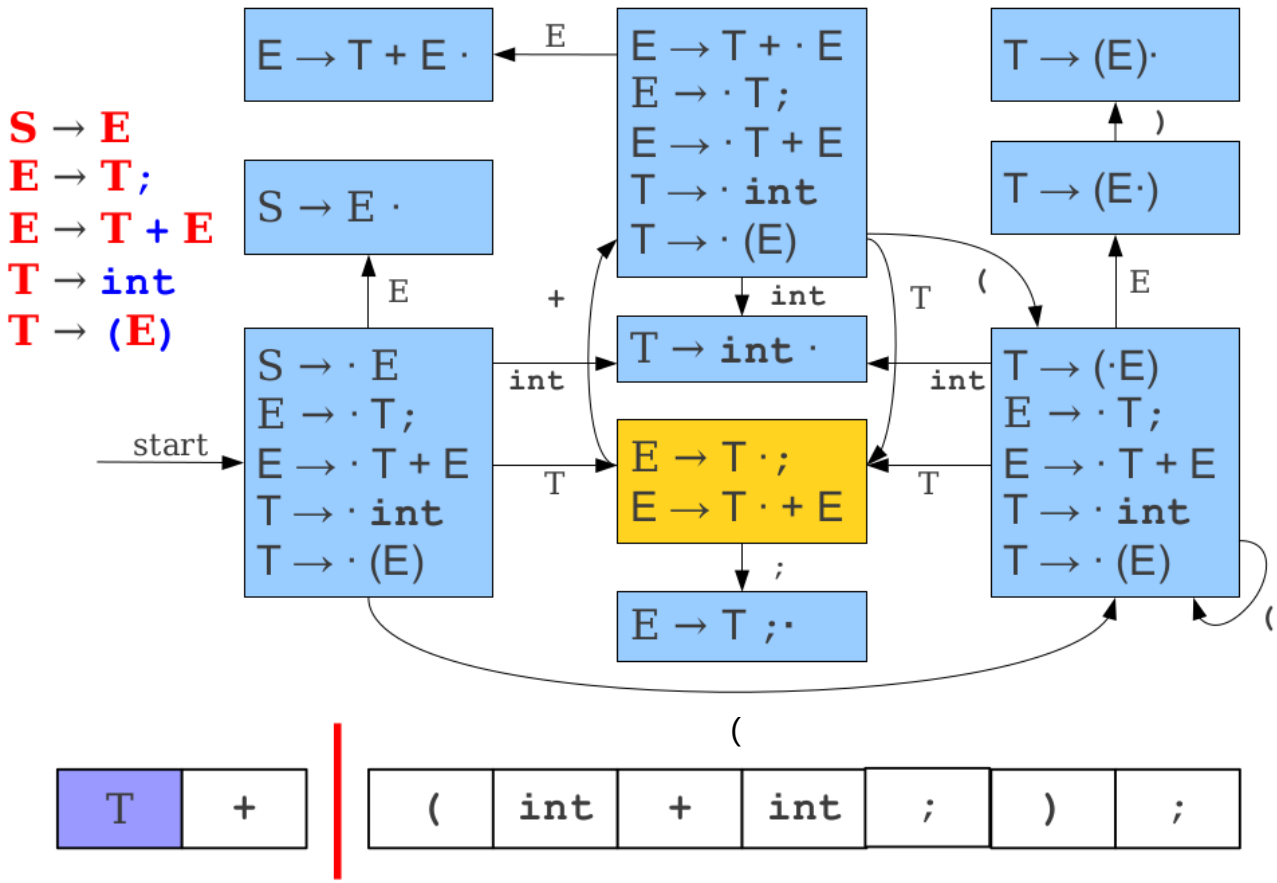
LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
E → T + E ·
S → E ·
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
T → (E)·
T → (E·)
S → · E
E → · T;
E → · T + E
T → · int
T → · (E)
T → int ·
E → T ·;
E → T · + E
E → T ;·
T → (·E)
E → · T;
E → · T + E
T → · int
T → · (E)
start
E
+
int
T
(
)
;
T
+
(
int
+
int
;
)
;

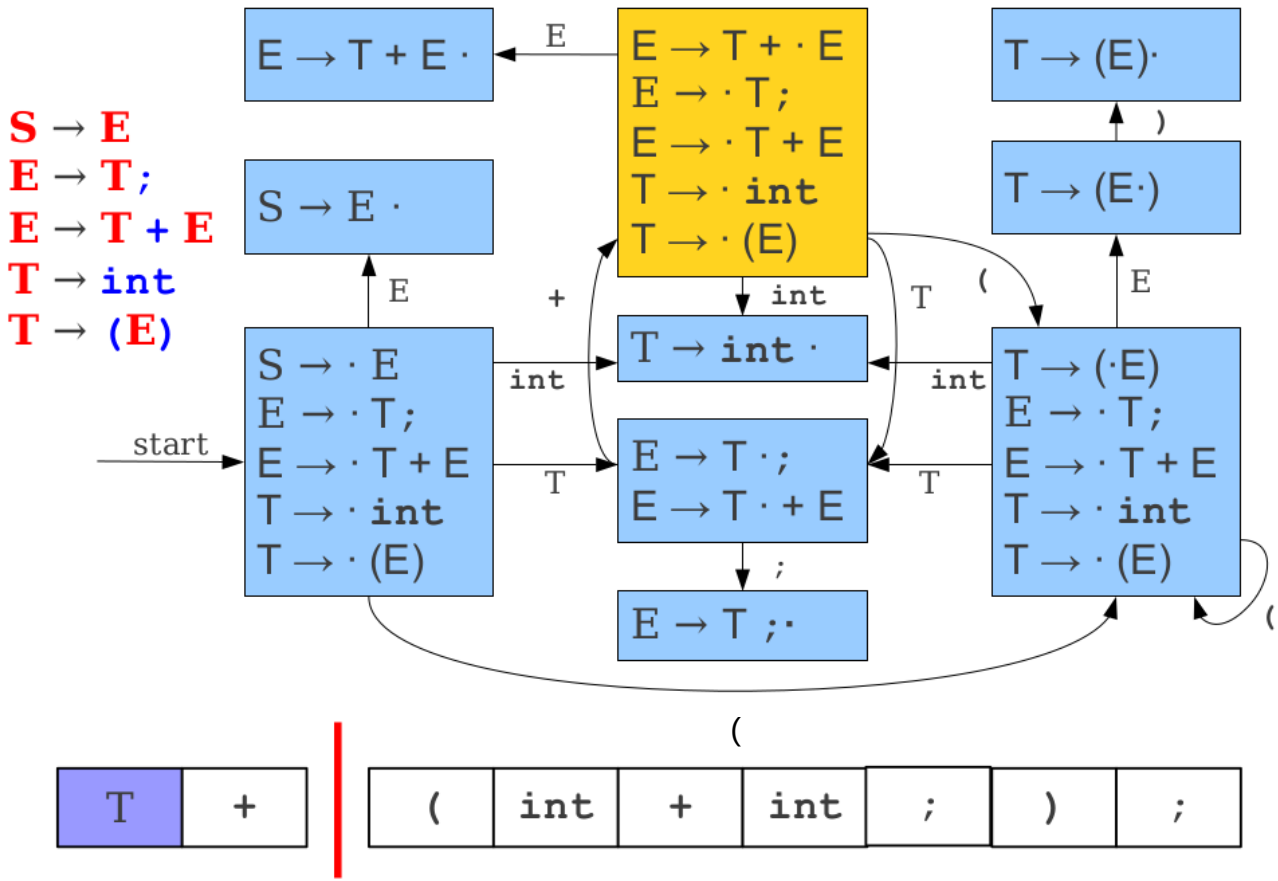
LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
E → T + E ·
S → E ·
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
T → (E)·
T → (E·)
S → · E
E → · T;
E → · T + E
T → · int
T → · (E)
T → int ·
E → T ·;
E → T · + E
E → T ;·
T → (·E)
E → · T;
E → · T + E
T → · int
T → · (E)
start
E
+
int
T
(
)
;
T
+
(
int
+
int
;
)
;

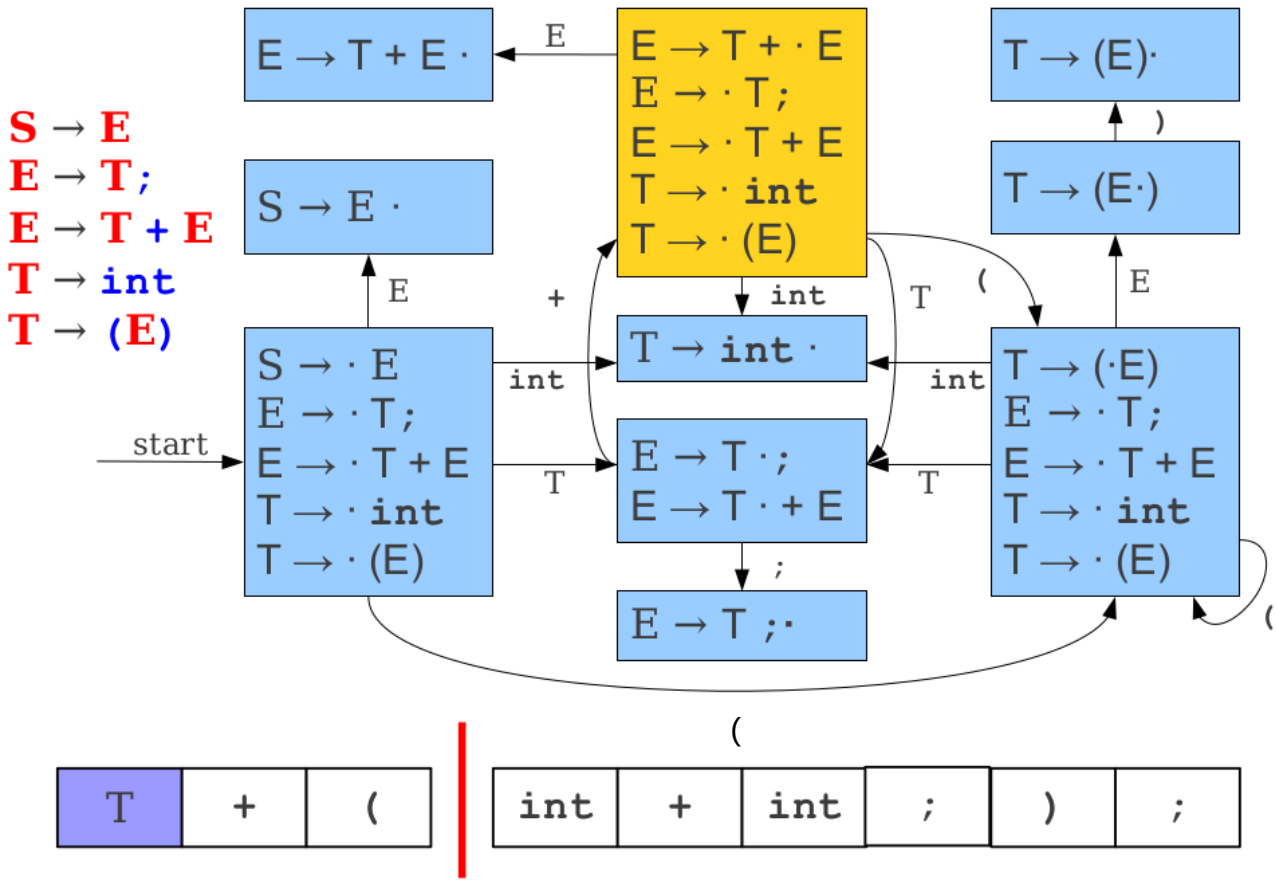
LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
E → T + E ·
S → E ·
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
T → (E)·
T → (E·)
S → · E
E → · T;
E → · T + E
T → · int
T → · (E)
T → int ·
E → T ·;
E → T · + E
E → T ;·
T → (·E)
E → · T;
E → · T + E
T → · int
T → · (E)
start
E
+
int
T
;
(
)
T
+
(
int
+
int
;
)
;

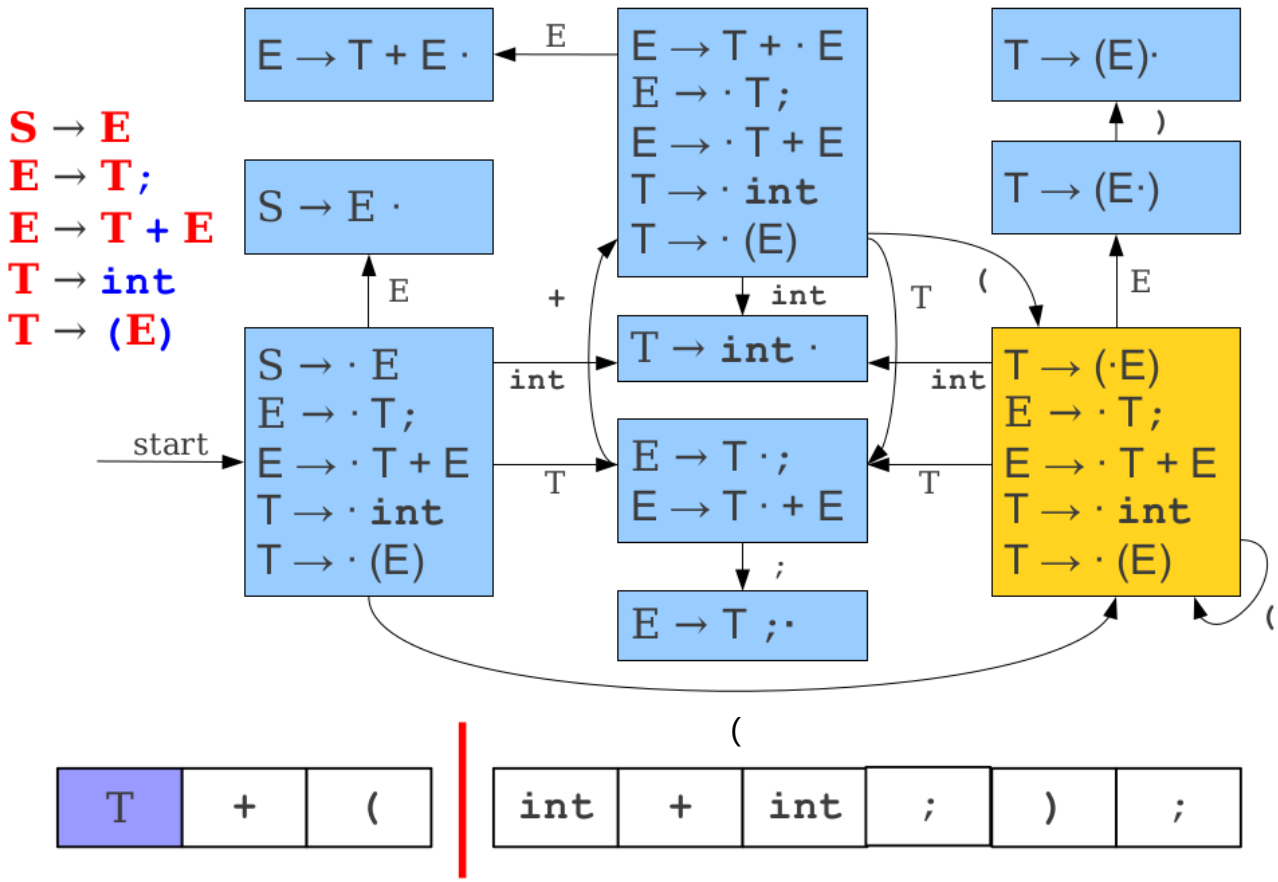
LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
E → T + E ·
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
T → (E)·
T → (E·)
S → E ·
S → · E
E → · T;
E → · T + E
T → · int
T → · (E)
T → int ·
E → T ·;
E → T · + E
E → T ;·
T → (·E)
E → · T;
E → · T + E
T → · int
T → · (E)
start
E
+
int
T
(
)
;
T
+
(
int
+
int
;
)
;

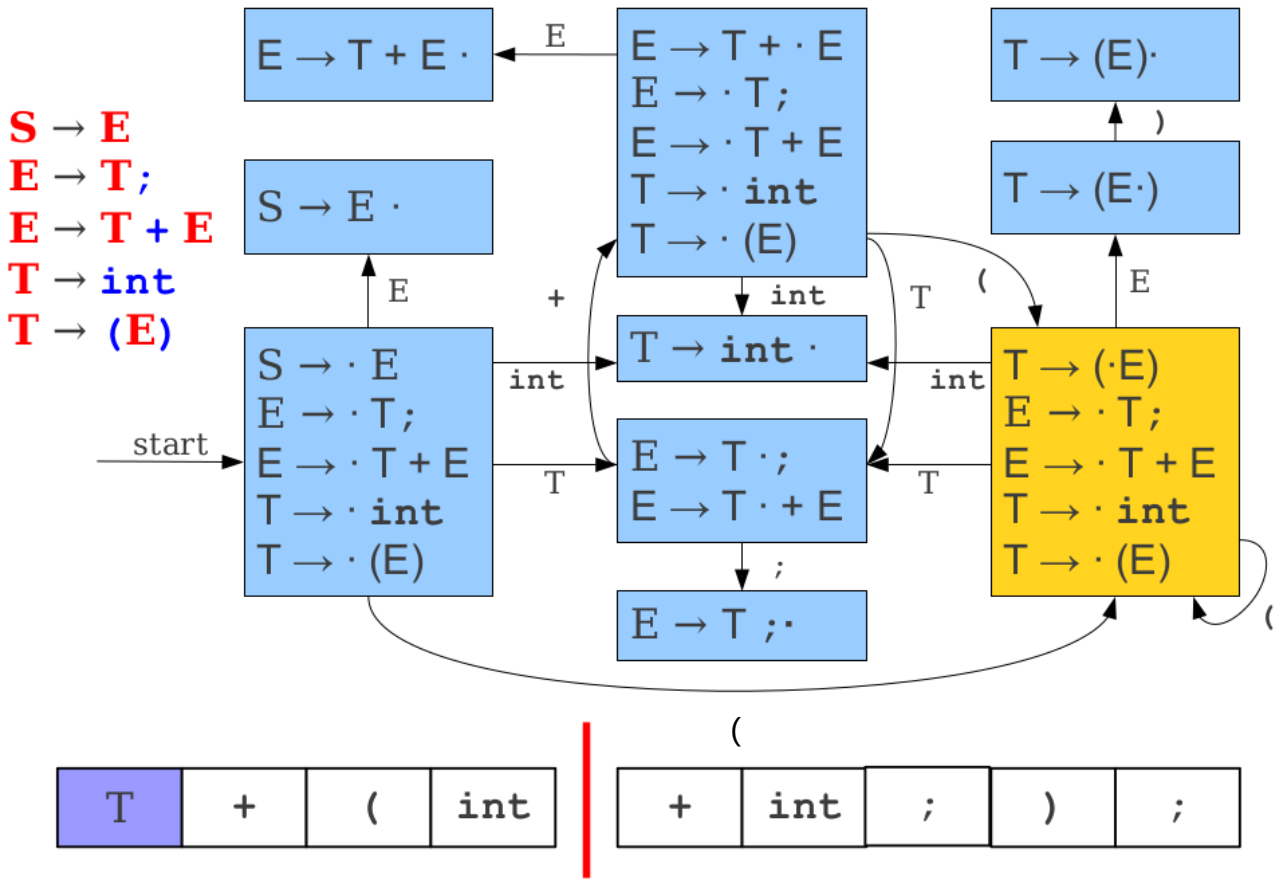
LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
E → T + E ·
S → E ·
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
T → (E)·
T → (E·)
S → · E
E → · T;
E → · T + E
T → · int
T → · (E)
T → int ·
E → T ·;
E → T · + E
E → T ;·
T → (·E)
E → · T;
E → · T + E
T → · int
T → · (E)
start
E
+
int
T
(
)
;
T
+
(
int
+
int
;
)
;

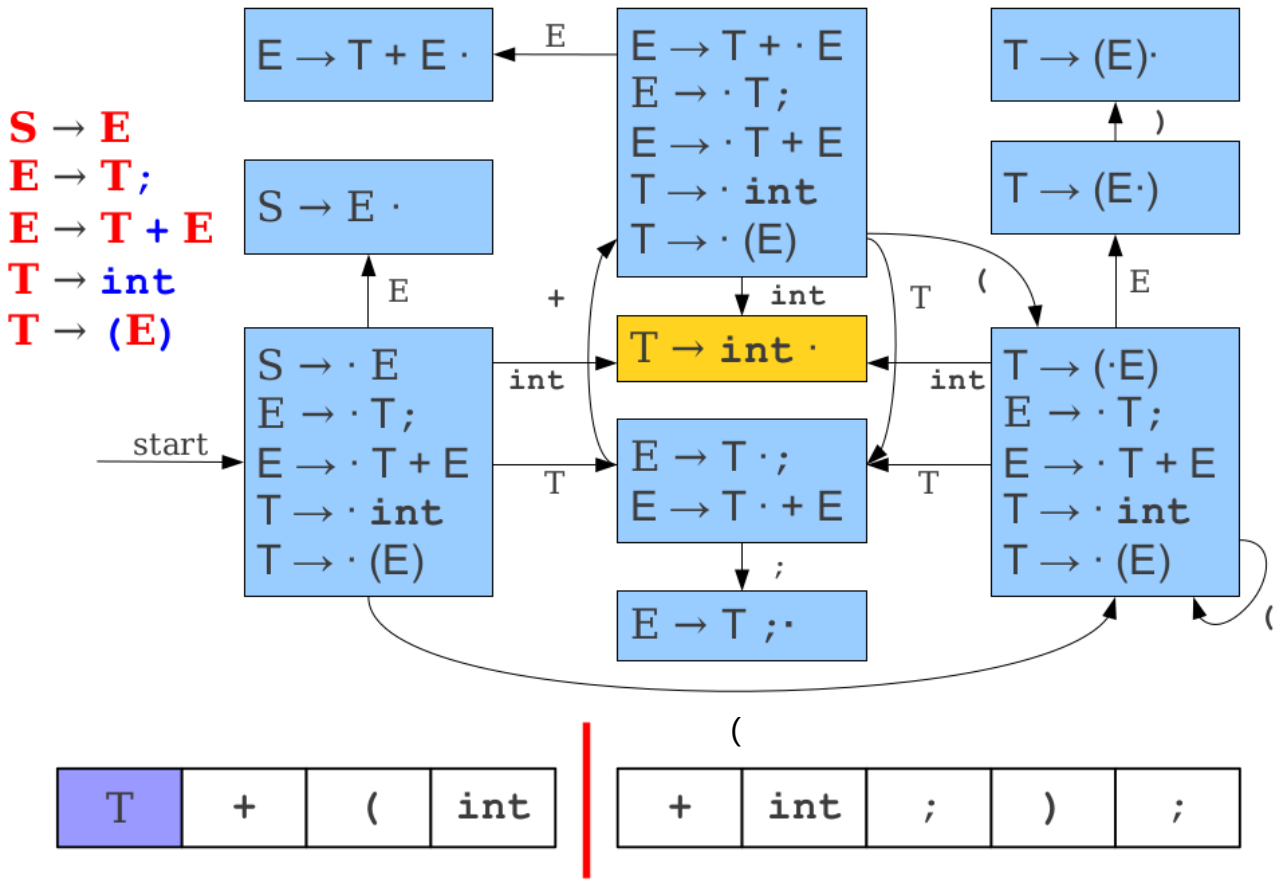
LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
E → T + E ·
S → E ·
S → · E
E → · T;
E → · T + E
T → · int
T → · (E)
start
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
T → int ·
E → T ·;
E → T · + E
E → T ;·
T → (E)·
T → (E·)
T → (·E)
E → · T;
E → · T + E
T → · int
T → · (E)
E
+
int
T
;
(
)
T
+
(
int
+
int
;
)
;

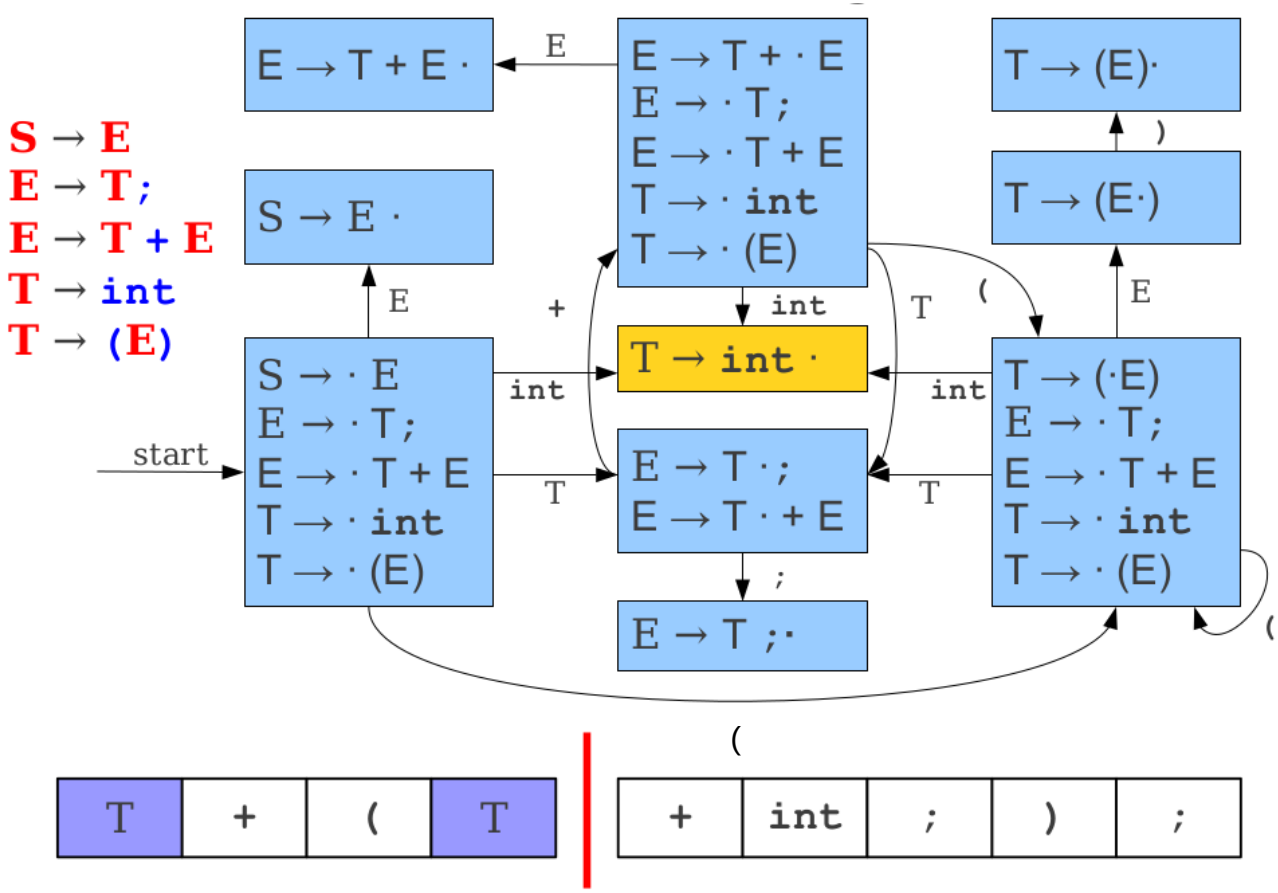
LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
E → T + E ·
S → E ·
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
T → (E)·
T → (E·)
S → · E
E → · T;
E → · T + E
T → · int
T → · (E)
T → int ·
E → T ·;
E → T · + E
E → T ;·
T → (·E)
E → · T;
E → · T + E
T → · int
T → · (E)
start
E
+
int
T
;
(
)
T
+
(
int
+
int
;
)
;

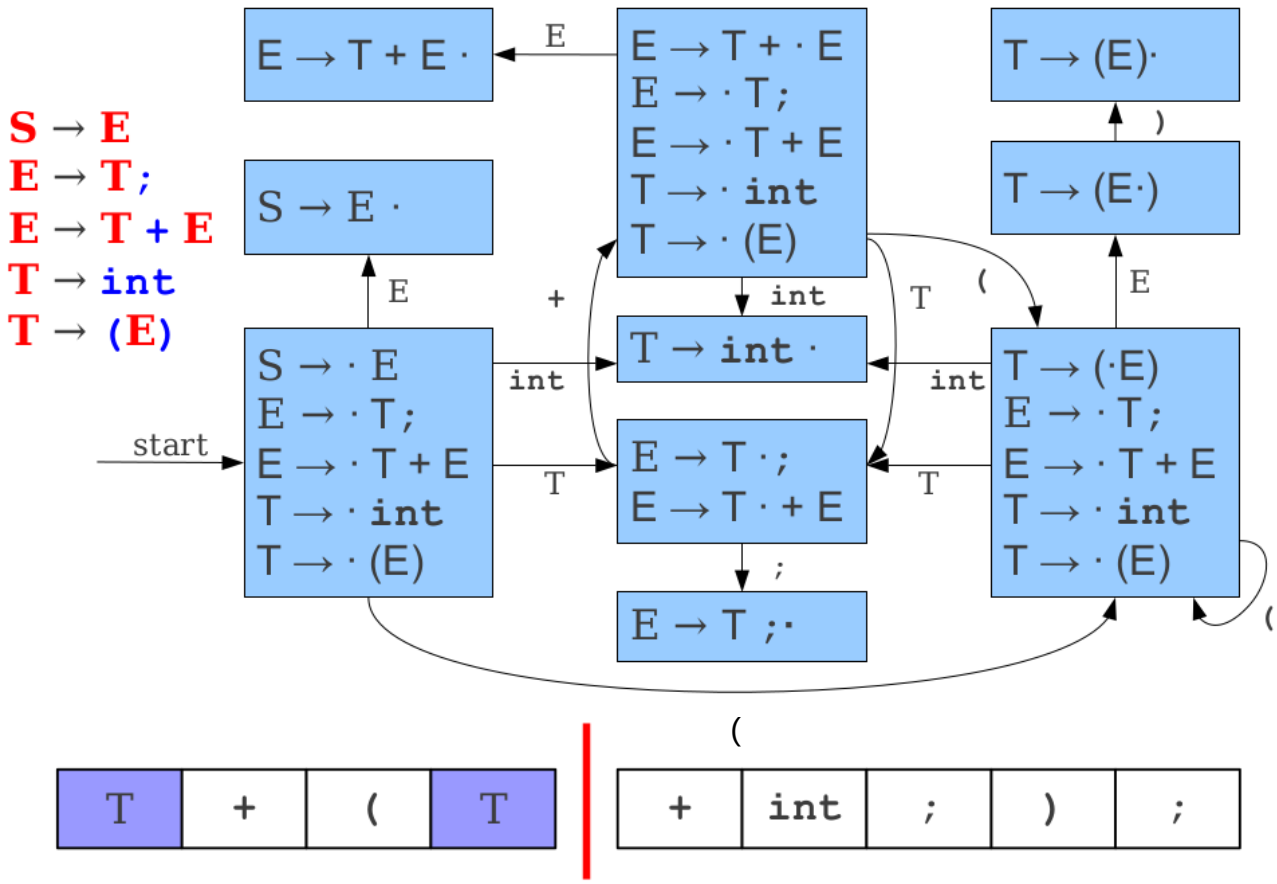
LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
E → T + E ·
S → E ·
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
T → (E)·
T → (E·)
T → int ·
E → T ·;
E → T · + E
E → T ;·
T → (·E)
E → · T;
E → · T + E
T → · int
T → · (E)
S → · E
E → · T;
E → · T + E
T → · int
T → · (E)
start
E
+
int
T
(
)
;
T
+
(
int
+
int
;
)
;

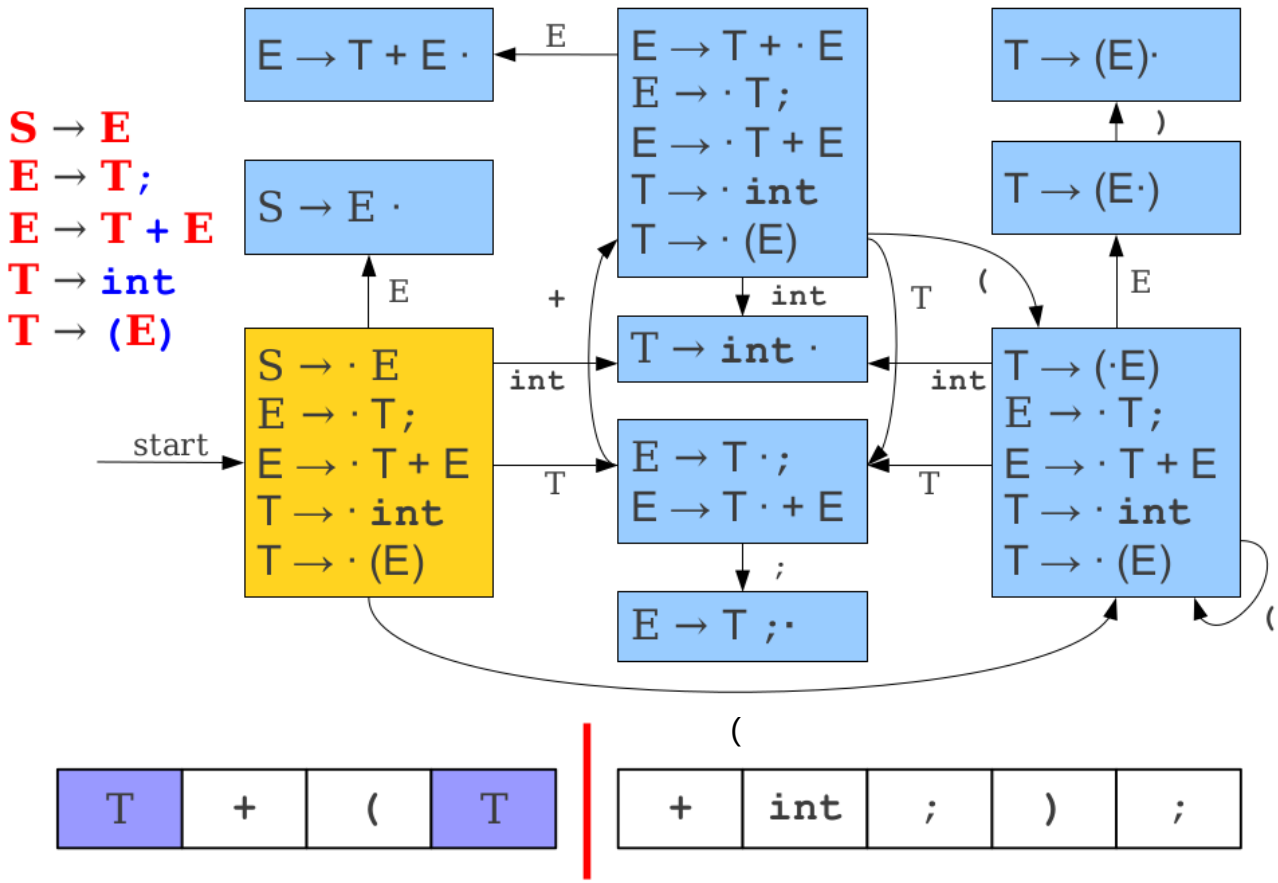
LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
E → T + E ·
S → E ·
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
T → (E)·
T → (E·)
T → int ·
E → T ·;
E → T · + E
E → T ;·
T → (·E)
E → · T;
E → · T + E
T → · int
T → · (E)
S → · E
E → · T;
E → · T + E
T → · int
T → · (E)
start
E
+
int
T
(
)
;
T
+
(
T
+
int
;
)
;

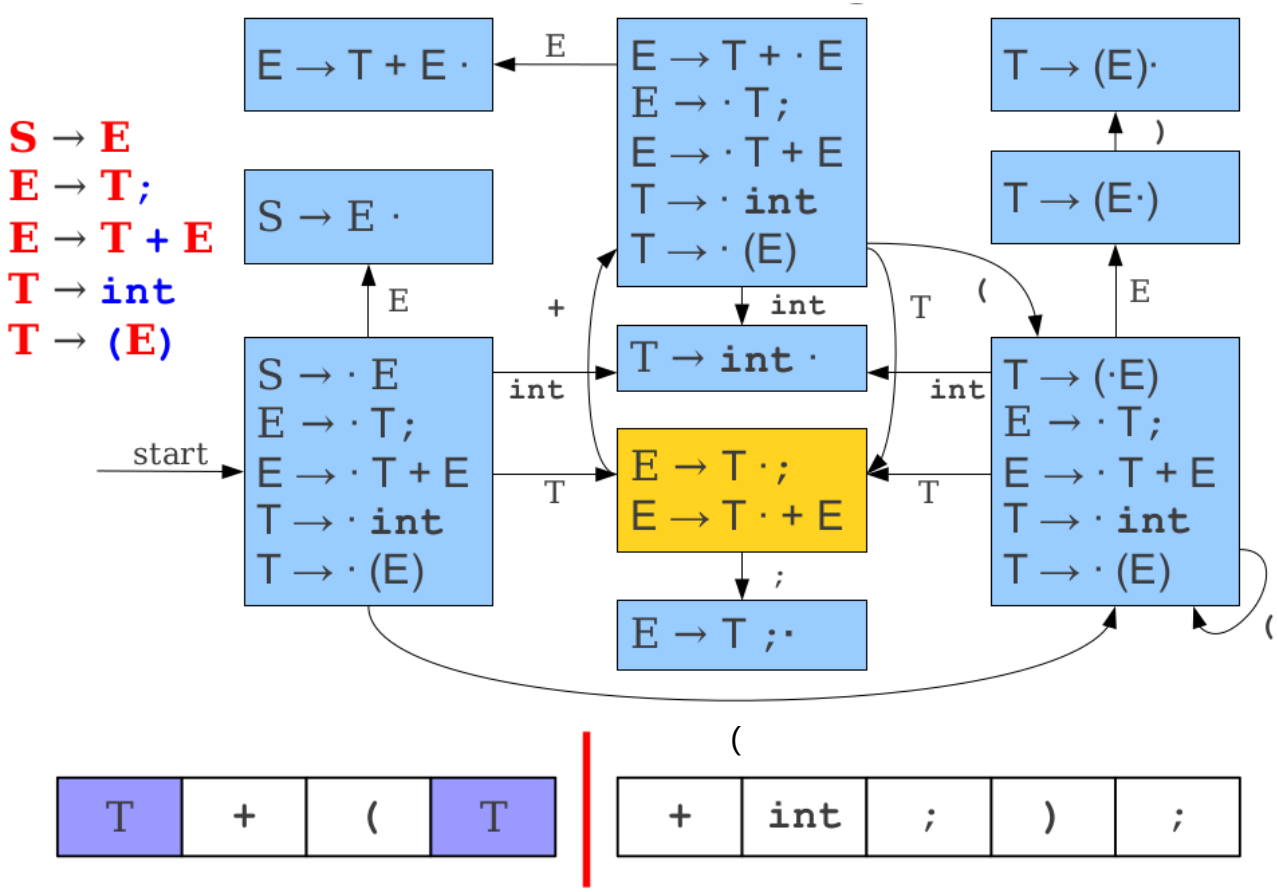
LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
E → T + E ·
S → E ·
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
T → (E)·
T → (E·)
S → · E
E → · T;
E → · T + E
T → · int
T → · (E)
T → int ·
E → T ·;
E → T · + E
E → T ;·
T → (·E)
E → · T;
E → · T + E
T → · int
T → · (E)
start
E
+
int
T
(
;
)
T
+
(
T
+
int
;
)
;

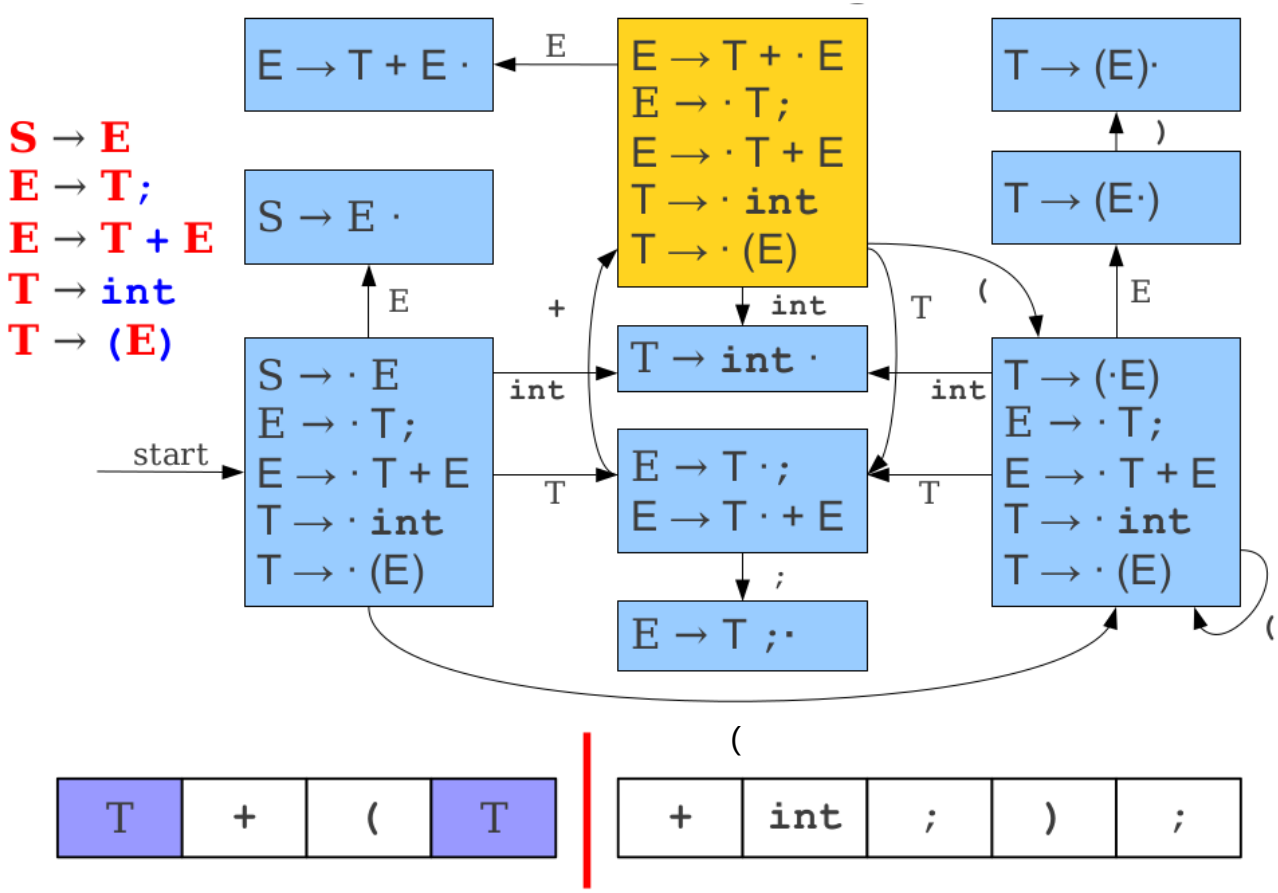
LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
E → T + E ·
S → E ·
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
T → (E)·
T → (E·)
S → · E
E → · T;
E → · T + E
T → · int
T → · (E)
T → int ·
E → T ·;
E → T · + E
E → T ;·
T → (·E)
E → · T;
E → · T + E
T → · int
T → · (E)
start
E
+
int
T
(
)
;
T
+
(
T
+
int
;
)
;

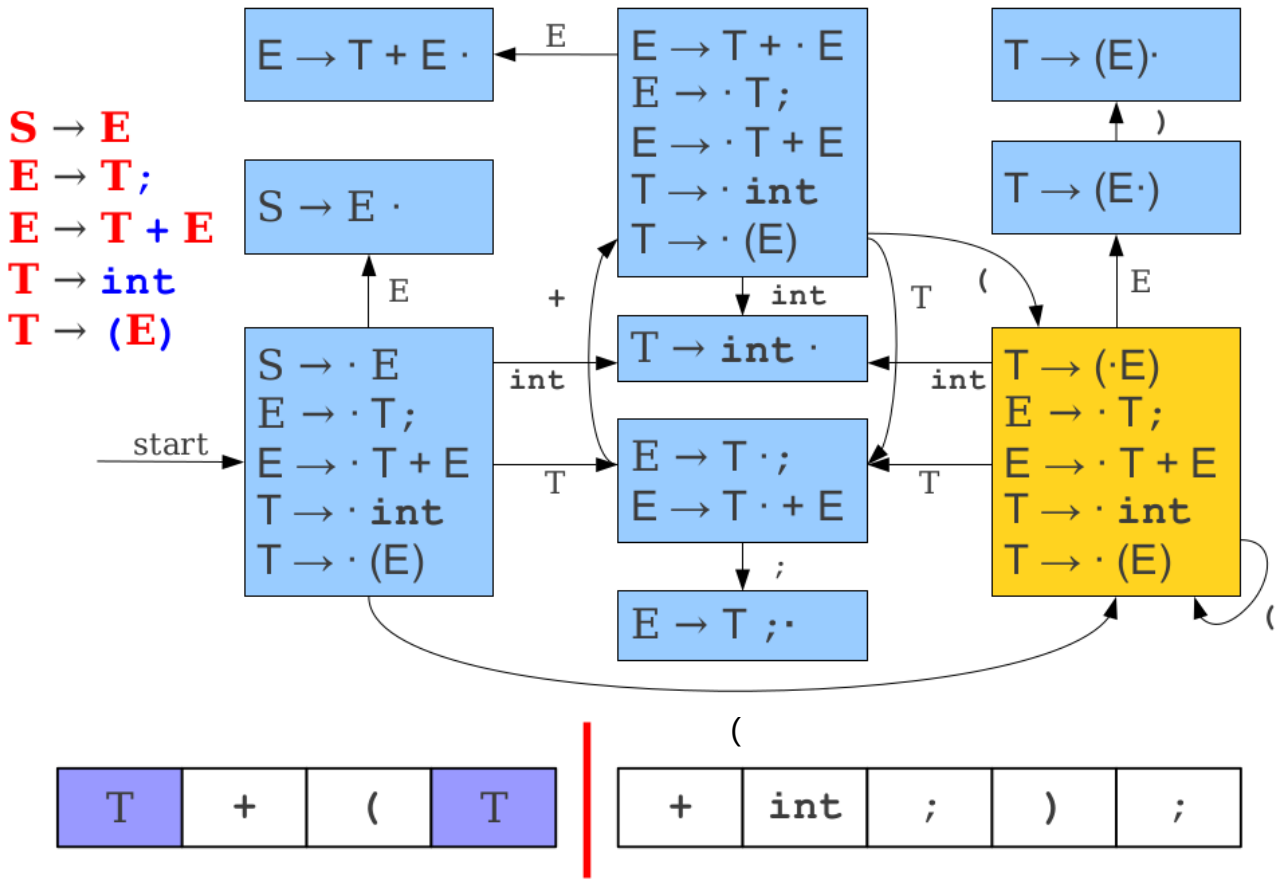
LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
E → T + E ·
S → E ·
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
T → (E)·
T → (E·)
S → · E
E → · T;
E → · T + E
T → · int
T → · (E)
T → int ·
E → T ·;
E → T · + E
E → T ;·
T → (·E)
E → · T;
E → · T + E
T → · int
T → · (E)
start
E
+
int
T
(
)
;
T
+
(
T
+
int
;
)
;

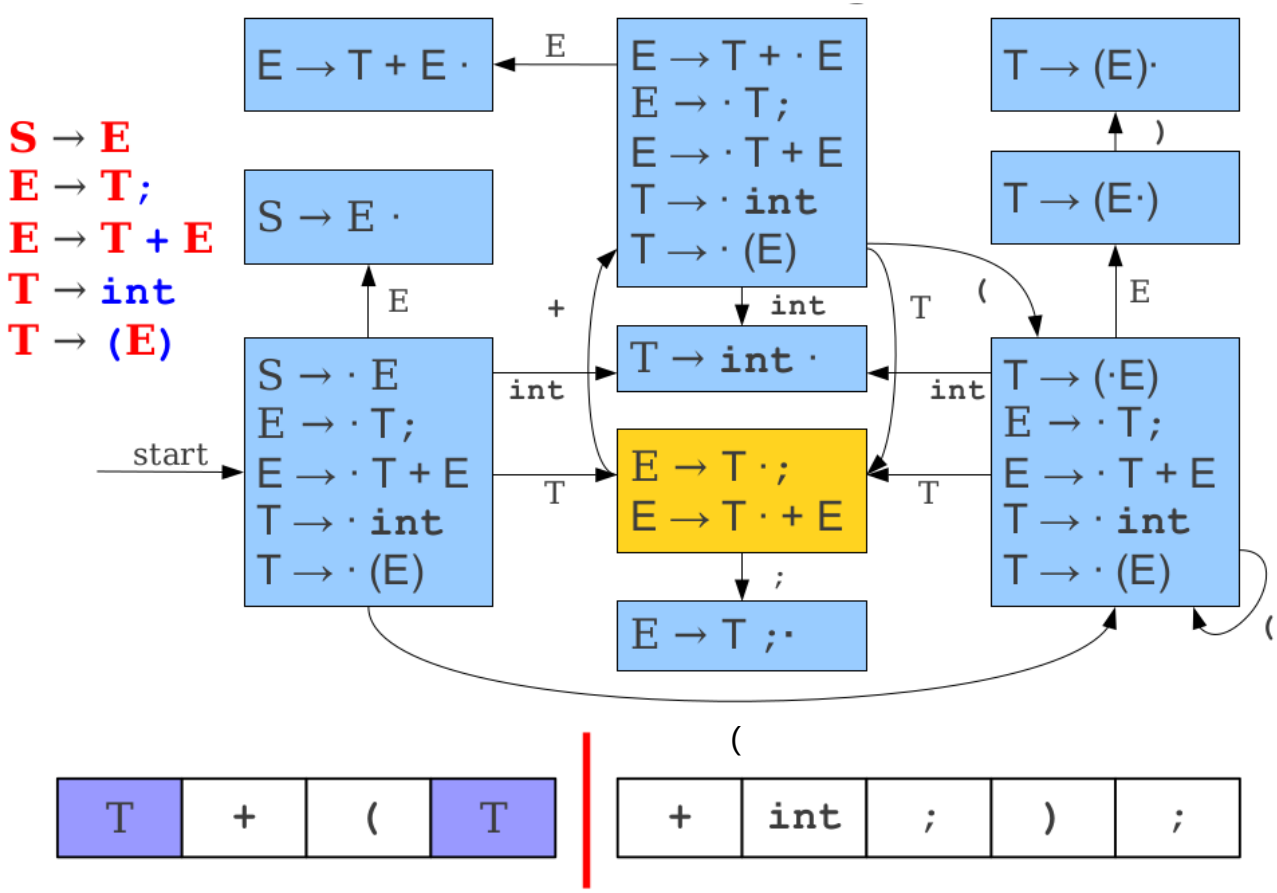
LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
E → T + E ·
S → E ·
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
T → (E)·
T → (E·)
S → · E
E → · T;
E → · T + E
T → · int
T → · (E)
T → int ·
E → T ·;
E → T · + E
E → T ;·
T → (·E)
E → · T;
E → · T + E
T → · int
T → · (E)
start
E
+
int
T
(
;
)
T
+
(
T
+
int
;
)
;

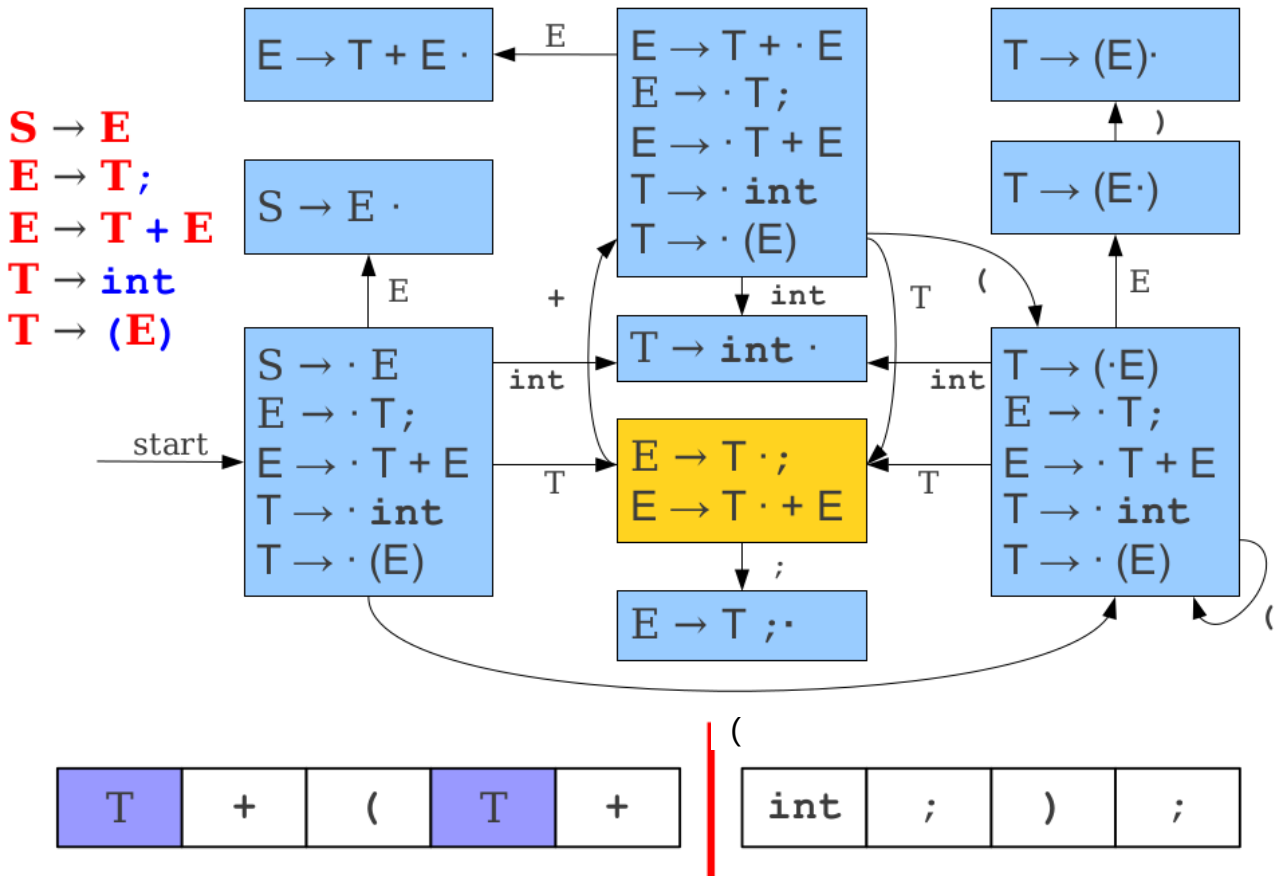
LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
E → T + E ·
S → E ·
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
T → (E)·
T → (E·)
S → · E
E → · T;
E → · T + E
T → · int
T → · (E)
T → int ·
E → T ·;
E → T · + E
E → T ;·
T → (·E)
E → · T;
E → · T + E
T → · int
T → · (E)
start
E
+
int
T
;
(
)
T
+
(
T
+
int
;
)
;

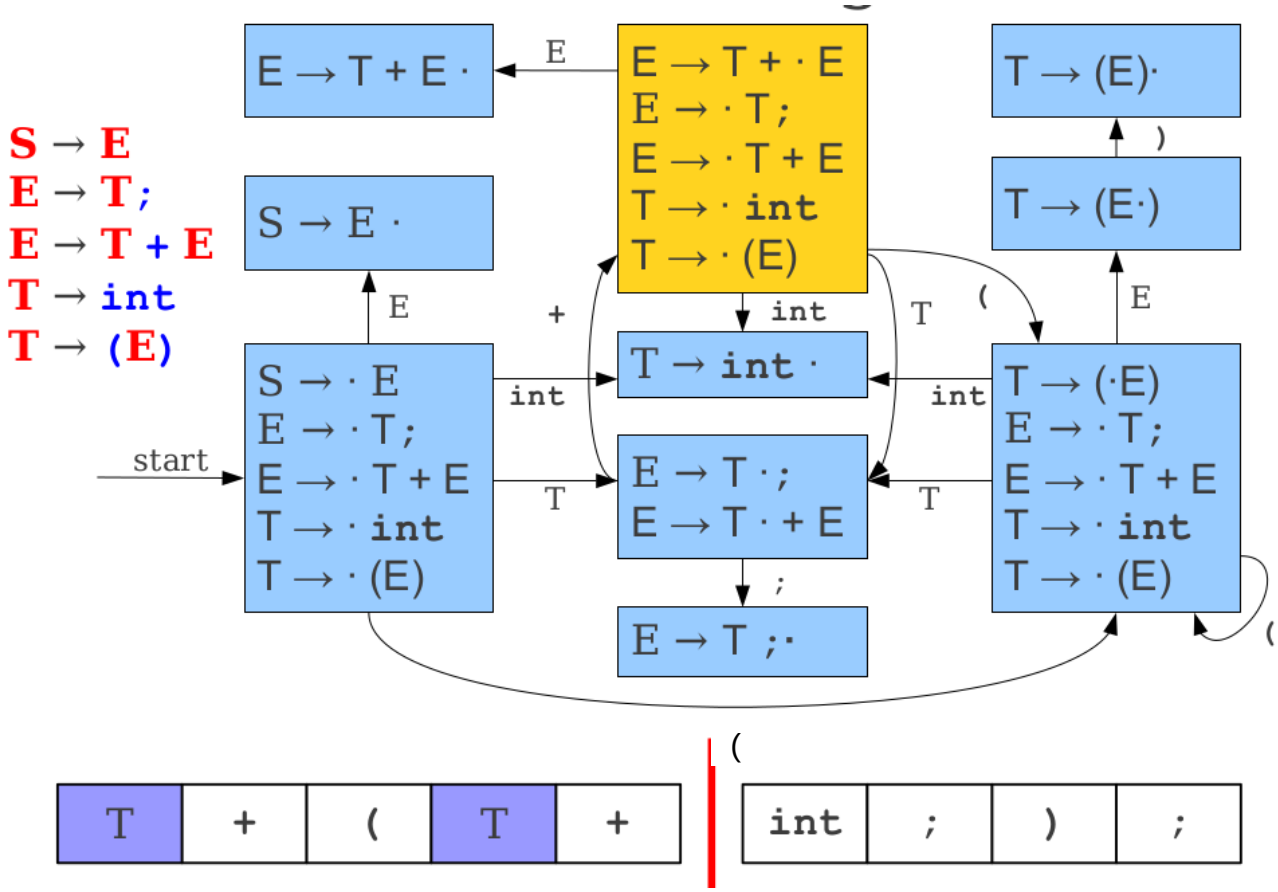
LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
E → T + E ·
S → E ·
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
T → (E)·
T → (E·)
S → · E
E → · T;
E → · T + E
T → · int
T → · (E)
T → int ·
E → T ·;
E → T · + E
E → T ;·
T → (·E)
E → · T;
E → · T + E
T → · int
T → · (E)
start
E
+
int
T
(
)
;
T
+
(
T
+
int
;
)
;

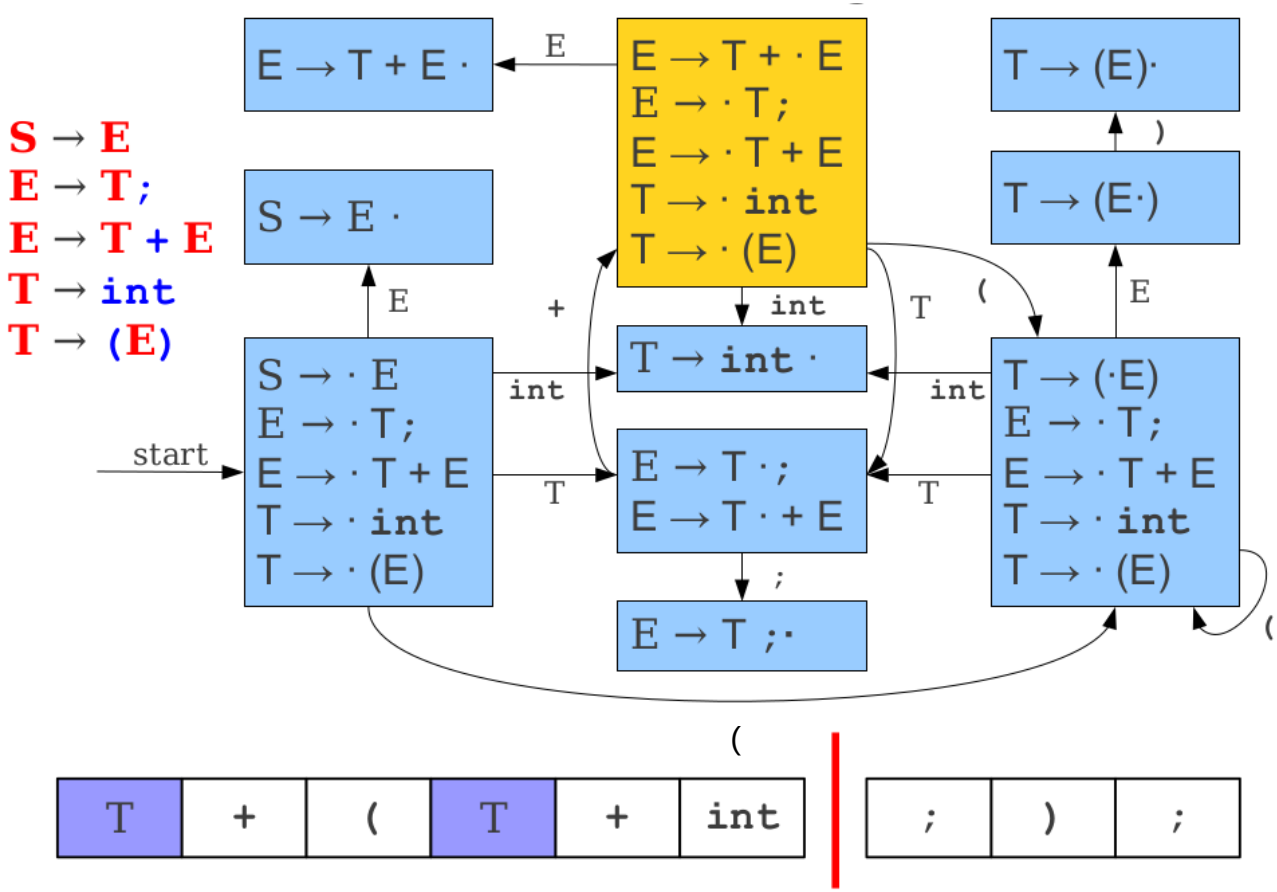
LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
E → T + E ·
S → E ·
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
T → (E)·
T → (E·)
S → · E
E → · T;
E → · T + E
T → · int
T → · (E)
T → int ·
E → T ·;
E → T · + E
E → T ;·
T → (·E)
E → · T;
E → · T + E
T → · int
T → · (E)
start
E
+
int
T
;
(
)
T
+
(
T
+
int
;
)
;

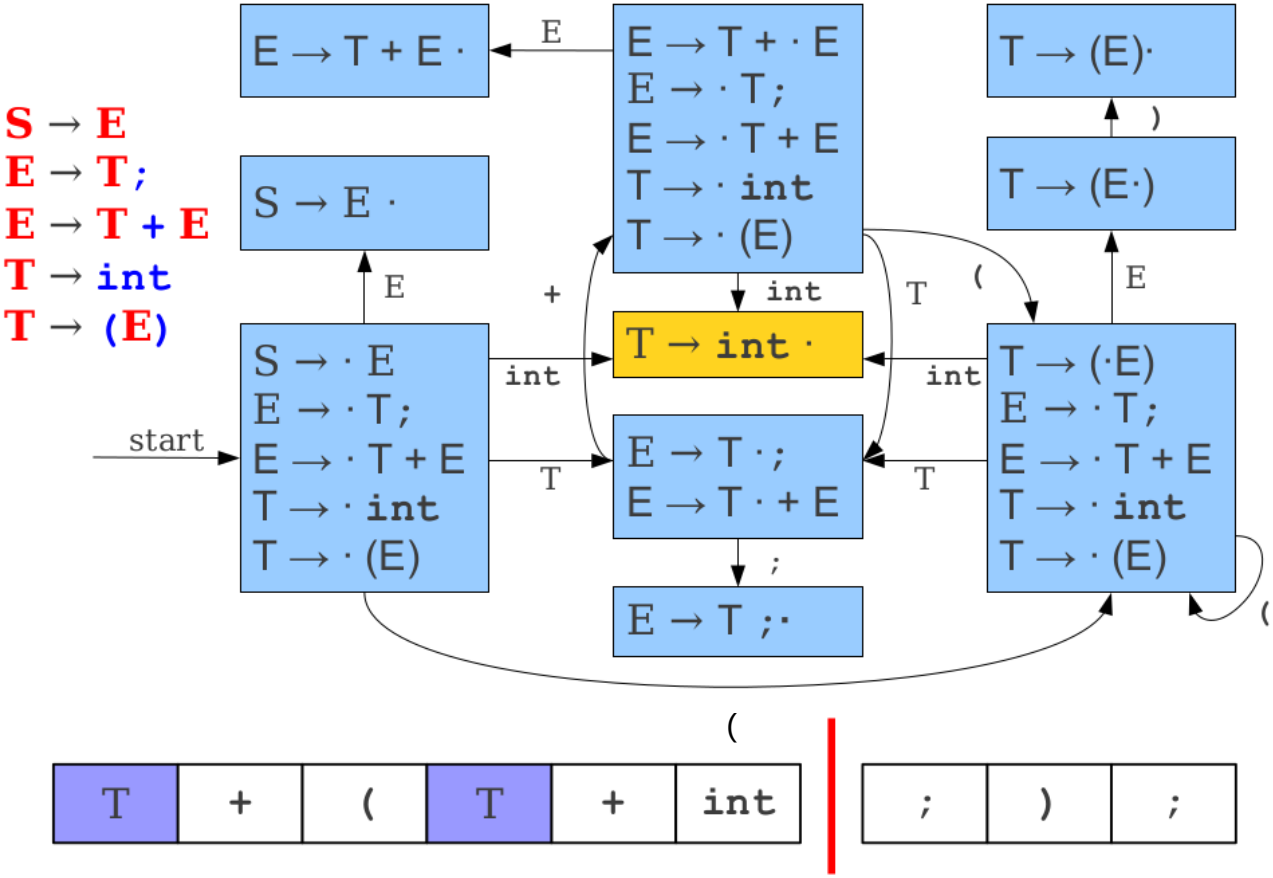
LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
E → T + E ·
S → E ·
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
T → (E)·
T → (E·)
S → · E
E → · T;
E → · T + E
T → · int
T → · (E)
T → int ·
E → T ·;
E → T · + E
E → T ;·
T → (·E)
E → · T;
E → · T + E
T → · int
T → · (E)
start
E
+
int
T
(
;
)
T
+
(
T
+
int
;
)
;

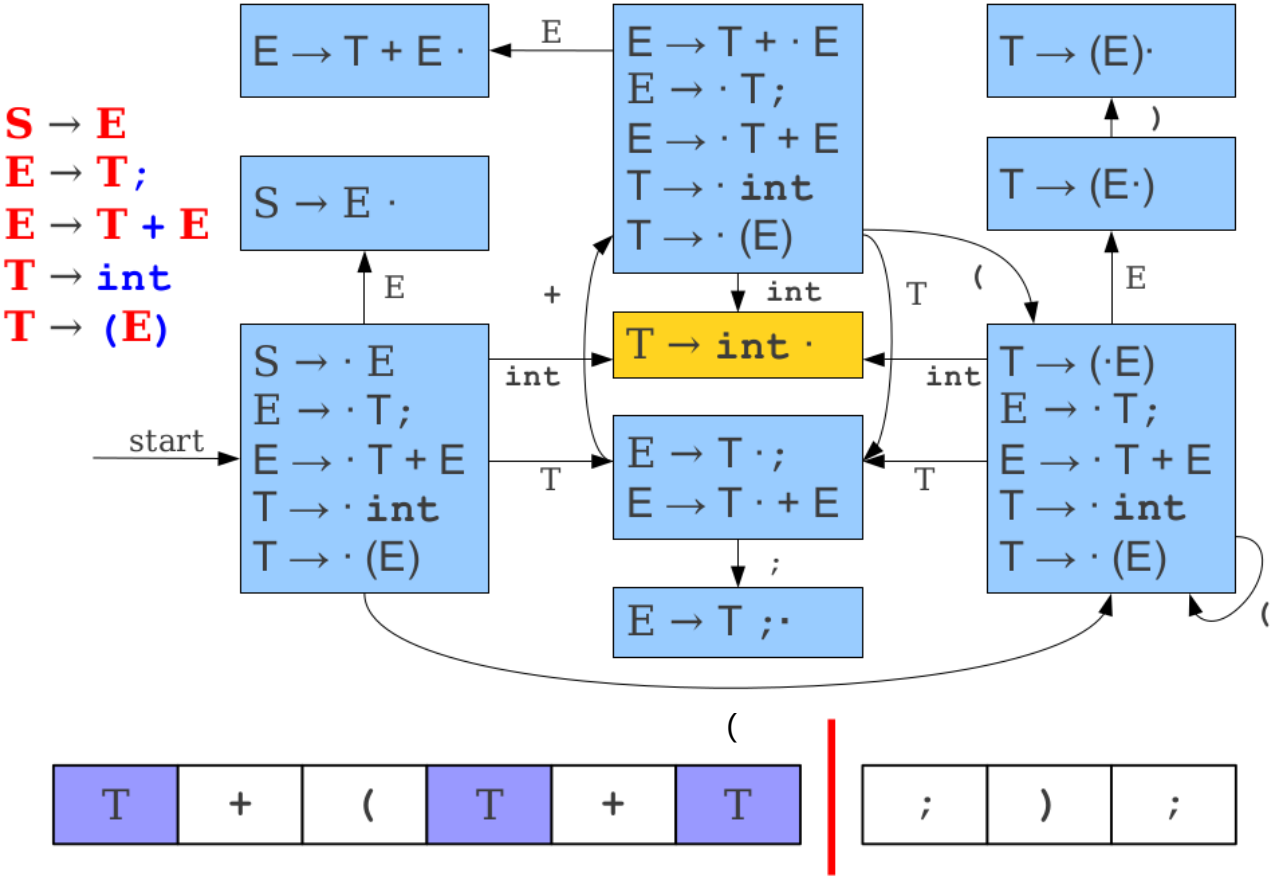
LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
E → T + E ·
S → E ·
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
T → (E)·
T → (E·)
S → · E
E → · T;
E → · T + E
T → · int
T → · (E)
T → int ·
E → T ·;
E → T · + E
E → T ;·
T → (·E)
E → · T;
E → · T + E
T → · int
T → · (E)
start
E
+
int
T
(
)
;
T
+
(
T
+
int
;
)
;

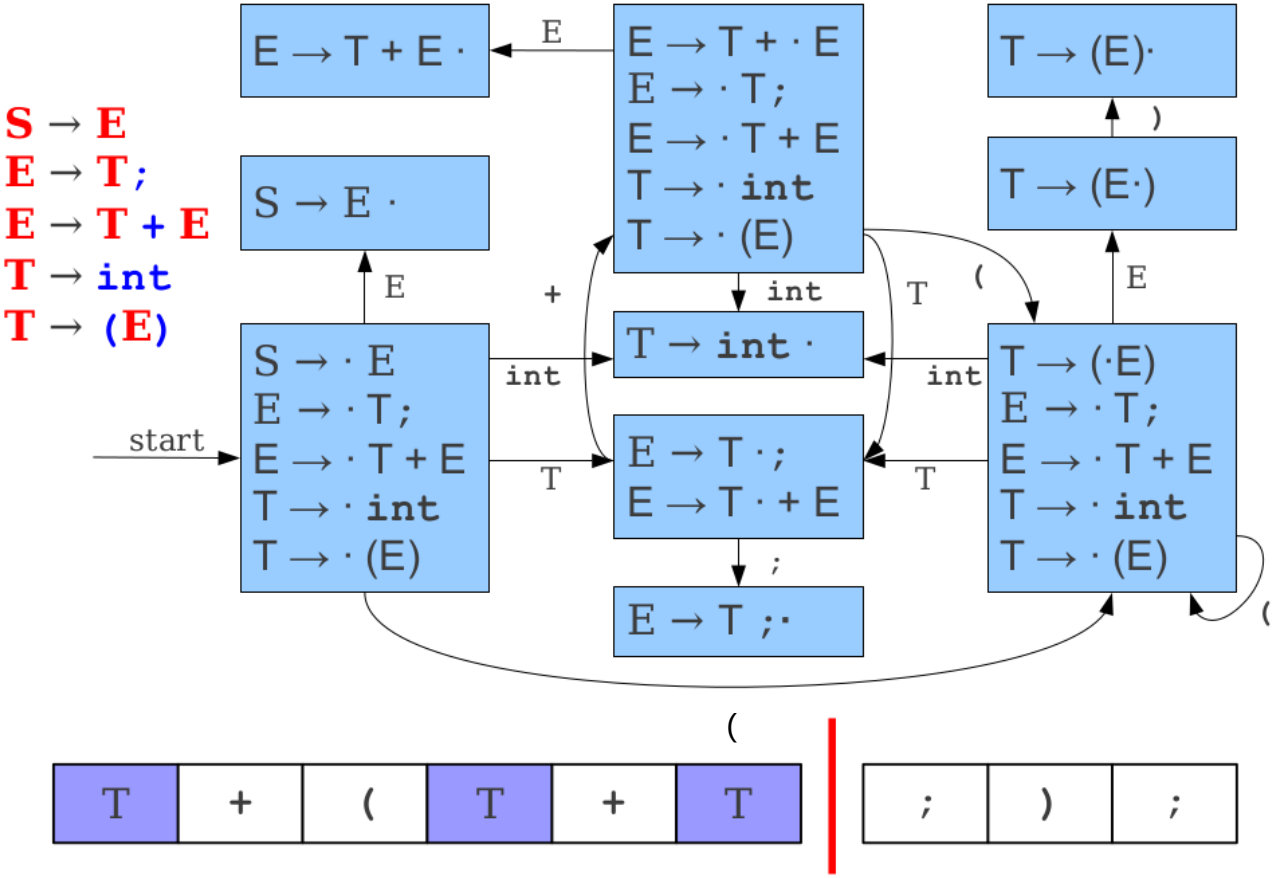
LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
E → T + E ·
S → E ·
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
T → (E)·
T → (E·)
T → int ·
E → T ·;
E → T · + E
E → T ;·
T → (·E)
E → · T;
E → · T + E
T → · int
T → · (E)
S → · E
E → · T;
E → · T + E
T → · int
T → · (E)
start
E
+
int
T
(
)
;
T
+
(
T
+
int
;
)
;

LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
E → T + E ·
S → E ·
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
T → (E)·
T → (E·)
T → int ·
E → T ·;
E → T · + E
E → T ;·
T → (·E)
E → · T;
E → · T + E
T → · int
T → · (E)
S → · E
E → · T;
E → · T + E
T → · int
T → · (E)
start
E
+
int
T
(
)
;
T
+
(
T
+
T
;
)
;

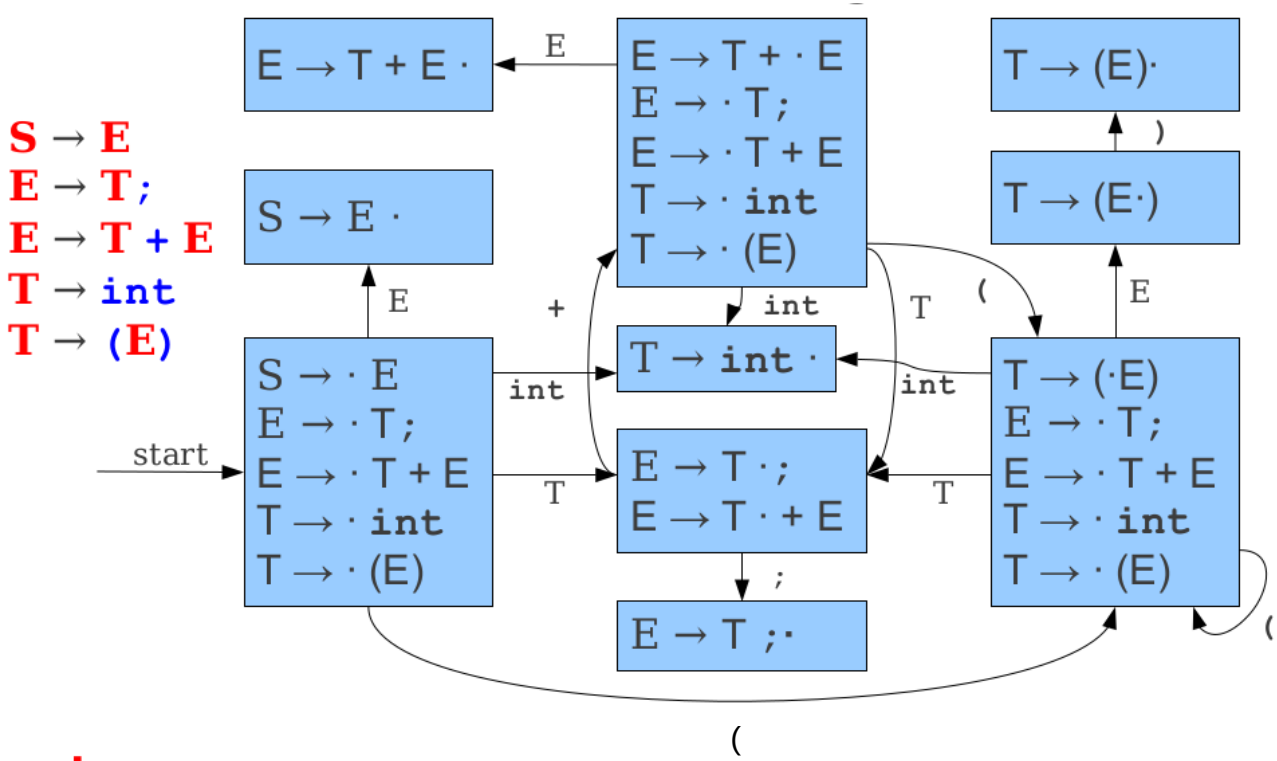
LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
E → T + E ·
S → E ·
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
T → (E)·
T → (E·)
S → · E
E → · T;
E → · T + E
T → · int
T → · (E)
T → int ·
E → T ·;
E → T · + E
E → T ;·
T → (·E)
E → · T;
E → · T + E
T → · int
T → · (E)
start
E
+
int
T
(
)
;
T
+
(
T
+
T
;
)
;

An Optimization

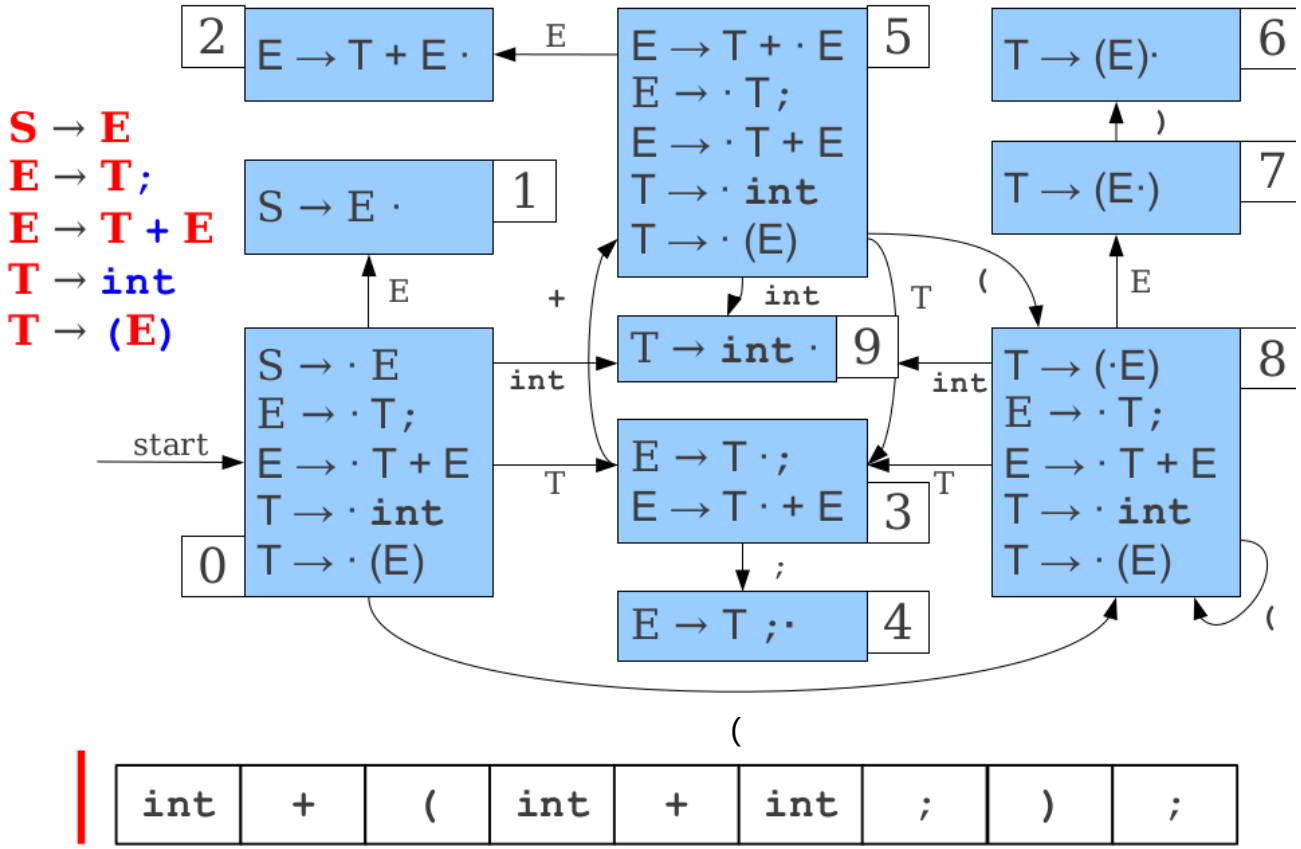
- Rather than restart the automaton on each reduction, remember what state we were in for each symbol.
- When applying a reduction, restart the automaton from the last known good state.

LR(0) Parsing



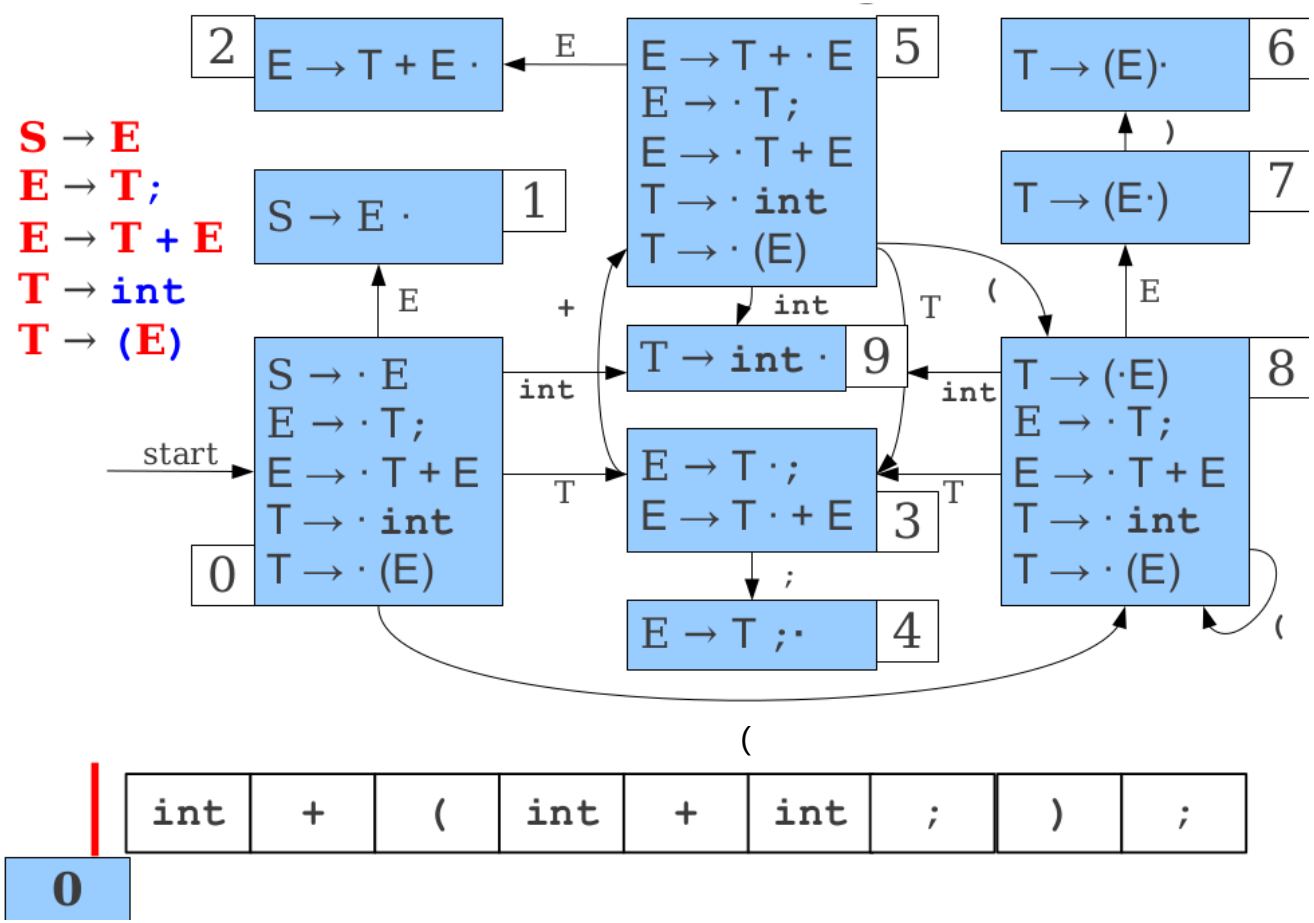
int	+	(	int	+	int	;	)	;

LR(0) Parsing



int	+	(	int	+	int	;	)	;

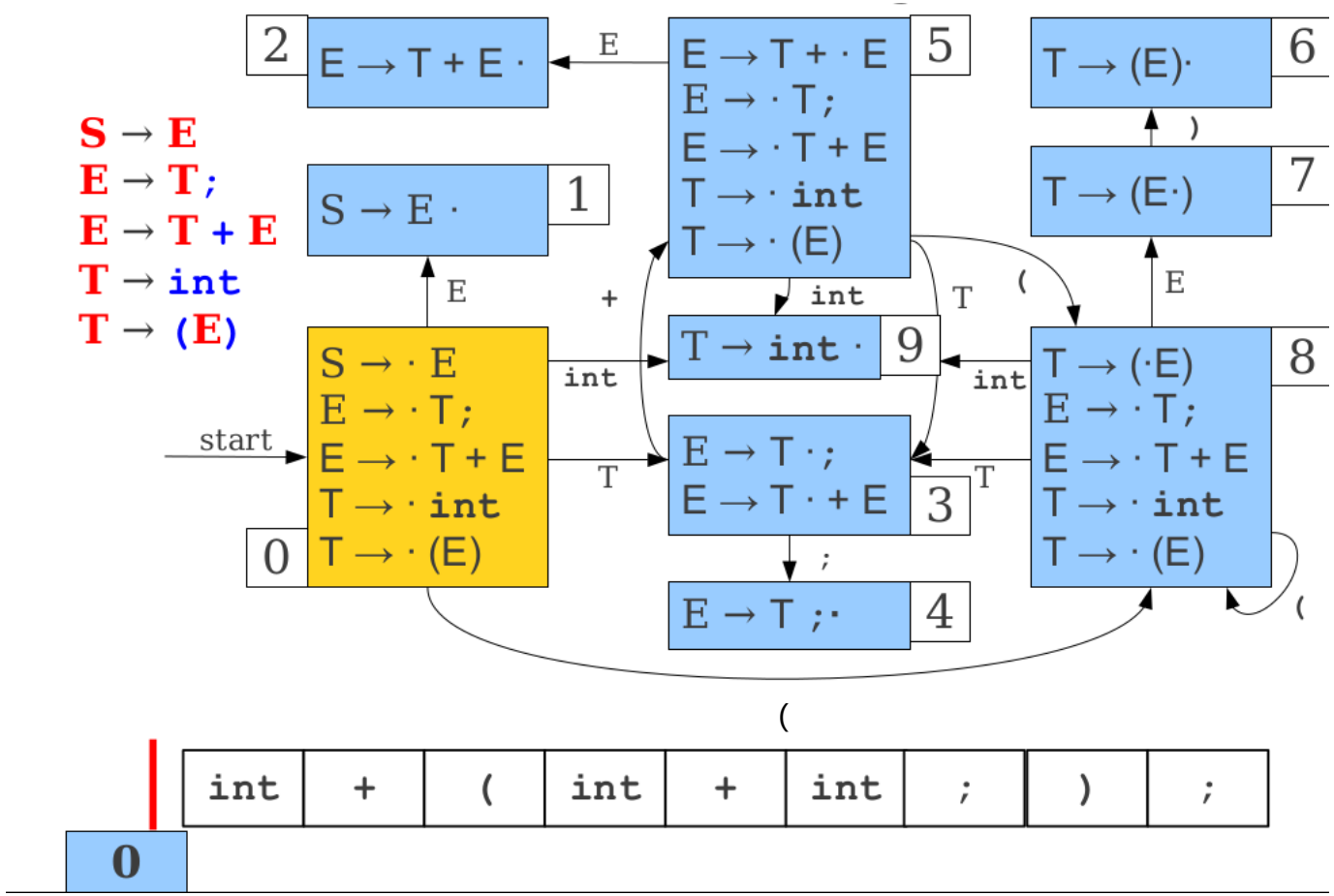
LR(0) Parsing



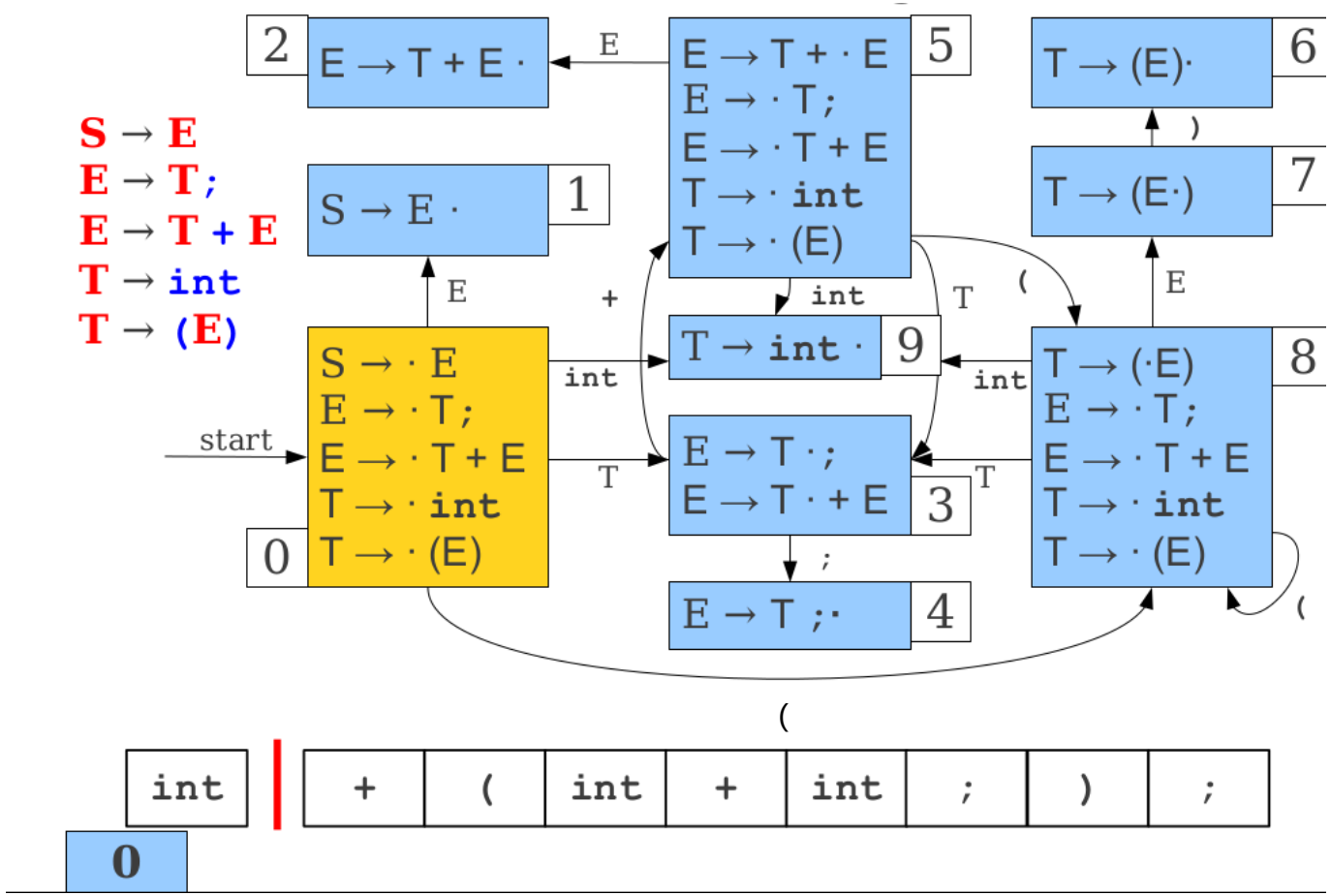
int	+	(	int	+	int	;	)	;

0

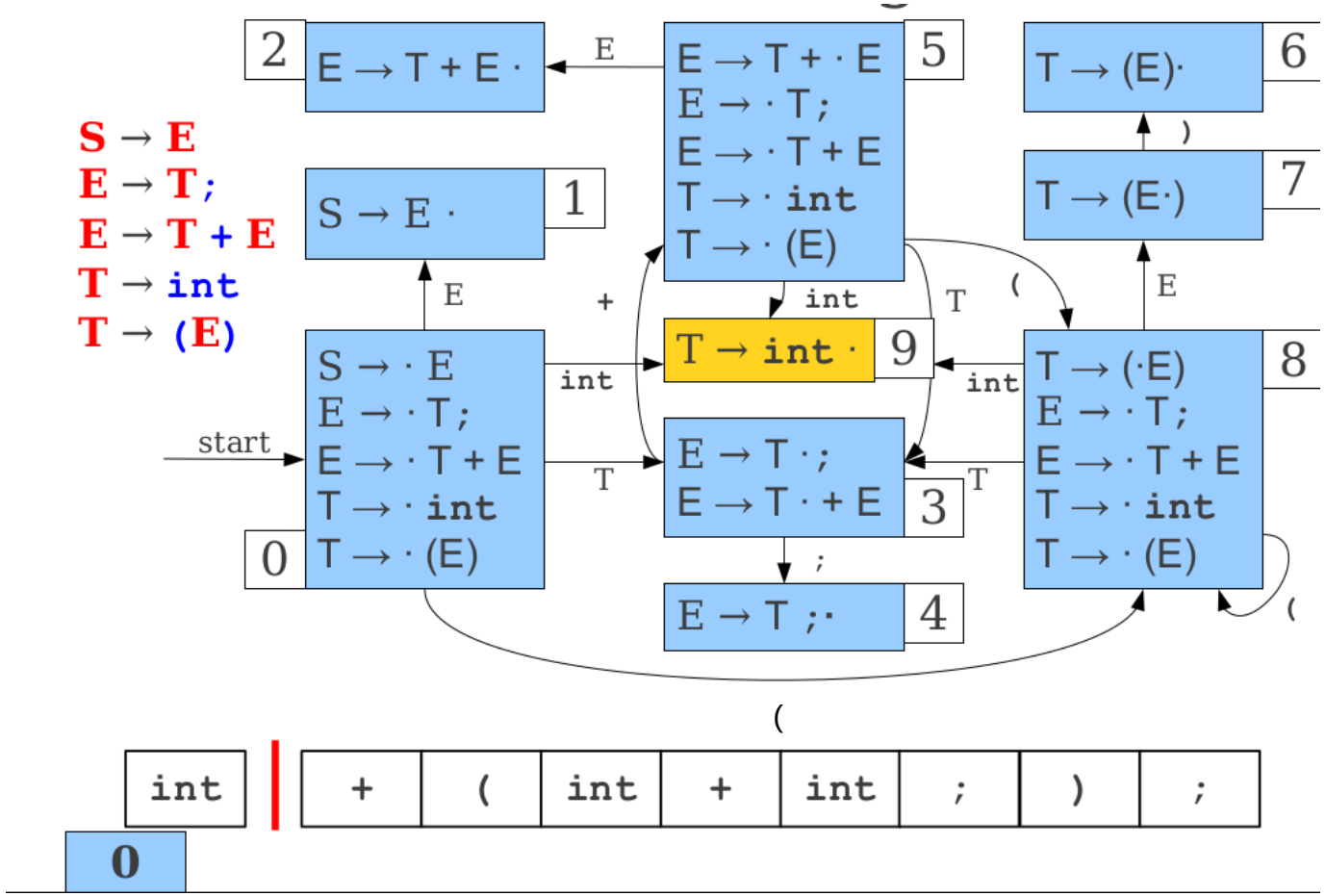
LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
start
0
S → · E
E → · T;
E → · T + E
T → · int
T → · (E)
1
S → E ·
2
E → T + E ·
3
E → T · ;
E → T · + E
4
E → T ; ·
5
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
6
T → (E)·
7
T → (E·)
8
T → (·E)
E → · T;
E → · T + E
T → · int
T → · (E)
9
T → int ·
E
+
int
T
(
)
;
int + (int + int ;) ;
0

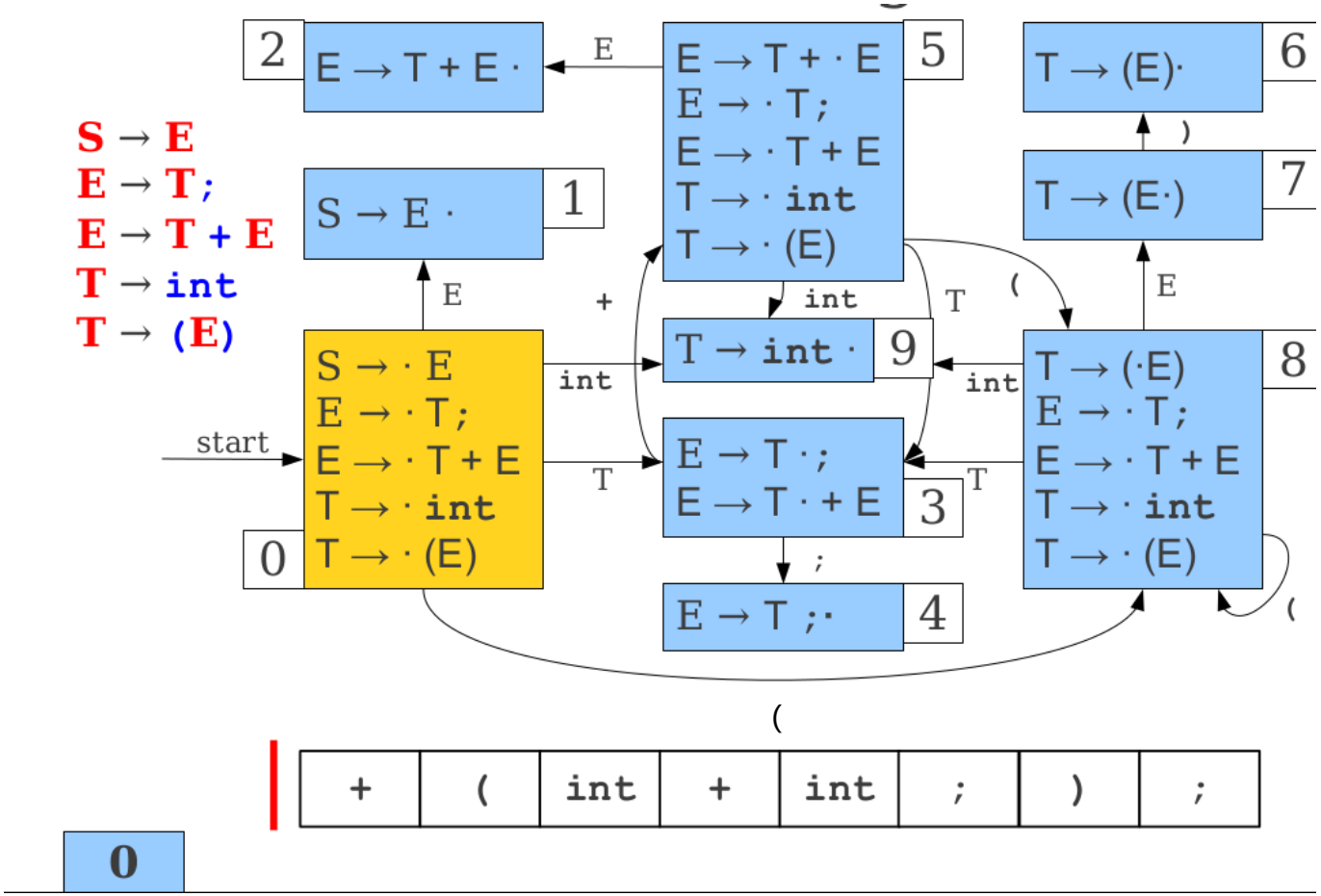
LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
start
0
S → · E
E → · T;
E → · T + E
T → · int
T → · (E)
1
S → E ·
2
E → T + E ·
3
E → T · ;
E → T · + E
4
E → T ; ·
5
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
6
T → (E)·
7
T → (E·)
8
T → (·E)
E → · T;
E → · T + E
T → · int
T → · (E)
9
T → int ·
E
+
int
T
;
(
)
int
+
(
int
+
int
;
)
;
0

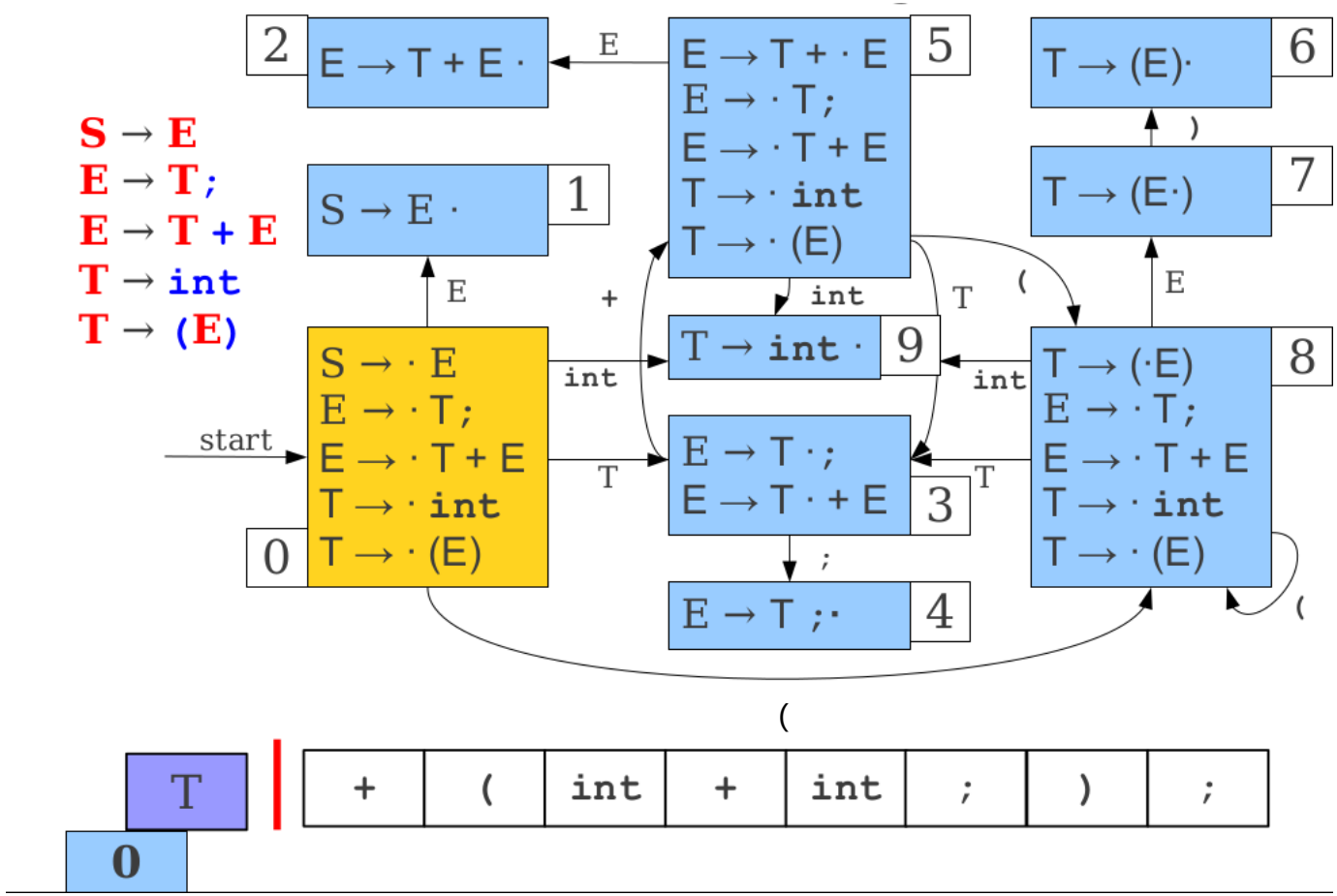
LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
2
E → T + E ·
E
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
5
T → (E)·
6
)
S → E ·
1
T → (E·)
7
E
+
int
T
(
E
T → int ·
9
int
T → (·E)
E → · T;
E → · T + E
T → · int
T → · (E)
8
start
S → · E
E → · T;
E → · T + E
T → · int
T → · (E)
0
int
T
E → T ·;
E → T · + E
3
T
;
E → T ;·
4
(
(
int
+
(
int
+
int
;
)
;
0

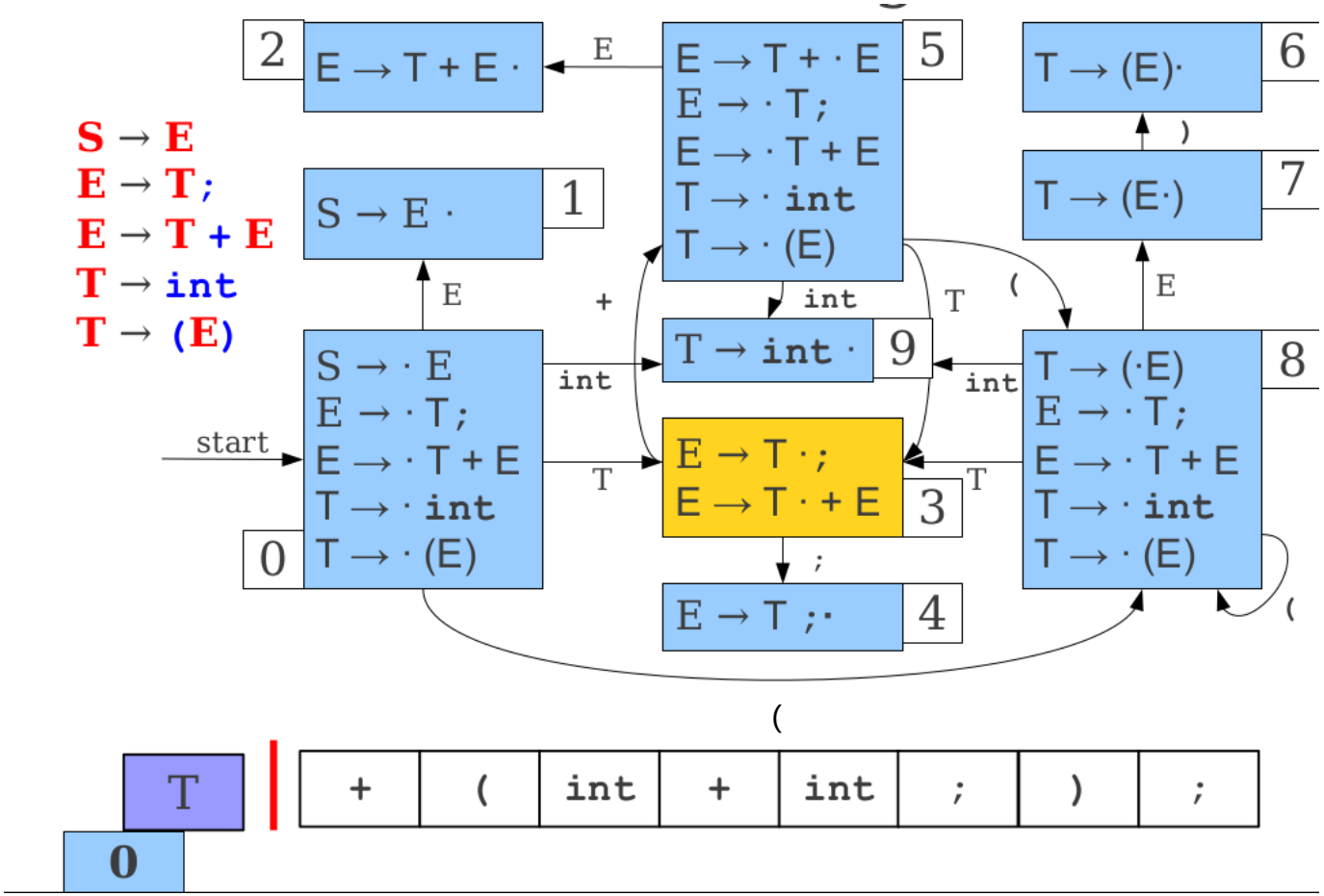
LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
start
0
S → · E
E → · T;
E → · T + E
T → · int
T → · (E)
1
S → E ·
2
E → T + E ·
3
E → T · ;
E → T · + E
4
E → T ; ·
5
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
6
T → (E)·
7
T → (E·)
8
T → (·E)
E → · T;
E → · T + E
T → · int
T → · (E)
9
T → int ·
E
+
int
T
;
(
)
+ (int + int ;) ;
0

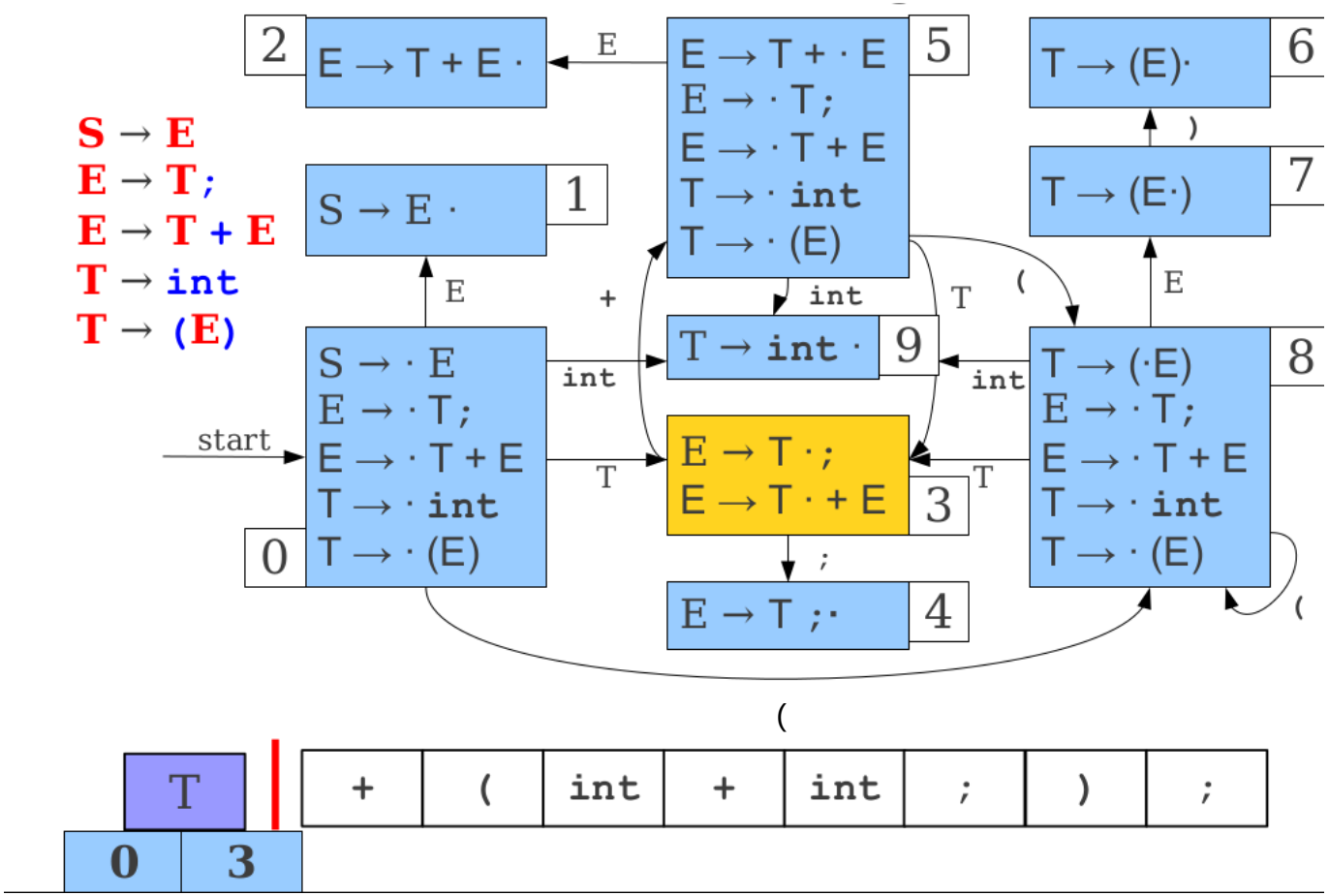
LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
2
E → T + E ·
E
5
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
6
T → (E)·
)
7
T → (E·)
1
S → E ·
E
+
int
T
(
E
9
T → int ·
int
8
T → (·E)
E → · T;
E → · T + E
T → · int
T → · (E)
start
S → · E
E → · T;
E → · T + E
T → · int
T → · (E)
0
int
T
E → T ·;
E → T · + E
3
T
;
4
E → T ;·
(
(
T
0
+
(
int
+
int
;
)
;

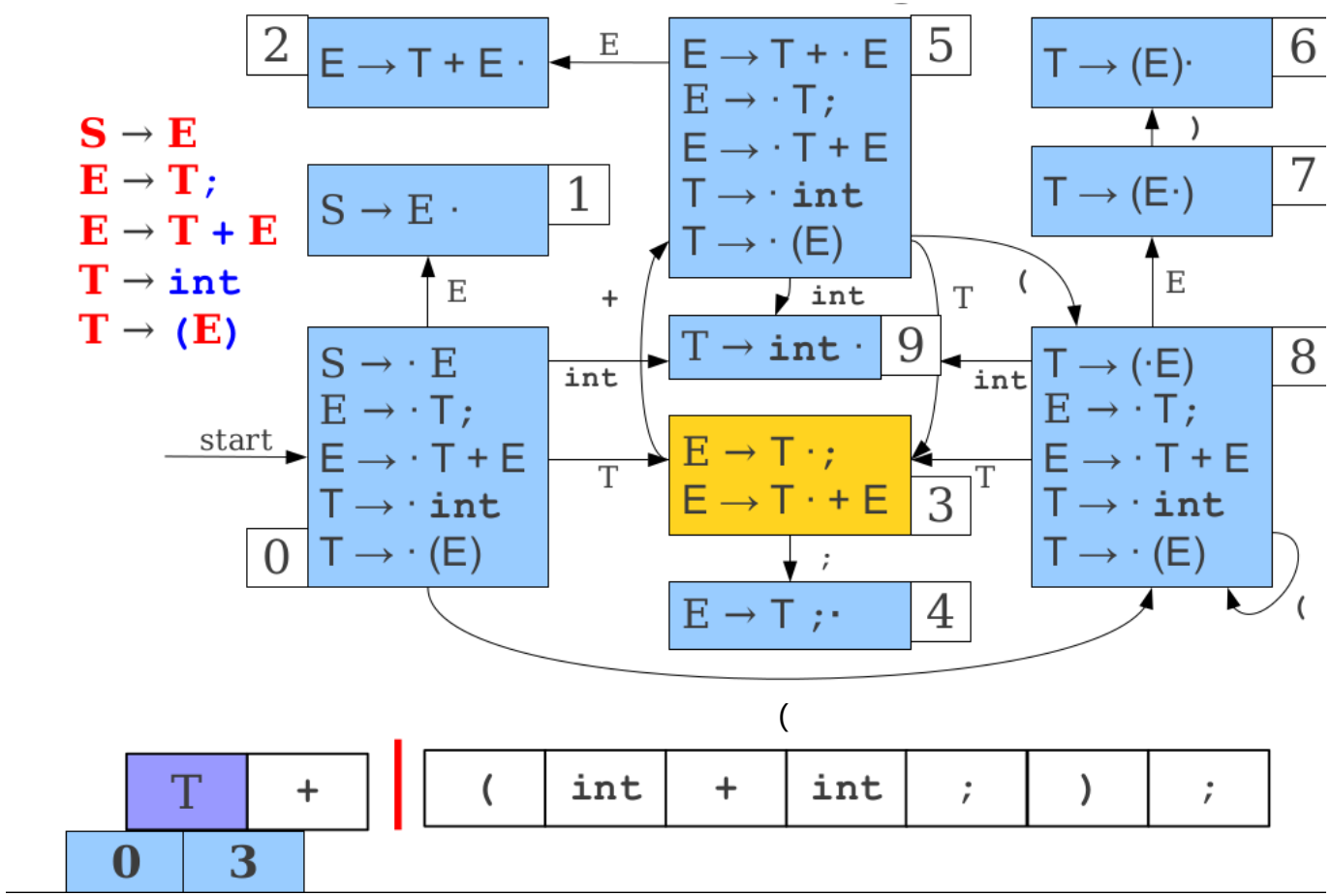
LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
2
E → T + E ·
E
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
5
T → (E)·
6
)
T → (E·)
7
S → E ·
1
E
+
int
T
(
E
S → · E
E → · T;
E → · T + E
T → · int
T → · (E)
0
start
int
T → int ·
9
int
T → (·E)
E → · T;
E → · T + E
T → · int
T → · (E)
8
E → T ·;
E → T · + E
3
T
;
E → T ;·
4
(
T
0
+
(
int
+
int
;
)
;

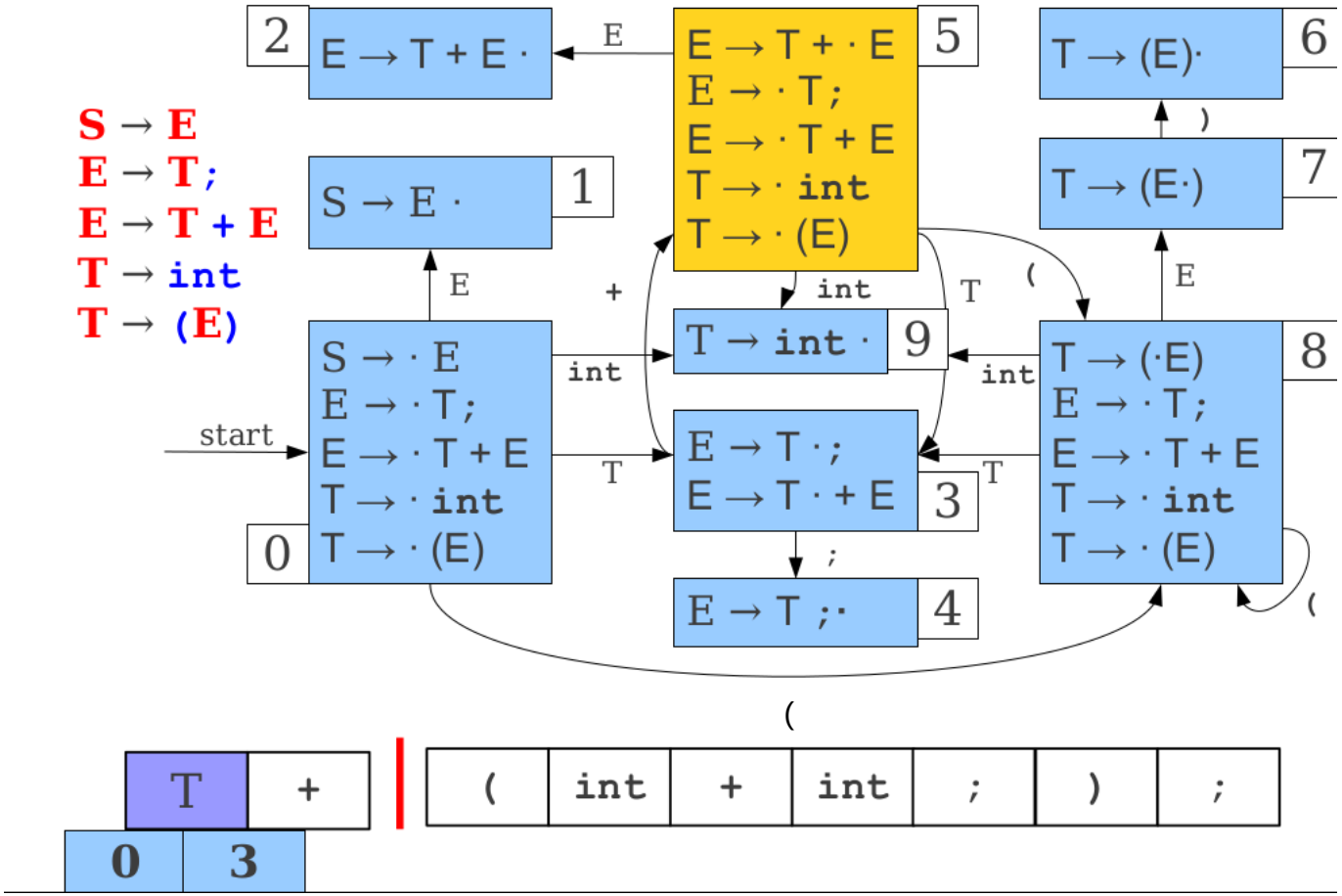
LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
start
0
S → · E
E → · T;
E → · T + E
T → · int
T → · (E)
1
S → E ·
2
E → T + E ·
3
E → T · ;
E → T · + E
4
E → T ; ·
5
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
6
T → (E)·
7
T → (E·)
8
T → (·E)
E → · T;
E → · T + E
T → · int
T → · (E)
9
T → int ·
E
+
int
T
;
(
)
T
0
3
+
(
int
+
int
;
)
;

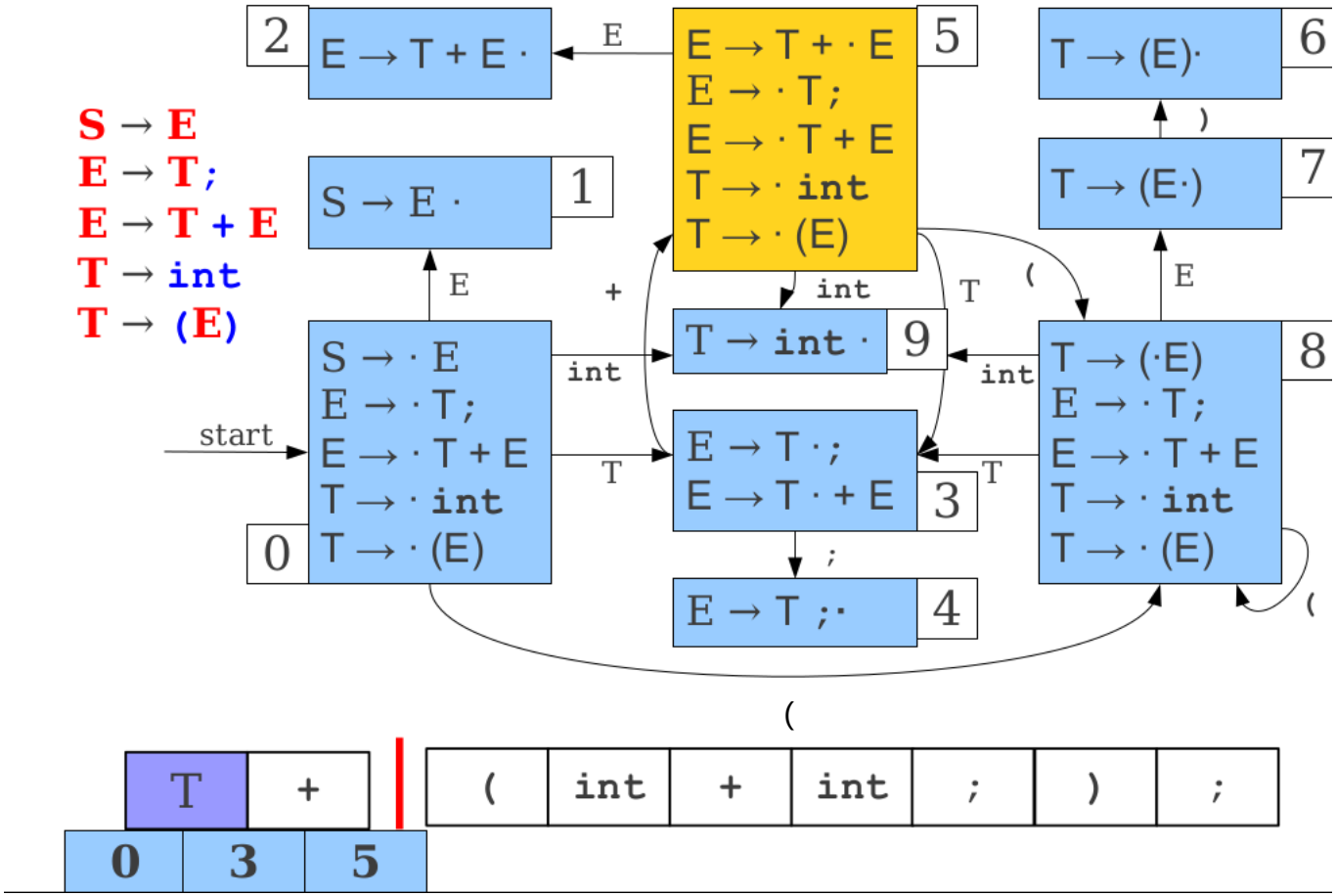
LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
2
E → T + E ·
E
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
5
T → (E)·
6
)
T → (E·)
7
S → E ·
1
E
+
int
T
(
E
T → int ·
9
int
T → (·E)
E → · T;
E → · T + E
T → · int
T → · (E)
8
start
S → · E
E → · T;
E → · T + E
T → · int
T → · (E)
0
int
T
E → T ·;
E → T · + E
3
T
;
E → T ;·
4
(
(
T
+
0
3
(
int
+
int
;
)
;

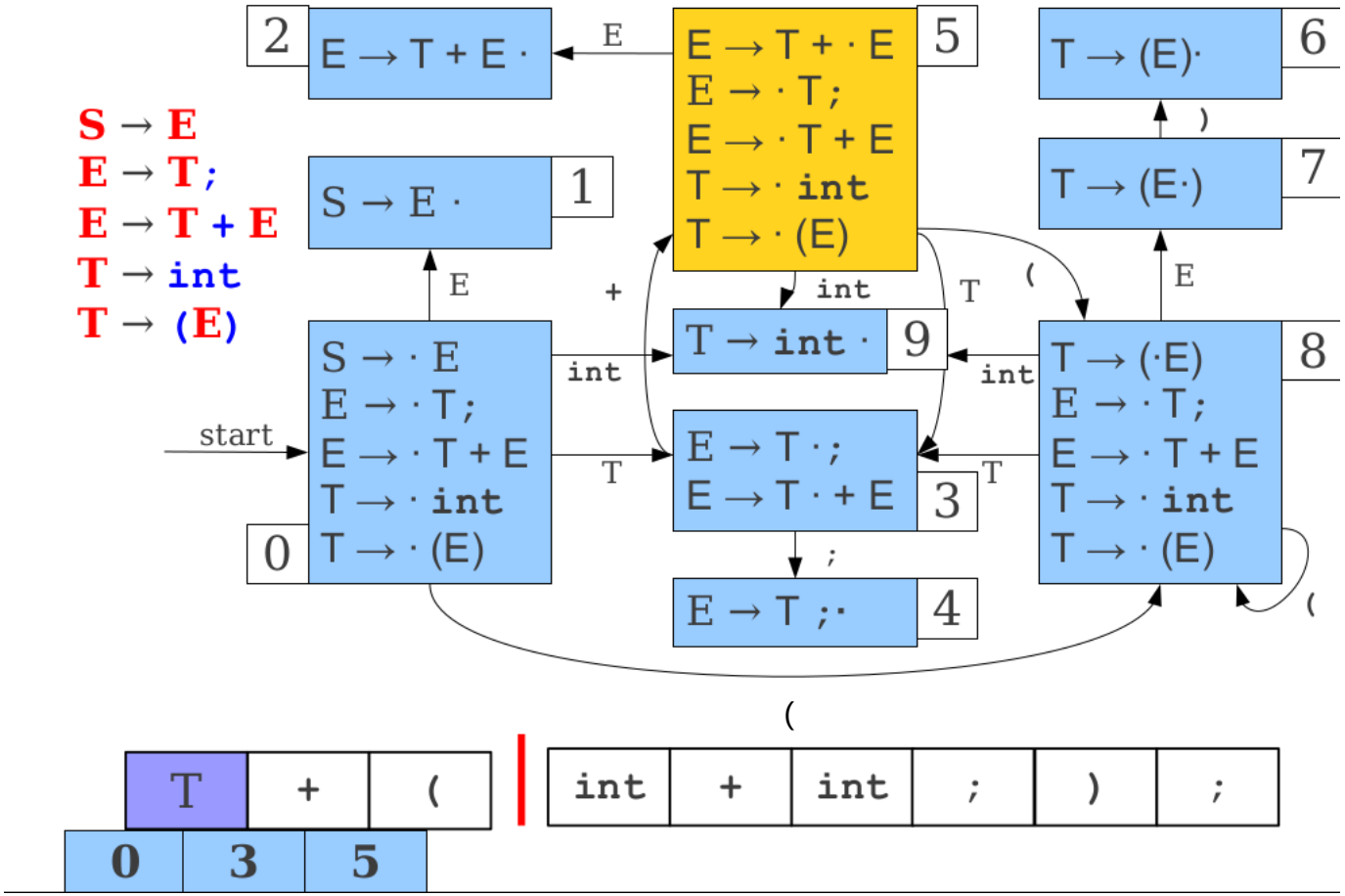
LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
2
E → T + E ·
E
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
5
T → (E)·
6
)
T → (E·)
7
S → E ·
1
E
+
int
T
(
E
T → int ·
9
int
T → (·E)
E → · T;
E → · T + E
T → · int
T → · (E)
8
start
S → · E
E → · T;
E → · T + E
T → · int
T → · (E)
0
int
T
E → T ·;
E → T · + E
3
T
;
E → T ;·
4
(
(
T
+
0
3
(
int
+
int
;
)
;

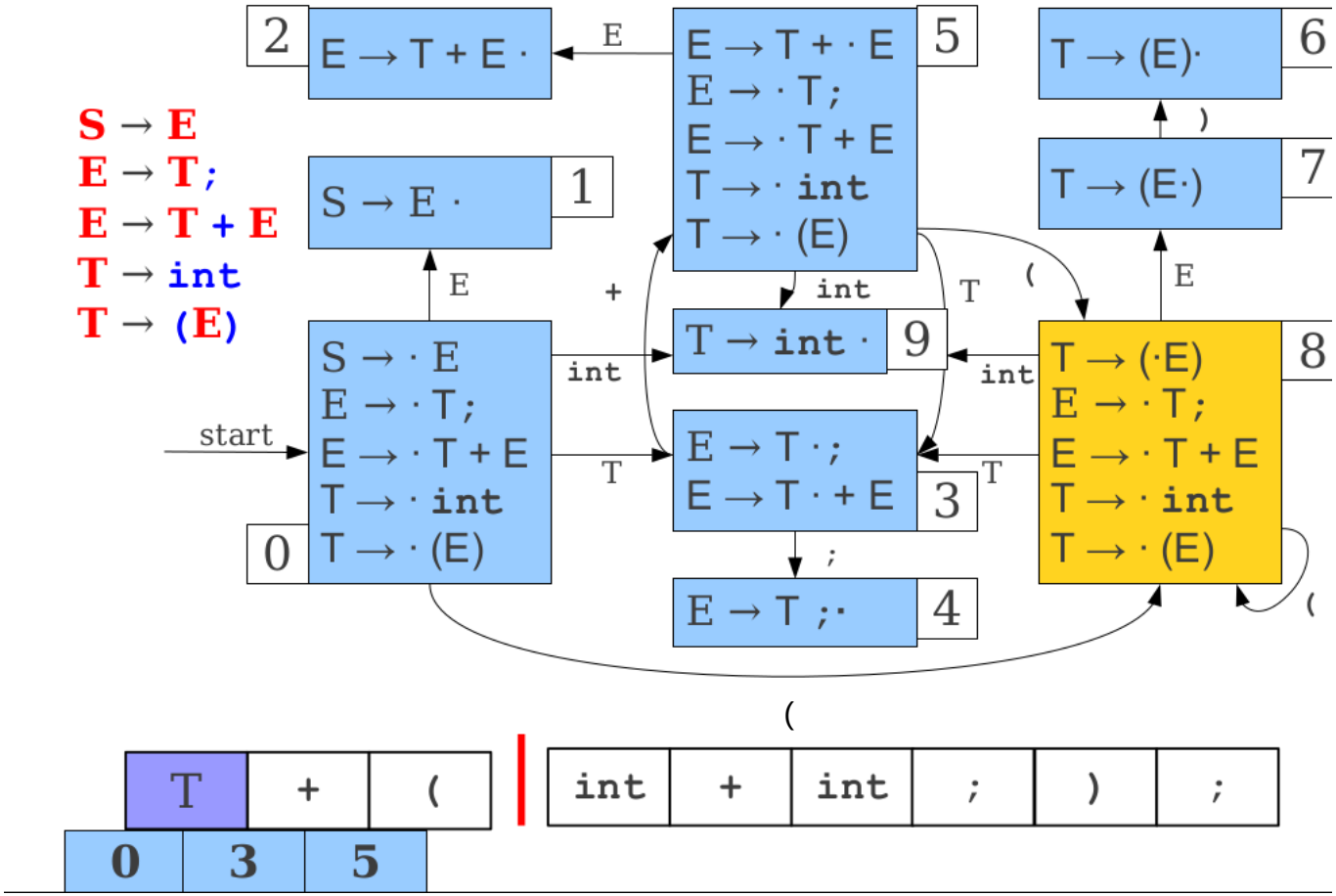
LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
2
E → T + E ·
E
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
5
T → (E)·
6
)
T → (E·)
7
S → E ·
1
E
+
int
T
(
E
S → · E
E → · T;
E → · T + E
T → · int
T → · (E)
0
start
int
T → int ·
9
int
T → (·E)
E → · T;
E → · T + E
T → · int
T → · (E)
8
E → T ·;
E → T · + E
3
T
;
E → T ;·
4
(
T
+
0
3
5
(
int
+
int
;
)
;

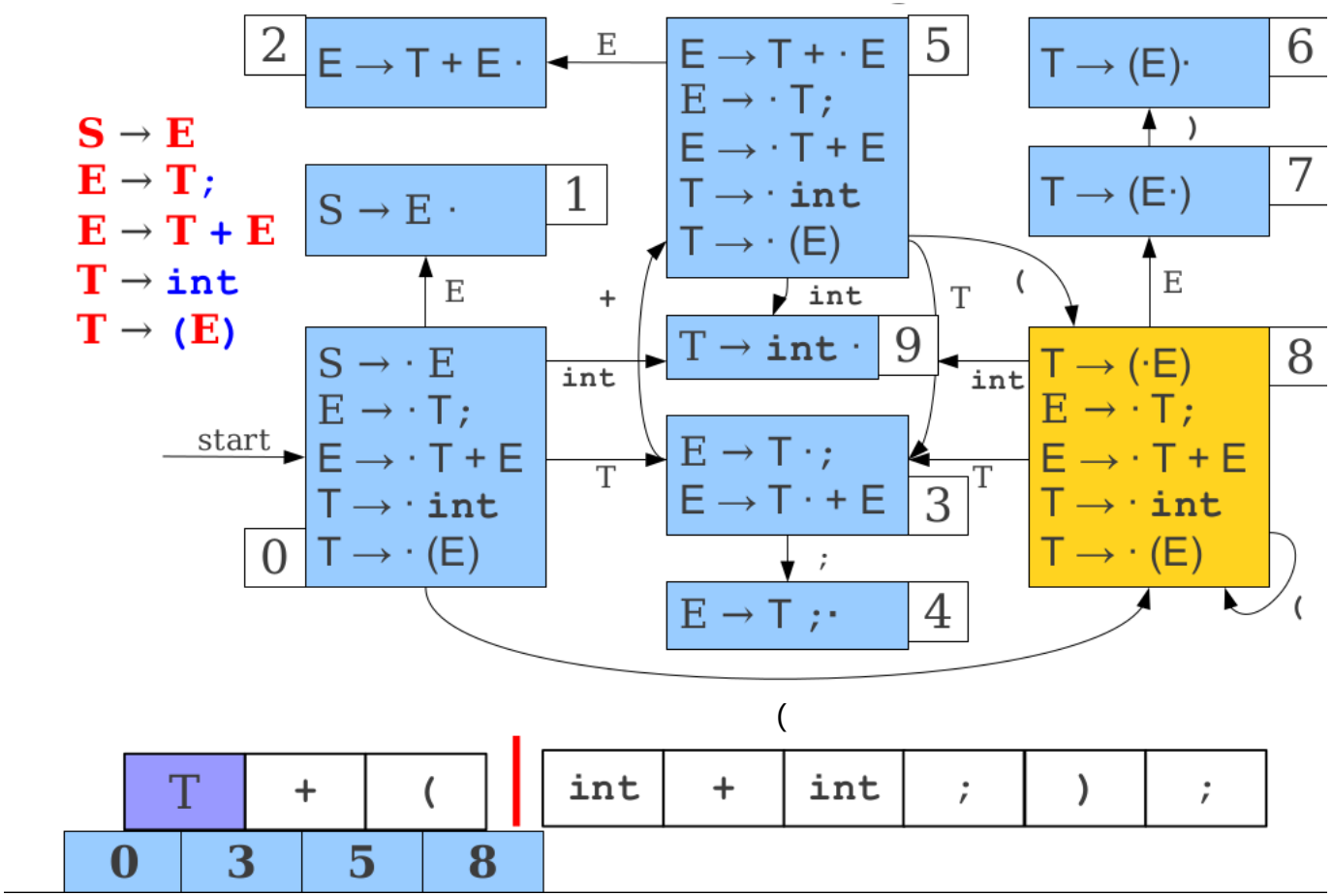
LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
2
E → T + E ·
E
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
5
T → (E)·
6
)
T → (E·)
7
S → E ·
1
E
+
int
T
(
E
S → · E
E → · T;
E → · T + E
T → · int
T → · (E)
0
start
int
T → int ·
9
int
T → (·E)
E → · T;
E → · T + E
T → · int
T → · (E)
8
E → T ·;
E → T · + E
3
T
T
;
E → T ;·
4
(
(
T
+
(
0
3
5
int
+
int
;
)
;

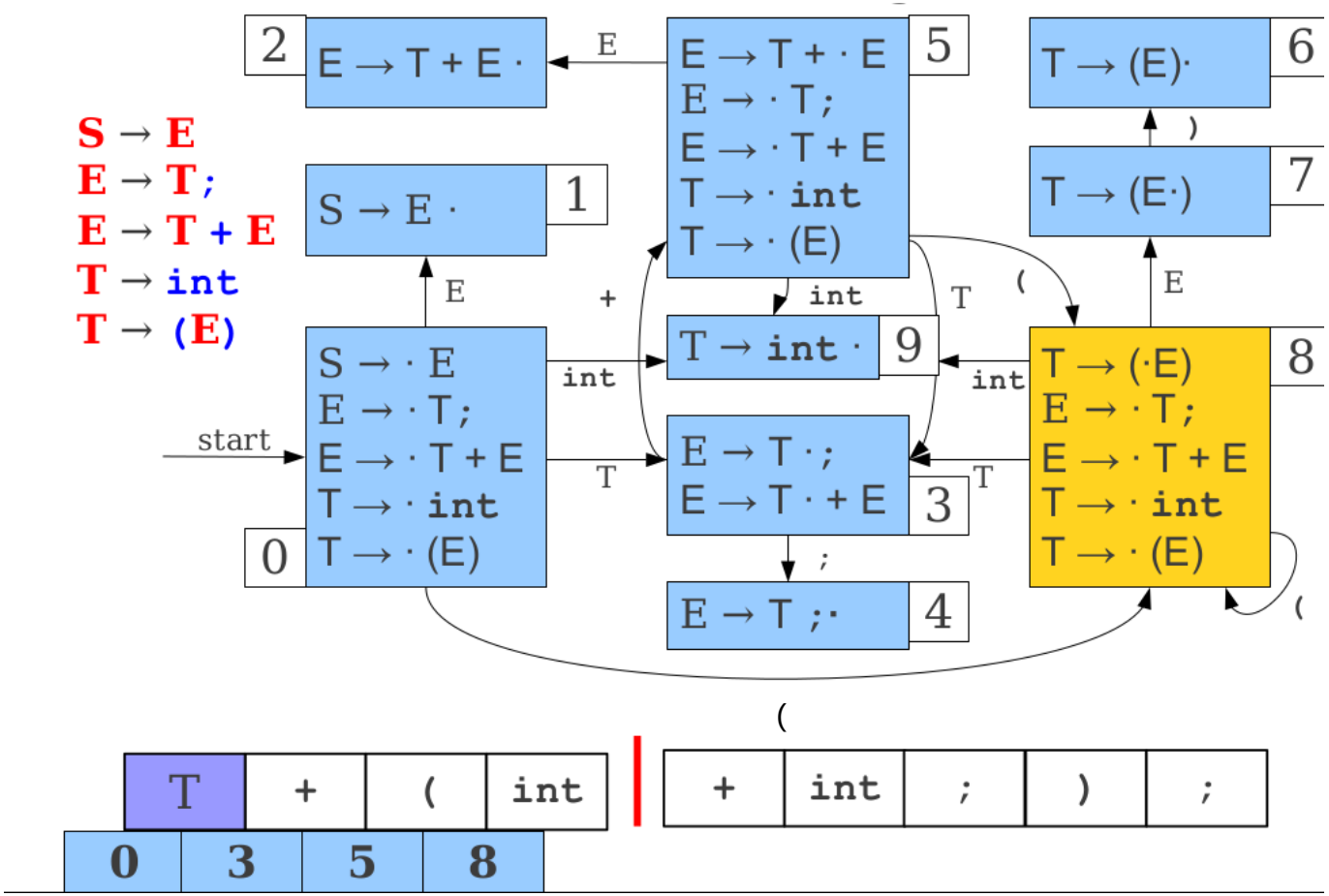
LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
start
0
S → · E
E → · T;
E → · T + E
T → · int
T → · (E)
1
S → E ·
2
E → T + E ·
3
E → T · ;
E → T · + E
4
E → T ; ·
5
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
6
T → (E) ·
7
T → (E ·)
8
T → (· E)
E → · T;
E → · T + E
T → · int
T → · (E)
9
T → int ·
E
+
int
T
;
(
)
T
+
(
0
3
5
int
+
int
;
)
;

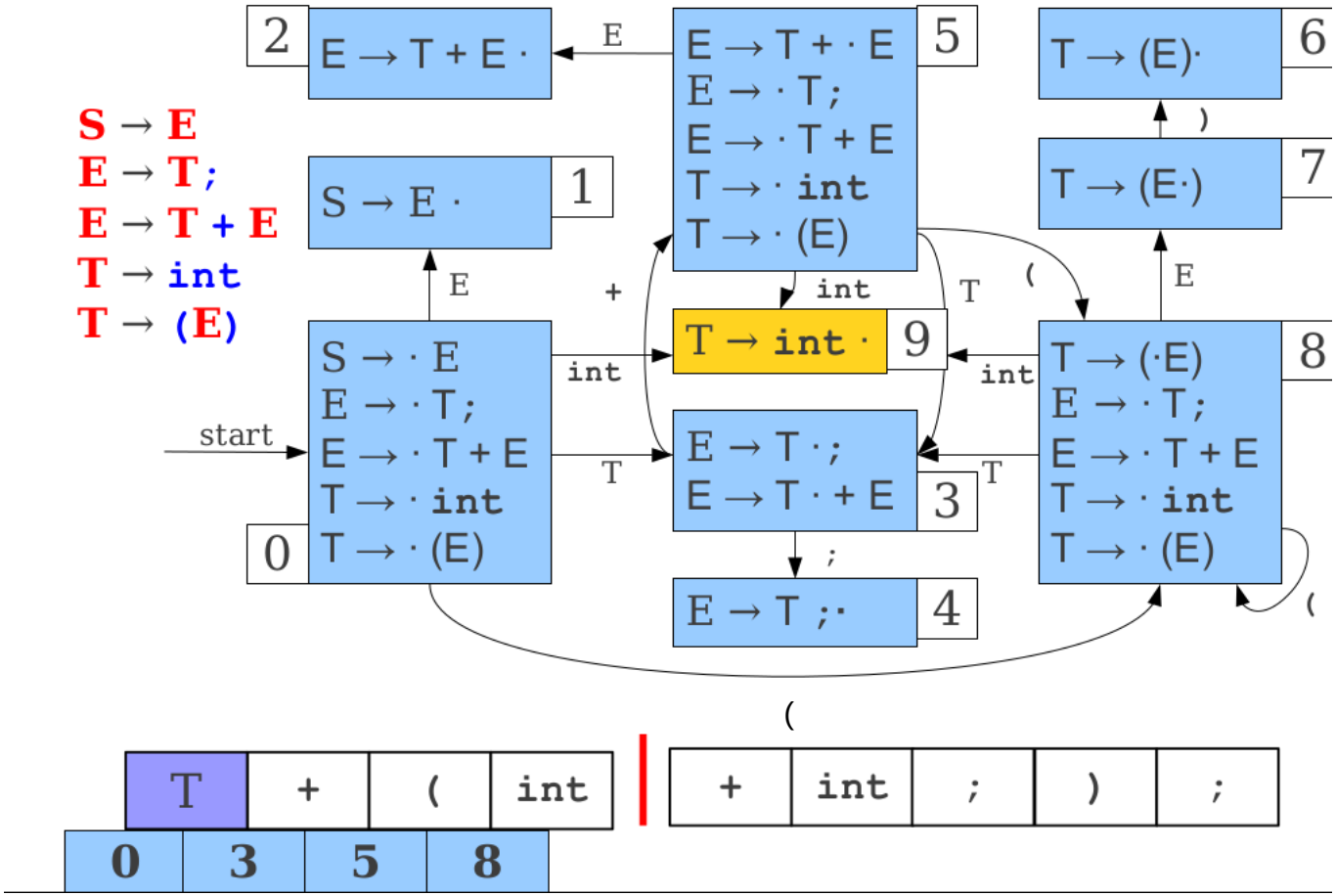
LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
start
0
S → · E
E → · T;
E → · T + E
T → · int
T → · (E)
1
S → E ·
2
E → T + E ·
3
E → T · ;
E → T · + E
4
E → T ; ·
5
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
6
T → (E) ·
7
T → (E ·)
8
T → (· E)
E → · T;
E → · T + E
T → · int
T → · (E)
9
T → int ·
E
+
int
T
;
(
)
T
+
(
0
3
5
8
int
+
int
;
)
;

LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
2
E → T + E ·
E
5
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
6
T → (E)·
)
7
T → (E·)
1
S → E ·
E
+
int
T
(
E
9
T → int ·
int
8
T → (·E)
E → · T;
E → · T + E
T → · int
T → · (E)
start
S → · E
E → · T;
E → · T + E
T → · int
T → · (E)
0
int
T
E → T ·;
E → T · + E
3
T
;
E → T ;·
4
(
(
T
+
(
int
0
3
5
8
+
int
;
)
;

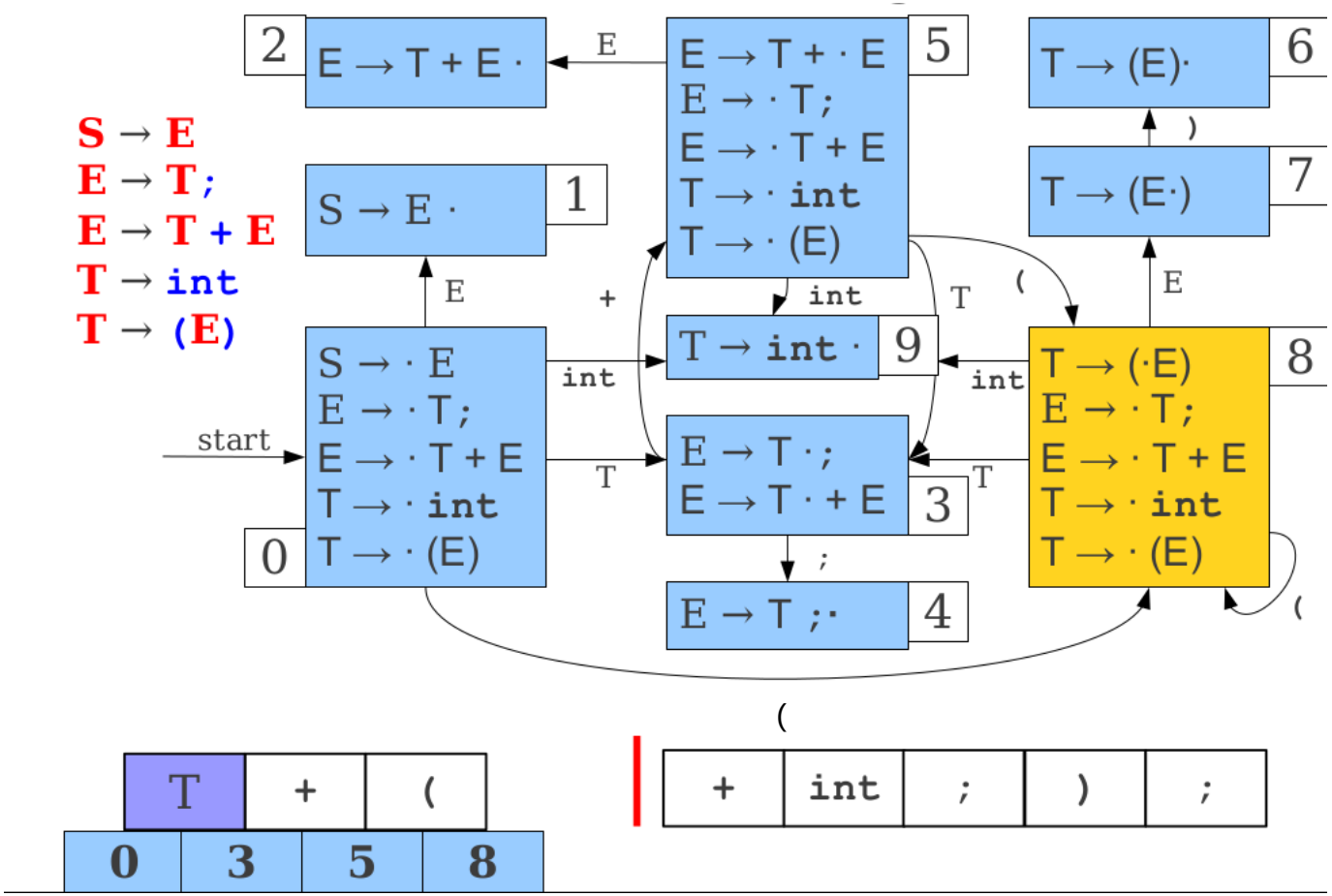
LR(0) Parsing



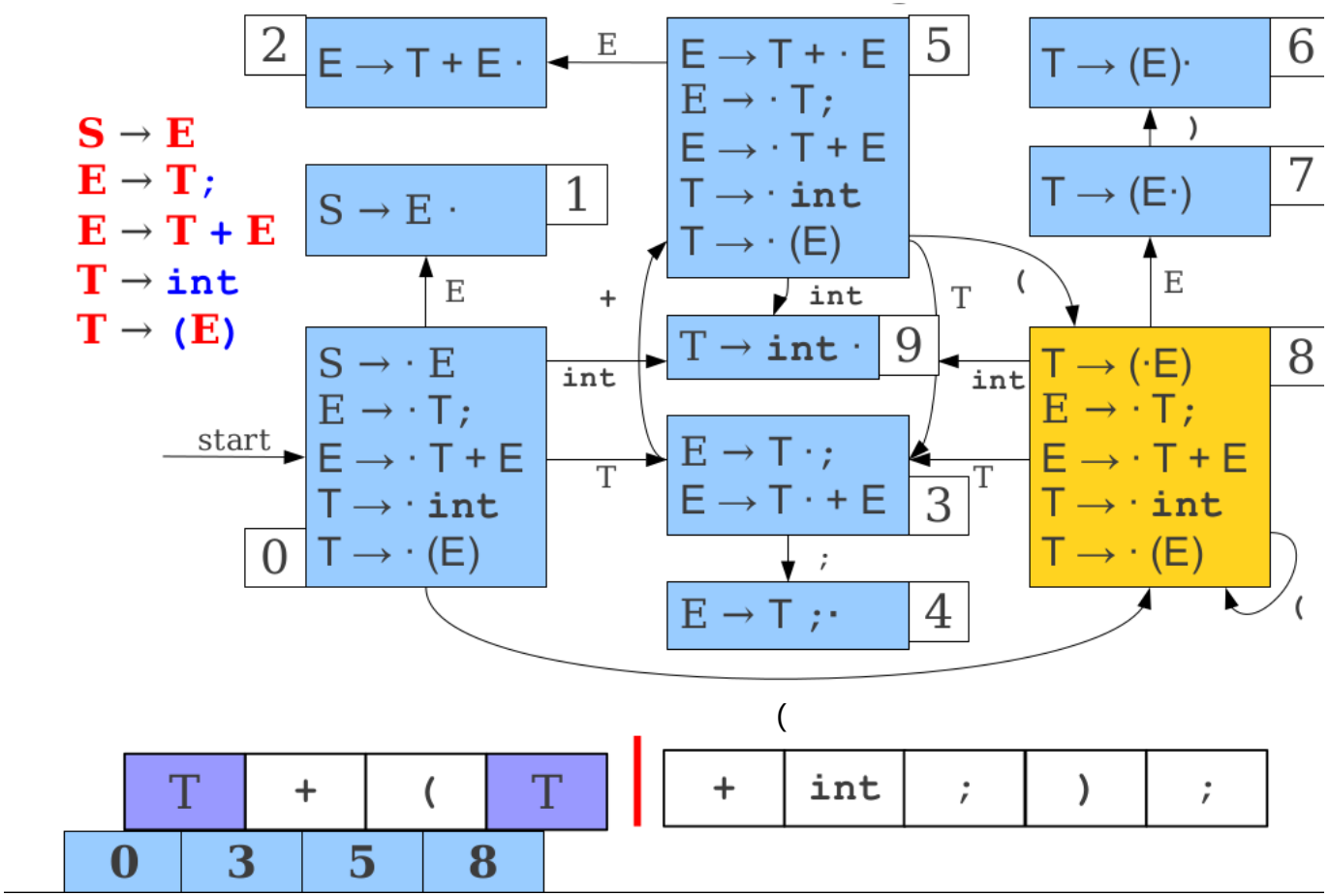
T	+	(	int
0	3	5	8

+	int	;	)	;

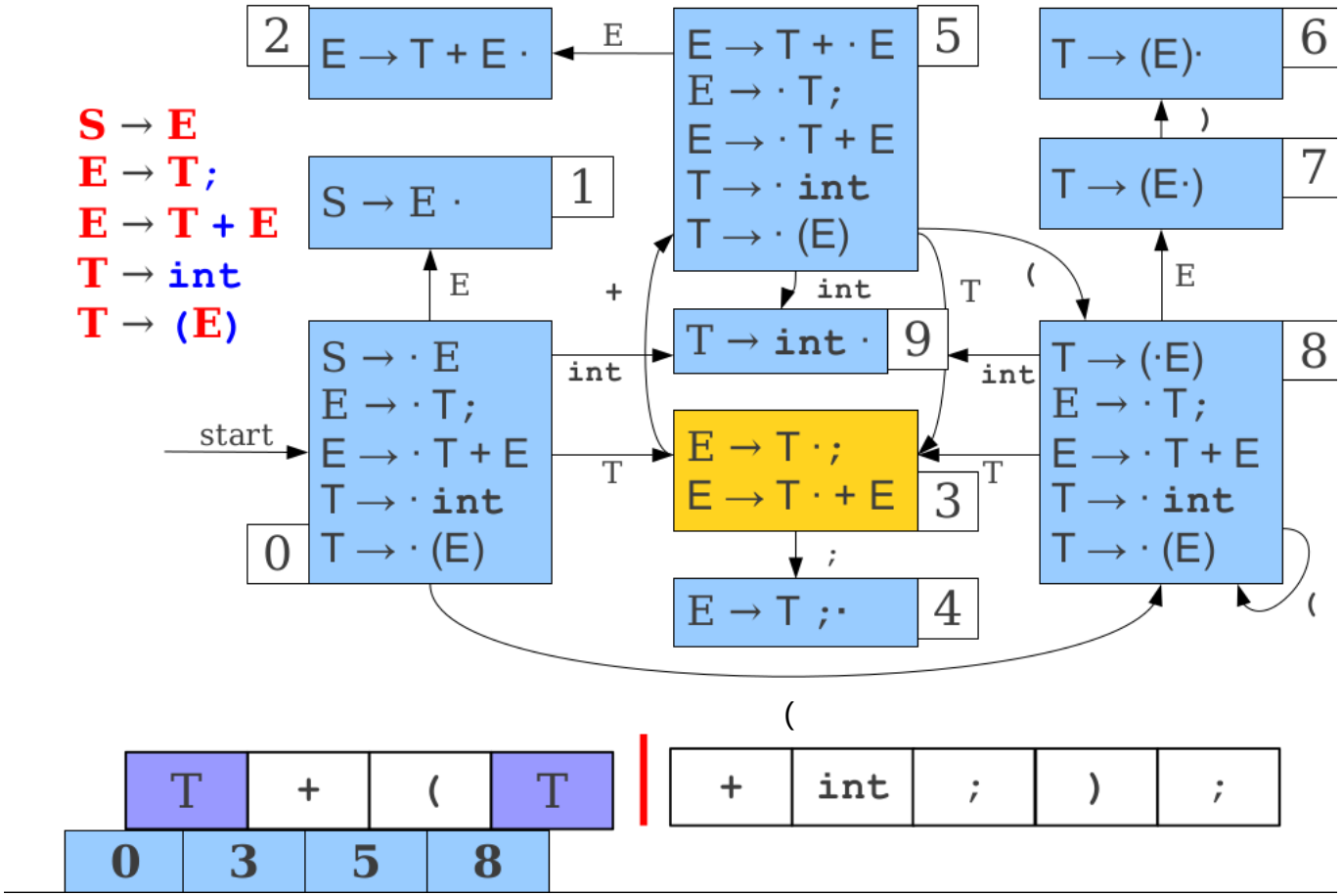
LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
2
E → T + E ·
E
5
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
6
T → (E)·
)
7
T → (E·)
1
S → E ·
E
+
int
T
(
E
9
T → int ·
int
8
T → (·E)
E → · T;
E → · T + E
T → · int
T → · (E)
start
S → · E
E → · T;
E → · T + E
T → · int
T → · (E)
0
int
T
E → T ·;
E → T · + E
3
T
;
4
E → T ;·
(
T
+
(
0
3
5
8
+
int
;
)
;

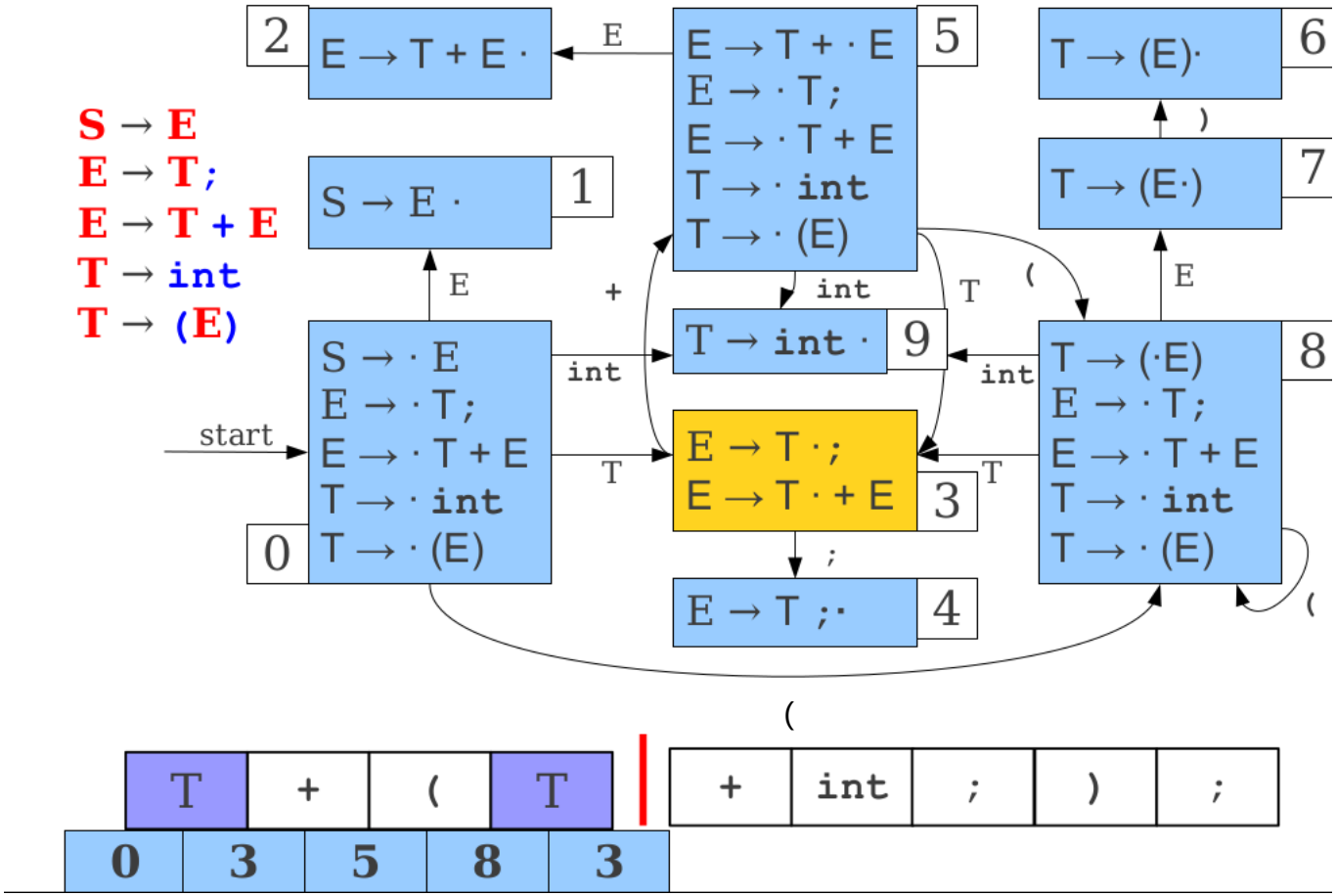
LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
start
0
S → · E
E → · T;
E → · T + E
T → · int
T → · (E)
1
S → E ·
2
E → T + E ·
3
E → T ·;
E → T · + E
4
E → T ;·
5
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
6
T → (E)·
7
T → (E·)
8
T → (·E)
E → · T;
E → · T + E
T → · int
T → · (E)
9
T → int ·
E
+
int
T
;
(
)
T
+
(
T
0
3
5
8
+
int
;
)
;

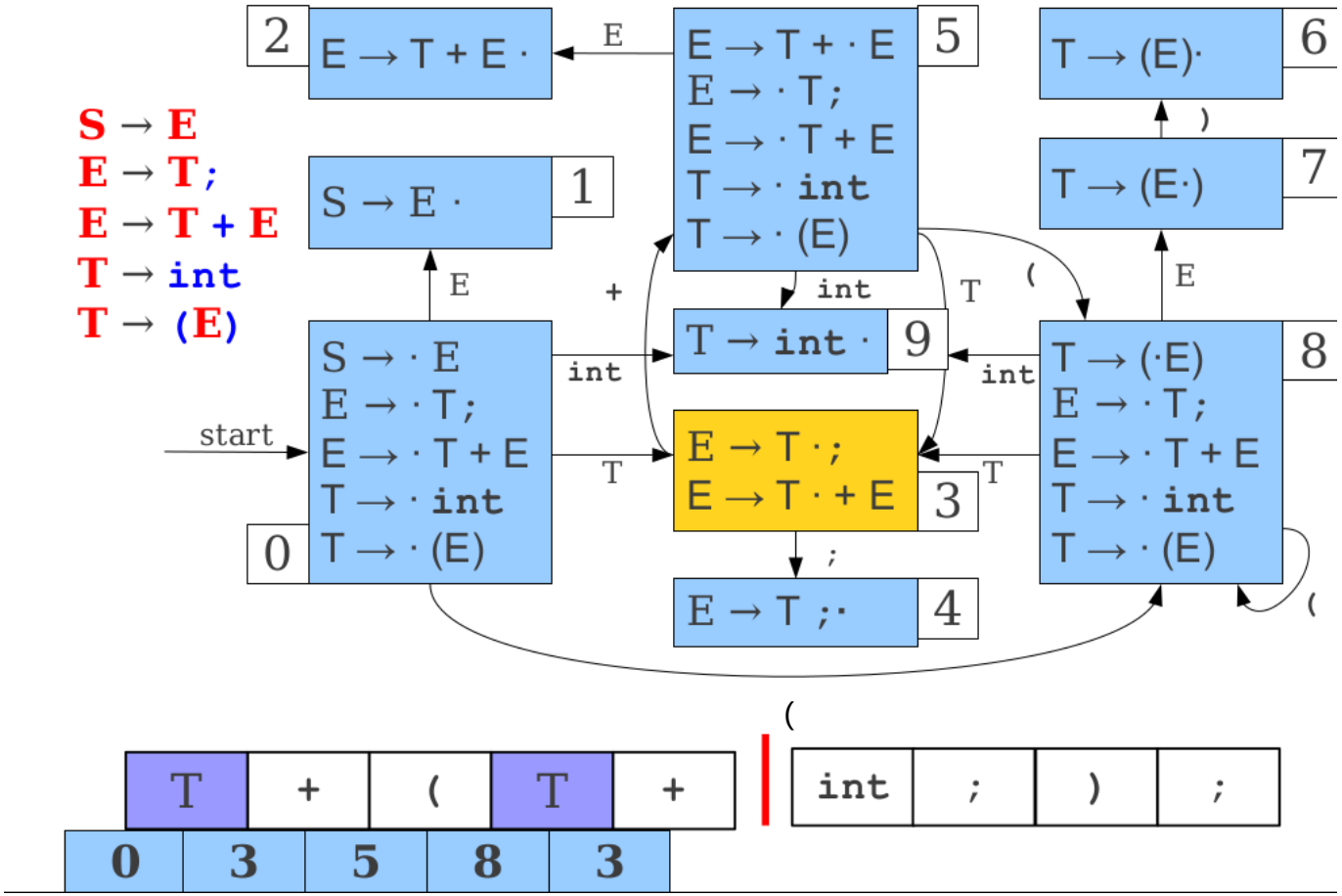
LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
start
0
S → · E
E → · T;
E → · T + E
T → · int
T → · (E)
1
S → E ·
2
E → T + E ·
3
E → T · ;
E → T · + E
4
E → T ;·
5
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
6
T → (E)·
7
T → (E·)
8
T → (·E)
E → · T;
E → · T + E
T → · int
T → · (E)
9
T → int ·
E
+
int
T
;
(
)
T
+
(
T
0
3
5
8
+
int
;
)
;

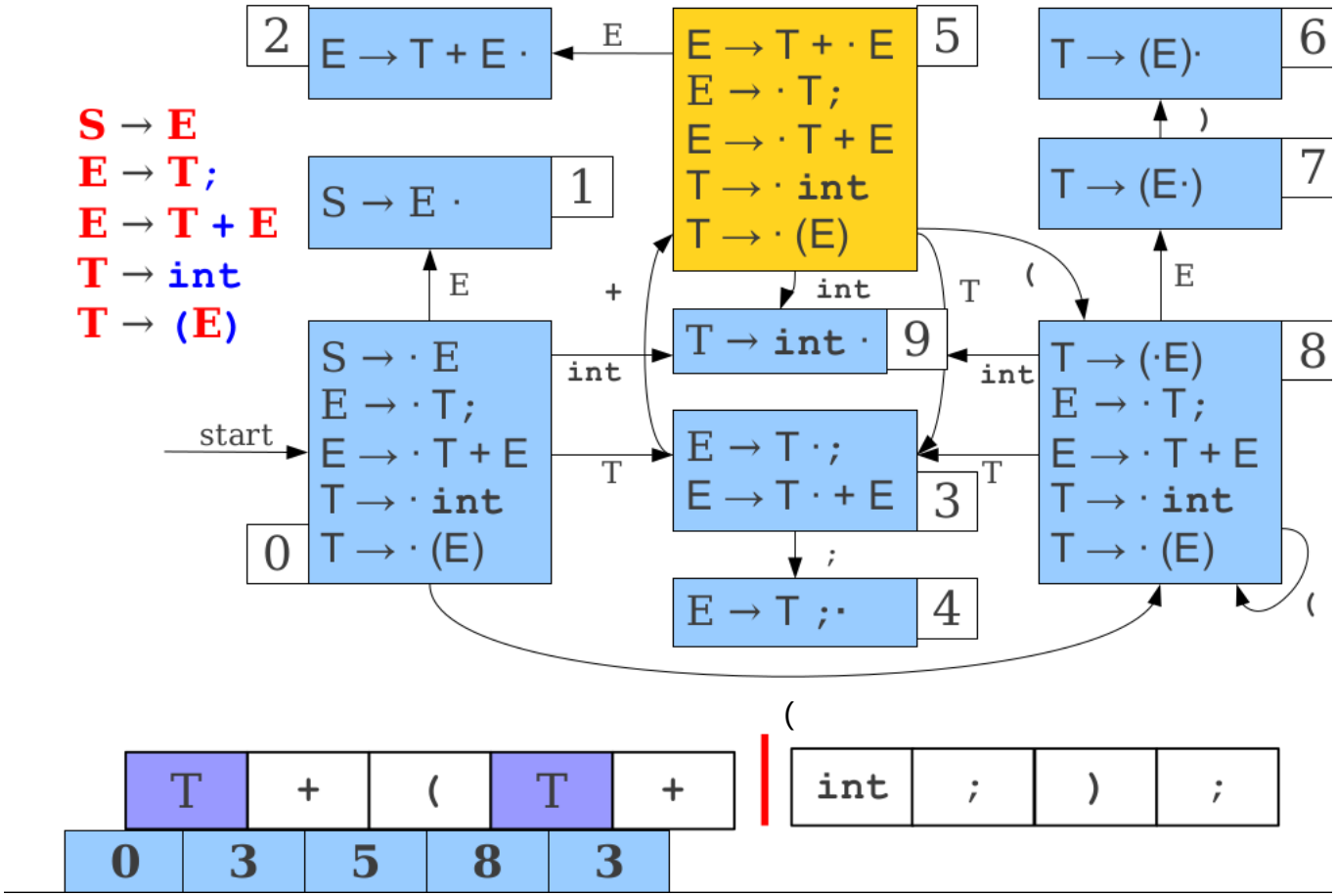
LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
2
E → T + E ·
E
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
5
T → (E)·
6
)
T → (E·)
7
S → E ·
1
E
+
int
T
(
E
S → · E
E → · T;
E → · T + E
T → · int
T → · (E)
0
start
int
T → int ·
9
int
T → (·E)
E → · T;
E → · T + E
T → · int
T → · (E)
8
E → T ·;
E → T · + E
3
T
;
E → T ;·
4
(
T
+
(
T
0
3
5
8
3
+
int
;
)
;

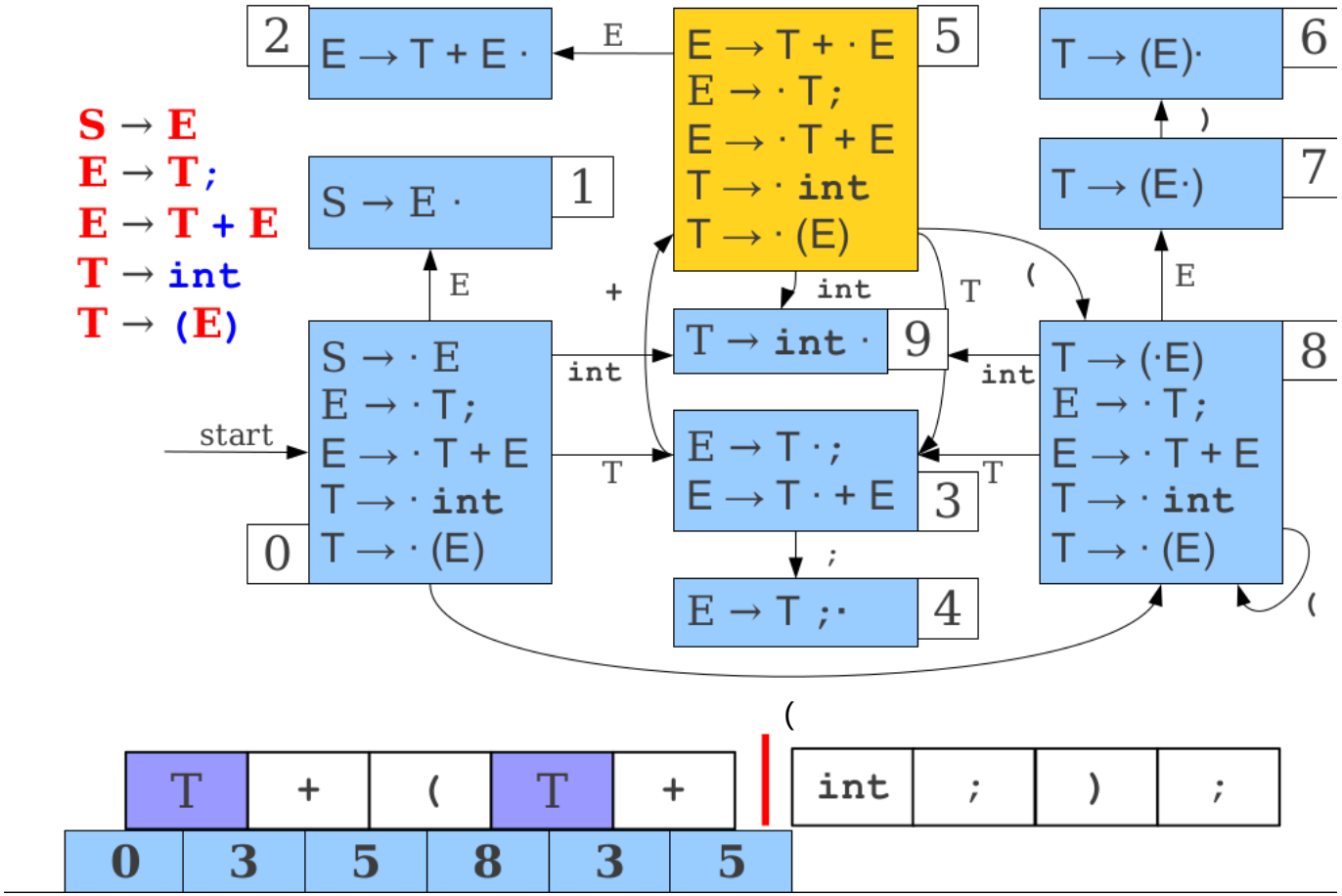
LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
start
0
S → · E
E → · T;
E → · T + E
T → · int
T → · (E)
1
S → E ·
2
E → T + E ·
3
E → T · ;
E → T · + E
4
E → T ;·
5
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
6
T → (E)·
7
T → (E·)
8
T → (·E)
E → · T;
E → · T + E
T → · int
T → · (E)
9
T → int ·
E
+
int
T
(
)
;
T
+
(
T
+
0
3
5
8
3
(
int
;
)
;

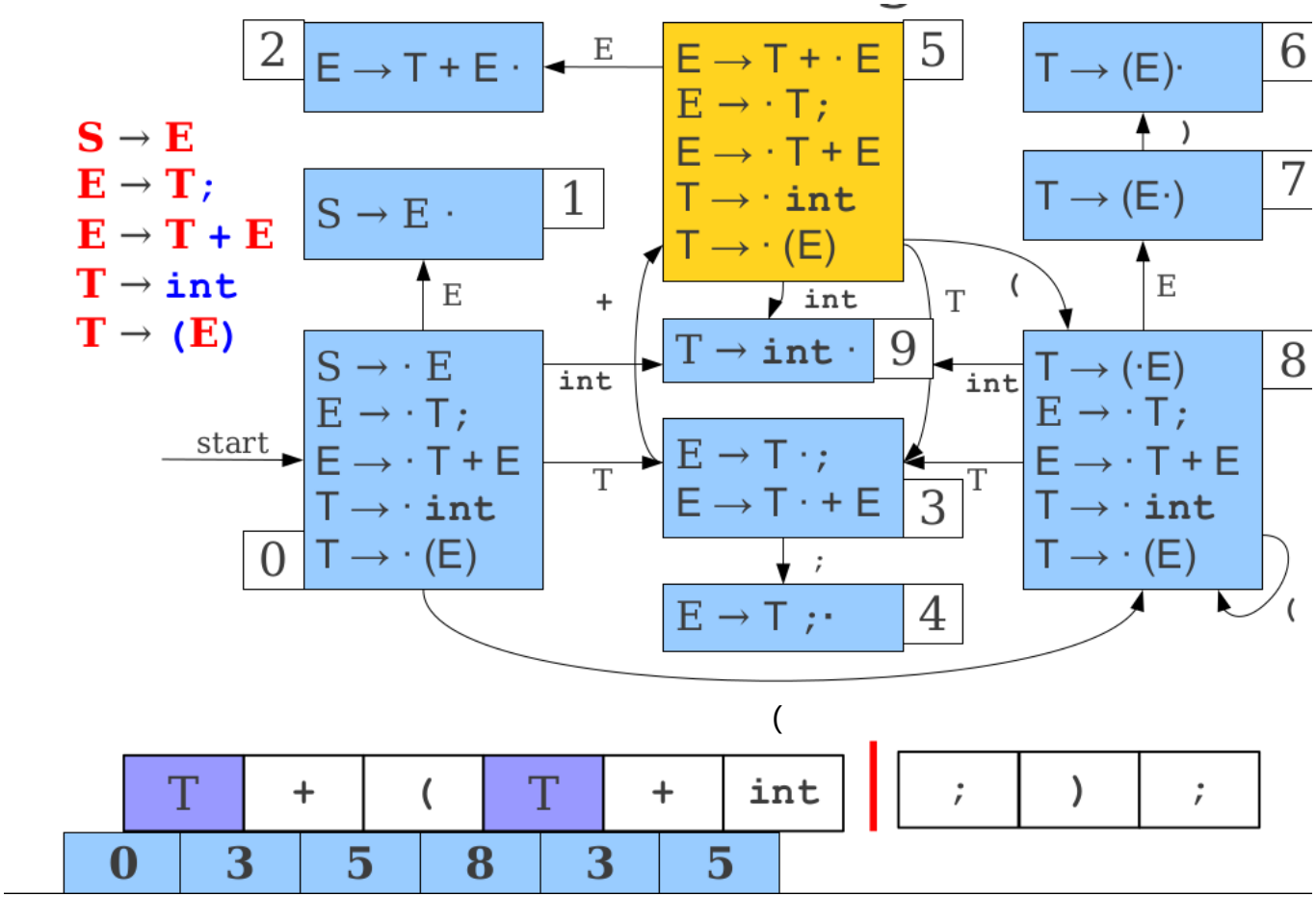
LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
start
0
S → · E
E → · T;
E → · T + E
T → · int
T → · (E)
1
S → E ·
2
E → T + E ·
3
E → T · ;
E → T · + E
4
E → T ; ·
5
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
6
T → (E) ·
7
T → (E ·)
8
T → (· E)
E → · T;
E → · T + E
T → · int
T → · (E)
9
T → int ·
E
+
int
T
(
)
;
T
+
(
int
;
)
;
0
3
5
8
3

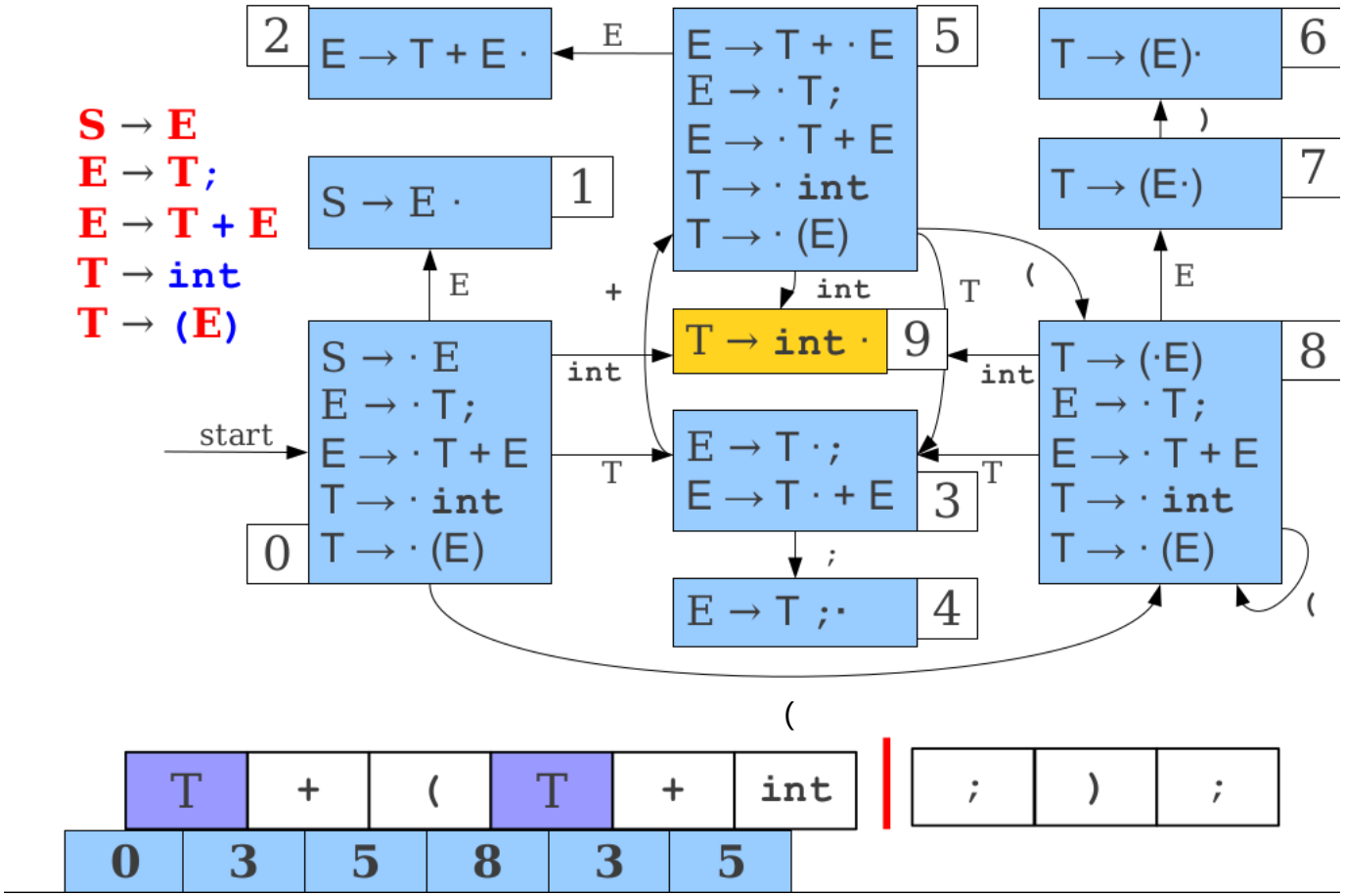
LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
start
0
S → · E
E → · T;
E → · T + E
T → · int
T → · (E)
1
S → E ·
2
E → T + E ·
3
E → T · ;
E → T · + E
4
E → T ;·
5
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
6
T → (E)·
7
T → (E·)
8
T → (·E)
E → · T;
E → · T + E
T → · int
T → · (E)
9
T → int ·
E
+
int
T
;
(
)
T
+
(
T
+
int
;
)
;
0
3
5
8
3
5

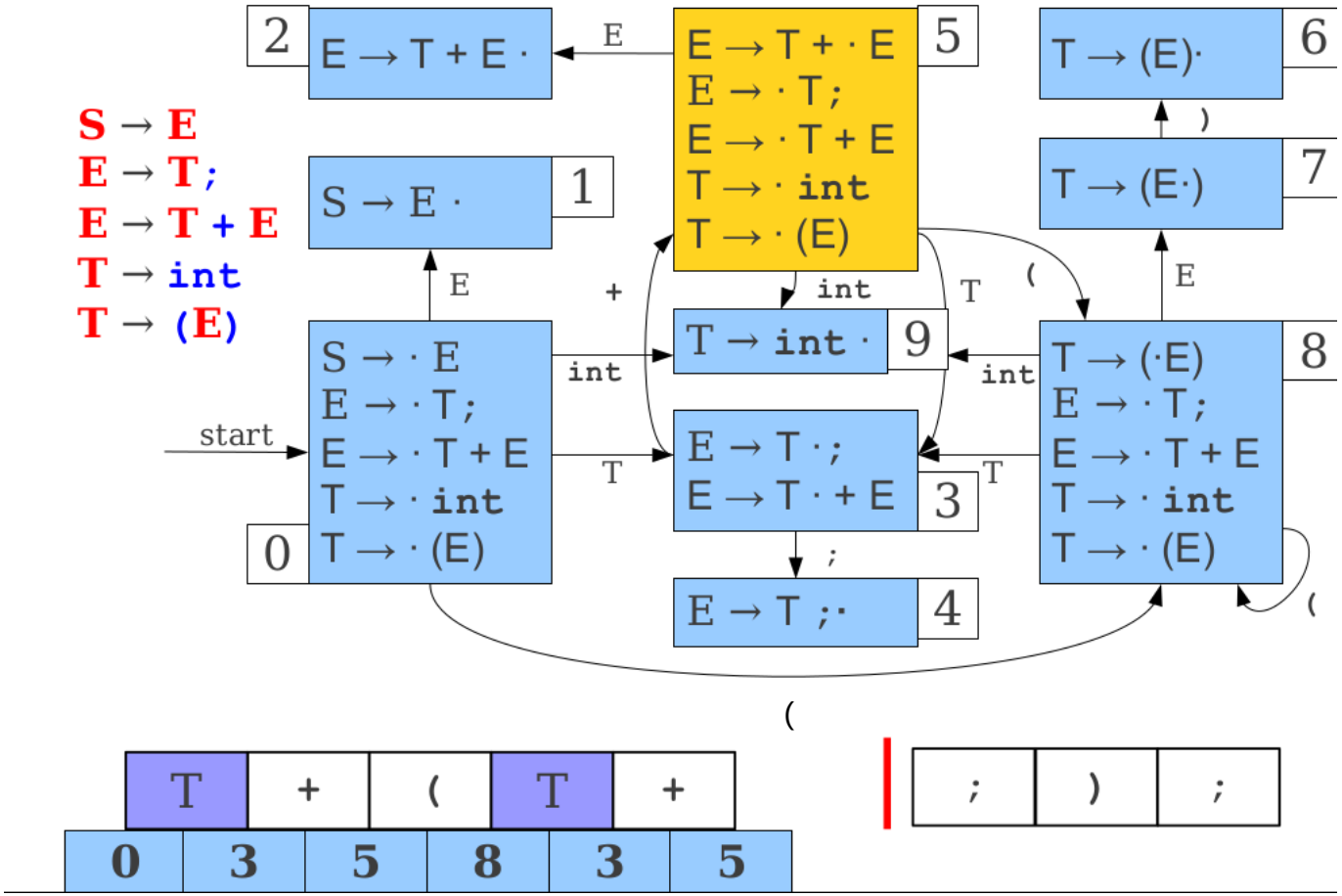
LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
2
E → T + E ·
E
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
5
T → (E)·
6
)
S → E ·
1
T → (E·)
7
E
+
int
T
(
E
T → int ·
9
int
T → (·E)
E → · T;
E → · T + E
T → · int
T → · (E)
8
start
S → · E
E → · T;
E → · T + E
T → · int
T → · (E)
0
int
T
E → T ·;
E → T · + E
3
T
;
E → T ;·
4
(
(
T
+
(
T
+
int
;
)
;
0
3
5
8
3
5

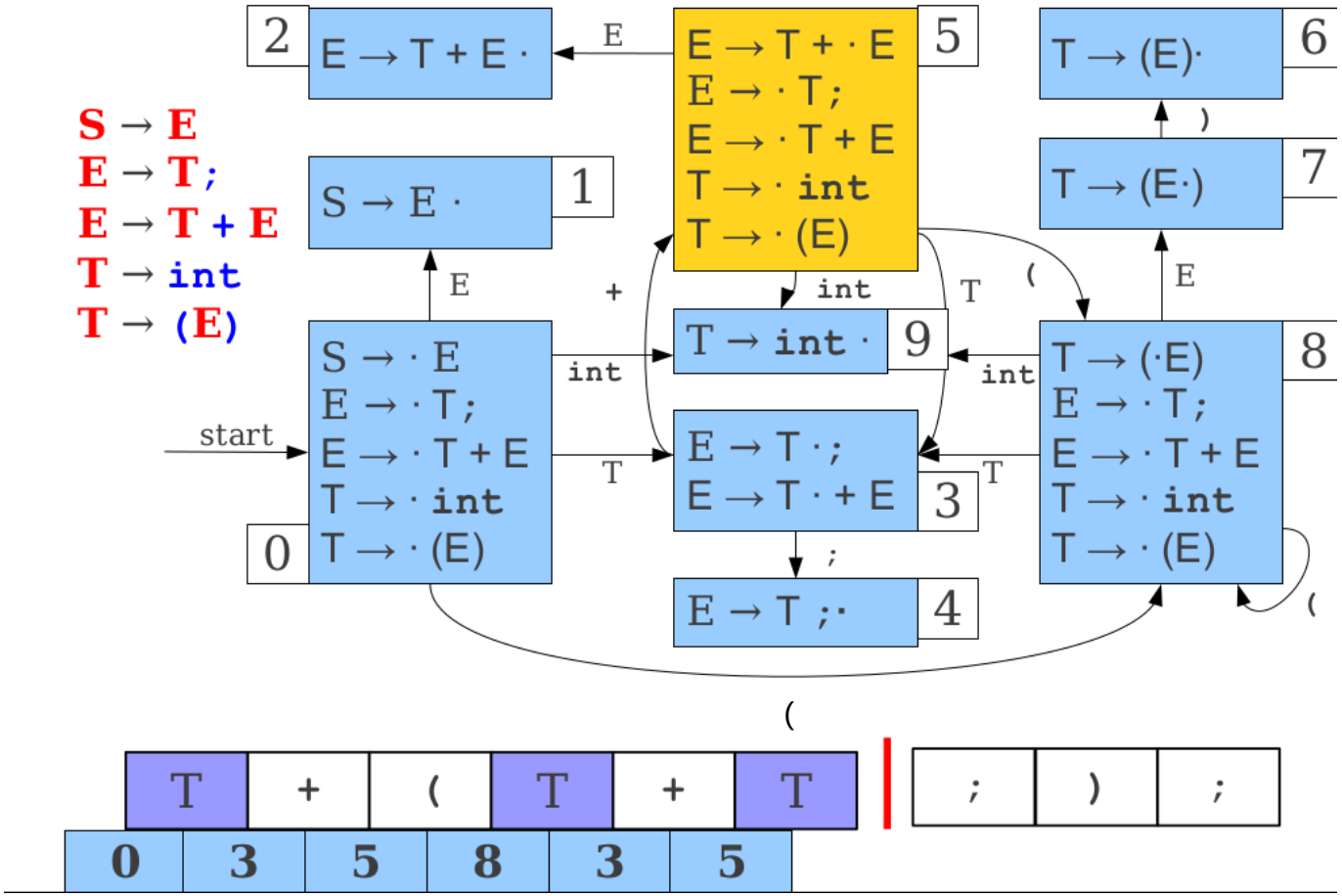
LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
2
E → T + E ·
E
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
5
T → (E)·
6
)
T → (E·)
7
S → E ·
1
E
+
int
T
(
E
T → int ·
9
int
T → (·E)
E → · T;
E → · T + E
T → · int
T → · (E)
8
start
S → · E
E → · T;
E → · T + E
T → · int
T → · (E)
0
int
T
E → T ·;
E → T · + E
3
T
;
E → T ;·
4
(
(
T
+
(
T
+
int
;
)
;
0
3
5
8
3
5

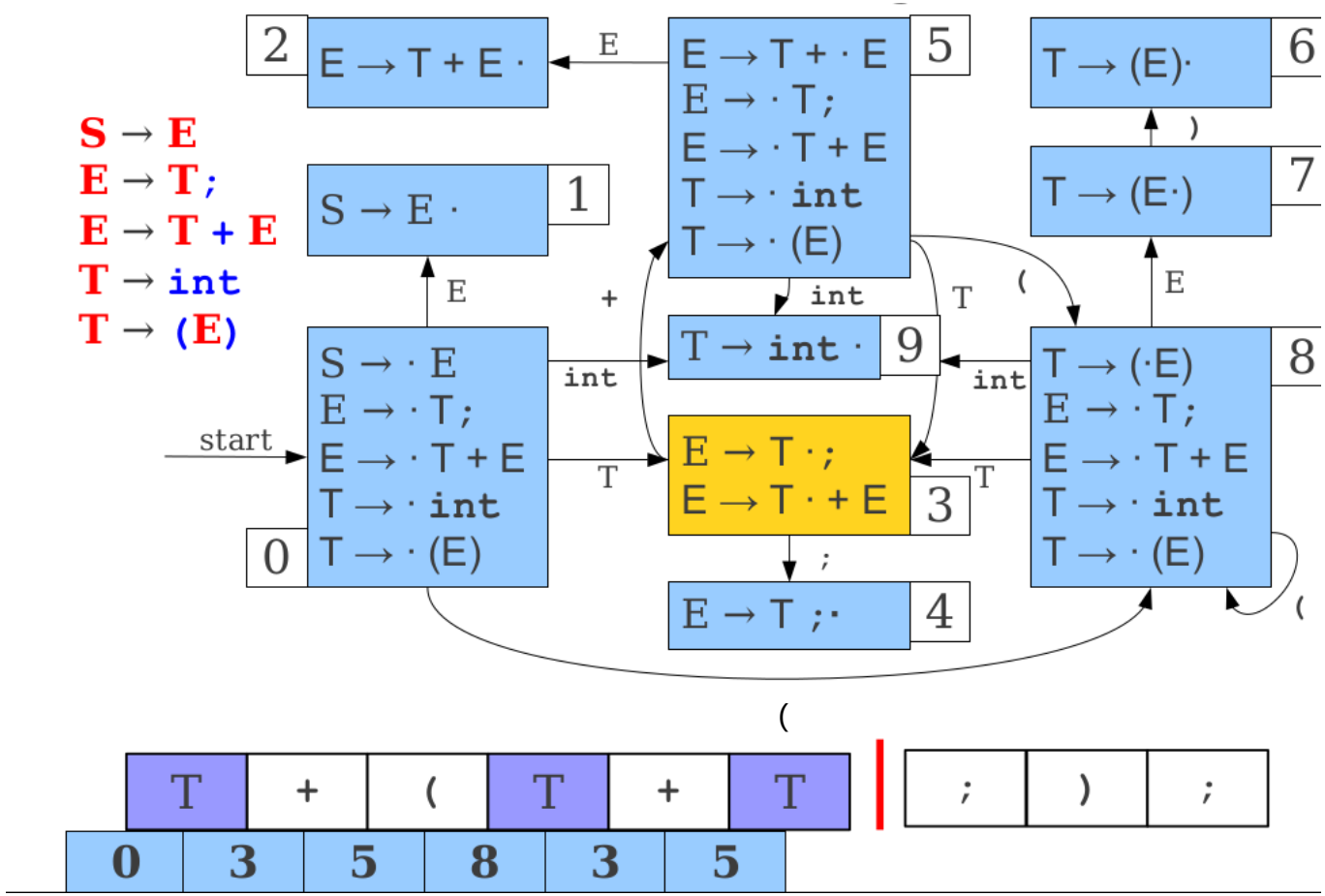
LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
2
E → T + E ·
E
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
5
T → (E)·
6
)
T → (E·)
7
S → E ·
1
E
+
int
T
(
E
T → int ·
9
int
T → (·E)
E → · T;
E → · T + E
T → · int
T → · (E)
8
start
S → · E
E → · T;
E → · T + E
T → · int
T → · (E)
0
int
T
E → T ·;
E → T · + E
3
T
;
E → T ;·
4
(
(
T
+
(
T
+
0
3
5
8
3
5
;
)
;

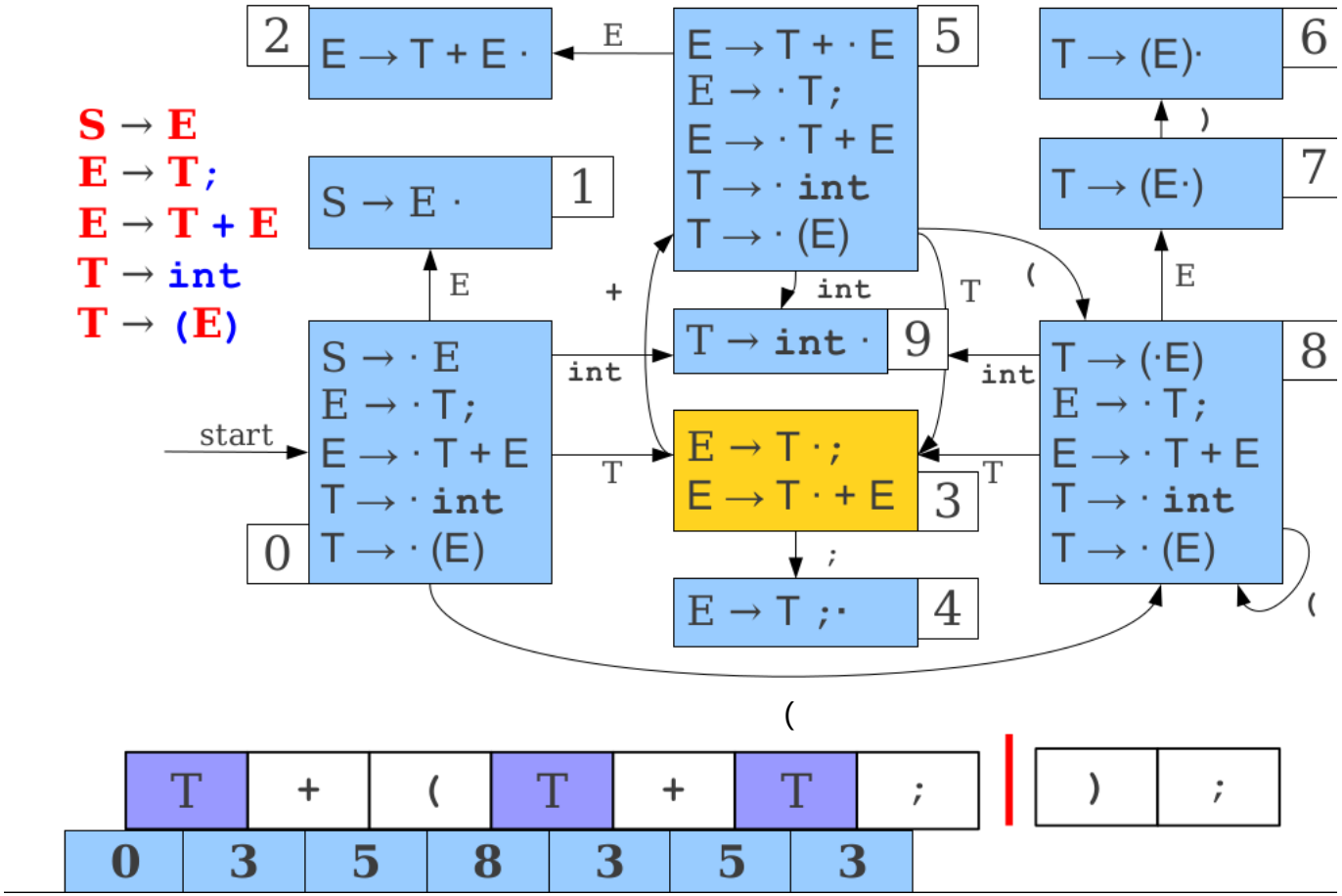
LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
start
0
S → · E
E → · T;
E → · T + E
T → · int
T → · (E)
1
S → E ·
2
E → T + E ·
3
E → T · ;
E → T · + E
4
E → T ; ·
5
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
6
T → (E)·
7
T → (E·)
8
T → (·E)
E → · T;
E → · T + E
T → · int
T → · (E)
9
T → int ·
E
+
int
T
;
(
)
T
+
(
T
+
T
;
)
;
0
3
5
8
3
5

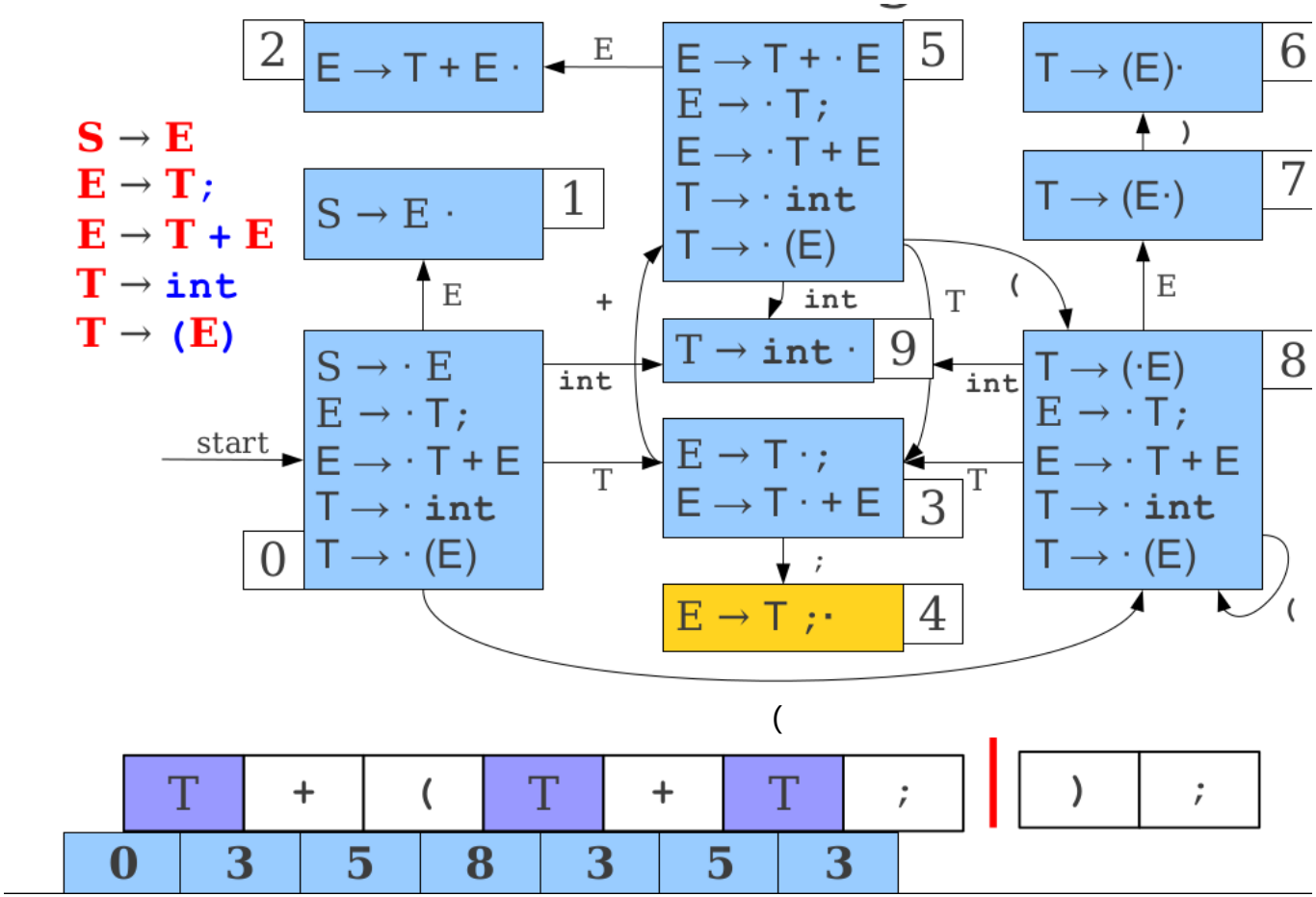
LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
start
0
S → · E
E → · T;
E → · T + E
T → · int
T → · (E)
1
S → E ·
2
E → T + E ·
3
E → T · ;
E → T · + E
4
E → T ; ·
5
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
6
T → (E)·
7
T → (E·)
8
T → (·E)
E → · T;
E → · T + E
T → · int
T → · (E)
9
T → int ·
E
+
int
T
(
;
)
T
+
(
T
+
T
;
)
;
0
3
5
8
3
5

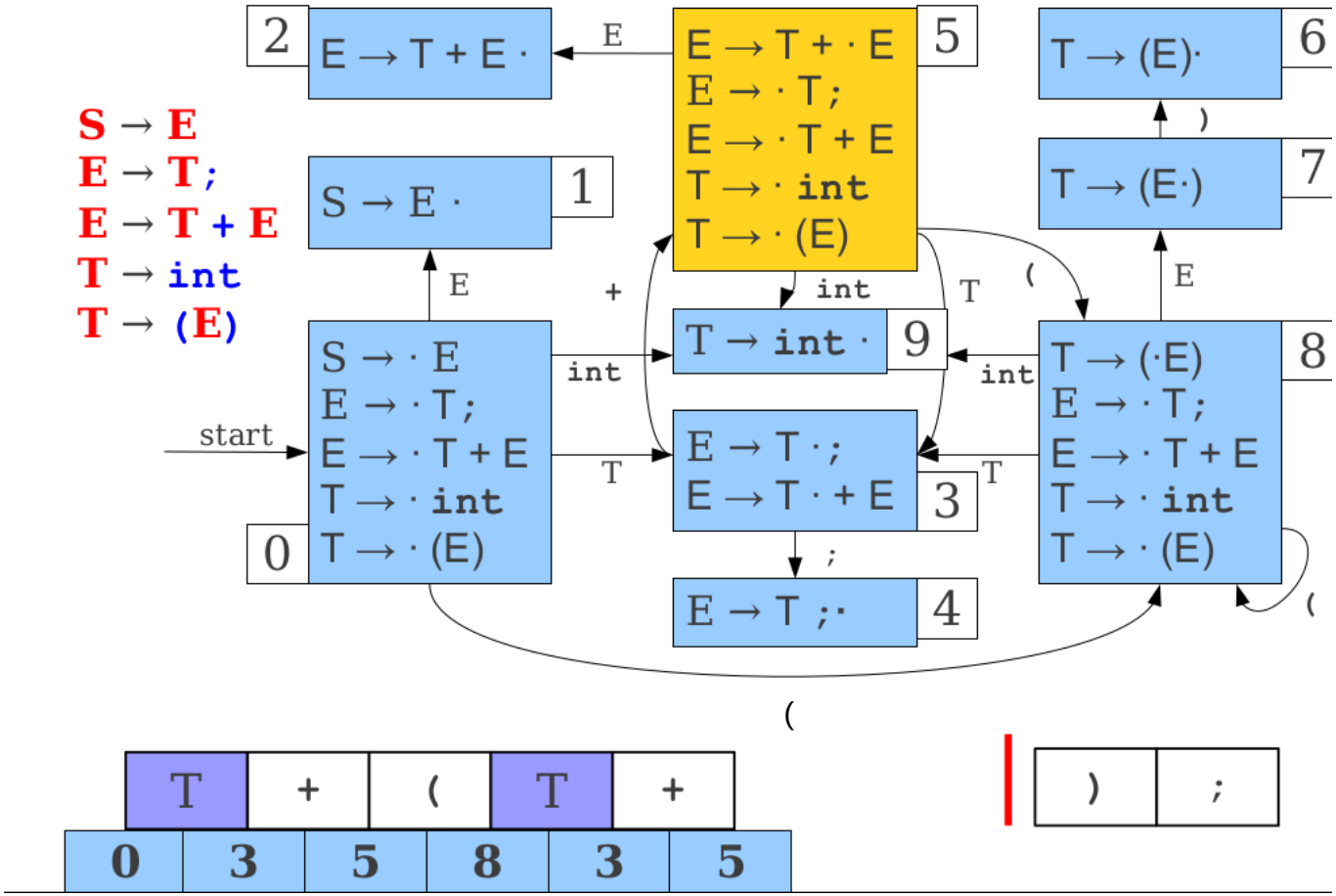
LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
start
0
S → · E
E → · T;
E → · T + E
T → · int
T → · (E)
1
S → E ·
2
E → T + E ·
3
E → T · ;
E → T · + E
4
E → T ; ·
5
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
6
T → (E)·
7
T → (E·)
8
T → (·E)
E → · T;
E → · T + E
T → · int
T → · (E)
9
T → int ·
E
+
int
T
;
(
)
T
+
(
T
+
T
;
)
;
0
3
5
8
3
5
3

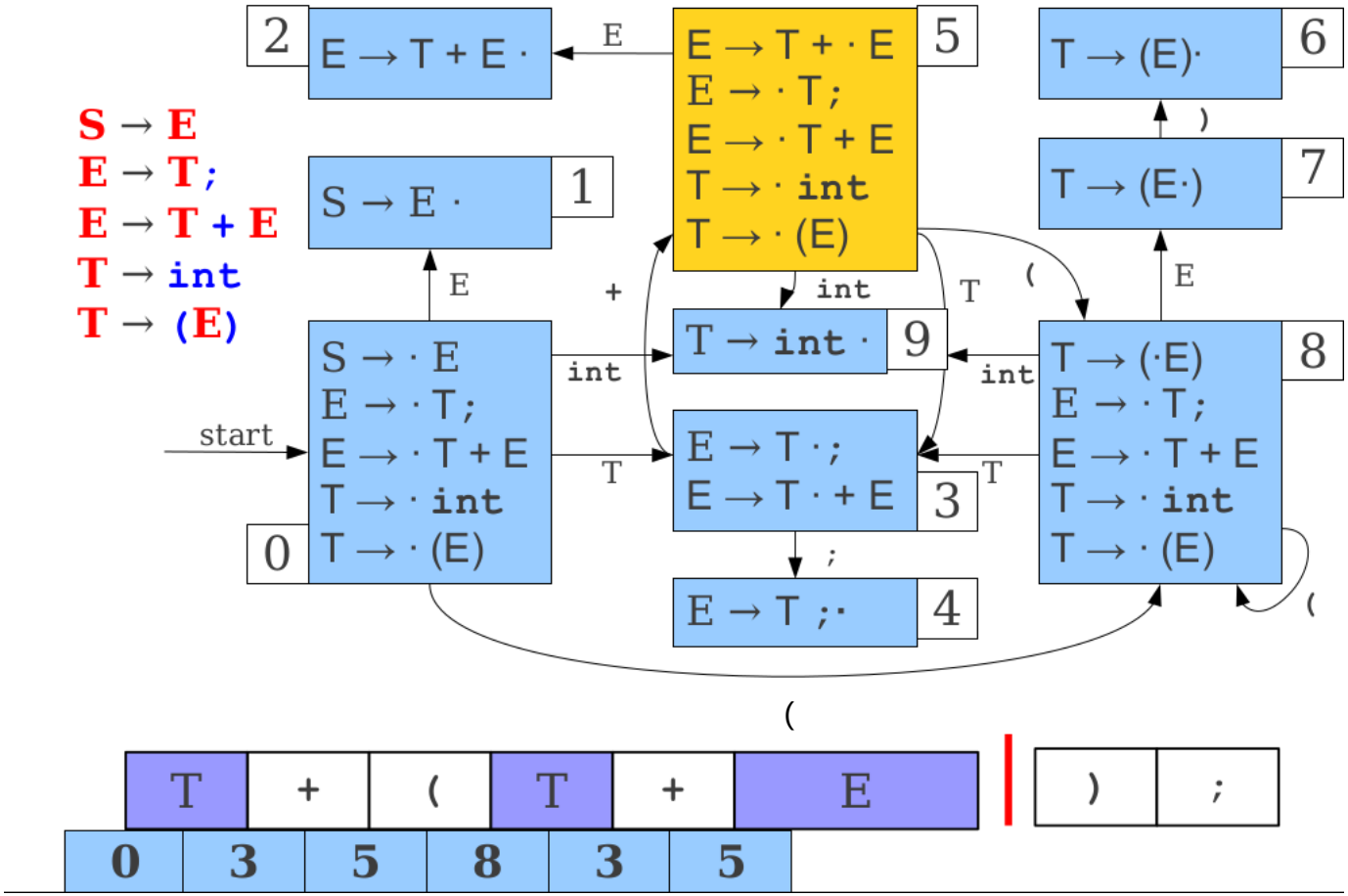
LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
start
0
S → · E
E → · T;
E → · T + E
T → · int
T → · (E)
1
S → E ·
2
E → T + E ·
3
E → T · ;
E → T · + E
4
E → T ; ·
5
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
6
T → (E) ·
7
T → (E ·)
8
T → (· E)
E → · T;
E → · T + E
T → · int
T → · (E)
9
T → int ·
E
+
int
T
(
;
)
T
+
(
T
+
T
;
)
;
0
3
5
8
3
5
3

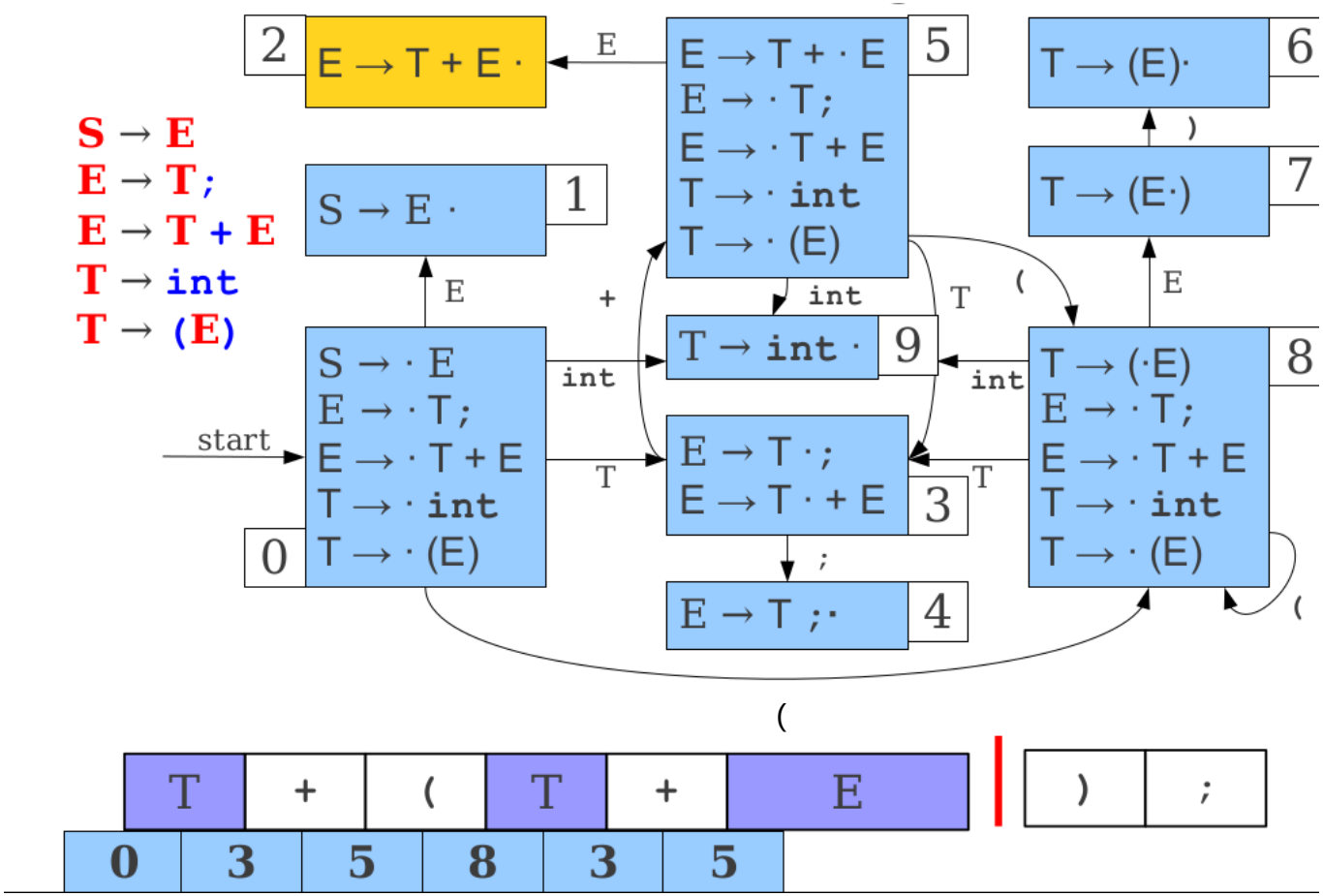
LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
2
E → T + E ·
E
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
5
T → (E)·
6
)
T → (E·)
7
S → E ·
1
E
+
int
T
(
E
S → · E
E → · T;
E → · T + E
T → · int
T → · (E)
0
start
int
T → int ·
9
int
T → (·E)
E → · T;
E → · T + E
T → · int
T → · (E)
8
E → T ·;
E → T · + E
3
T
;
E → T ;·
4
(
(
T
+
(
T
+
0
3
5
8
3
5
)
;

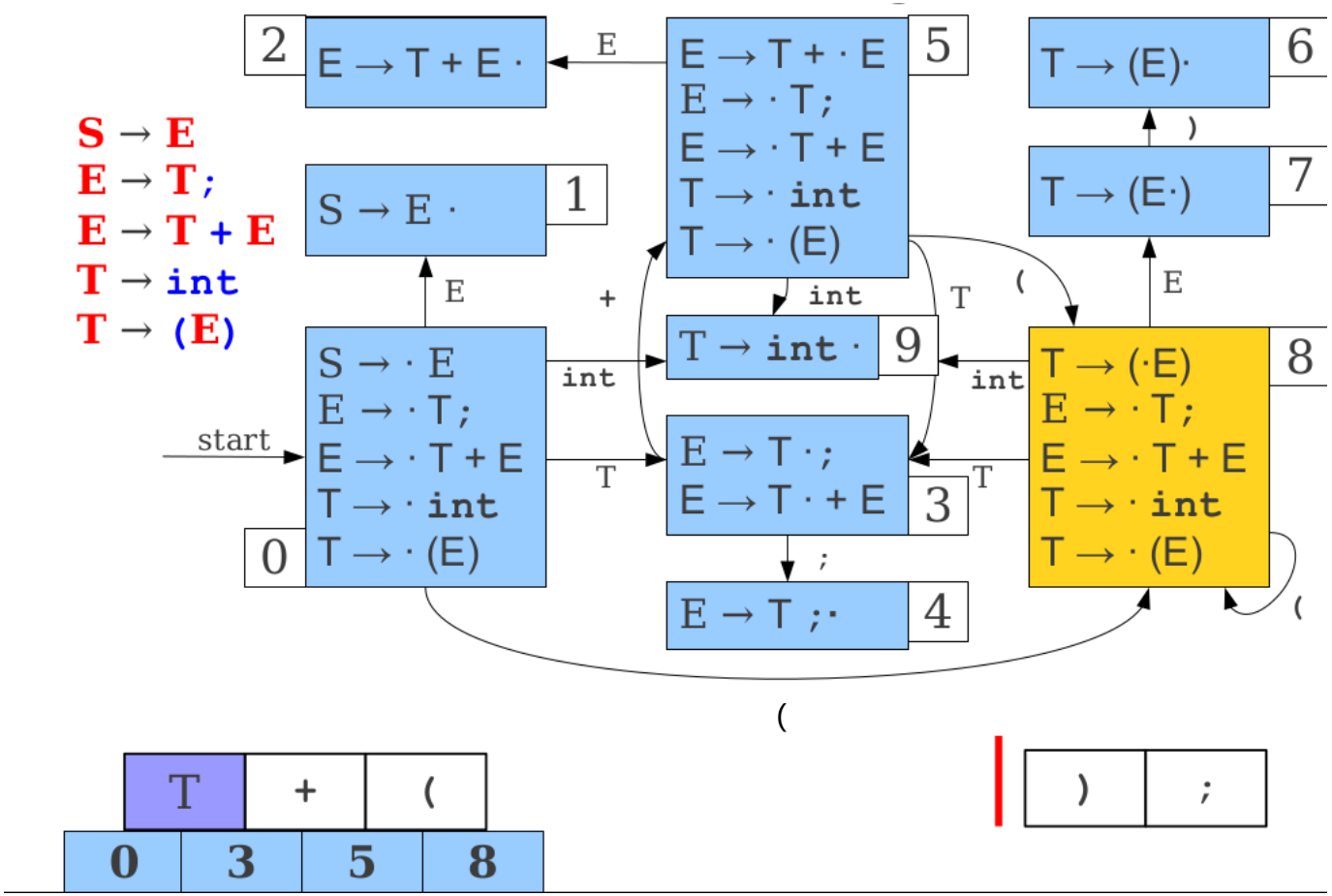
LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
2
E → T + E ·
E
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
5
T → (E)·
6
)
T → (E·)
7
S → E ·
1
E
+
int
T
(
E
S → · E
E → · T;
E → · T + E
T → · int
T → · (E)
0
start
int
T → int ·
9
int
T → (·E)
E → · T;
E → · T + E
T → · int
T → · (E)
8
E → T ·;
E → T · + E
3
T
;
E → T ;·
4
(
(
T
+
(
T
+
E
)
;
0
3
5
8
3
5

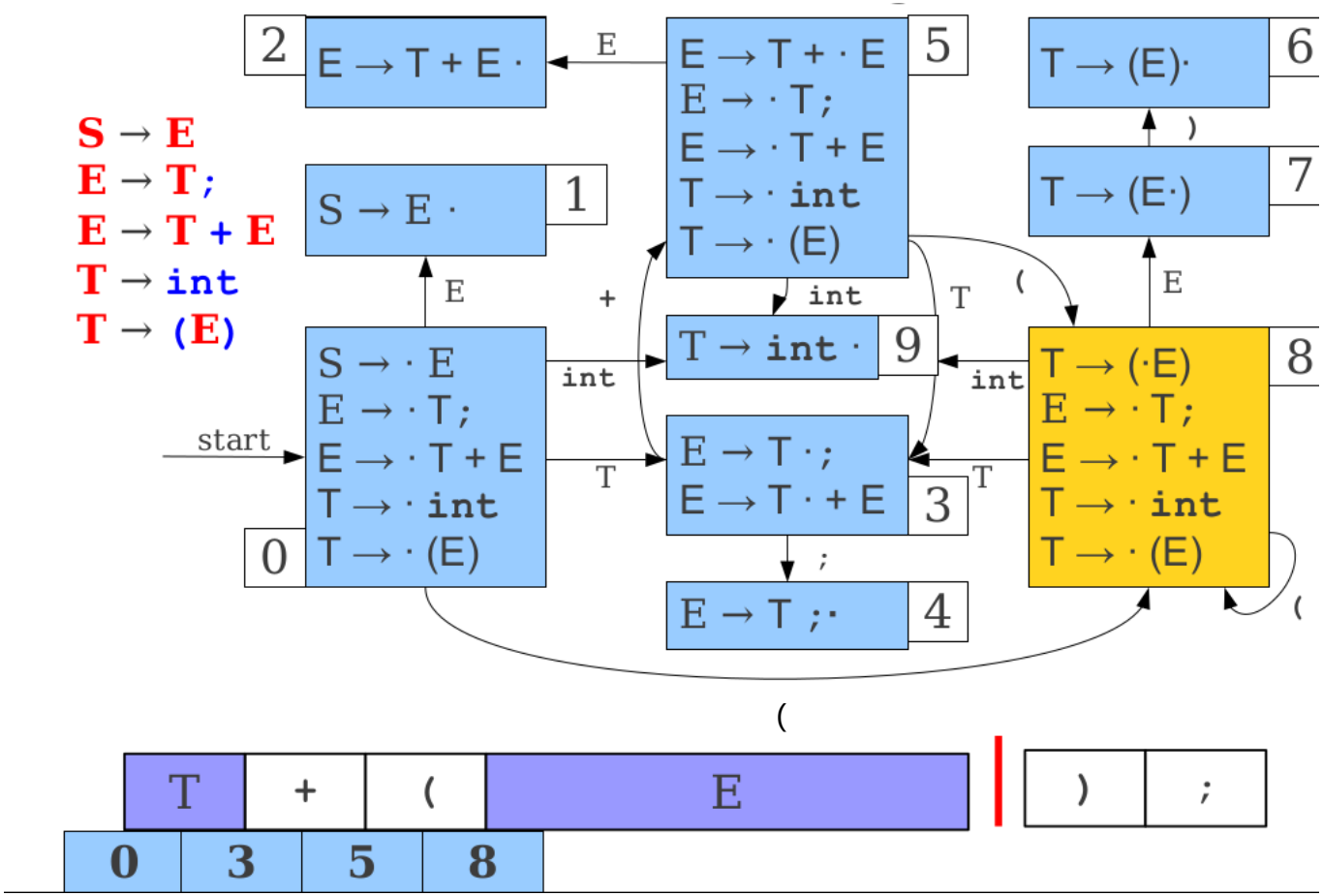
LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
2
E → T + E ·
E
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
5
T → (E)·
6
)
T → (E·)
7
S → E ·
1
E
+
int
T
(
E
T → int ·
9
int
T → (·E)
E → · T;
E → · T + E
T → · int
T → · (E)
8
start
S → · E
E → · T;
E → · T + E
T → · int
T → · (E)
0
int
T
E → T ·;
E → T · + E
3
T
;
E → T ;·
4
(
(
T
+
(
T
+
E
)
;
0
3
5
8
3
5

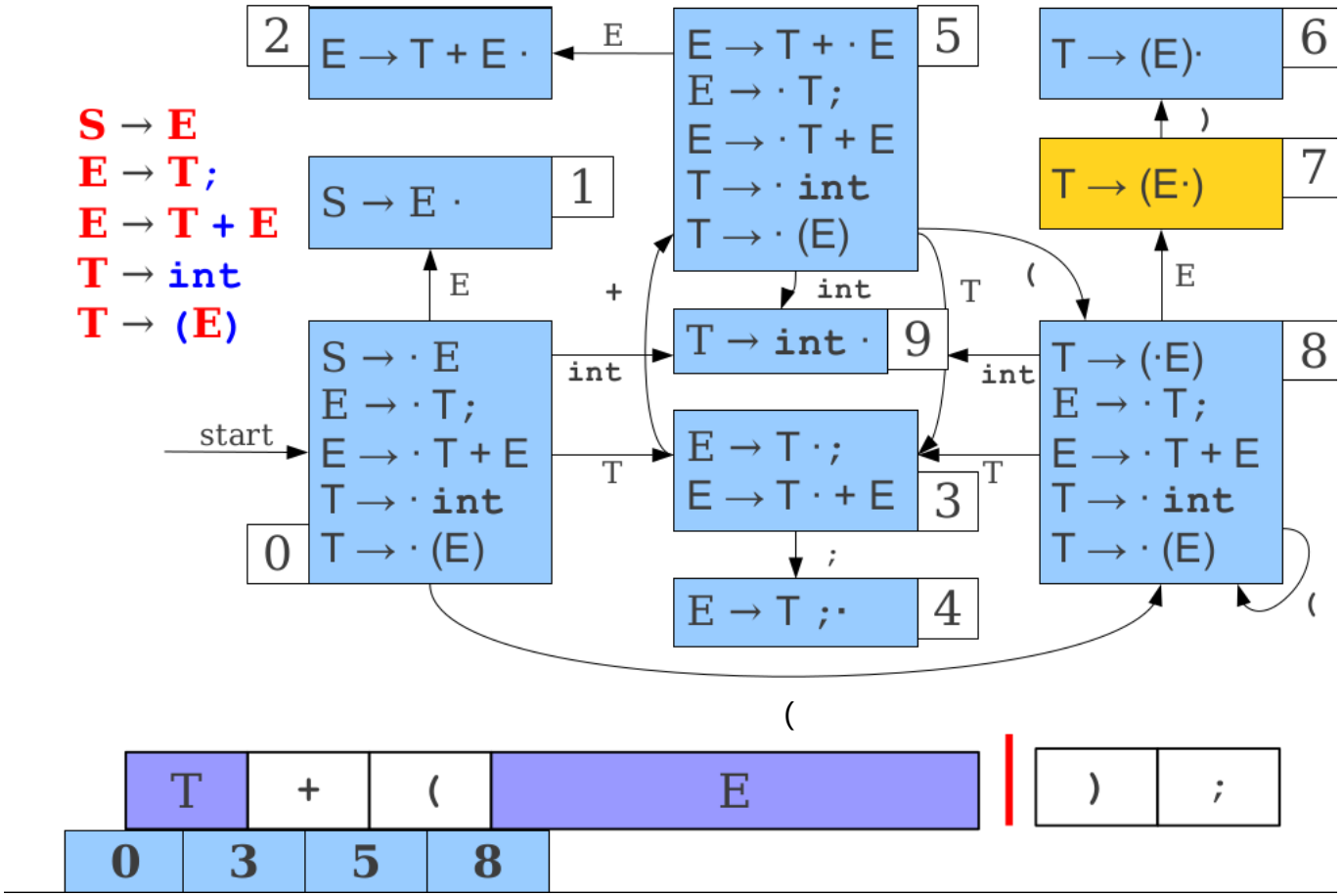
LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
2
E → T + E ·
E
5
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
6
T → (E)·
)
7
T → (E·)
1
S → E ·
E
+
int
T
(
E
9
T → int ·
int
8
T → (·E)
E → · T;
E → · T + E
T → · int
T → · (E)
start
S → · E
E → · T;
E → · T + E
T → · int
T → · (E)
0
int
T
E → T ·;
E → T · + E
3
T
;
4
E → T ;·
(
(
T
+
(
0
3
5
8
)
;

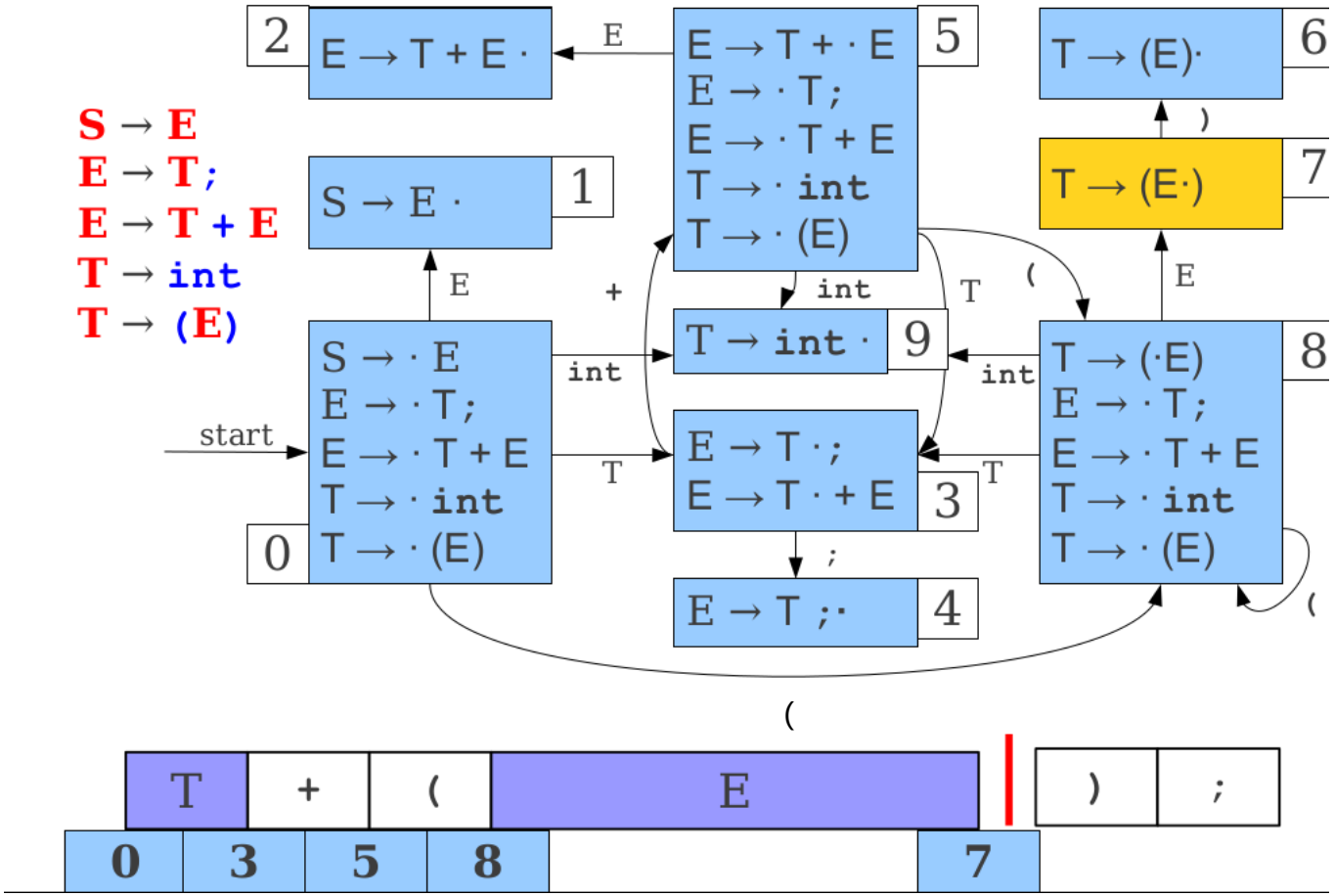
LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
start
0
S → · E
E → · T;
E → · T + E
T → · int
T → · (E)
1
S → E ·
2
E → T + E ·
3
E → T · ;
E → T · + E
4
E → T ; ·
5
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
6
T → (E)·
7
T → (E·)
8
T → (·E)
E → · T;
E → · T + E
T → · int
T → · (E)
9
T → int ·
E
+
int
T
(
)
;
T
+
(
E
)
;
0
3
5
8

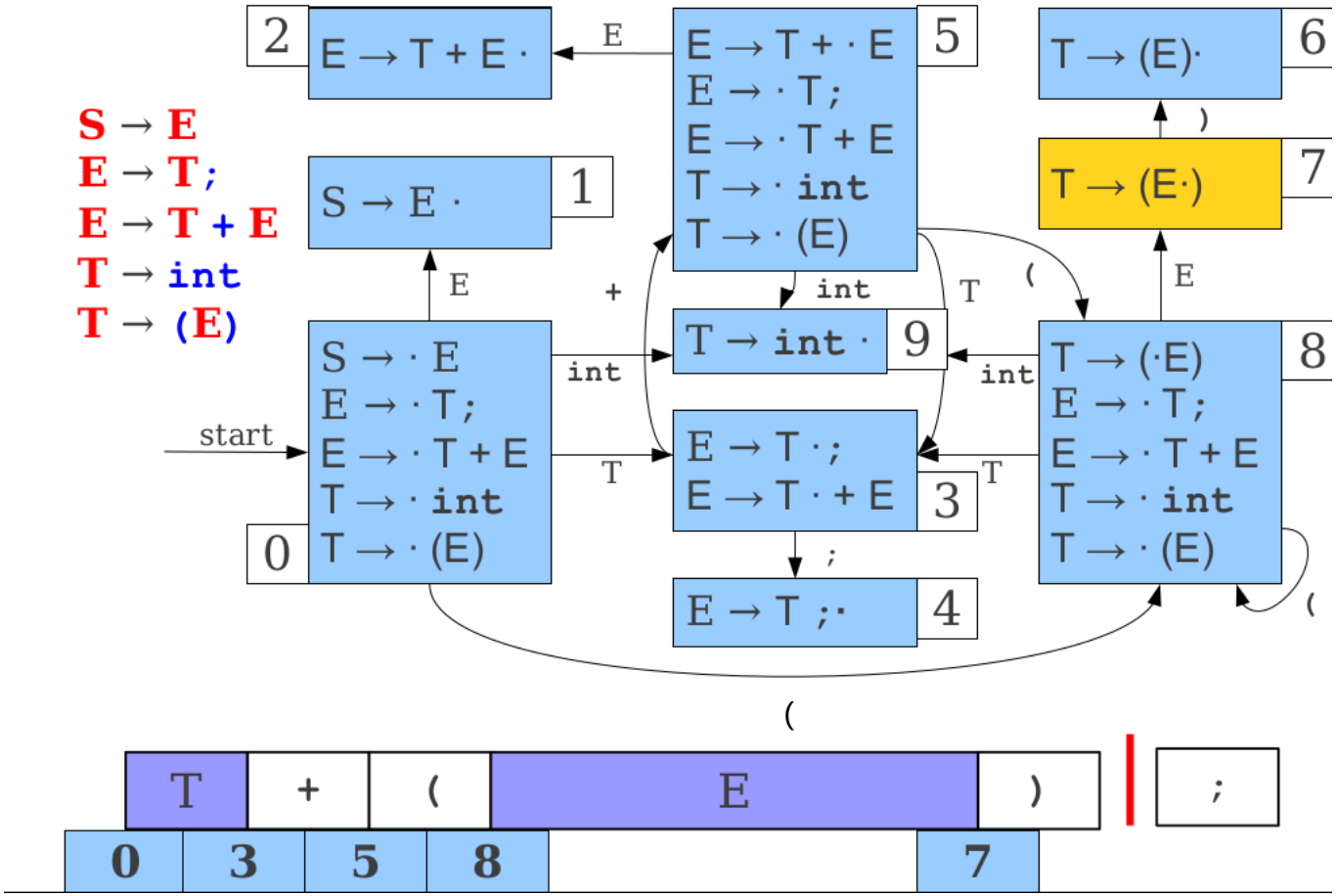
LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
2
E → T + E ·
E
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
5
T → (E)·
6
)
T → (E·)
7
S → E ·
1
E
+
int
T
(
E
S → · E
E → · T;
E → · T + E
T → · int
T → (E)
0
start
int
T → int ·
9
int
T → (·E)
E → · T;
E → · T + E
T → · int
T → · (E)
8
E → T ·;
E → T · + E
3
T
;
E → T ;·
4
(
T
+
(
E
)
;
0
3
5
8

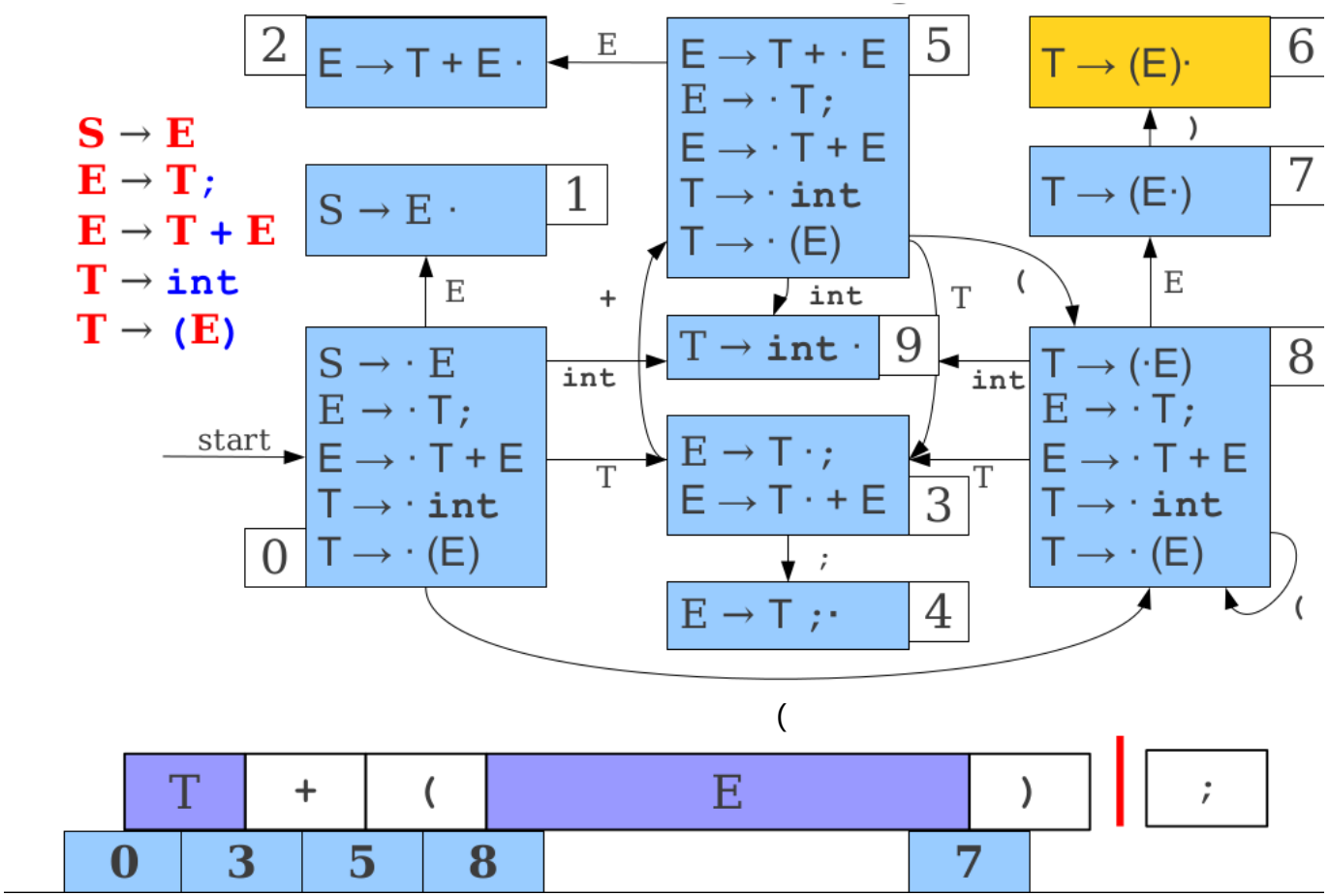
LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
2
E → T + E ·
E
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
5
T → (E)·
6
)
T → (E·)
7
S → E ·
1
E
+
int
T
(
E
S → · E
E → · T;
E → · T + E
T → · int
T → (E)
0
start
int
T → int ·
9
int
T → (·E)
E → · T;
E → · T + E
T → · int
T → · (E)
8
E → T ·;
E → T · + E
3
T
;
E → T ;·
4
(
T
+
(
E
)
;
0
3
5
8
7

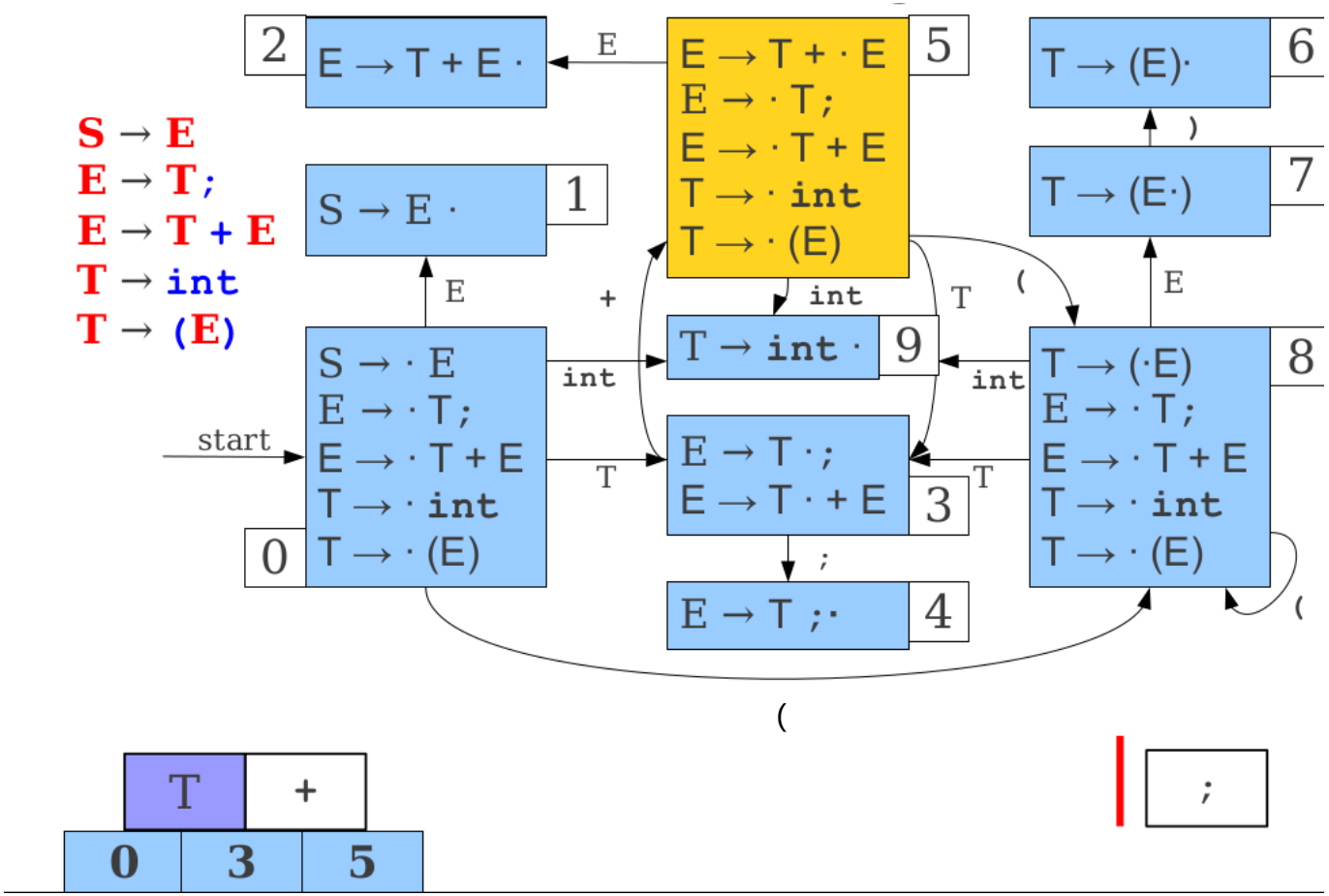
LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
2
E → T + E ·
E
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
5
T → (E)·
6
)
T → (E·)
7
S → E ·
1
E
+
int
T
(
E
S → · E
E → · T;
E → · T + E
T → · int
T → · (E)
0
start
int
T → int ·
9
int
T → (·E)
E → · T;
E → · T + E
T → · int
T → · (E)
8
E → T ·;
E → T · + E
3
T
;
E → T ;·
4
(
T
+
(
E
)
;
0
3
5
8
7

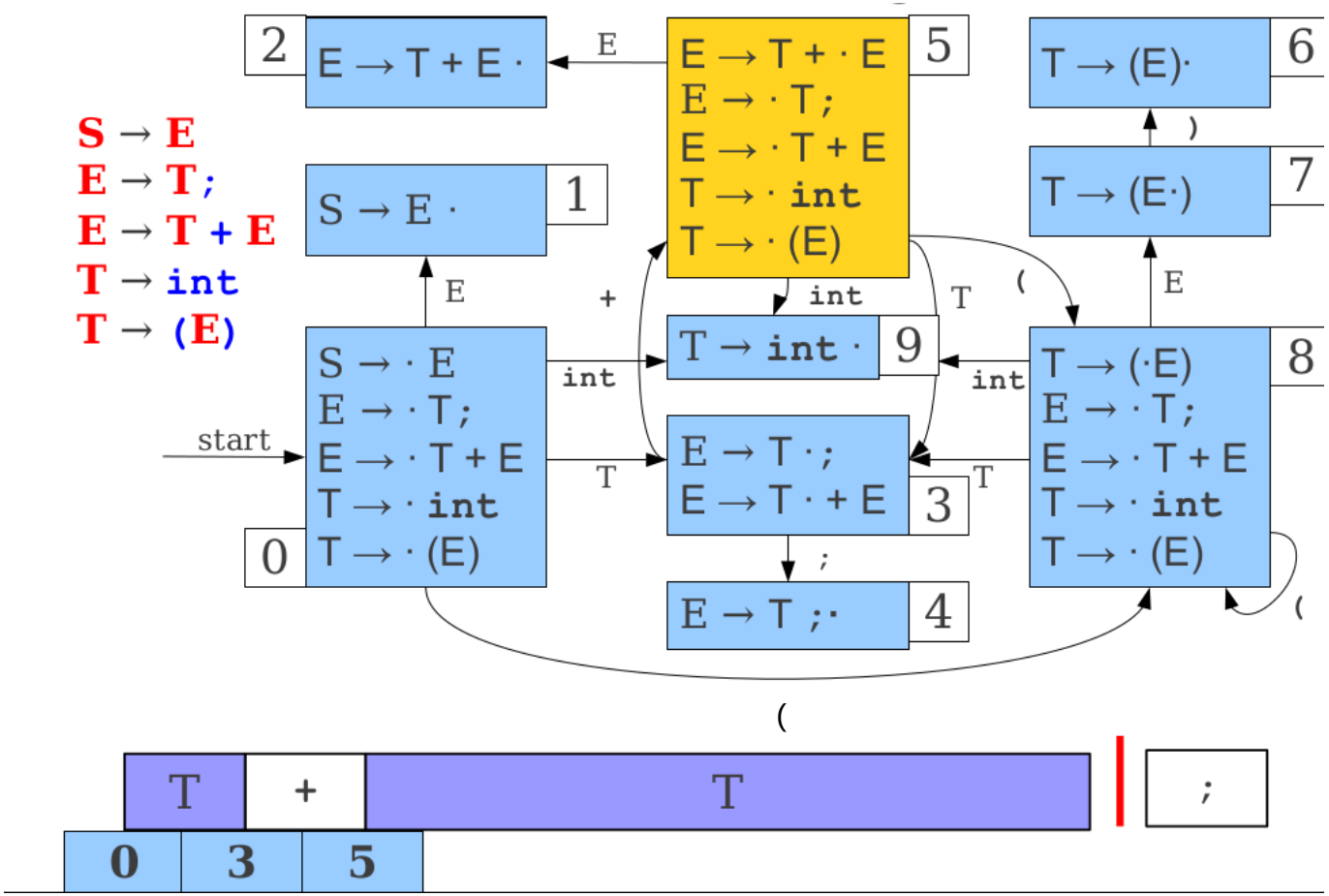
LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
start
0
S → · E
E → · T;
E → · T + E
T → · int
T → · (E)
1
S → E ·
2
E → T + E ·
3
E → T ·;
E → T · + E
4
E → T ;·
5
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
6
T → (E)·
7
T → (E·)
8
T → (·E)
E → · T;
E → · T + E
T → · int
T → · (E)
9
T → int ·
E
+
int
T
;
(
)
T
+
(
E
)
;
0
3
5
8
7

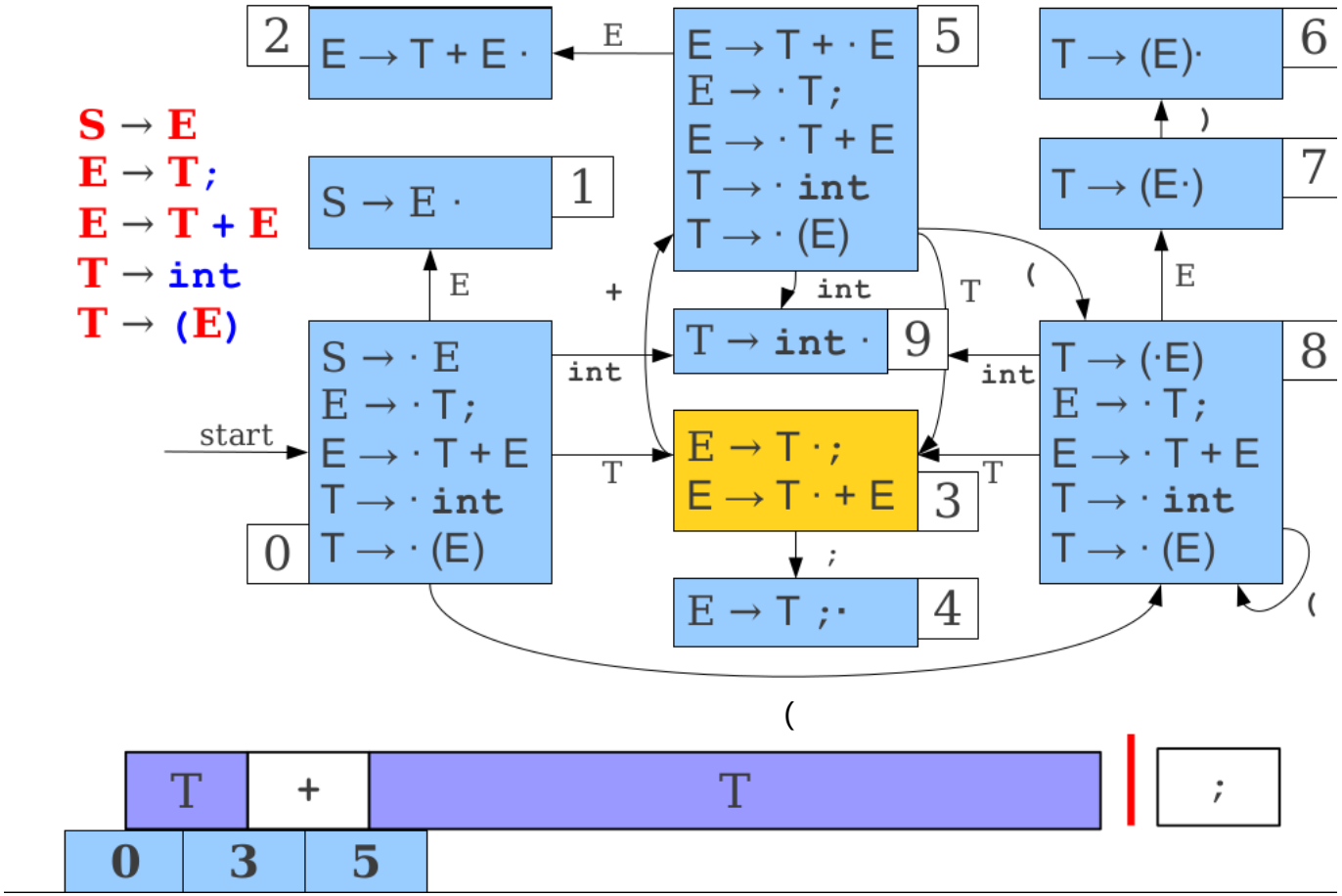
LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
2
E → T + E ·
E
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
5
T → (E)·
6
)
T → (E·)
7
S → E ·
1
E
+
int
T
(
E
T → int ·
9
int
T → (·E)
E → · T;
E → · T + E
T → · int
T → · (E)
8
start
S → · E
E → · T;
E → · T + E
T → · int
T → · (E)
0
int
T
E → T ·;
E → T · + E
3
T
;
E → T ;·
4
(
(
T
+
0
3
5
;

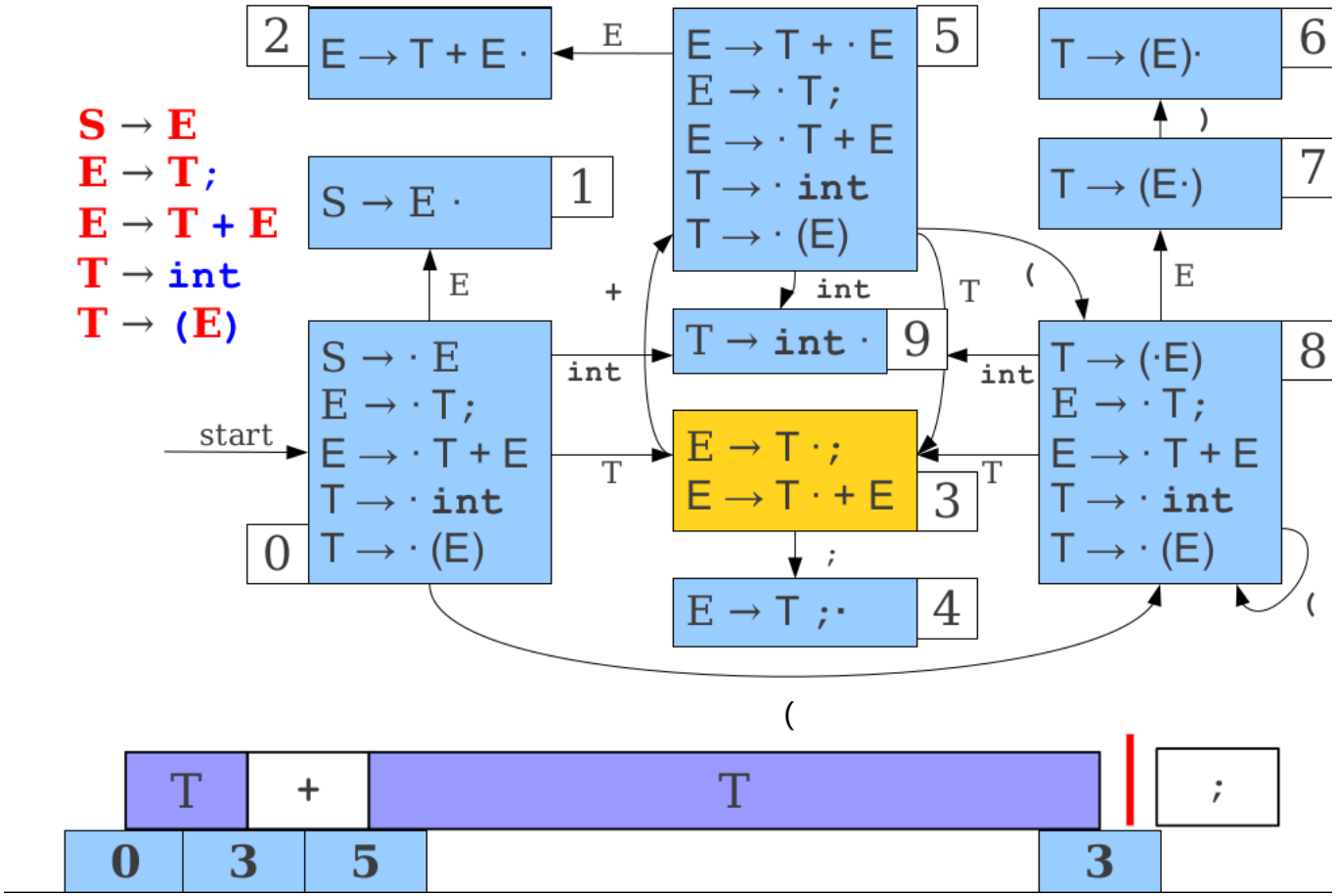
LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
2
E → T + E ·
E
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
5
T → (E)·
6
)
T → (E·)
7
S → E ·
1
E
+
int
T
(
E
T → int ·
9
int
T → (·E)
E → · T;
E → · T + E
T → · int
T → · (E)
8
start
S → · E
E → · T;
E → · T + E
T → · int
T → · (E)
0
int
T
E → T ·;
E → T · + E
3
T
;
E → T ;·
4
(
(
T
+
T
;
0
3
5

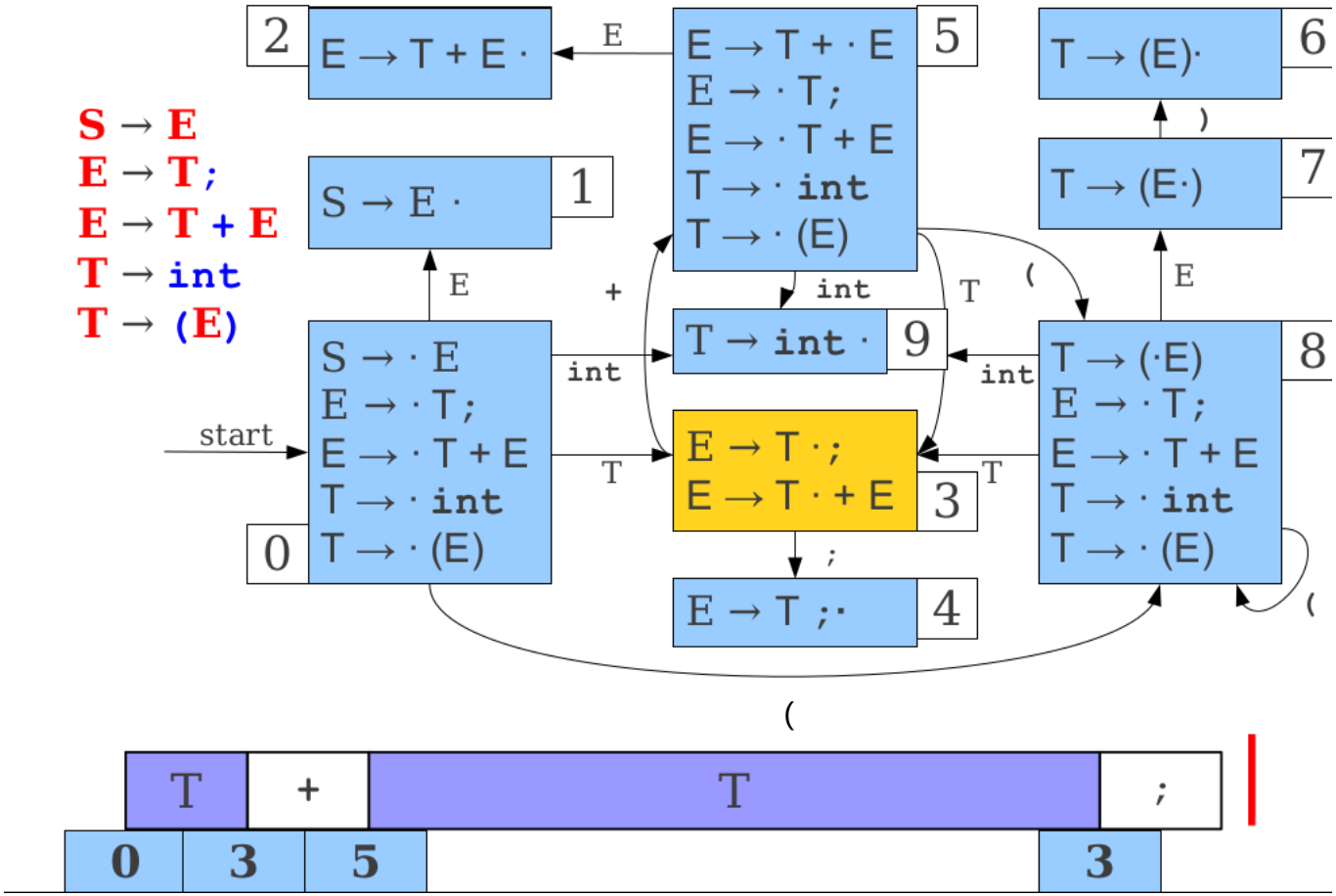
LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
2
E → T + E ·
E
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
5
T → (E)·
6
)
T → (E·)
7
S → E ·
1
E
+
int
T
(
E
S → · E
E → · T;
E → · T + E
T → · int
T → · (E)
0
start
int
T → int ·
9
int
T → (·E)
E → · T;
E → · T + E
T → · int
T → · (E)
8
E → T ·;
E → T · + E
3
T
;
E → T ;·
4
(
(
T
+
T
;
0
3
5

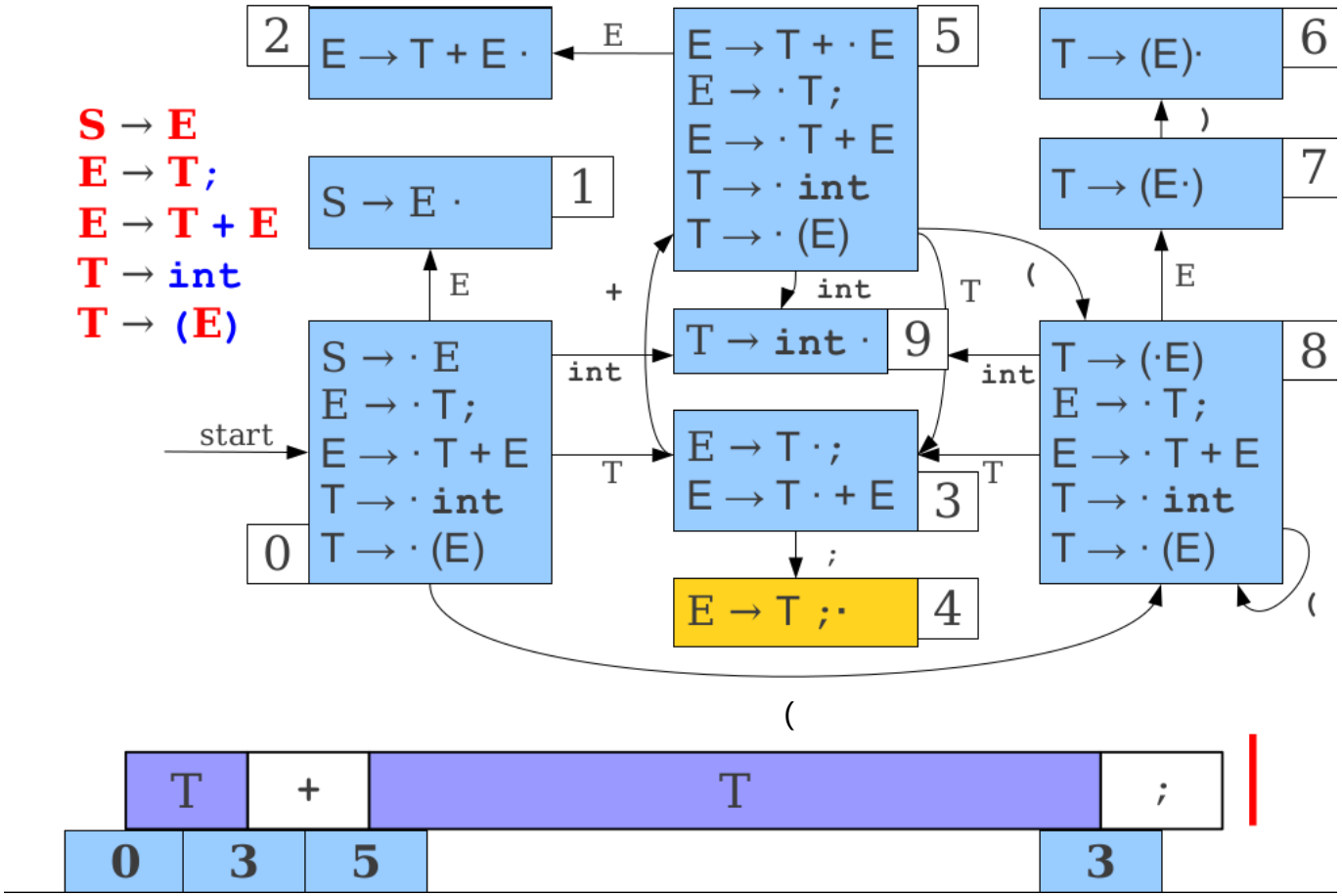
LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
2
E → T + E ·
E
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
5
T → (E)·
6
)
T → (E·)
7
S → E ·
1
E
+
int
T
(
E
S → · E
E → · T;
E → · T + E
T → · int
T → · (E)
0
start
int
T
T → int ·
9
int
T → (·E)
E → · T;
E → · T + E
T → · int
T → · (E)
8
E → T ·;
E → T · + E
3
T
;
E → T ;·
4
(
(
T
+
T
;
0
3
5
3

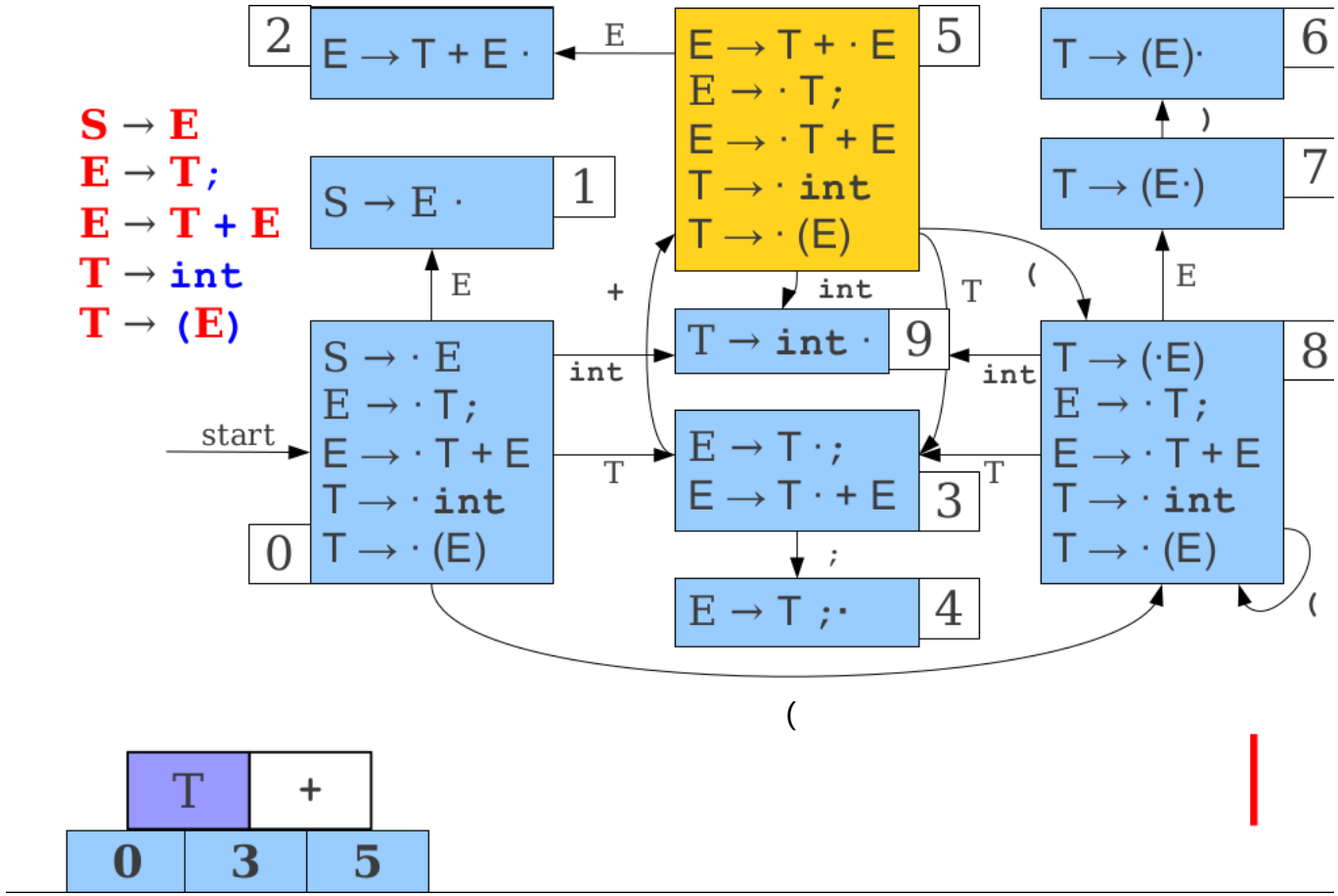
LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
2
E → T + E ·
E
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
5
T → (E)·
6
)
T → (E·)
7
S → E ·
1
E
+
int
T
(
E
S → · E
E → · T;
E → · T + E
T → · int
T → · (E)
start
0
int
T → int ·
9
int
T → (·E)
E → · T;
E → · T + E
T → · int
T → · (E)
8
E → T ·;
E → T · + E
3
T
;
E → T ;·
4
(
(
T
+
T
;
0
3
5
3

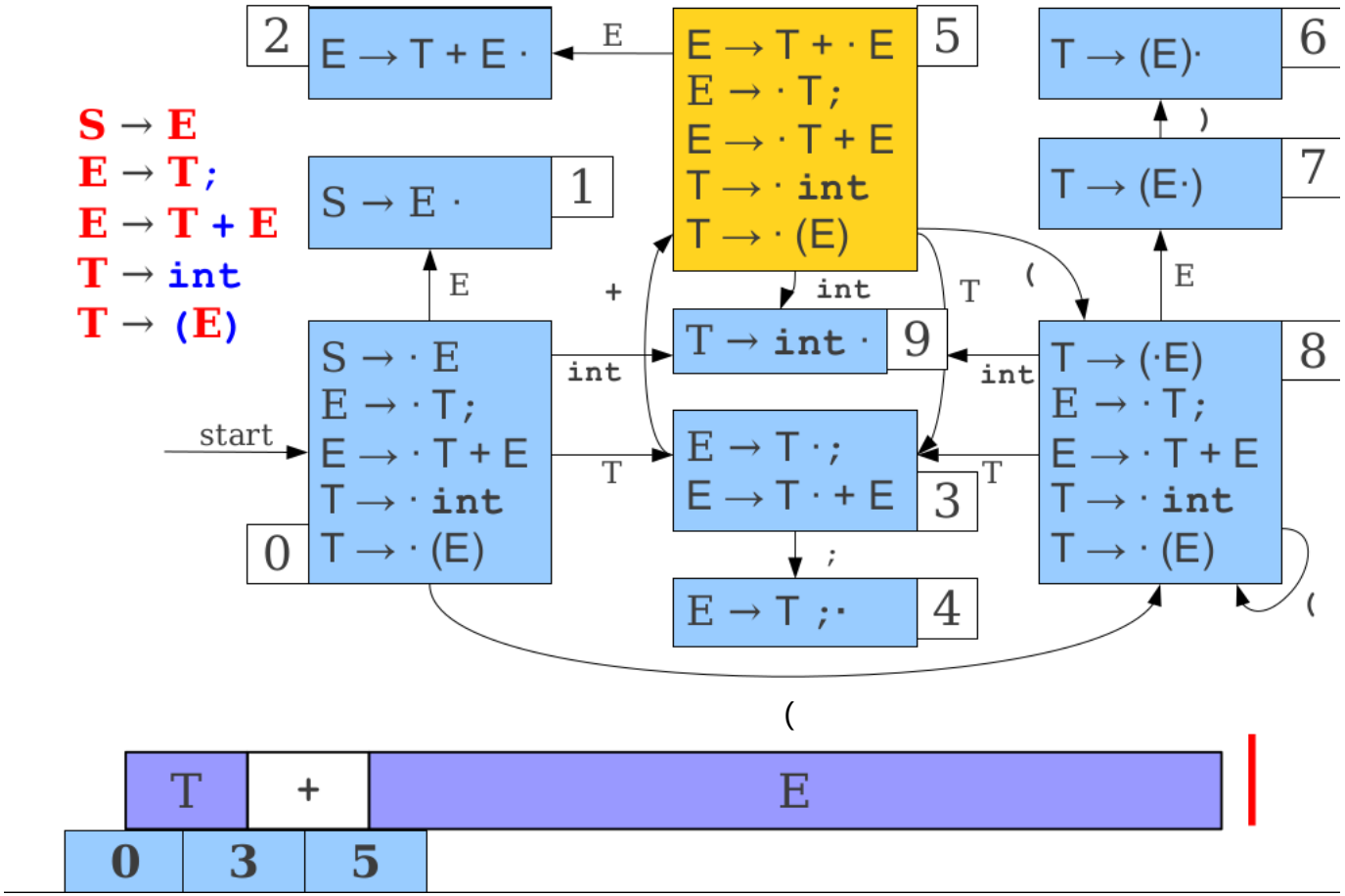
LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
2
E → T + E ·
E
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
5
T → (E)·
6
)
T → (E·)
7
S → E ·
1
E
+
int
T
(
E
T → int ·
9
int
T → (·E)
E → · T;
E → · T + E
T → · int
T → · (E)
8
start
S → · E
E → · T;
E → · T + E
T → · int
T → · (E)
0
int
T
E → T ·;
E → T · + E
3
T
;
E → T ;·
4
(
(
T
+
T
;
0
3
5
3

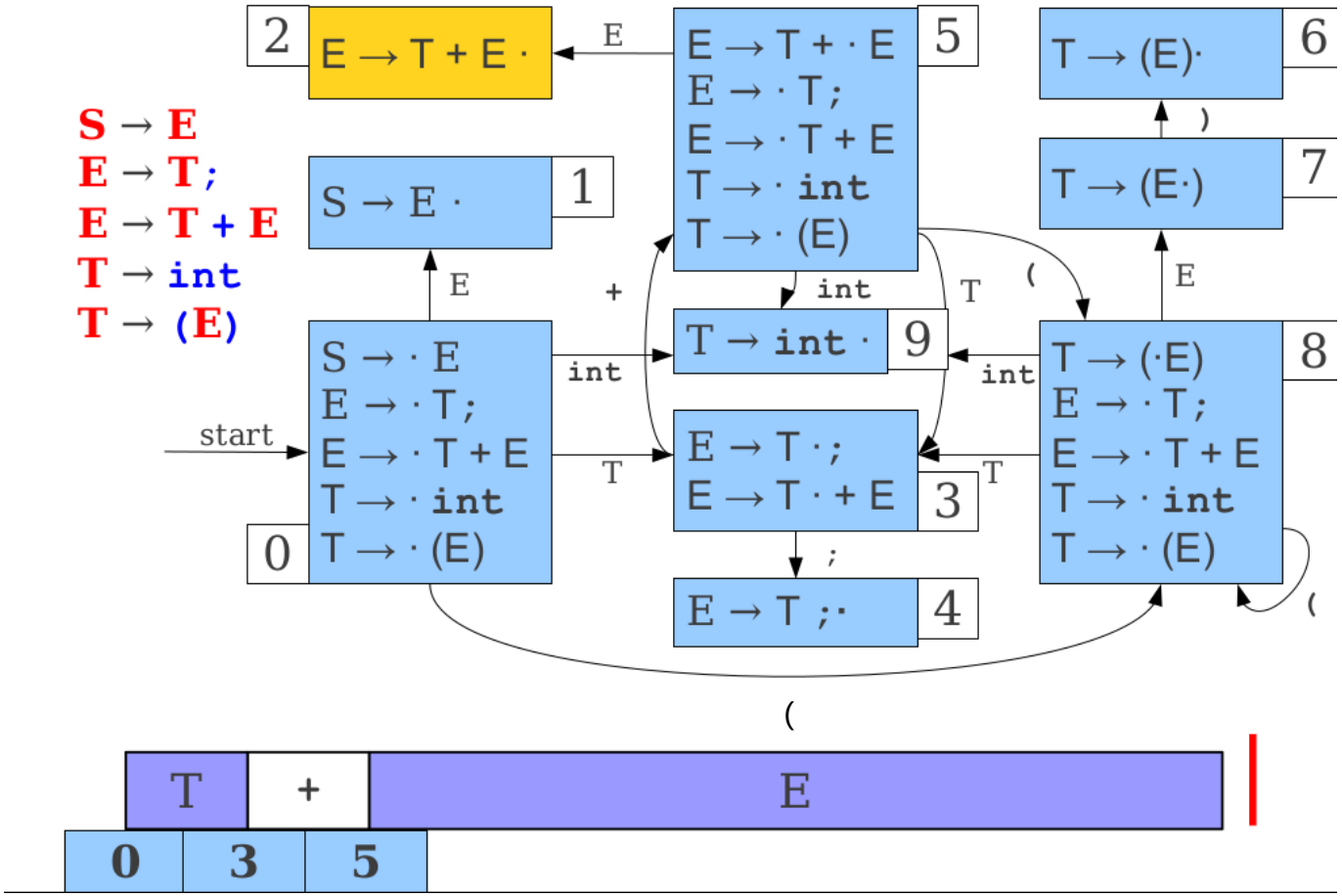
LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
2
E → T + E ·
E
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
5
T → (E)·
6
)
T → (E·)
7
S → E ·
1
E
+
int
T
(
E
T → int ·
9
int
T → (·E)
E → · T;
E → · T + E
T → · int
T → · (E)
8
start
S → · E
E → · T;
E → · T + E
T → · int
T → · (E)
0
int
T
E → T ·;
E → T · + E
3
T
;
E → T ;·
4
(
(
T
+
0
3
5

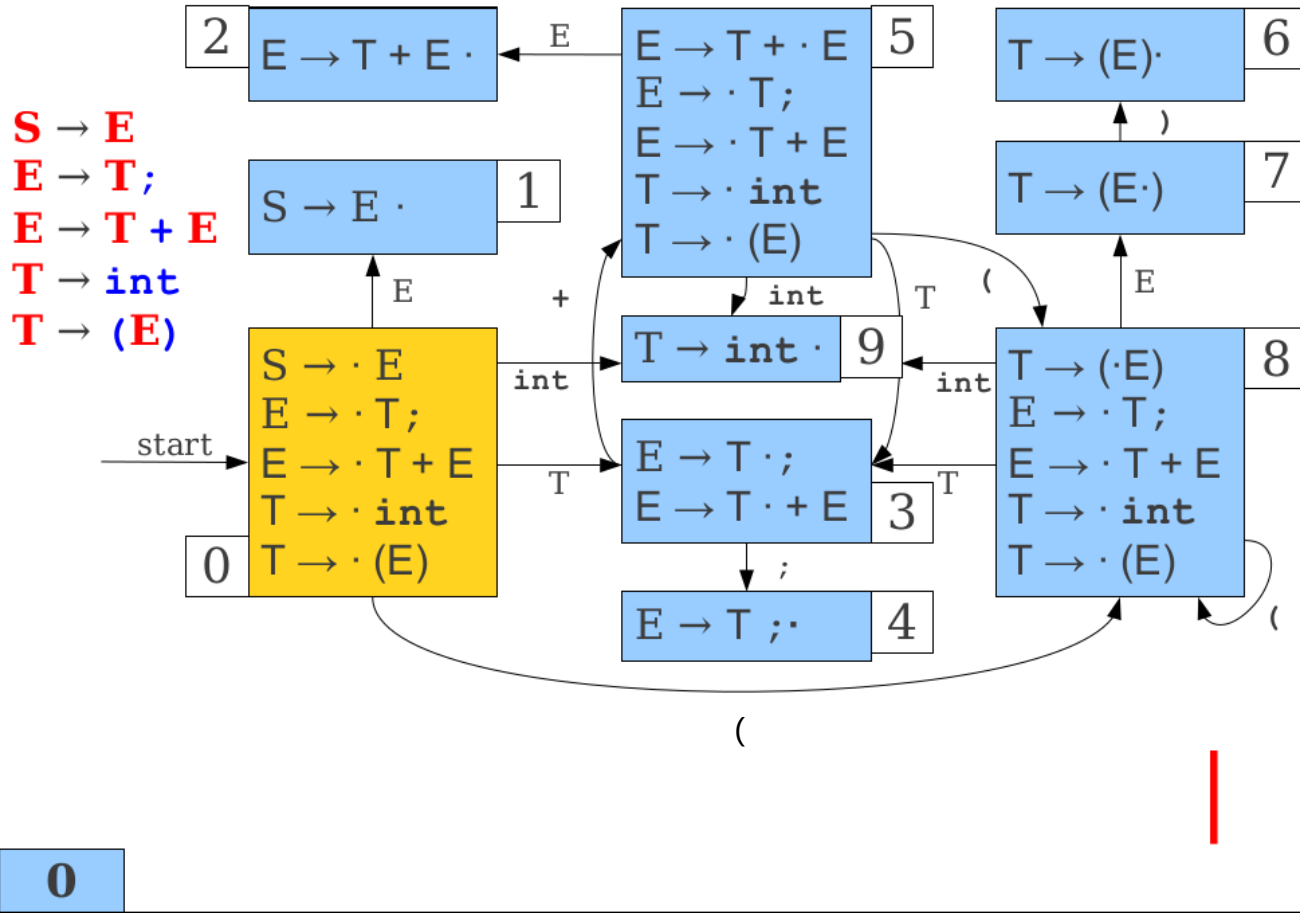
LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
start
0
S → · E
E → · T;
E → · T + E
T → · int
T → · (E)
1
S → E ·
2
E → T + E ·
3
E → T · ;
E → T · + E
4
E → T ; ·
5
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
6
T → (E)·
7
T → (E·)
8
T → (·E)
E → · T;
E → · T + E
T → · int
T → · (E)
9
T → int ·
E
+
int
T
;
(
)
T
+
E
0
3
5

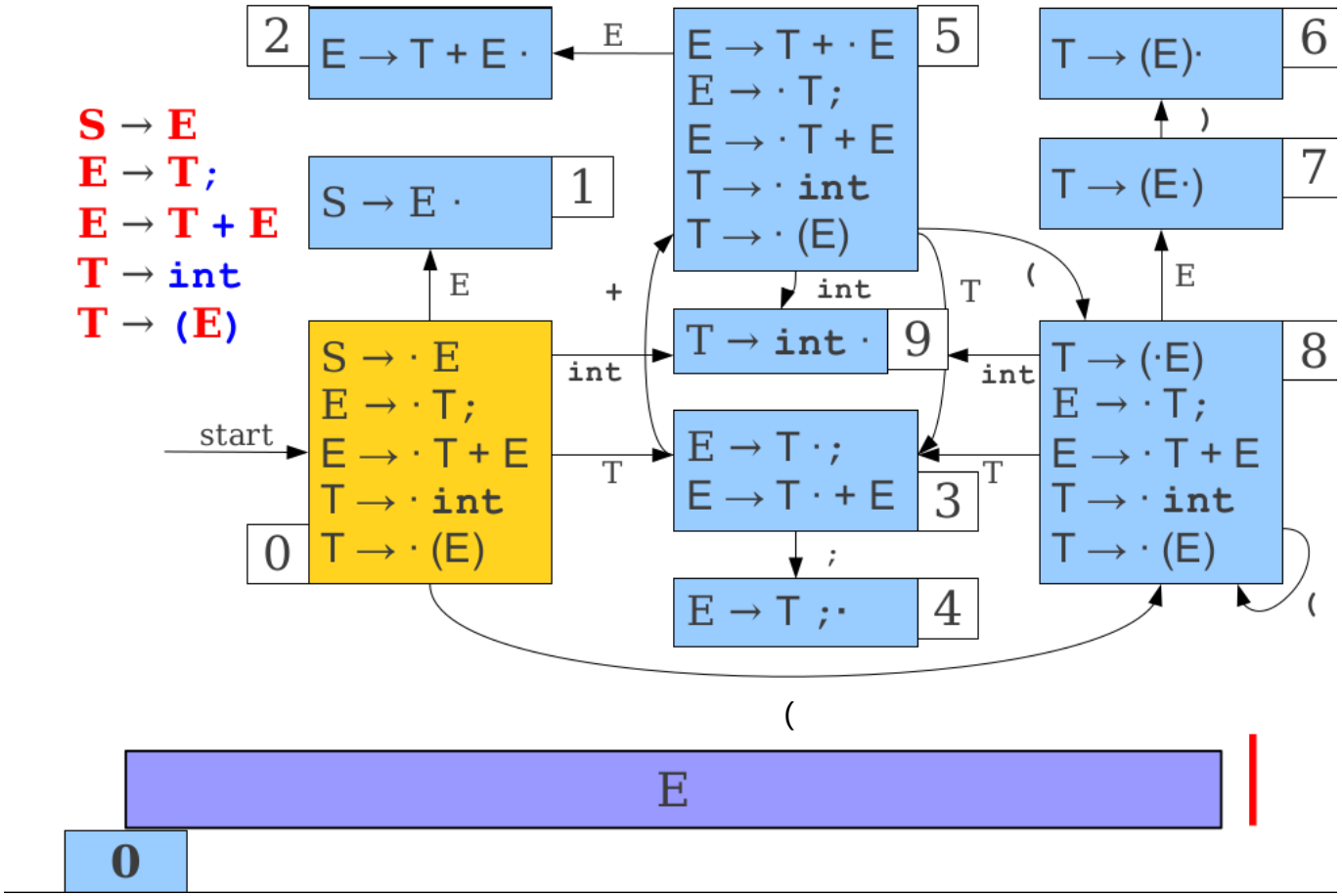
LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
2
E → T + E ·
E
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
5
T → (E)·
6
)
T → (E·)
7
S → E ·
1
E
+
int
T
(
E
T → int ·
9
int
T → (·E)
E → · T;
E → · T + E
T → · int
T → · (E)
8
start
S → · E
E → · T;
E → · T + E
T → · int
T → · (E)
0
int
T
E → T ·;
E → T · + E
3
T
;
E → T ;·
4
(
(
T
+
E
0
3
5

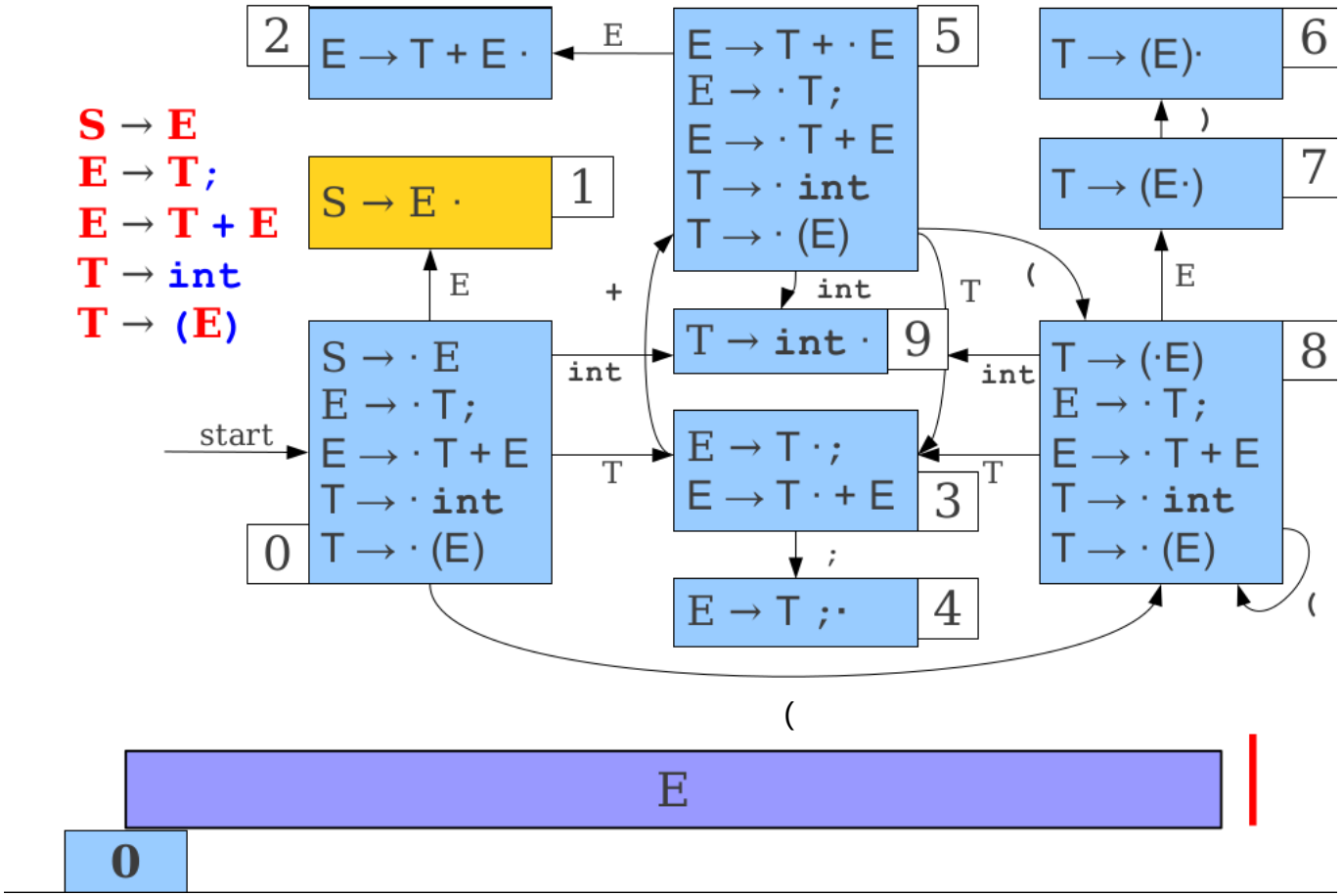
LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
2
E → T + E ·
E
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
5
T → (E)·
6
)
S → E ·
1
T → (E·)
7
E
+
int
T
(
E
T → int ·
9
int
T → (·E)
E → · T;
E → · T + E
T → · int
T → · (E)
8
start
S → · E
E → · T;
E → · T + E
T → · int
T → · (E)
0
int
T
E → T ·;
E → T · + E
3
T
;
E → T ;·
4
(
(
0

LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
start
0
S → · E
E → · T;
E → · T + E
T → · int
T → · (E)
1
S → E ·
2
E → T + E ·
3
E → T · ;
E → T · + E
4
E → T ; ·
5
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
6
T → (E) ·
7
T → (E ·)
8
T → (· E)
E → · T;
E → · T + E
T → · int
T → · (E)
9
T → int ·
E
+
int
T
;
(
)
E
0

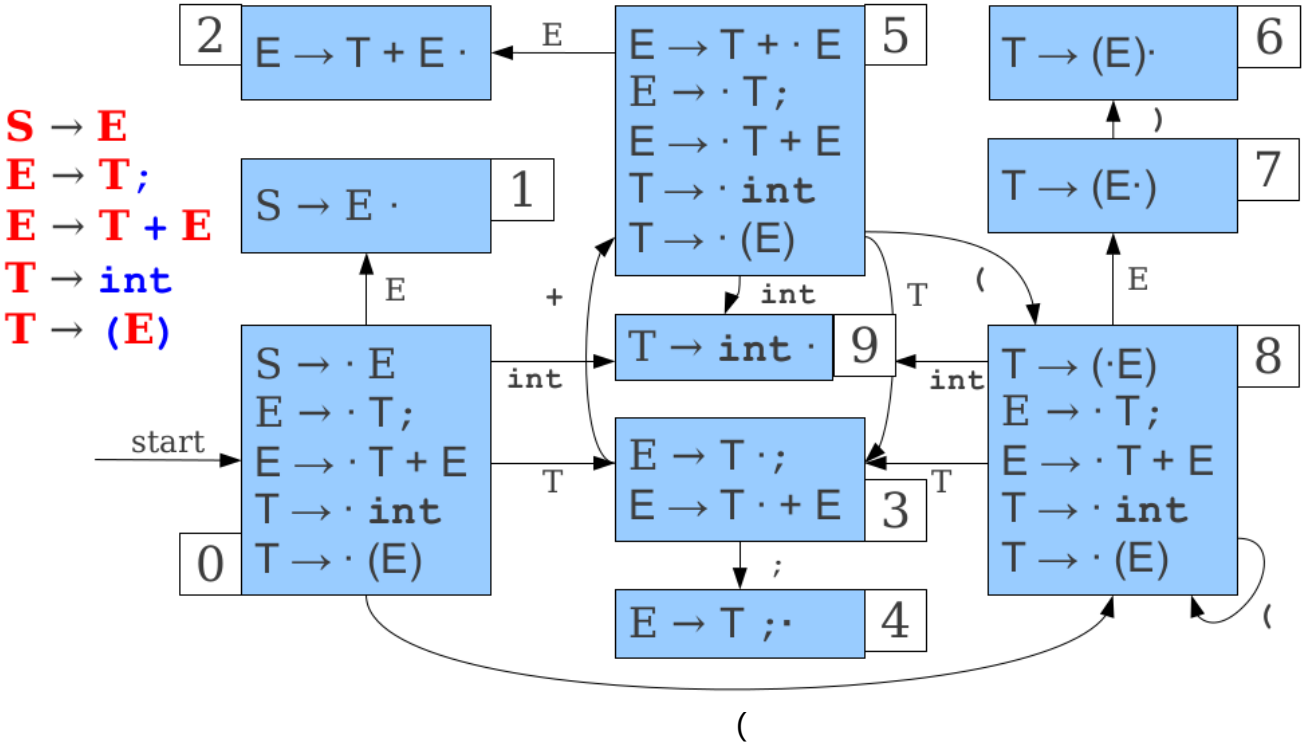
LR(0) Parsing


S → E
E → T;
E → T + E
T → int
T → (E)
start
0
S → · E
E → · T;
E → · T + E
T → · int
T → · (E)
1
S → E ·
2
E → T + E ·
3
E → T · ;
E → T · + E
4
E → T ; ·
5
E → T + · E
E → · T;
E → · T + E
T → · int
T → · (E)
6
T → (E) ·
7
T → (E ·)
8
T → (· E)
E → · T;
E → · T + E
T → · int
T → · (E)
9
T → int ·
E
+
int
T
(
)
;
E
0

Representing the Automaton

- LR(0) parsers are usually represented via two tables: an action table and a goto table.
- The action table maps each state to an action:
  - shift, which shifts the next terminal, and
  - reduce $A \rightarrow \omega$, which performs reduction $A \rightarrow \omega$.
  - Any state of the form $A \rightarrow \omega \cdot$ does that reduction; everything else shifts.
- The goto table maps state/symbol pairs to a next state.
  - This is just the transition table for the automaton.

Building LR(0) Tables



LR(0) Tables

	int	+	;	(	)	E	T	Action
0	9			8		1	3	Shift
1								Accept
2								Reduce **E** → **T + E**
3		5	4					Shift
4								Reduce **E** → **T ;**
5	9			8		2	3	Shift
6								Reduce **T** → **(E)**
7					6			Shift
8	9			8		7	3	Shift
9								Reduce **T** → **int**

The LR(0) Algorithm

- Maintain a stack of (symbol, state) pairs, which is initially (?, 1) for some dummy symbol ?.
- While the stack is not empty:
  - Let state be the top state.
  - If action[state] is shift:
    - Let t be the next symbol in the input.
    - Push (t, goto[state, t]) atop the stack.
  - If action[state] is reduce $A \rightarrow \omega$:
    - Remove $|\omega|$ symbols from the top of the stack.
    - Let top-state be the state on top of the stack.
    - Push (A, goto[top-state, A]) atop the stack.
  - Otherwise, report an error.

LR(0) Parser

- Input: (int;);

Stack	Input	Action
0	(int;);$	

	int	+	;	(	)	E	T	Action
0	9			8		1	3	Shift
1								Accept
2								Reduce E → T + E
3		5	4					Shift
4								Reduce E → T ;
5	9			8		2	3	Shift
6								Reduce T → (E)
7					6			Shift
8	9			8		7	3	Shift
9								Reduce T → int

S → E
E → T;
E → T + E
T → int
T → (E)

LR(0) Parser

- Input: (int;);

Stack	Input	Action
0 0(8	(int;);$ int;);$	s8

	int	+	;	(	)	E	T	Action
0	9			8		1	3	Shift
1								Accept
2								Reduce E → T + E
3		5	4					Shift
4								Reduce E → T ;
5	9			8		2	3	Shift
6								Reduce T → (E)
7					6			Shift
8	9			8		7	3	Shift
9								Reduce T → int

S → E
E → T;
E → T + E
T → int
T → (E)

LR(0) Parser

- Input: (int;);

Stack	Input	Action
0	(int;);$	s8
0(8	int;);$	s9
0(8int9	;);$	

	int	+	;	(	)	E	T	Action
0	9			8		1	3	Shift
1								Accept
2								Reduce E → T + E
3		5	4					Shift
4								Reduce E → T ;
5	9			8		2	3	Shift
6								Reduce T → (E)
7					6			Shift
8	9			8		7	3	Shift
9								Reduce T → int

S → E
E → T;
E → T + E
T → int
T → (E)

LR(0) Parser

- Input: (int;);

Stack	Input	Action
0	(int;);$	s8
0(8	int;);$	s9
0(8int9	;);$	r4
0(8T	;);$	

	int	+	;	(	)	E	T	Action
0	9			8		1	3	Shift
1								Accept
2								Reduce E → T + E
3		5	4					Shift
4								Reduce E → T ;
5	9			8		2	3	Shift
6								Reduce T → (E)
7					6			Shift
8	9			8		7	3	Shift
9								Reduce T → int

S → E
E → T;
E → T + E
T → int
T → (E)

LR(0) Parser

- Input: (int;);

Stack	Input	Action
0	(int;);$	s8
0(8	int;);$	s9
0(8int9	;);$	r4 (g3)
0(8T3	;);$	

	int	+	;	(	)	E	T	Action
0	9			8		1	3	Shift
1								Accept
2								Reduce E → T + E
3		5	4					Shift
4								Reduce E → T ;
5	9			8		2	3	Shift
6								Reduce T → (E)
7					6			Shift
8	9			8		7	3	Shift
9								Reduce T → int

S → E
E → T;
E → T + E
T → int
T → (E)

LR(0) Parser

- Input: (int;);

Stack	Input	Action
0	(int;);$	s8
0(8	int;);$	s9
0(8int9	;);$	r4 (g3)
0(8T3	;);$	s4
0(8T3;4	);$	

	int	+	;	(	)	E	T	Action
0	9			8		1	3	Shift
1								Accept
2								Reduce E → T + E
3		5	4					Shift
4								Reduce E → T ;
5	9			8		2	3	Shift
6								Reduce T → (E)
7					6			Shift
8	9			8		7	3	Shift
9								Reduce T → int

S → E
E → T;
E → T + E
T → int
T → (E)

LR(0) Parser

- Input: (int;);

Stack	Input	Action
0	(int;);$	s8
0(8	int;);$	s9
0(8int9	;);$	r4 (g3)
0(8T3	;);$	s4
0(8T3;4	);$	r2
0(8E	);$	

	int	+	;	(	)	E	T	Action
0	9			8		1	3	Shift
1								Accept
2								Reduce E → T + E
3		5	4					Shift
4								Reduce E → T ;
5	9			8		2	3	Shift
6								Reduce T → (E)
7					6			Shift
8	9			8		7	3	Shift
9								Reduce T → int

S → E
E → T;
E → T + E
T → int
T → (E)

LR(0) Parser

- Input: (int;);

Stack	Input	Action
0	(int;);$	s8
0(8	int;);$	s9
0(8int9	;);$	r4 (g3)
0(8T3	;);$	s4
0(8T3;4	);$	r2 (g7)
0(8E7	);$	

	int	+	;	(	)	E	T	Action
0	9			8		1	3	Shift
1								Accept
2								Reduce E → T + E
3		5	4					Shift
4								Reduce E → T ;
5	9			8		2	3	Shift
6								Reduce T → (E)
7					6			Shift
8	9			8		7	3	Shift
9								Reduce T → int

S → E
E → T;
E → T + E
T → int
T → (E)

LR(0) Parser

- Input: (int;);

Stack	Input	Action
0	(int;);$	s8
0(8	int;);$	s9
0(8int9	;);$	r4 (g3)
0(8T3	;);$	s4
0(8T3;4	);$	r2 (g7)
0(8E7	);$	s6
0(8E7)6	;$	

	int	+	;	(	)	E	T	Action
0	9			8		1	3	Shift
1								Accept
2								Reduce E → T + E
3		5	4					Shift
4								Reduce E → T ;
5	9			8		2	3	Shift
6								Reduce T → (E)
7					6			Shift
8	9			8		7	3	Shift
9								Reduce T → int

S → E
E → T;
E → T + E
T → int
T → (E)

LR(0) Parser

- Input: (int;);

Stack	Input	Action
0	(int;);$	s8
0(8	int;);$	s9
0(8int9	;);$	r4 (g3)
0(8T3	;);$	s4
0(8T3;4	);$	r2 (g7)
0(8E7	);$	s6
0(8E7)6	;$	r5
0T	;$	

	int	+	;	(	)	E	T	Action
0	9			8		1	3	Shift
1								Accept
2								Reduce E → T + E
3		5	4					Shift
4								Reduce E → T ;
5	9			8		2	3	Shift
6								Reduce T → (E)
7					6			Shift
8	9			8		7	3	Shift
9								Reduce T → int

S → E
E → T;
E → T + E
T → int
T → (E)

LR(0) Parser

- Input: (int;);

Stack	Input	Action
0	(int;);$	s8
0(8	int;);$	s9
0(8int9	;);$	r4 (g3)
0(8T3	;);$	s4
0(8T3;4	);$	r2 (g7)
0(8E7	);$	s6
0(8E7)6	;$	r5 (g3)
0T3	;$	

	int	+	;	(	)	E	T	Action
0	9			8		1	3	Shift
1								Accept
2								Reduce E → T + E
3		5	4					Shift
4								Reduce E → T ;
5	9			8		2	3	Shift
6								Reduce T → (E)
7					6			Shift
8	9			8		7	3	Shift
9								Reduce T → int

S → E
E → T;
E → T + E
T → int
T → (E)

LR(0) Parser

- Input: (int;);

Stack	Input	Action
0	(int;);$	s8
0(8	int;);$	s9
0(8int9	;);$	r4 (g3)
0(8T3	;);$	s4
0(8T3;4	);$	r2 (g7)
0(8E7	);$	s6
0(8E7)6	;$	r5 (g3)
0T3	;$	s4
0T3;4	$	

	int	+	;	(	)	E	T	Action
0	9			8		1	3	Shift
1								Accept
2								Reduce E → T + E
3		5	4					Shift
4								Reduce E → T ;
5	9			8		2	3	Shift
6								Reduce T → (E)
7					6			Shift
8	9			8		7	3	Shift
9								Reduce T → int

S → E
E → T;
E → T + E
T → int
T → (E)

LR(0) Parser

- Input: (int;);

Stack	Input	Action
0	(int;);$	s8
0(8	int;);$	s9
0(8int9	;);$	r4 (g3)
0(8T3	;);$	s4
0(8T3;4	);$	r2 (g7)
0(8E7	);$	s6
0(8E7)6	;$	r5 (g3)
0T3	;$	s4
0T3;4	$	r2
0E	$	

	int	+	;	(	)	E	T	Action
0	9			8		1	3	Shift
1								Accept
2								Reduce E → T + E
3		5	4					Shift
4								Reduce E → T ;
5	9			8		2	3	Shift
6								Reduce T → (E)
7					6			Shift
8	9			8		7	3	Shift
9								Reduce T → int

S → E
E → T;
E → T + E
T → int
T → (E)

LR(0) Parser

- Input: (int;);

Stack	Input	Action
0	(int;);$	s8
0(8	int;);$	s9
0(8int9	;);$	r4 (g3)
0(8T3	;);$	s4
0(8T3;4	);$	r2 (g7)
0(8E7	);$	s6
0(8E7)6	;$	r5 (g3)
0T3	;$	s4
0T3;4	$	r2 (g1)
0E1	$	

	int	+	;	(	)	E	T	Action
0	9			8		1	3	Shift
1								Accept
2								Reduce E → T + E
3		5	4					Shift
4								Reduce E → T ;
5	9			8		2	3	Shift
6								Reduce T → (E)
7					6			Shift
8	9			8		7	3	Shift
9								Reduce T → int

S → E
E → T;
E → T + E
T → int
T → (E)

LR(0) Parser

- Input: (int;);

Stack	Input	Action
0	(int;);$	s8
0(8	int;);$	s9
0(8int9	;);$	r4 (g3)
0(8T3	;);$	s4
0(8T3;4	);$	r2 (g7)
0(8E7	);$	s6
0(8E7)6	;$	r5 (g3)
0T3	;$	s4
0T3;4	$	r2 (g1)
0E1	$	Accept

	int	+	;	(	)	E	T	Action
0	9			8		1	3	Shift
1								Accept
2								Reduce E → T + E
3		5	4					Shift
4								Reduce E → T ;
5	9			8		2	3	Shift
6								Reduce T → (E)
7					6			Shift
8	9			8		7	3	Shift
9								Reduce T → int

S → E
E → T;
E → T + E
T → int
T → (E)

LR(0) Parser

(1) **S** → **X**

(2) **X** → **(X)**

(3) **X** → **( )**

0

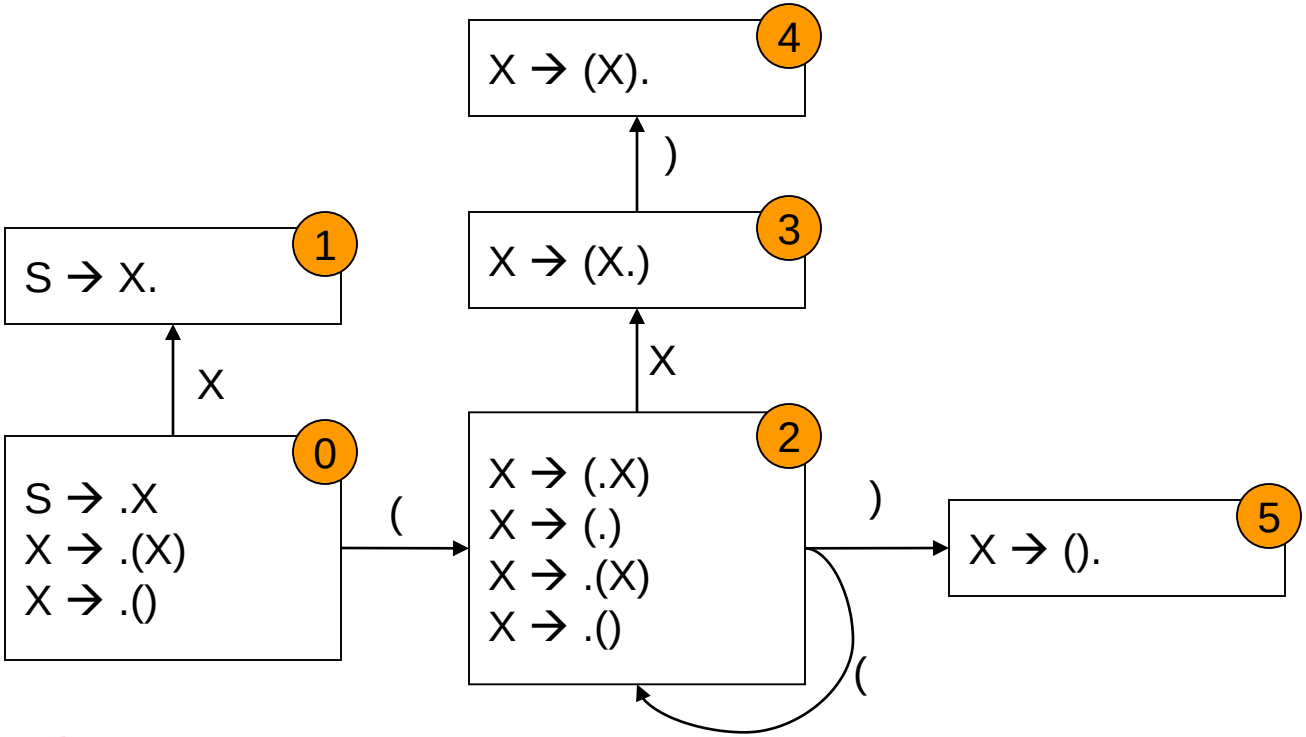
S → .X
X → .(X)
X → .()

LR(0) Parser

(1) **S → X**

(2) **X → (X)**

(3) **X → ( )**



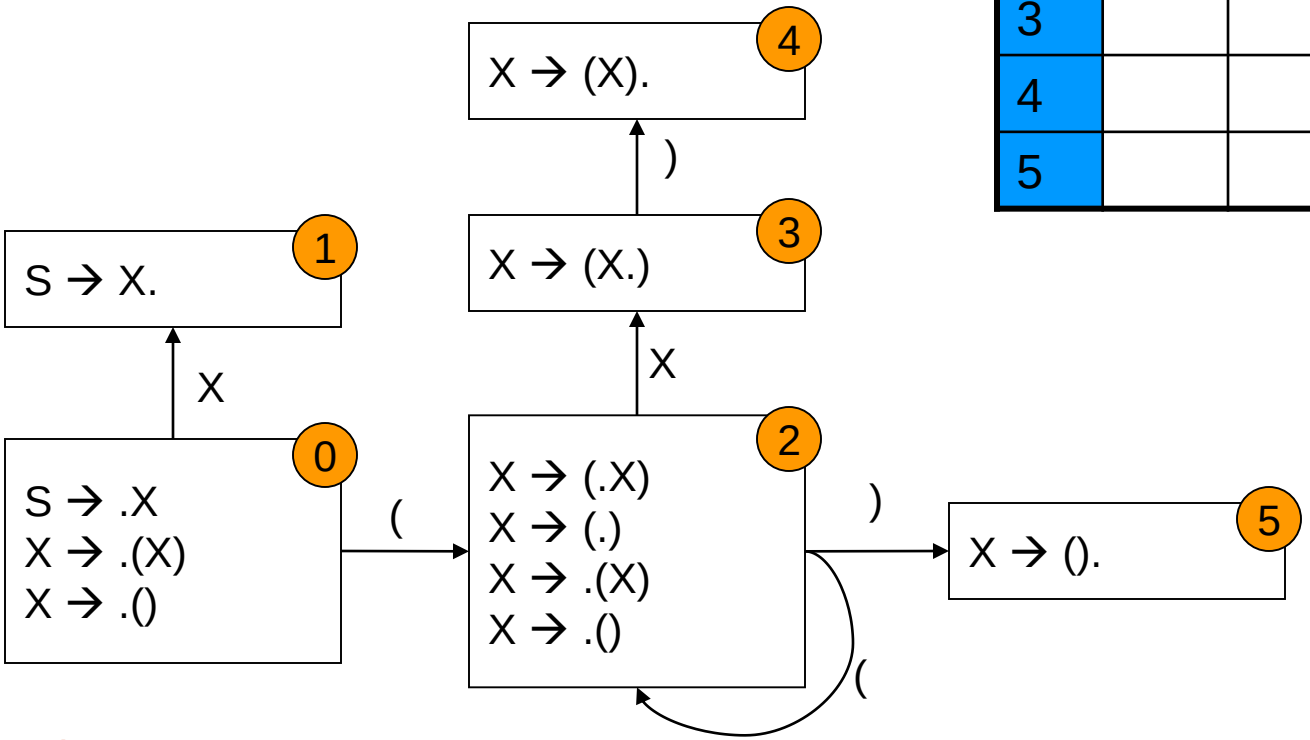
LR(0) Parser

(1) **S → X**

(2) **X → (X)**

(3) **X → ( )**

	(	)	x	ACTION
0				
1				
2				
3				
4				
5				



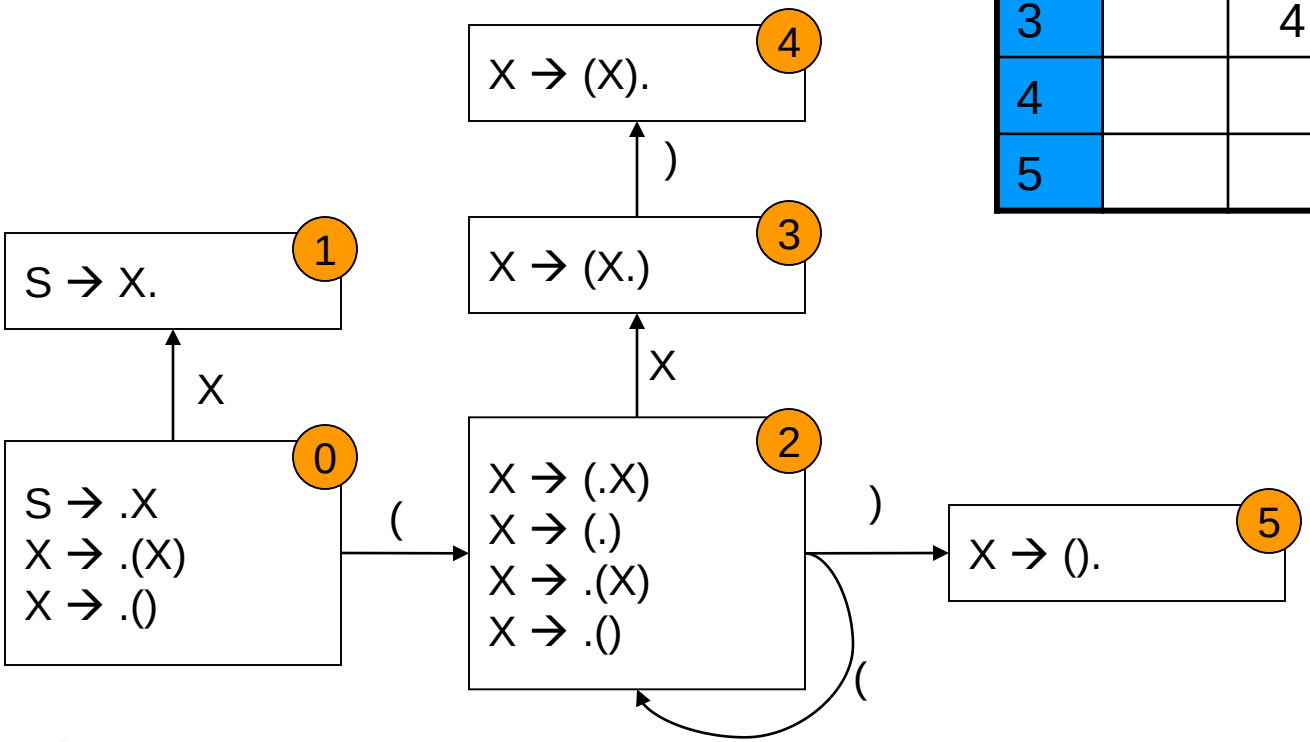
LR(0) Parser

(1) **S → X**

(2) **X → (X)**

(3) **X → ( )**

	(	)	x	ACTION
0	2		1	Shift
1				Accept
2	2	5	3	Shift
3		4		Shift
4				Reduce 2
5				Reduce 3



LR(0) Parser

- Input: (())

	(	)	x	ACTION
0	2		1	Shift
1				Accept
2	2	5	3	Shift
3		4		Shift
4				Reduce 2
5				Reduce 3

Stack	Input	Action
0	(())$	

(1) $\mathbf{S} \rightarrow \mathbf{X}$

(2) $\mathbf{X} \rightarrow \mathbf{(X)}$

(3) $\mathbf{X} \rightarrow \mathbf{( )}$

LR(0) Parser

- Input: (())

	(	)	x	ACTION
0	2		1	Shift
1				Accept
2	2	5	3	Shift
3		4		Shift
4				Reduce 2
5				Reduce 3

Stack	Input	Action
0 0(2	(())$ ())$	s2

(1) $\mathbf{S} \rightarrow \mathbf{X}$

(2) $\mathbf{X} \rightarrow \mathbf{(X)}$

(3) $\mathbf{X} \rightarrow \mathbf{( )}$

LR(0) Parser

- Input: (())

	(	)	x	ACTION
0	2		1	Shift
1				Accept
2	2	5	3	Shift
3		4		Shift
4				Reduce 2
5				Reduce 3

Stack	Input	Action
0	(())$	s2
0(2	())$	s2
0(2(2	))$	

(1) $\mathbf{S} \rightarrow \mathbf{X}$

(2) $\mathbf{X} \rightarrow \mathbf{(X)}$

(3) $\mathbf{X} \rightarrow \mathbf{( )}$

LR(0) Parser

- Input: (())

	(	)	x	ACTION
0	2		1	Shift
1				Accept
2	2	5	3	Shift
3		4		Shift
4				Reduce 2
5				Reduce 3

Stack	Input	Action
0	(())$	s2
0(2	())$	s2
0(2(2	))$	s5
0(2(2)5	)$	

(1) $\mathbf{S} \rightarrow \mathbf{X}$

(2) $\mathbf{X} \rightarrow \mathbf{(\ X\ )}$

(3) $\mathbf{X} \rightarrow \mathbf{(\ )}$

LR(0) Parser

- Input: (())

	(	)	x	ACTION
0	2		1	Shift
1				Accept
2	2	5	3	Shift
3		4		Shift
4				Reduce 2
5				Reduce 3

Stack	Input	Action
0	(())$	s2
0(2	())$	s2
0(2(2	))$	s5
0(2(2)5	)$	r3
0(2X	)$	

(1) $\mathbf{S} \rightarrow \mathbf{X}$

(2) $\mathbf{X} \rightarrow \mathbf{(\ X\ )}$

(3) $\mathbf{X} \rightarrow \mathbf{(\ )}$

LR(0) Parser

- Input: (())

	(	)	x	ACTION
0	2		1	Shift
1				Accept
2	2	5	3	Shift
3		4		Shift
4				Reduce 2
5				Reduce 3

Stack	Input	Action
0	(())$	s2
0(2	())$	s2
0(2(2	))$	s5
0(2(2)5	)$	r3 (g3)
0(2X3	)$	

(1) $\mathbf{S} \rightarrow \mathbf{X}$

(2) $\mathbf{X} \rightarrow \mathbf{(X)}$

(3) $\mathbf{X} \rightarrow \mathbf{( )}$

LR(0) Parser

- Input: (())

	(	)	x	ACTION
0	2		1	Shift
1				Accept
2	2	5	3	Shift
3		4		Shift
4				Reduce 2
5				Reduce 3

Stack	Input	Action
0	(())$	s2
0(2	())$	s2
0(2(2	))$	s5
0(2(2)5	)$	r3 (g3)
0(2X3	)$	s4
0(2X3)4	$	

(1) $\mathbf{S} \rightarrow \mathbf{X}$

(2) $\mathbf{X} \rightarrow \mathbf{(\ X\ )}$

(3) $\mathbf{X} \rightarrow \mathbf{(\ )}$

LR(0) Parser

- Input: (())

	(	)	x	ACTION
0	2		1	Shift
1				Accept
2	2	5	3	Shift
3		4		Shift
4				Reduce 2
5				Reduce 3

Stack	Input	Action
0	(())$	s2
0(2	())$	s2
0(2(2	))$	s5
0(2(2)5	)$	r3 (g3)
0(2X3	)$	s4
0(2X3)4	$	r2
0X	$	

(1) $\mathbf{S} \rightarrow \mathbf{X}$

(2) $\mathbf{X} \rightarrow \mathbf{(\ X\ )}$

(3) $\mathbf{X} \rightarrow \mathbf{(\ )}$

LR(0) Parser

- Input: (())

	(	)	x	ACTION
0	2		1	Shift
1				Accept
2	2	5	3	Shift
3		4		Shift
4				Reduce 2
5				Reduce 3

Stack	Input	Action
0	(())$	s2
0(2	())$	s2
0(2(2	))$	s5
0(2(2)5	)$	r3 (g3)
0(2X3	)$	s4
0(2X3)4	$	r2 (g1)
0X1	$	

(1) $\mathbf{S} \rightarrow \mathbf{X}$

(2) $\mathbf{X} \rightarrow \mathbf{(\ X\ )}$

(3) $\mathbf{X} \rightarrow \mathbf{(\ )}$

LR(0) Parser

- Input: (())

	(	)	x	ACTION
0	2		1	Shift
1				Accept
2	2	5	3	Shift
3		4		Shift
4				Reduce 2
5				Reduce 3

Stack	Input	Action
0	(())$	s2
0(2	())$	s2
0(2(2	))$	s5
0(2(2)5	)$	r3 (g3)
0(2X3	)$	s4
0(2X3)4	$	r2 (g1)
0X1	$	Accept

(1) $\mathbf{S} \rightarrow \mathbf{X}$

(2) $\mathbf{X} \rightarrow \mathbf{(\ X\ )}$

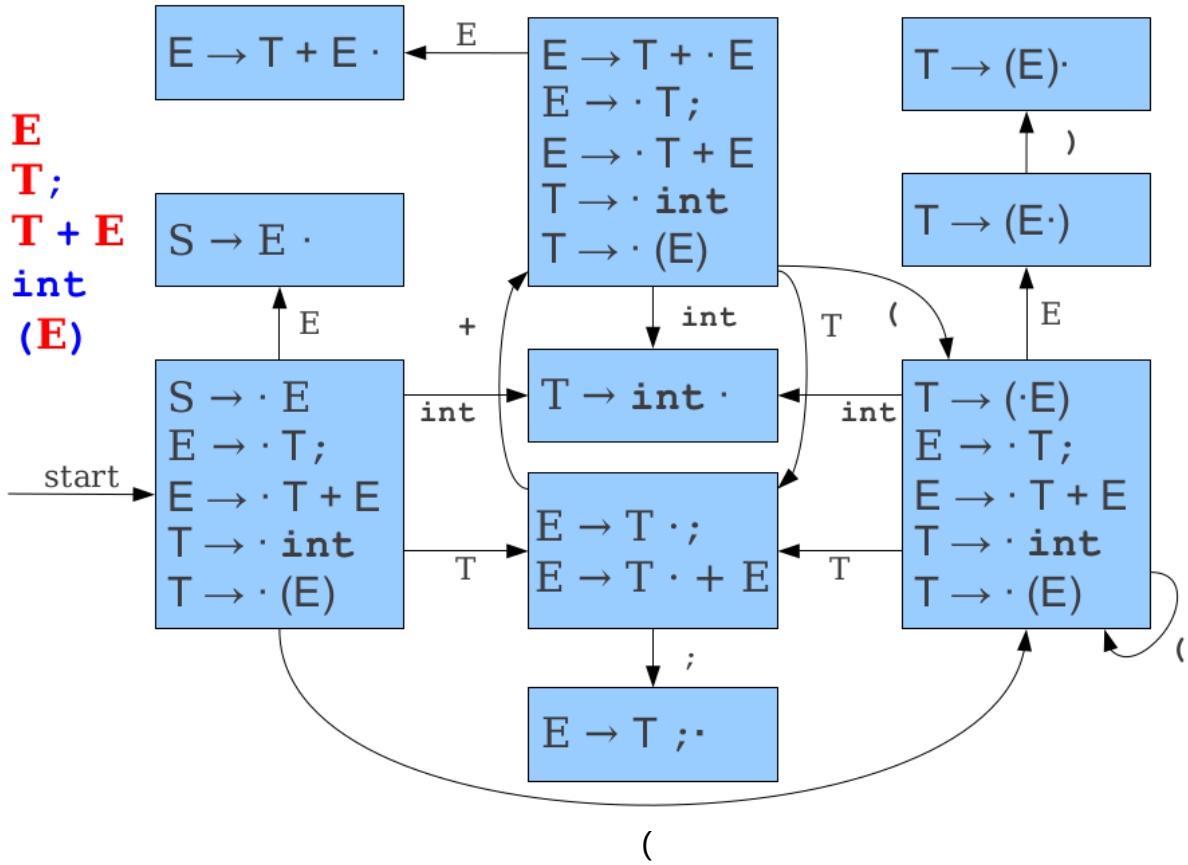
(3) $\mathbf{X} \rightarrow \mathbf{(\ )}$

The Limits of LR(0)

- A Non-LR(0) Grammar

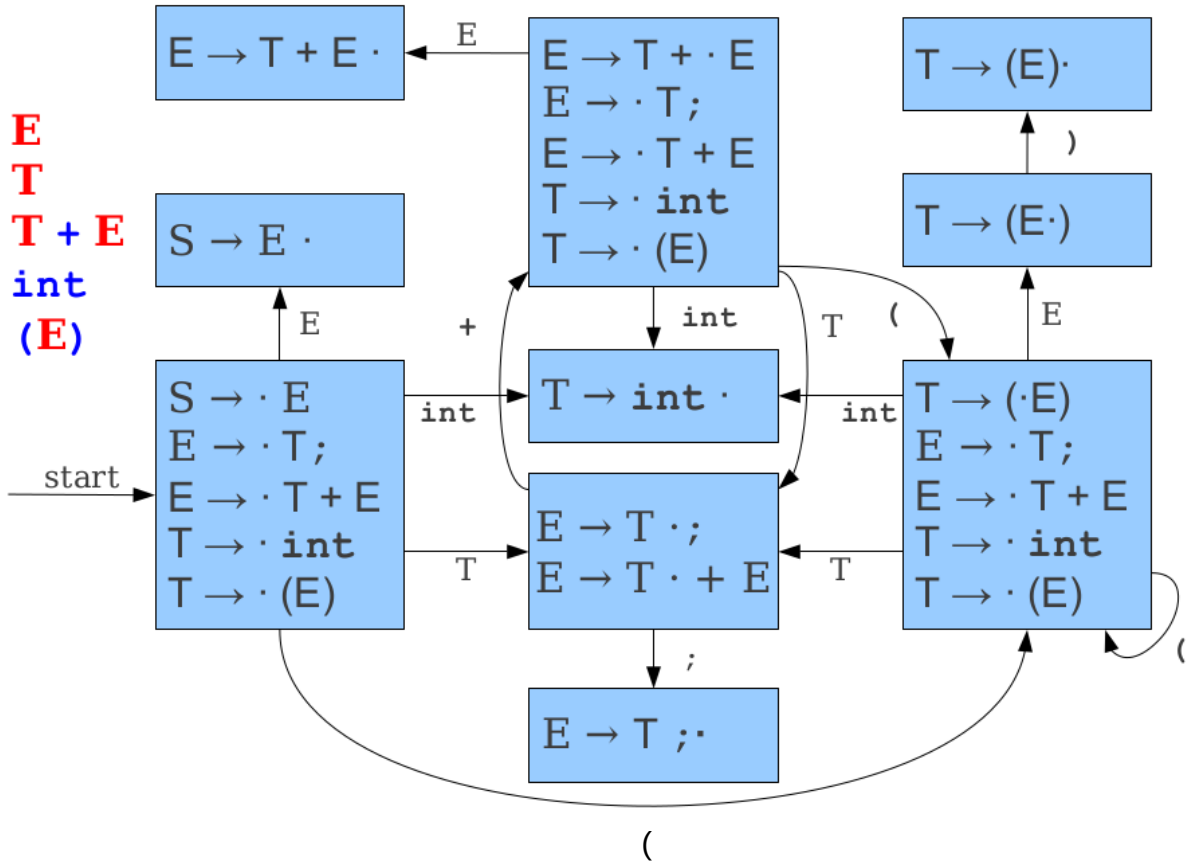
A Non-LR(0) Grammar

S → E
E → T;
E → T + E
T → int
T → (E)



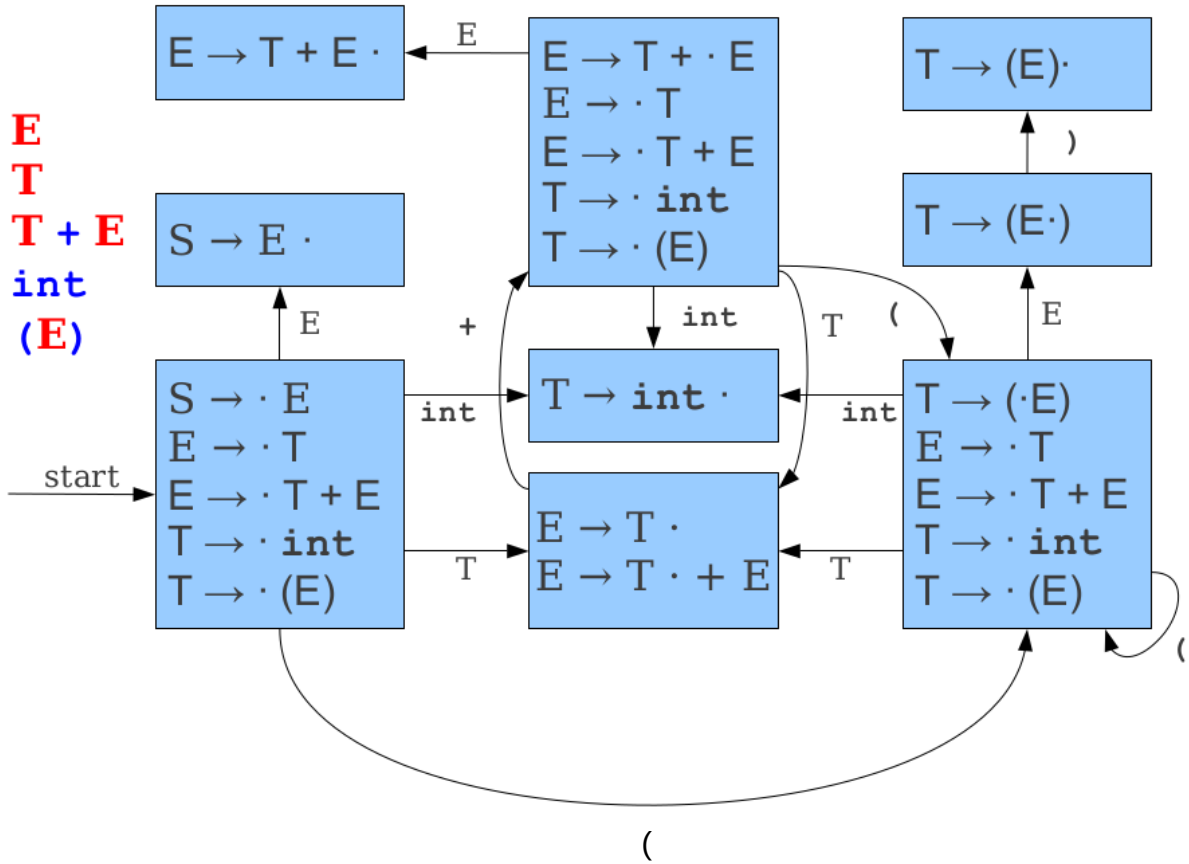
A Non-LR(0) Grammar

S → E
E → T
E → T + E
T → int
T → (E)



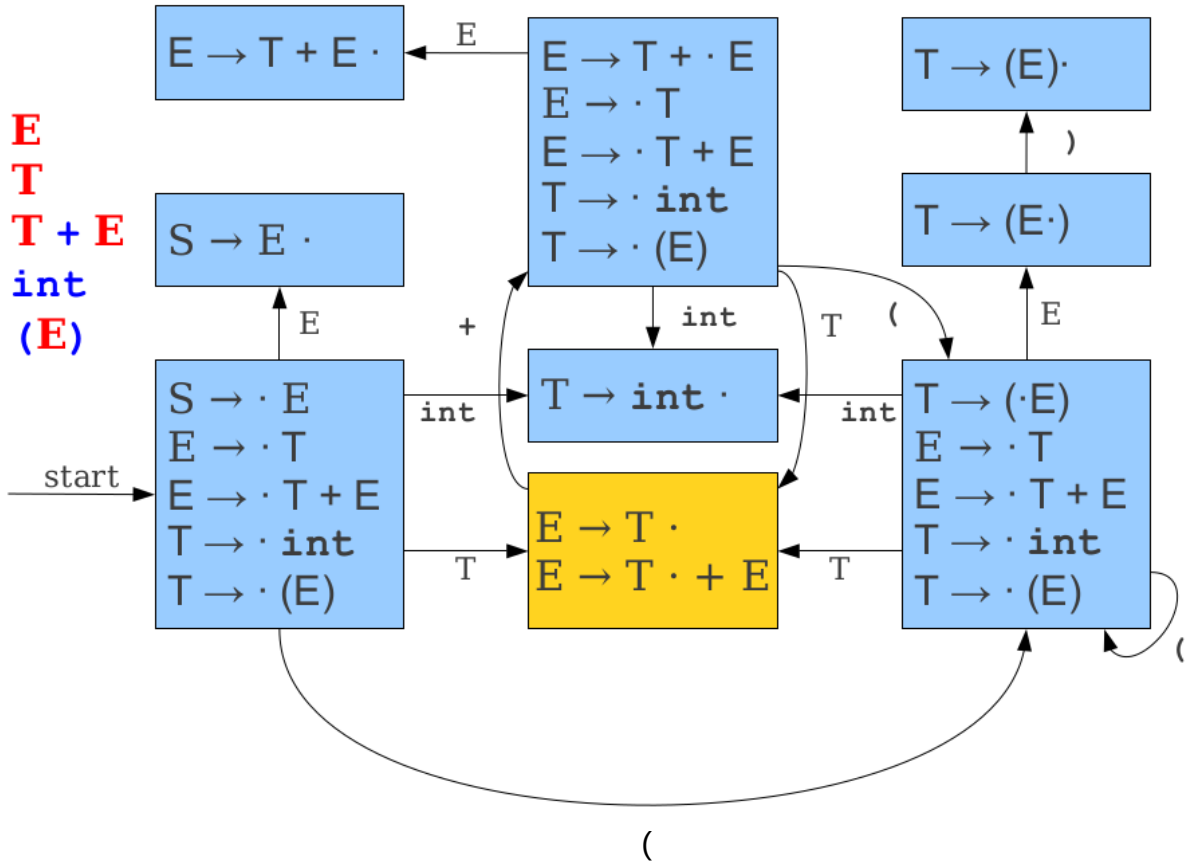
A Non-LR(0) Grammar

S → E
E → T
E → T + E
T → int
T → (E)



A Non-LR(0) Grammar

S → E
E → T
E → T + E
T → int
T → (E)



LR Conflicts

- A shift/reduce conflict is an error where a shift/reduce parser cannot tell whether to shift a token or perform a reduction.
  - Often happens when two productions overlap.
- A reduce/reduce conflict is an error where a shift/reduce parser cannot tell which of many reductions to perform.
  - Often the result of ambiguous grammars.
- A grammar whose handle-finding automaton contains a shift/reduce conflict or a reduce/reduce conflict is not LR(0).

What do these conflicts mean?

- Recall: our automaton was constructed by looking for viable prefixes.
- Each accepting state represents a point where the handle might occur.
- A shift/reduce conflict is a state where the handle might occur, but we might actually need to keep searching.
- A reduce/reduce conflict is a state where we know we have found the handle, but can't tell which reduction to apply.

Why LR(0) is Weak

- LR(0) only accepts languages where the handle can be found with no right context.
- Our shift/reduce parser only looks to the left of the handle, not to the right.
- How do we exploit the tokens after a possible handle to determine what to do?

Outline

- Introduction
- Shift-reduce Parsing
- **LR Parsing**
  - LR(0)
  - **LR(1)**
  - SLR
  - LALR
- Ambiguity
- Error-handling