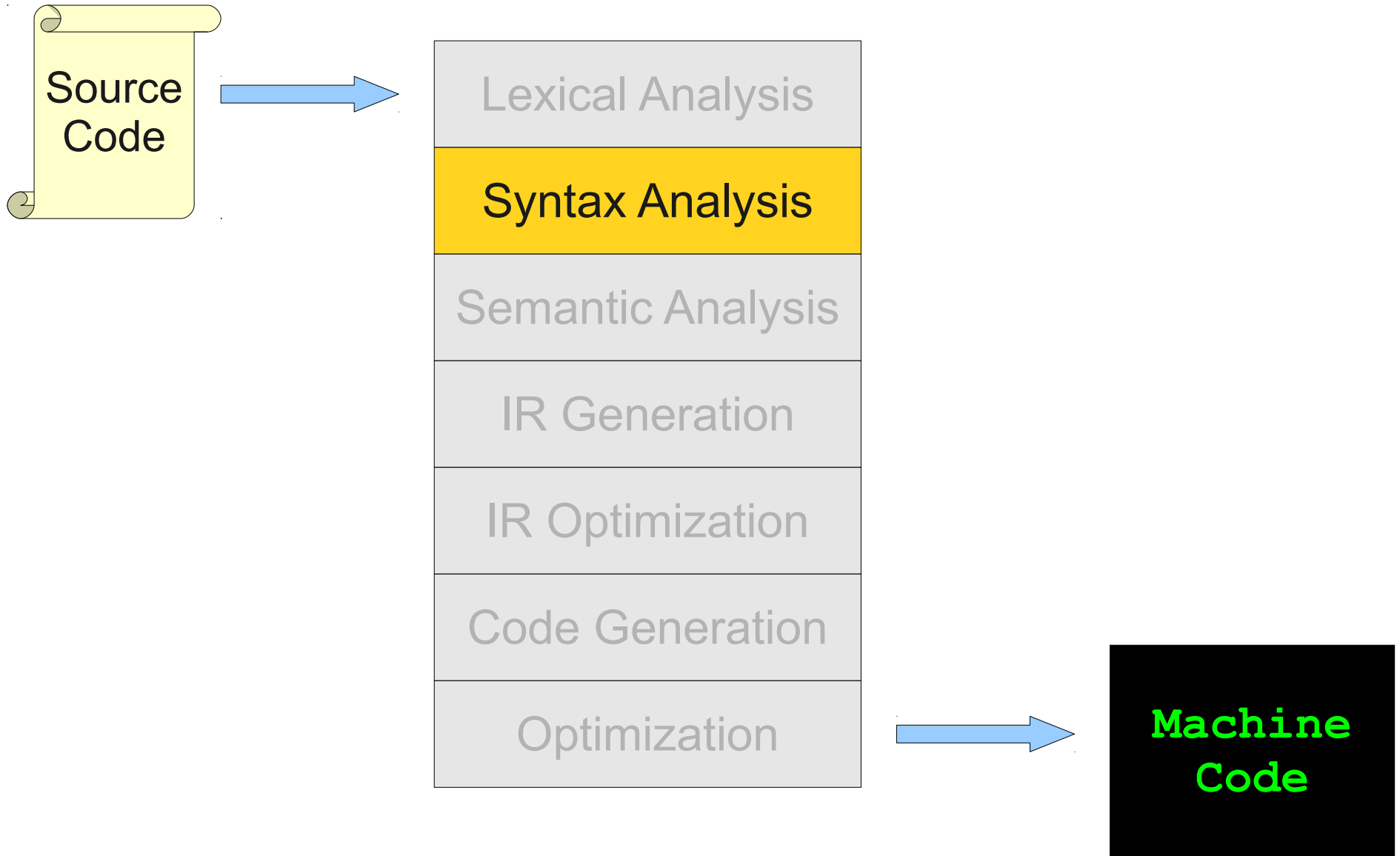


Bottom-Up Parsing II

Announcements

- Programming Project 1 due **tonight** at 11:59PM.
- Written Assignment 1 due next Wednesday, 5:00PM.
- If you need help:
 - Hrysoula will be holding OH from 1:00 – 3:00 today in Gates 24A.
 - Feel free to ask on www.piazza.com
 - Or email the staff directly at cs143-sum1011-staff@lists.stanford.edu.
- Submission instructions on course website.
- Programming Project 2 out, due **Wednesday, July 13** at 11:59PM.

Where We Are



Recap from Last Time

- A left-to-right, bottom-up parse is a **rightmost derivation traced in reverse**.
- A **handle** is a substring of a sentential form that can be reduced without getting stuck.
- Bottom-up parsing is often done as **shift/reduce** parsing:
 - Split the text to parse into two groups.
 - **Shift**: Push a new token atop the stack.
 - **Reduce**: Update the top of the stack by applying a reduction.
- Question: How do we find the handle?

Viable Prefixes

- A **viable prefix** is a string w such that, for some v , $w \mid v$ is a legal configuration in a shift/reduce parser.
- In other words, a viable prefix is
a string of terminals and nonterminals,
that does not extend past a handle,
that can be completed with terminals into a
sentential form.

Critical Fact

The set of viable prefixes for any grammar is a **regular language**.

Critical Fact

The set of viable prefixes for any grammar is a **regular language**.

This is a nontrivial result!

Tracking our Position in Productions


$S \rightarrow E\$$

$E \rightarrow T$

$E \rightarrow E + T$

$T \rightarrow \text{int}$

$T \rightarrow (E)$



<code>int</code>	<code>+</code>	<code>(</code>	<code>int</code>	<code>+</code>	<code>int</code>	<code>)</code>	<code>\$</code>
------------------	----------------	----------------	------------------	----------------	------------------	----------------	-----------------

Tracking our Position in Productions

$S \rightarrow E\$$

$E \rightarrow T$

$E \rightarrow E + T$

$T \rightarrow \text{int}$

$T \rightarrow (E)$

$S \rightarrow \cdot E\$$

| int + (int + int) \$

Tracking our Position in Productions

$S \rightarrow E\$$


$E \rightarrow T$

$E \rightarrow E + T$

$T \rightarrow \text{int}$

$T \rightarrow (E)$

$S \rightarrow \cdot E\$$
$E \rightarrow \cdot E + T$



<code>int</code>	<code>+</code>	<code>(</code>	<code>int</code>	<code>+</code>	<code>int</code>	<code>)</code>	<code>\$</code>
------------------	----------------	----------------	------------------	----------------	------------------	----------------	-----------------

Tracking our Position in Productions

$S \rightarrow E\$$


$E \rightarrow T$

$E \rightarrow E + T$

$T \rightarrow \text{int}$

$T \rightarrow (E)$

$S \rightarrow \cdot E\$$
$E \rightarrow \cdot E + T$
$E \rightarrow \cdot T$



<code>int</code>	<code>+</code>	<code>(</code>	<code>int</code>	<code>+</code>	<code>int</code>	<code>)</code>	<code>\$</code>
------------------	----------------	----------------	------------------	----------------	------------------	----------------	-----------------

Tracking our Position in Productions

$S \rightarrow E\$$

$E \rightarrow T$

$E \rightarrow E + T$

$T \rightarrow \text{int}$

$T \rightarrow (E)$

$S \rightarrow \cdot E\$$
$E \rightarrow \cdot E + T$
$E \rightarrow \cdot T$
$T \rightarrow \cdot \text{int}$

| int + (int + int) \$

Tracking our Position in Productions

$S \rightarrow E\$$

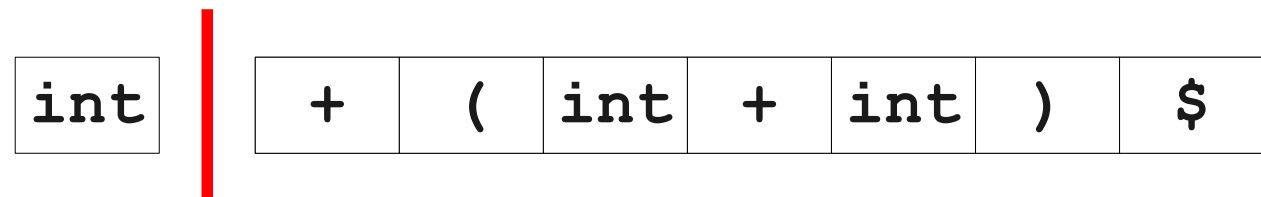
$E \rightarrow T$

$E \rightarrow E + T$

$T \rightarrow \text{int}$

$T \rightarrow (E)$

$S \rightarrow \cdot E\$$
$E \rightarrow \cdot E + T$
$E \rightarrow \cdot T$
$T \rightarrow \cdot \text{int}$



Tracking our Position in Productions

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$

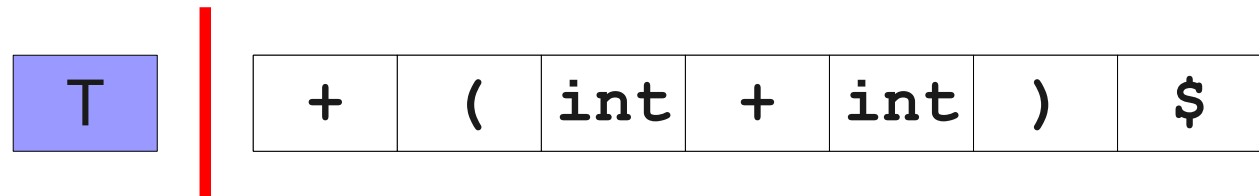
$S \rightarrow \cdot E\$$
$E \rightarrow \cdot E + T$
$E \rightarrow \cdot T$
$T \rightarrow \text{int} \cdot$

int		+	(int	+	int)	\$
-----	--	---	---	-----	---	-----	---	----

Tracking our Position in Productions

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$

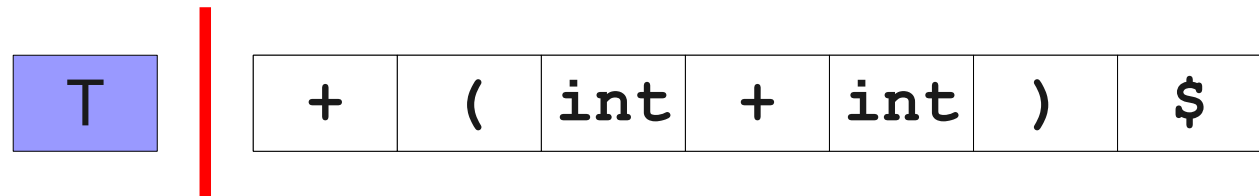
$S \rightarrow \cdot E\$$
$E \rightarrow \cdot E + T$
$E \rightarrow \cdot T$
$T \rightarrow \text{int} \cdot$



Tracking our Position in Productions

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$

$S \rightarrow \cdot E\$$
$E \rightarrow \cdot E + T$
$E \rightarrow T \cdot$



Tracking our Position in Productions

$S \rightarrow E\$$

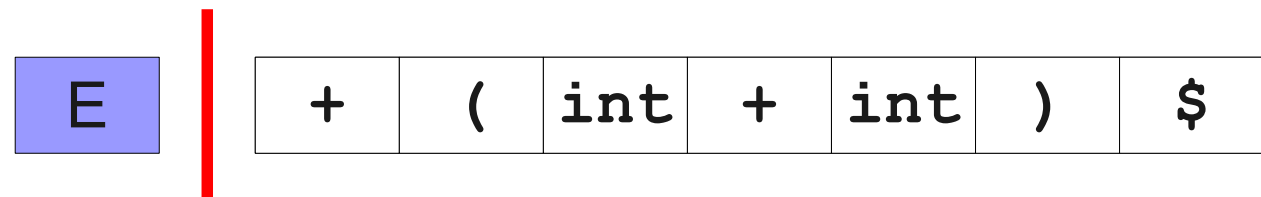
$E \rightarrow T$

$E \rightarrow E + T$

$T \rightarrow \text{int}$

$T \rightarrow (E)$

$S \rightarrow \cdot E\$$
$E \rightarrow \cdot E + T$
$E \rightarrow T \cdot$



Tracking our Position in Productions

$S \rightarrow E\$$

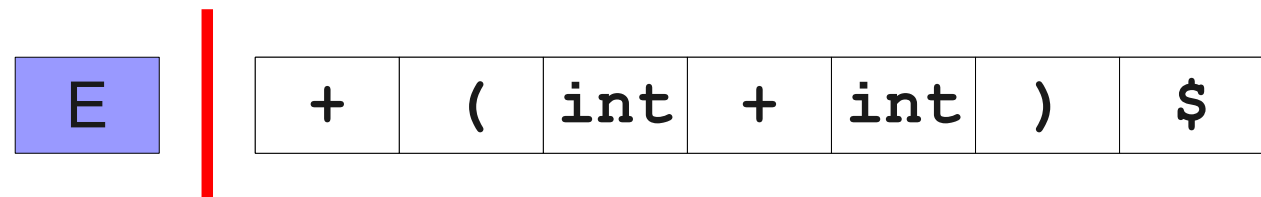
$E \rightarrow T$

$E \rightarrow E + T$

$T \rightarrow \text{int}$

$T \rightarrow (E)$

$S \rightarrow \cdot E\$$
$E \rightarrow E \cdot + T$



Tracking our Position in Productions

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$

$S \rightarrow \cdot E\$$
$E \rightarrow E \cdot + T$

E	+		(int	+	int)	\$
---	---	--	---	-----	---	-----	---	----

Tracking our Position in Productions

$S \rightarrow E\$$

$E \rightarrow T$

$E \rightarrow E + T$

$T \rightarrow \text{int}$

$T \rightarrow (E)$

$S \rightarrow \cdot E\$$
$E \rightarrow E + \cdot T$

E	+
---	---



(int	+	int)	\$
---	-----	---	-----	---	----

Tracking our Position in Productions

$S \rightarrow E\$$

$E \rightarrow T$

$E \rightarrow E + T$

$T \rightarrow \text{int}$

$T \rightarrow (E)$

$S \rightarrow \cdot E\$$
$E \rightarrow E + \cdot T$
$T \rightarrow \cdot (E)$

E	+
---	---

(int	+	int)	\$
---	-----	---	-----	---	----

Tracking our Position in Productions

$S \rightarrow E\$$

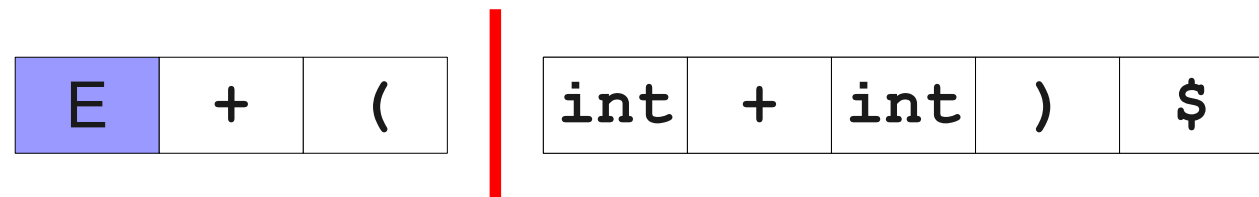
$E \rightarrow T$

$E \rightarrow E + T$

$T \rightarrow \text{int}$

$T \rightarrow (E)$

$S \rightarrow \cdot E\$$
$E \rightarrow E + \cdot T$
$T \rightarrow \cdot (E)$



Tracking our Position in Productions

$S \rightarrow E\$$

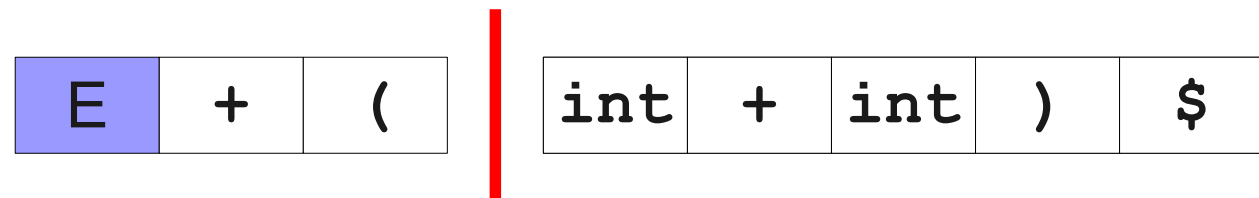
$E \rightarrow T$

$E \rightarrow E + T$

$T \rightarrow \text{int}$

$T \rightarrow (E)$

$S \rightarrow \cdot E\$$
$E \rightarrow E + \cdot T$
$T \rightarrow (\cdot E)$



Tracking our Position in Productions

$S \rightarrow E\$$

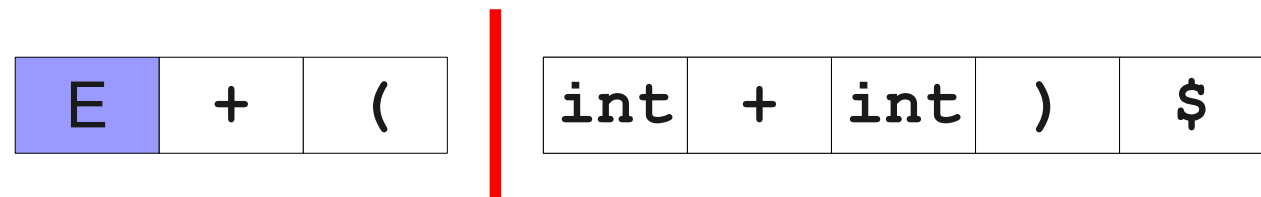
$E \rightarrow T$

$E \rightarrow E + T$

$T \rightarrow \text{int}$

$T \rightarrow (E)$

$S \rightarrow \cdot E\$$
$E \rightarrow E + \cdot T$
$T \rightarrow (\cdot E)$
$E \rightarrow \cdot E + T$



Tracking our Position in Productions

$S \rightarrow E\$$

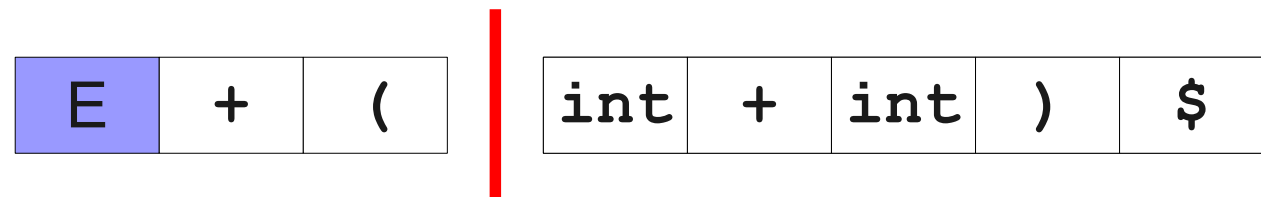
$E \rightarrow T$

$E \rightarrow E + T$

$T \rightarrow \text{int}$

$T \rightarrow (E)$

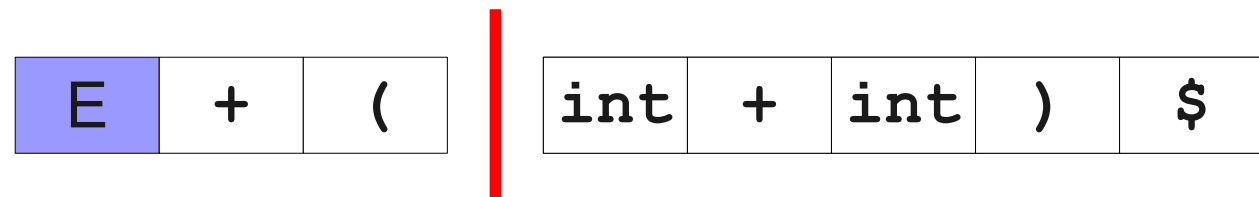
$S \rightarrow \cdot E\$$
$E \rightarrow E + \cdot T$
$T \rightarrow (\cdot E)$
$E \rightarrow \cdot E + T$
$E \rightarrow \cdot T$



Tracking our Position in Productions

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$

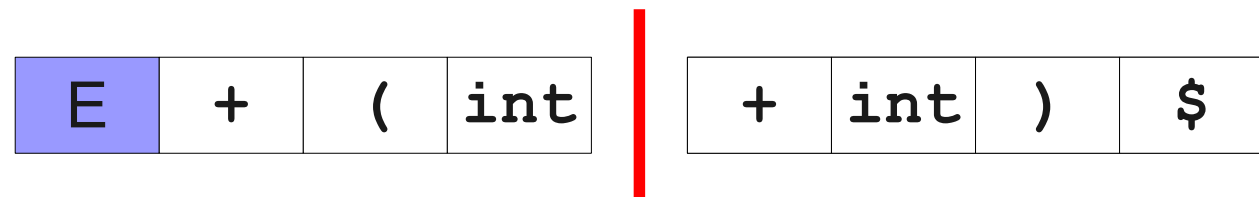
$S \rightarrow \cdot E\$$
$E \rightarrow E + \cdot T$
$T \rightarrow (\cdot E)$
$E \rightarrow \cdot E + T$
$E \rightarrow \cdot T$
$T \rightarrow \cdot \text{int}$



Tracking our Position in Productions

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$

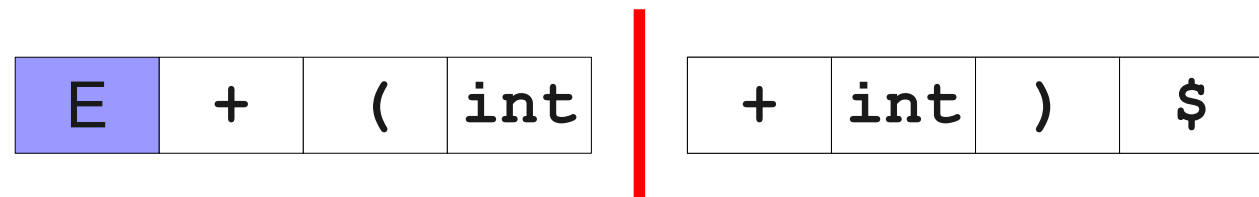
$S \rightarrow \cdot E\$$
$E \rightarrow E + \cdot T$
$T \rightarrow (\cdot E)$
$E \rightarrow \cdot E + T$
$E \rightarrow \cdot T$
$T \rightarrow \cdot \text{int}$



Tracking our Position in Productions

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$

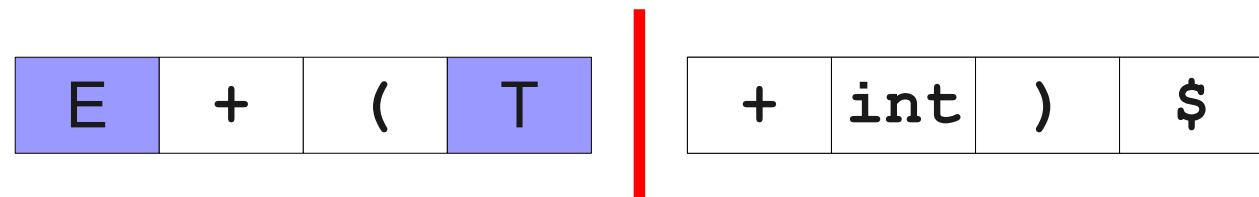
$S \rightarrow \cdot E\$$
$E \rightarrow E + \cdot T$
$T \rightarrow (\cdot E)$
$E \rightarrow \cdot E + T$
$E \rightarrow \cdot T$
$T \rightarrow \text{int} \cdot$



Tracking our Position in Productions

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$

$S \rightarrow \cdot E\$$
$E \rightarrow E + \cdot T$
$T \rightarrow (\cdot E)$
$E \rightarrow \cdot E + T$
$E \rightarrow \cdot T$
$T \rightarrow \text{int} \cdot$



Tracking our Position in Productions

$S \rightarrow E\$$

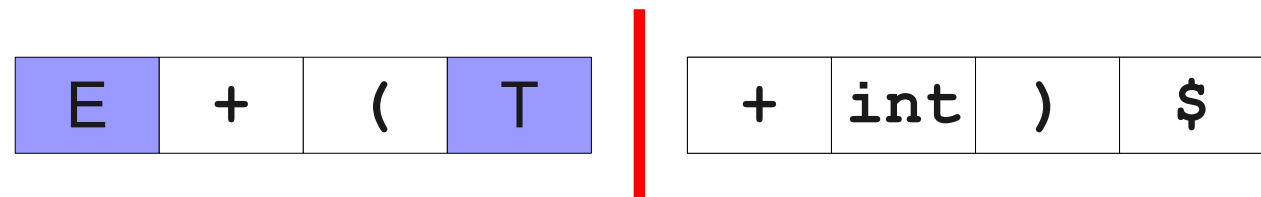
$E \rightarrow T$

$E \rightarrow E + T$

$T \rightarrow \text{int}$

$T \rightarrow (E)$

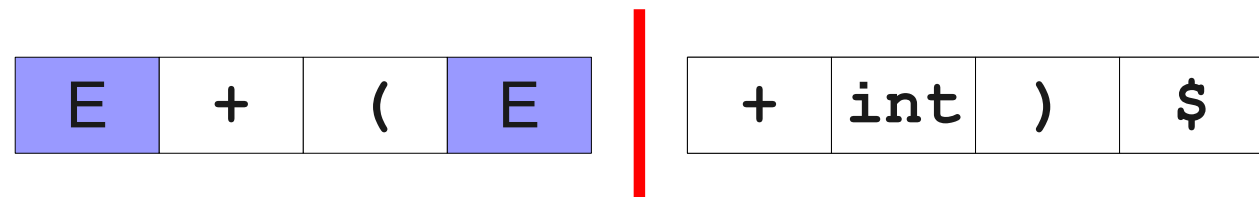
$S \rightarrow \cdot E\$$
$E \rightarrow E + \cdot T$
$T \rightarrow (\cdot E)$
$E \rightarrow \cdot E + T$
$E \rightarrow T \cdot$



Tracking our Position in Productions

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$

$S \rightarrow \cdot E\$$
$E \rightarrow E + \cdot T$
$T \rightarrow (\cdot E)$
$E \rightarrow \cdot E + T$
$E \rightarrow T \cdot$



Tracking our Position in Productions

$S \rightarrow E\$$

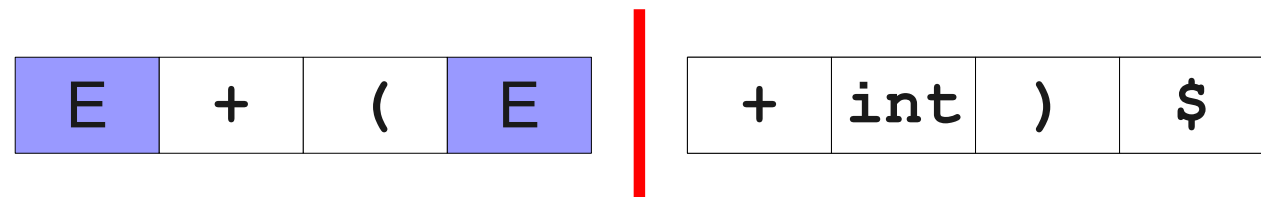
$E \rightarrow T$

$E \rightarrow E + T$

$T \rightarrow \text{int}$

$T \rightarrow (E)$

$S \rightarrow \cdot E\$$
$E \rightarrow E + \cdot T$
$T \rightarrow (\cdot E)$
$E \rightarrow \cdot E + T$



Tracking our Position in Productions

$S \rightarrow E\$$

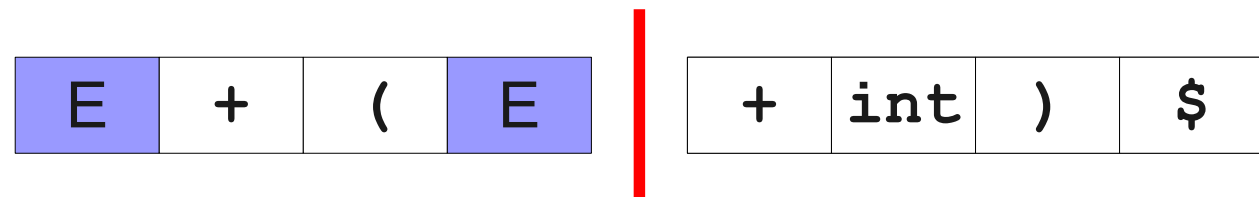
$E \rightarrow T$

$E \rightarrow E + T$

$T \rightarrow \text{int}$

$T \rightarrow (E)$

$S \rightarrow \cdot E\$$
$E \rightarrow E + \cdot T$
$T \rightarrow (\cdot E)$
$E \rightarrow E \cdot + T$



Tracking our Position in Productions

$S \rightarrow E\$$

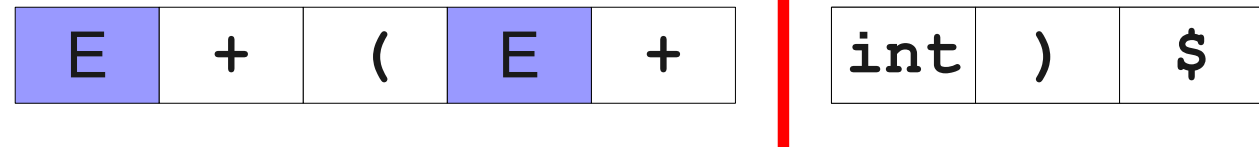
$E \rightarrow T$

$E \rightarrow E + T$

$T \rightarrow \text{int}$

$T \rightarrow (E)$

$S \rightarrow \cdot E\$$
$E \rightarrow E + \cdot T$
$T \rightarrow (\cdot E)$
$E \rightarrow E \cdot + T$



Tracking our Position in Productions

$S \rightarrow E\$$

$E \rightarrow T$

$E \rightarrow E + T$

$T \rightarrow \text{int}$

$T \rightarrow (E)$

$S \rightarrow \cdot E\$$
$E \rightarrow E + \cdot T$
$T \rightarrow (\cdot E)$
$E \rightarrow E + \cdot T$

E	+	(E	+
---	---	---	---	---

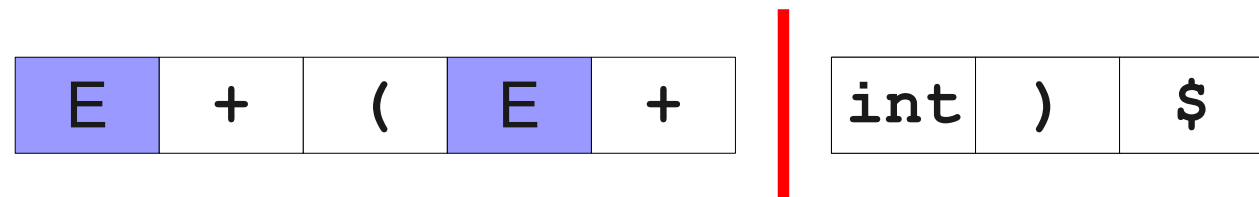


int)	\$
-----	---	----

Tracking our Position in Productions

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$

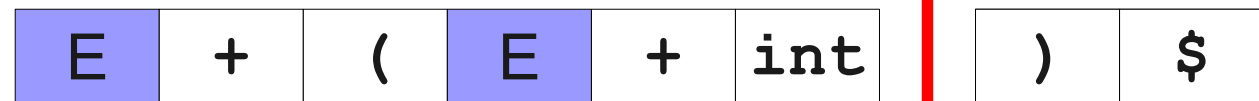
$S \rightarrow \cdot E\$$
$E \rightarrow E + \cdot T$
$T \rightarrow (\cdot E)$
$E \rightarrow E + \cdot T$
$T \rightarrow \cdot \text{int}$



Tracking our Position in Productions

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$

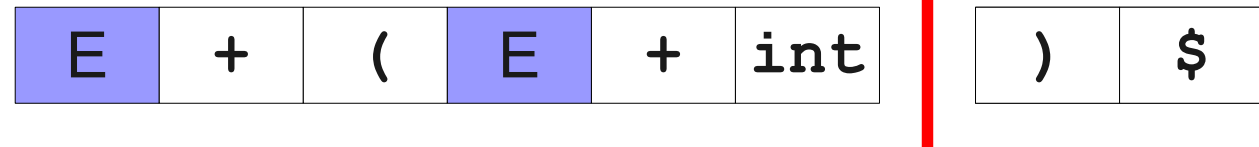
$S \rightarrow \cdot E\$$
$E \rightarrow E + \cdot T$
$T \rightarrow (\cdot E)$
$E \rightarrow E + \cdot T$
$T \rightarrow \cdot \text{int}$



Tracking our Position in Productions

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$

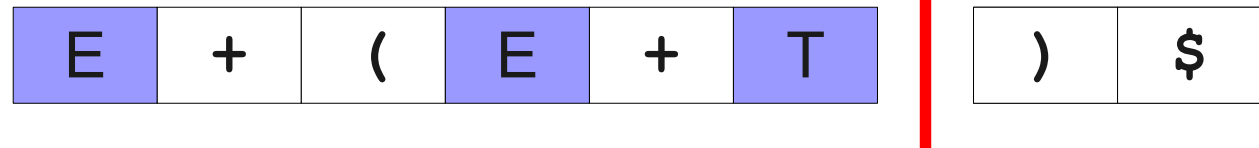
$S \rightarrow \cdot E\$$
$E \rightarrow E + \cdot T$
$T \rightarrow (\cdot E)$
$E \rightarrow E + \cdot T$
$T \rightarrow \text{int} \cdot$



Tracking our Position in Productions

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$

$S \rightarrow \cdot E\$$
$E \rightarrow E + \cdot T$
$T \rightarrow (\cdot E)$
$E \rightarrow E + \cdot T$
$T \rightarrow \text{int} \cdot$



Tracking our Position in Productions

$S \rightarrow E\$$

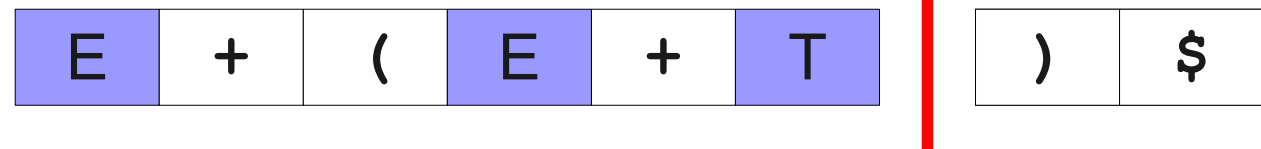
$E \rightarrow T$

$E \rightarrow E + T$

$T \rightarrow \text{int}$

$T \rightarrow (E)$

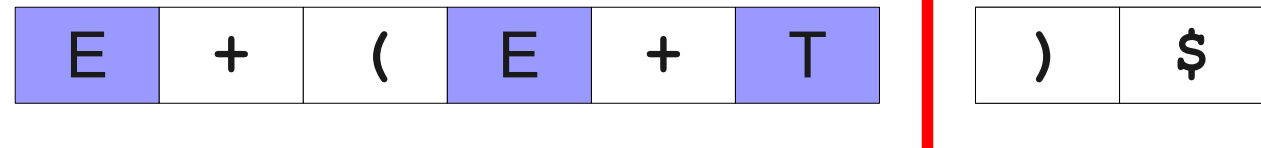
$S \rightarrow \cdot E\$$
$E \rightarrow E + \cdot T$
$T \rightarrow (\cdot E)$
$E \rightarrow E + \cdot T$



Tracking our Position in Productions

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$

$S \rightarrow \cdot E\$$
$E \rightarrow E + \cdot T$
$T \rightarrow (\cdot E)$
$E \rightarrow E + T \cdot$



Tracking our Position in Productions

$S \rightarrow E\$$

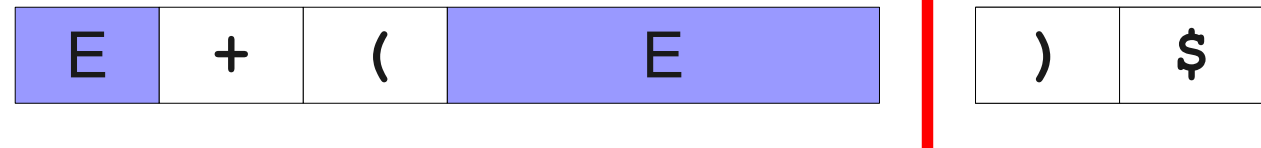
$E \rightarrow T$

$E \rightarrow E + T$

$T \rightarrow \text{int}$

$T \rightarrow (E)$

$S \rightarrow \cdot E\$$
$E \rightarrow E + \cdot T$
$T \rightarrow (\cdot E)$
$E \rightarrow E + T \cdot$



Tracking our Position in Productions

$S \rightarrow E\$$

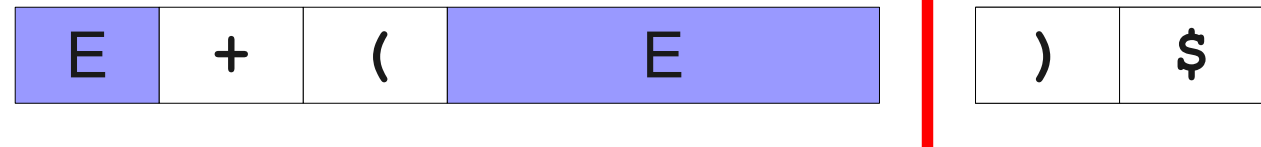
$E \rightarrow T$

$E \rightarrow E + T$

$T \rightarrow \text{int}$

$T \rightarrow (E)$

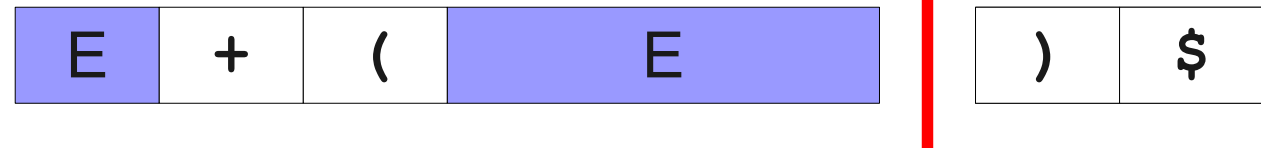
$S \rightarrow \cdot E\$$
$E \rightarrow E + \cdot T$
$T \rightarrow (\cdot E)$



Tracking our Position in Productions

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$

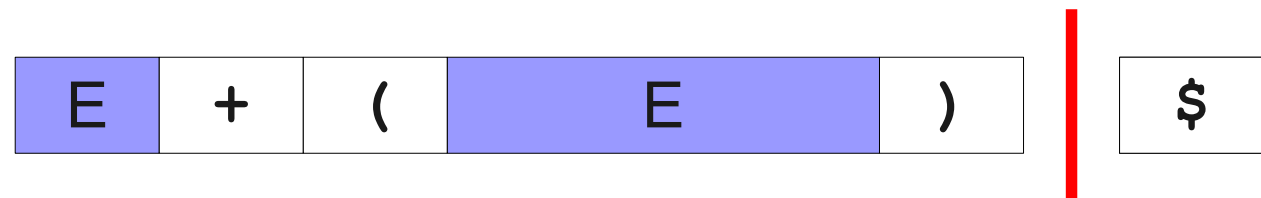
$S \rightarrow \cdot E\$$
$E \rightarrow E + \cdot T$
$T \rightarrow (E \cdot)$



Tracking our Position in Productions

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$

$S \rightarrow \cdot E\$$
$E \rightarrow E + \cdot T$
$T \rightarrow (E \cdot)$



Tracking our Position in Productions

$S \rightarrow E\$$

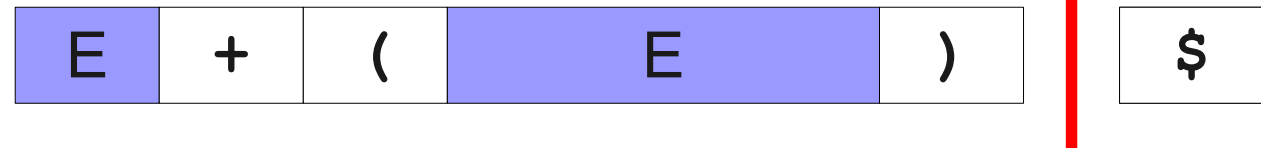
$E \rightarrow T$

$E \rightarrow E + T$

$T \rightarrow \text{int}$

$T \rightarrow (E)$

$S \rightarrow \cdot E\$$
$E \rightarrow E + \cdot T$
$T \rightarrow (E) \cdot$



Tracking our Position in Productions

$S \rightarrow E\$$

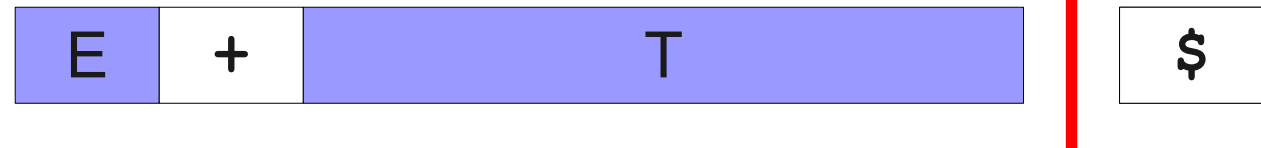
$E \rightarrow T$

$E \rightarrow E + T$

$T \rightarrow \text{int}$

$T \rightarrow (E)$

$S \rightarrow \cdot E\$$
$E \rightarrow E + \cdot T$
$T \rightarrow (E) \cdot$



Tracking our Position in Productions

$S \rightarrow E\$$

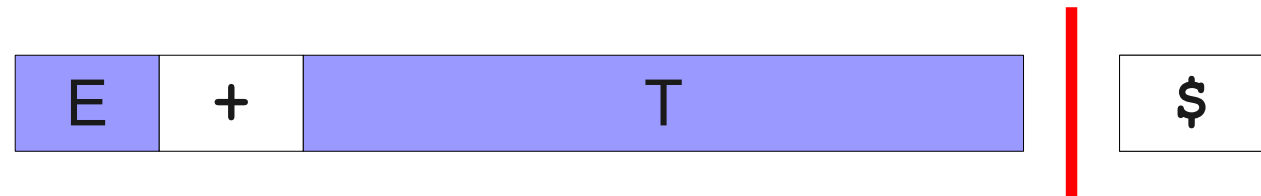
$E \rightarrow T$

$E \rightarrow E + T$

$T \rightarrow \text{int}$

$T \rightarrow (E)$

$S \rightarrow \cdot E\$$
$E \rightarrow E + \cdot T$



Tracking our Position in Productions

$S \rightarrow E\$$

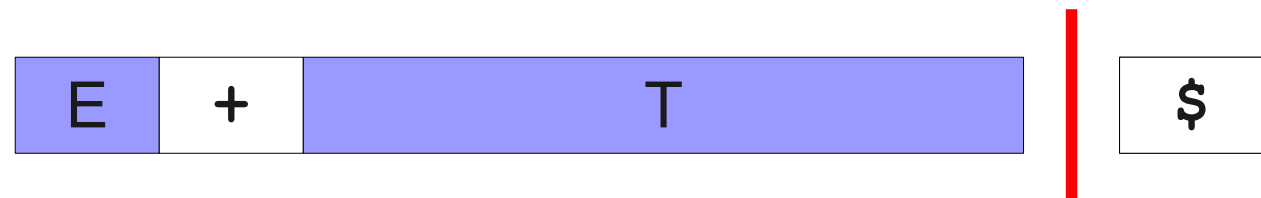
$E \rightarrow T$

$E \rightarrow E + T$

$T \rightarrow \text{int}$

$T \rightarrow (E)$

$S \rightarrow \cdot E\$$
$E \rightarrow E + T \cdot$



Tracking our Position in Productions

$S \rightarrow E\$$

$E \rightarrow T$

$E \rightarrow E + T$

$T \rightarrow \text{int}$

$T \rightarrow (E)$

$S \rightarrow \cdot E\$$
$E \rightarrow E + T \cdot$

E



\$

Tracking our Position in Productions

$S \rightarrow E\$$

$E \rightarrow T$

$E \rightarrow E + T$

$T \rightarrow \text{int}$

$T \rightarrow (E)$

$S \rightarrow \cdot E\$$

E

\$

Tracking our Position in Productions

$S \rightarrow E\$$

$E \rightarrow T$

$E \rightarrow E + T$

$T \rightarrow \mathbf{int}$

$T \rightarrow (E)$

$S \rightarrow E \cdot \$$

E

\$

Tracking our Position in Productions

$S \rightarrow E\$$

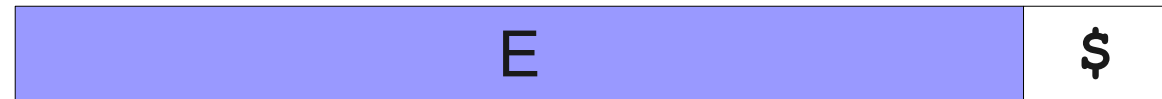
$E \rightarrow T$

$E \rightarrow E + T$

$T \rightarrow \text{int}$

$T \rightarrow (E)$

$S \rightarrow E \cdot \$$



Tracking our Position in Productions

$S \rightarrow E\$$

$E \rightarrow T$

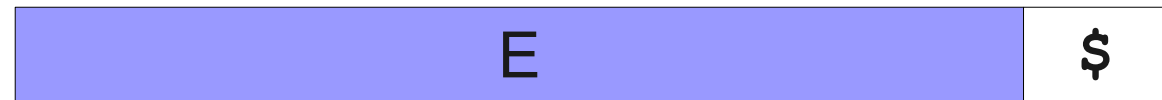
$E \rightarrow E + T$

$T \rightarrow \text{int}$

$T \rightarrow (E)$

$S \rightarrow E \$ \cdot$

E \$



Tracking our Position in Productions

$S \rightarrow E\$$

$E \rightarrow T$

$E \rightarrow E + T$

$T \rightarrow \text{int}$

$T \rightarrow (E)$

$S \rightarrow E \$ \cdot$

S

Generating Viable Prefixes

$S \rightarrow E\$$

$E \rightarrow T$

$E \rightarrow E + T$

$T \rightarrow \mathbf{int}$

$T \rightarrow (E)$

Generating Viable Prefixes

$S \rightarrow E\$$

$E \rightarrow T$

$E \rightarrow E + T$

$T \rightarrow \mathbf{int}$

$T \rightarrow (E)$

$S \rightarrow \cdot E\$$

Generating Viable Prefixes

$S \rightarrow E\$$

$E \rightarrow T$

$E \rightarrow E + T$

$T \rightarrow \mathbf{int}$

$T \rightarrow (E)$

$S \rightarrow \cdot E\$$

$E \rightarrow \cdot E + T$

Generating Viable Prefixes

$S \rightarrow E\$$

$E \rightarrow T$

$E \rightarrow E + T$

$T \rightarrow \mathbf{int}$

$T \rightarrow (E)$

$S \rightarrow \cdot E\$$

$E \rightarrow E \cdot + T$

Generating Viable Prefixes

$S \rightarrow E\$$

$E \rightarrow T$

$E \rightarrow E + T$

$T \rightarrow \text{int}$

$T \rightarrow (E)$

$S \rightarrow \cdot E\$$
$E \rightarrow E \cdot + T$

E

Generating Viable Prefixes

$S \rightarrow E\$$

$E \rightarrow T$

$E \rightarrow E + T$

$T \rightarrow \mathbf{int}$

$T \rightarrow (E)$

$S \rightarrow \cdot E\$$

$E \rightarrow E + \cdot T$

E

Generating Viable Prefixes

$S \rightarrow E\$$

$E \rightarrow T$

$E \rightarrow E + T$

$T \rightarrow \text{int}$

$T \rightarrow (E)$

$S \rightarrow \cdot E\$$
$E \rightarrow E + \cdot T$

E	+
---	---

Generating Viable Prefixes

$S \rightarrow E\$$

$E \rightarrow T$

$E \rightarrow E + T$

$T \rightarrow \mathbf{int}$

$T \rightarrow (E)$

$S \rightarrow \cdot E\$$

$E \rightarrow E + \cdot T$

$T \rightarrow \cdot (E)$

E

+

Generating Viable Prefixes

$S \rightarrow E\$$

$E \rightarrow T$

$E \rightarrow E + T$

$T \rightarrow \text{int}$

$T \rightarrow (E)$

$S \rightarrow \cdot E\$$
$E \rightarrow E + \cdot T$
$T \rightarrow (\cdot E)$

E	+
---	---

Generating Viable Prefixes

$S \rightarrow E\$$

$E \rightarrow T$

$E \rightarrow E + T$

$T \rightarrow \text{int}$

$T \rightarrow (E)$

$S \rightarrow \cdot E\$$
$E \rightarrow E + \cdot T$
$T \rightarrow (\cdot E)$

E	+	(
---	---	---

Generating Viable Prefixes

$S \rightarrow E\$$

$E \rightarrow T$

$E \rightarrow E + T$

$T \rightarrow \text{int}$

$T \rightarrow (E)$

$S \rightarrow \cdot E\$$
$E \rightarrow E + \cdot T$
$T \rightarrow (\cdot E)$
$E \rightarrow \cdot T$

E	+	(
---	---	---

Generating Viable Prefixes

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$

$S \rightarrow \cdot E\$$
$E \rightarrow E + \cdot T$
$T \rightarrow (\cdot E)$
$E \rightarrow T \cdot$

E	+	(
---	---	---

Generating Viable Prefixes

$S \rightarrow E\$$

$E \rightarrow T$

$E \rightarrow E + T$

$T \rightarrow \text{int}$

$T \rightarrow (E)$

$S \rightarrow \cdot E\$$
$E \rightarrow E + \cdot T$
$T \rightarrow (\cdot E)$
$E \rightarrow T \cdot$

E	+	(T
---	---	---	---

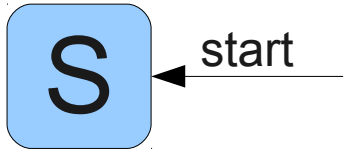
Maintaining our Current Position

- Begin just before the first symbol in the start state.
- If the symbol after our position is a nonterminal, nondeterministically guess which production of that nonterminal to use.
- If the symbol after our position is a terminal, transition past it if we encounter it next.
- **This can be simulated via an automaton.**

An Automaton for Viable Prefixes

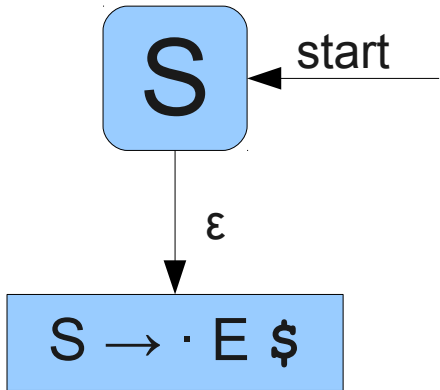
$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \mathbf{int}$
 $T \rightarrow (E)$

An Automaton for Viable Prefixes



$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \mathbf{int}$
 $T \rightarrow (E)$

An Automaton for Viable Prefixes



S \rightarrow E \$

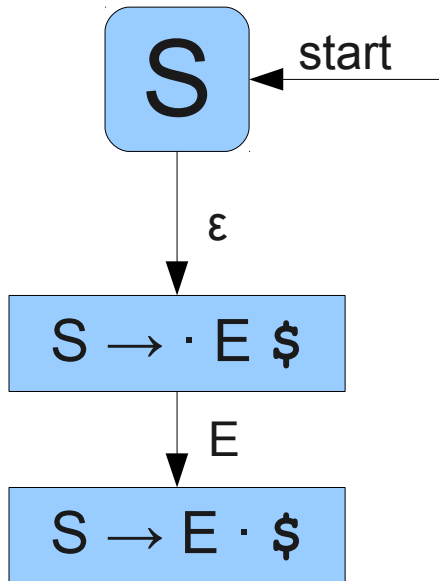
E \rightarrow T

E \rightarrow E + T

T \rightarrow int

T \rightarrow (E)

An Automaton for Viable Prefixes



$S \rightarrow E \$$

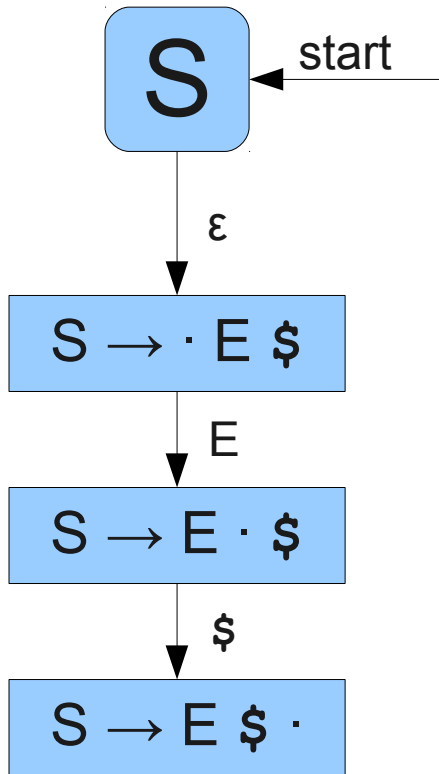
$E \rightarrow T$

$E \rightarrow E + T$

$T \rightarrow \mathbf{int}$

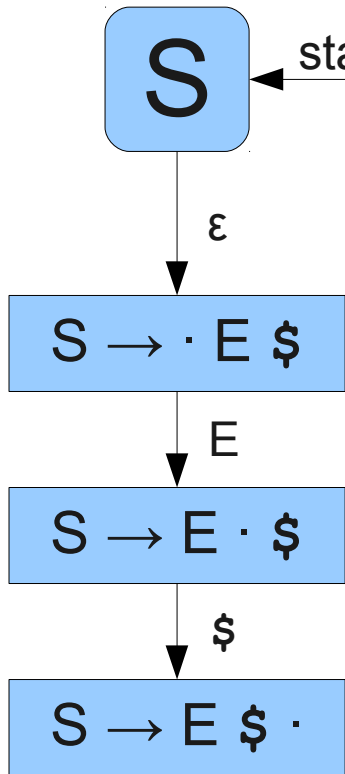
$T \rightarrow (E)$

An Automaton for Viable Prefixes



$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \mathbf{int}$
 $T \rightarrow (E)$

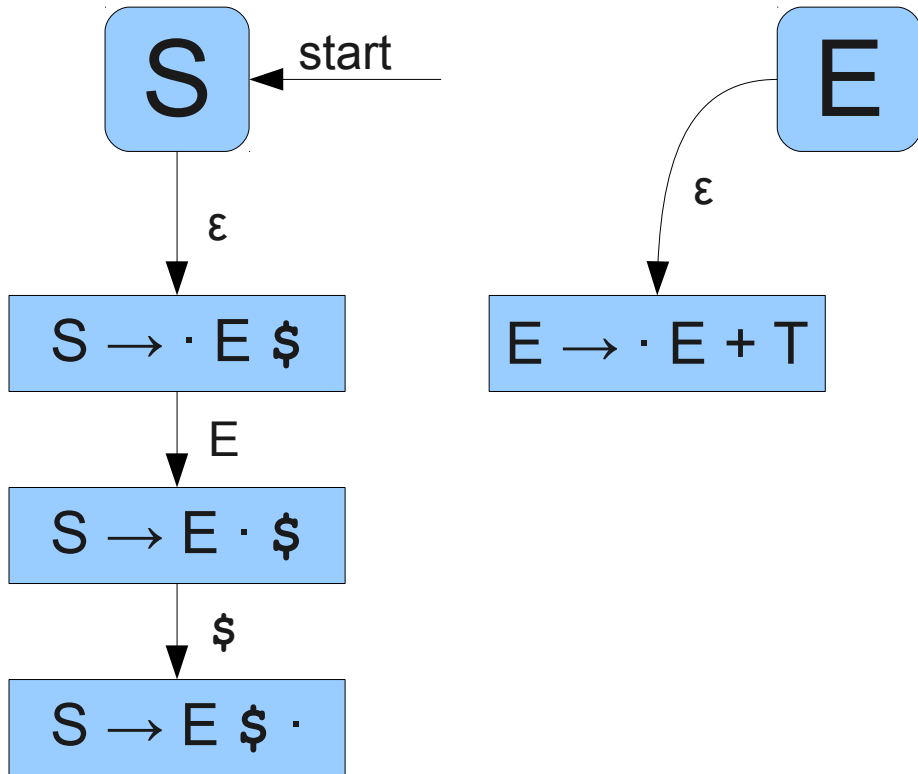
An Automaton for Viable Prefixes



E

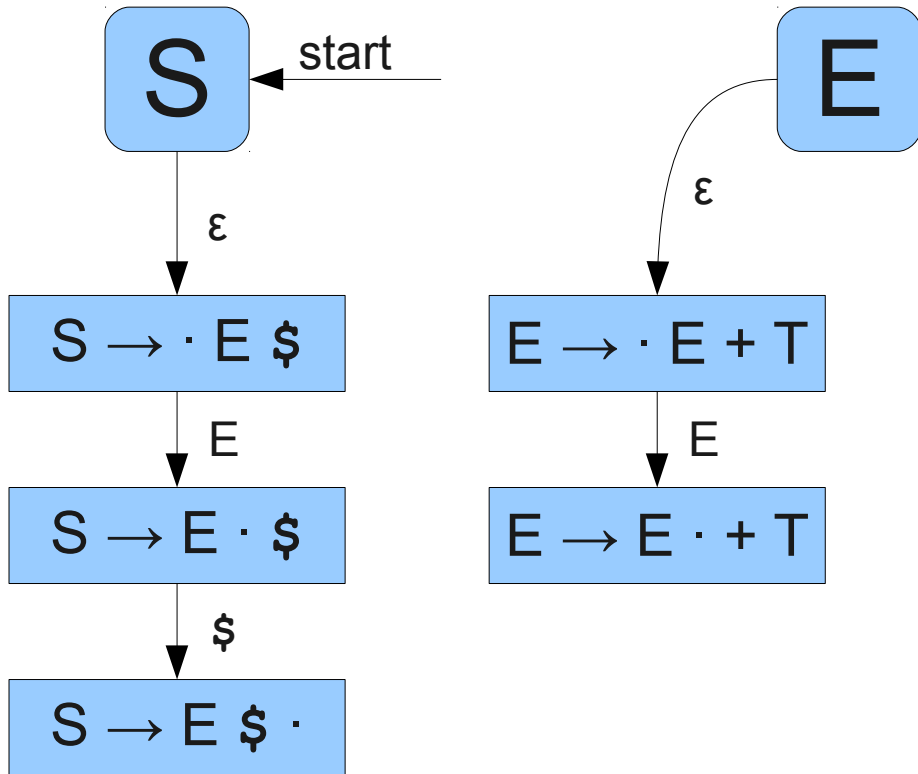
$S \rightarrow E \$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$

An Automaton for Viable Prefixes



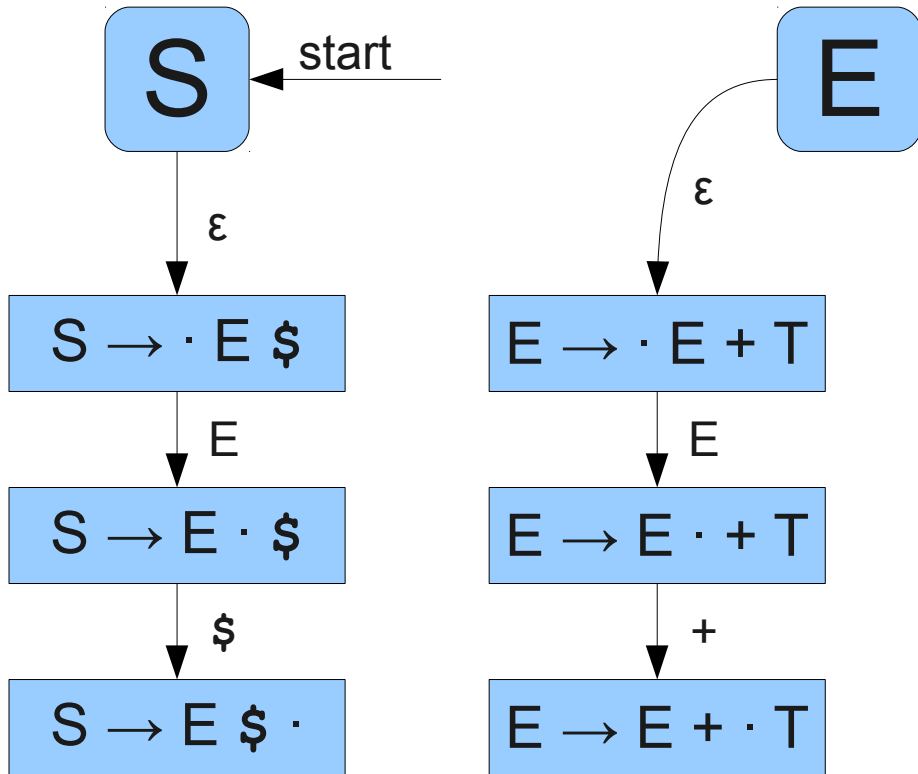
$S \rightarrow E \$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \mathbf{int}$
 $T \rightarrow (E)$

An Automaton for Viable Prefixes



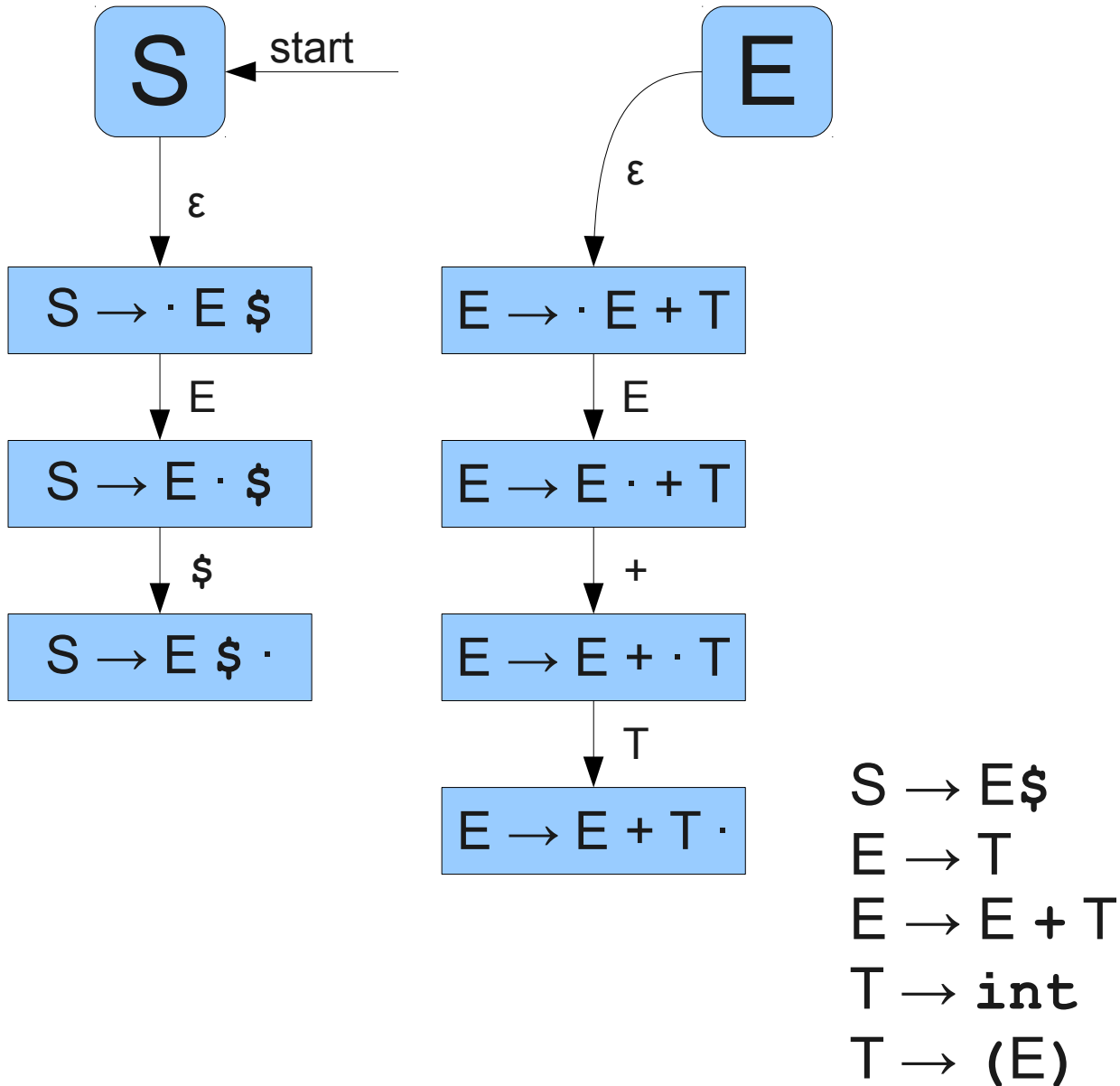
$S \rightarrow E \$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$

An Automaton for Viable Prefixes

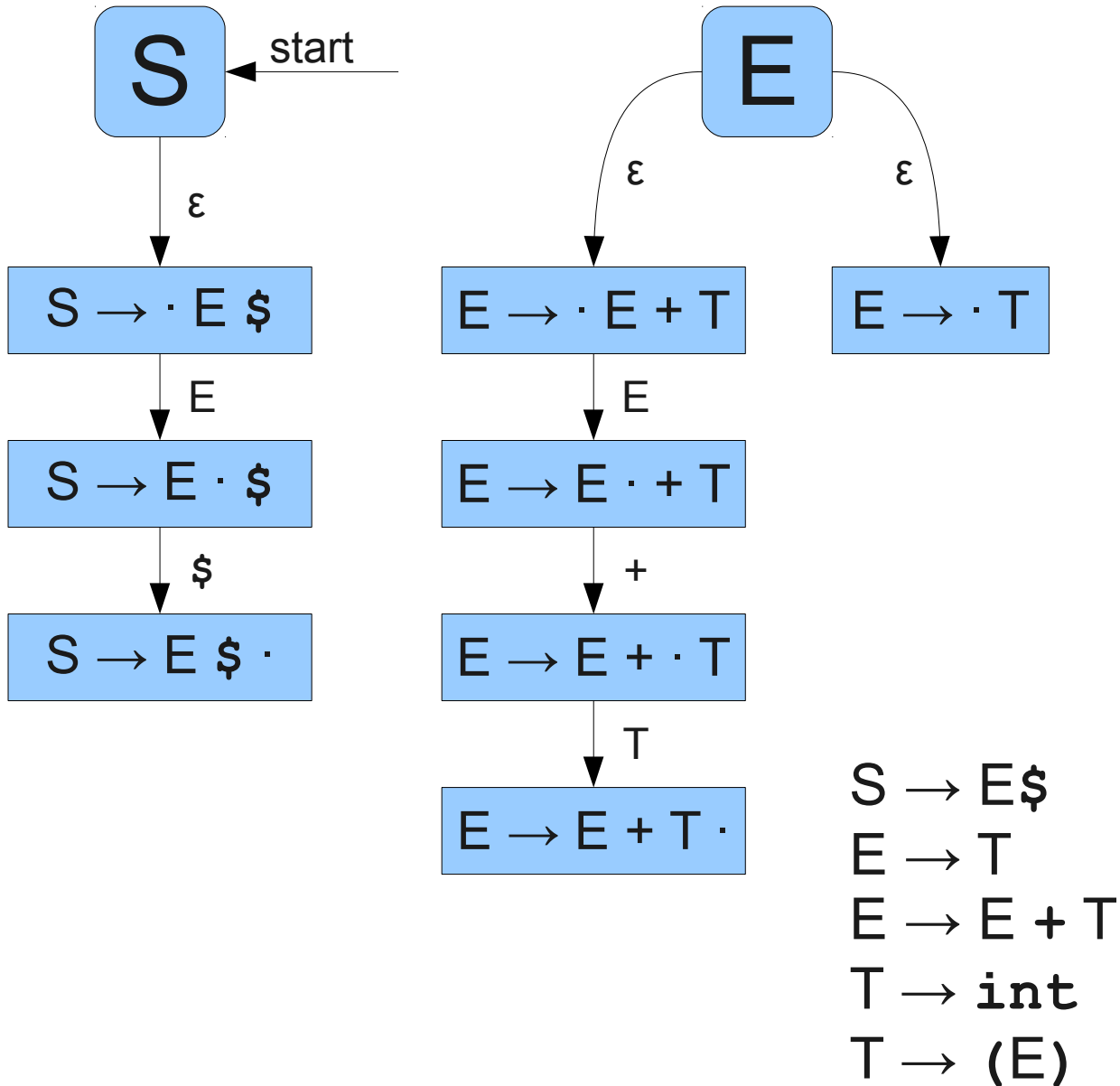


$S \rightarrow E \$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \mathbf{int}$
 $T \rightarrow (E)$

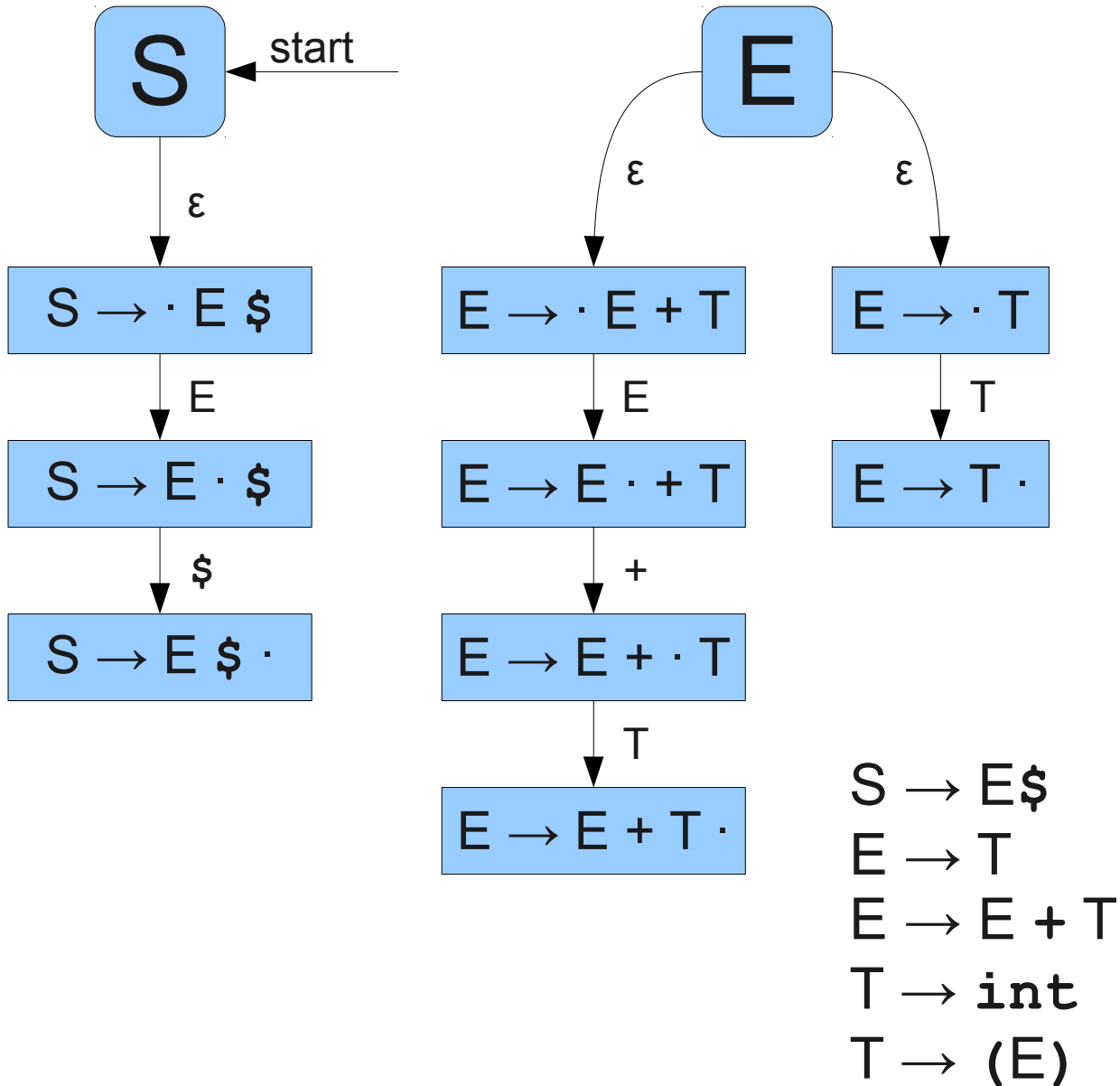
An Automaton for Viable Prefixes



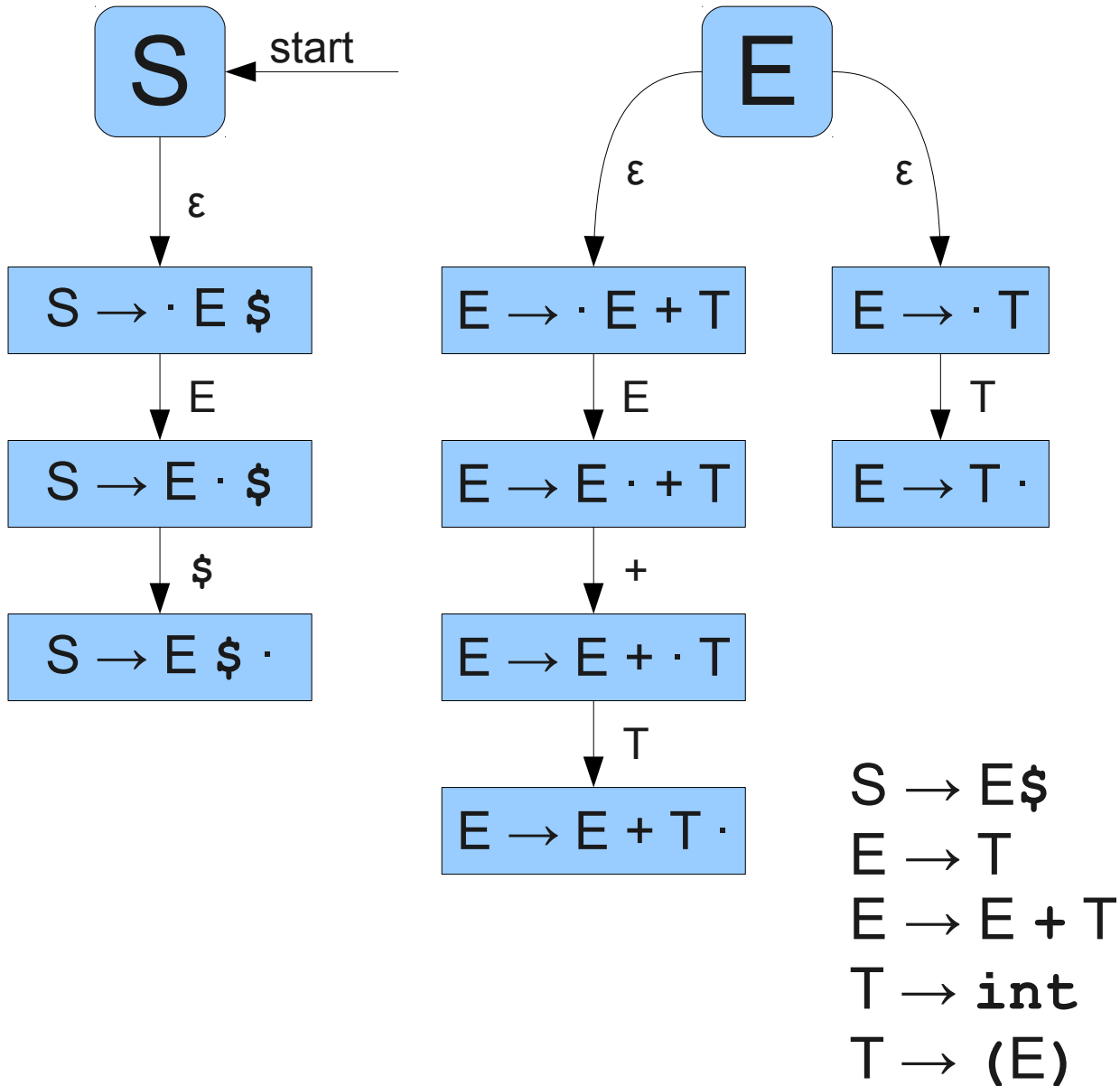
An Automaton for Viable Prefixes



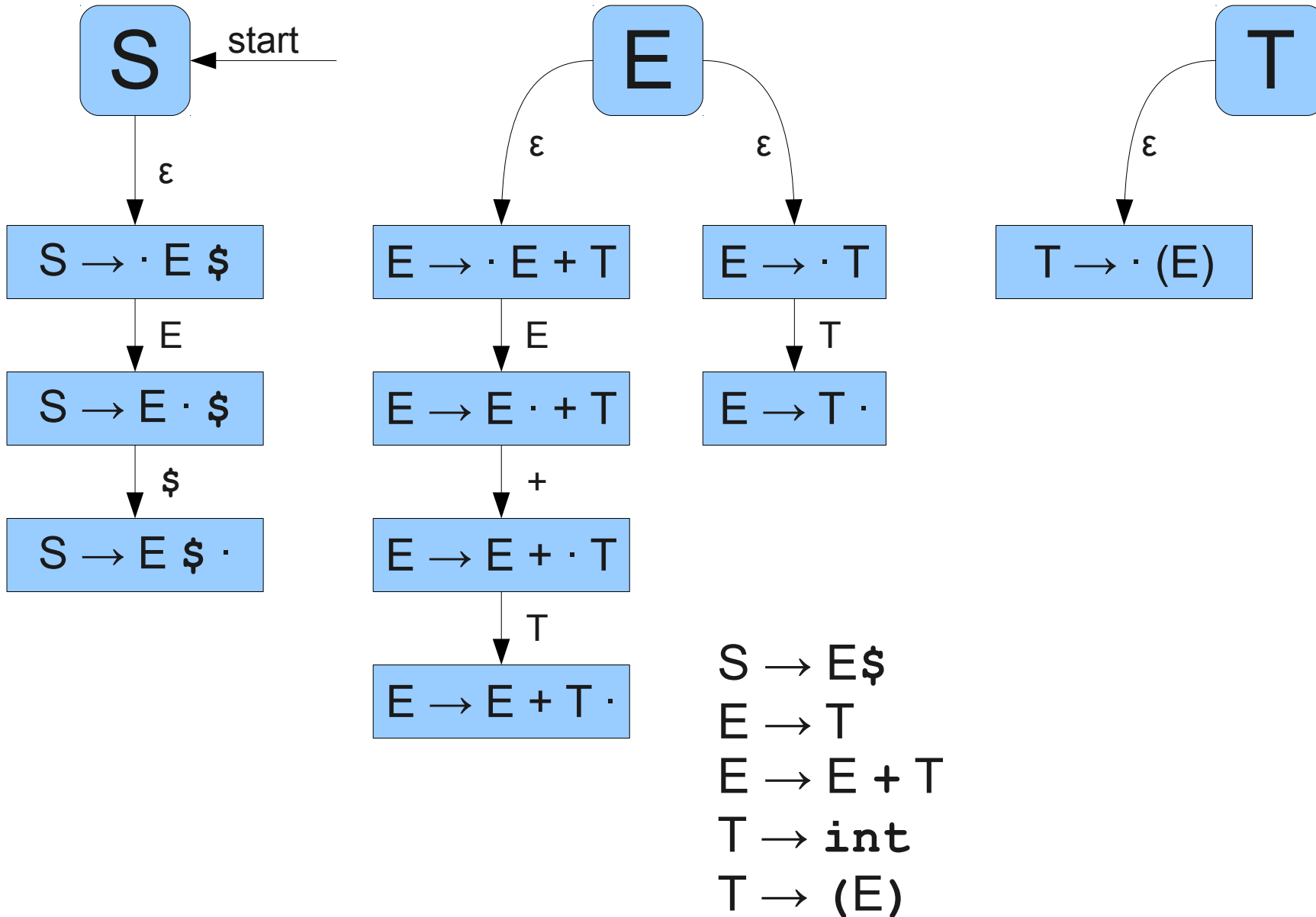
An Automaton for Viable Prefixes



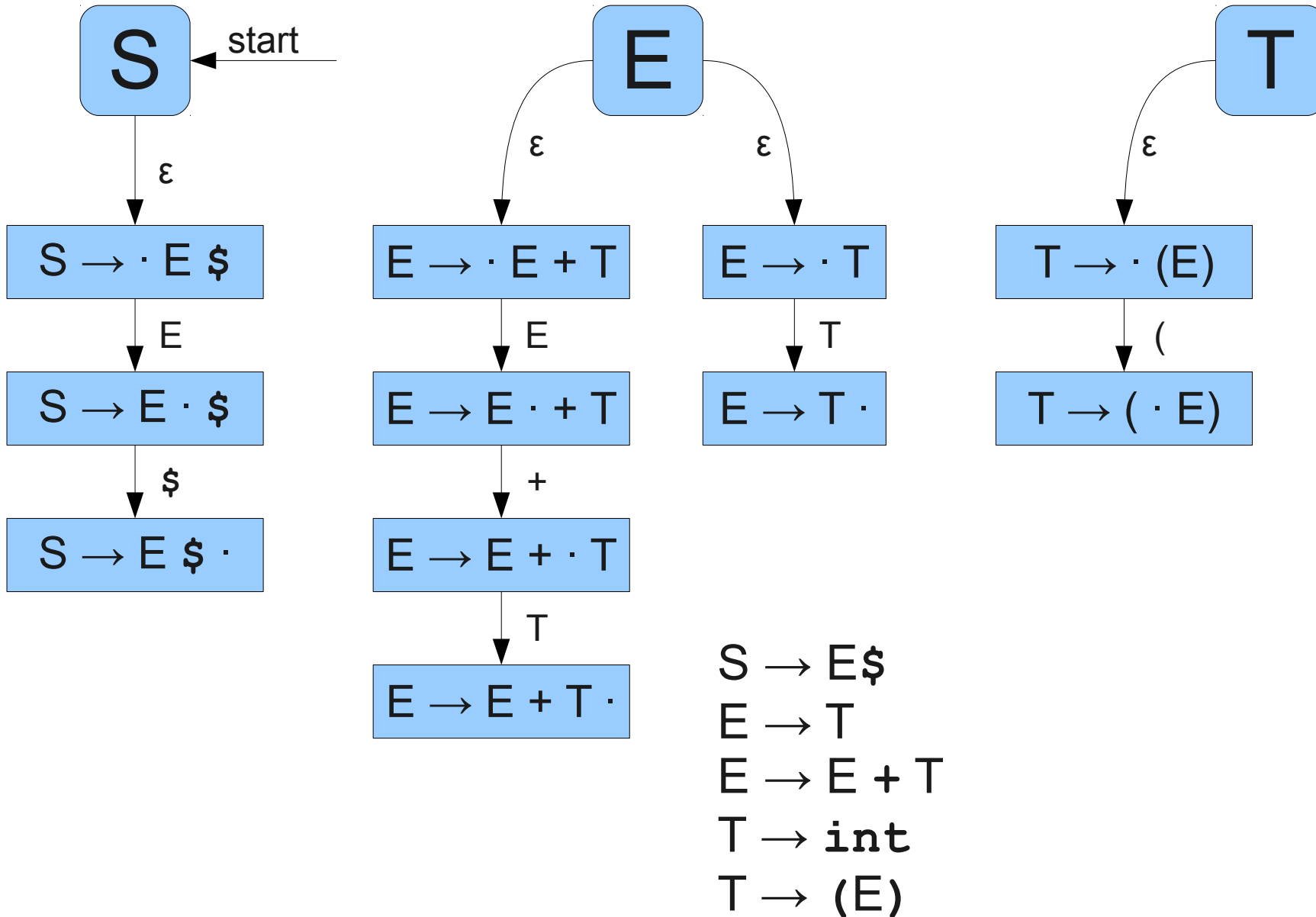
An Automaton for Viable Prefixes



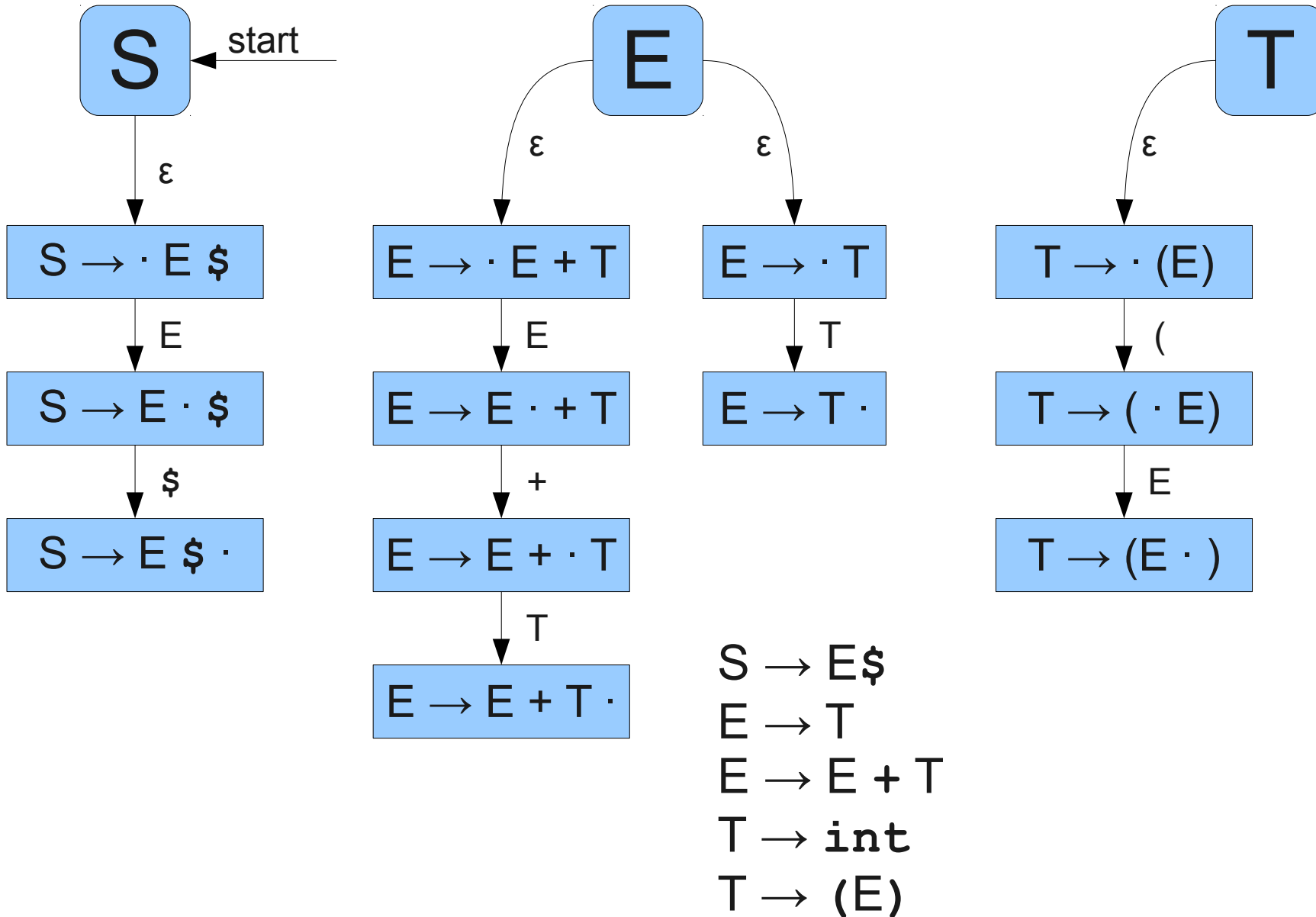
An Automaton for Viable Prefixes



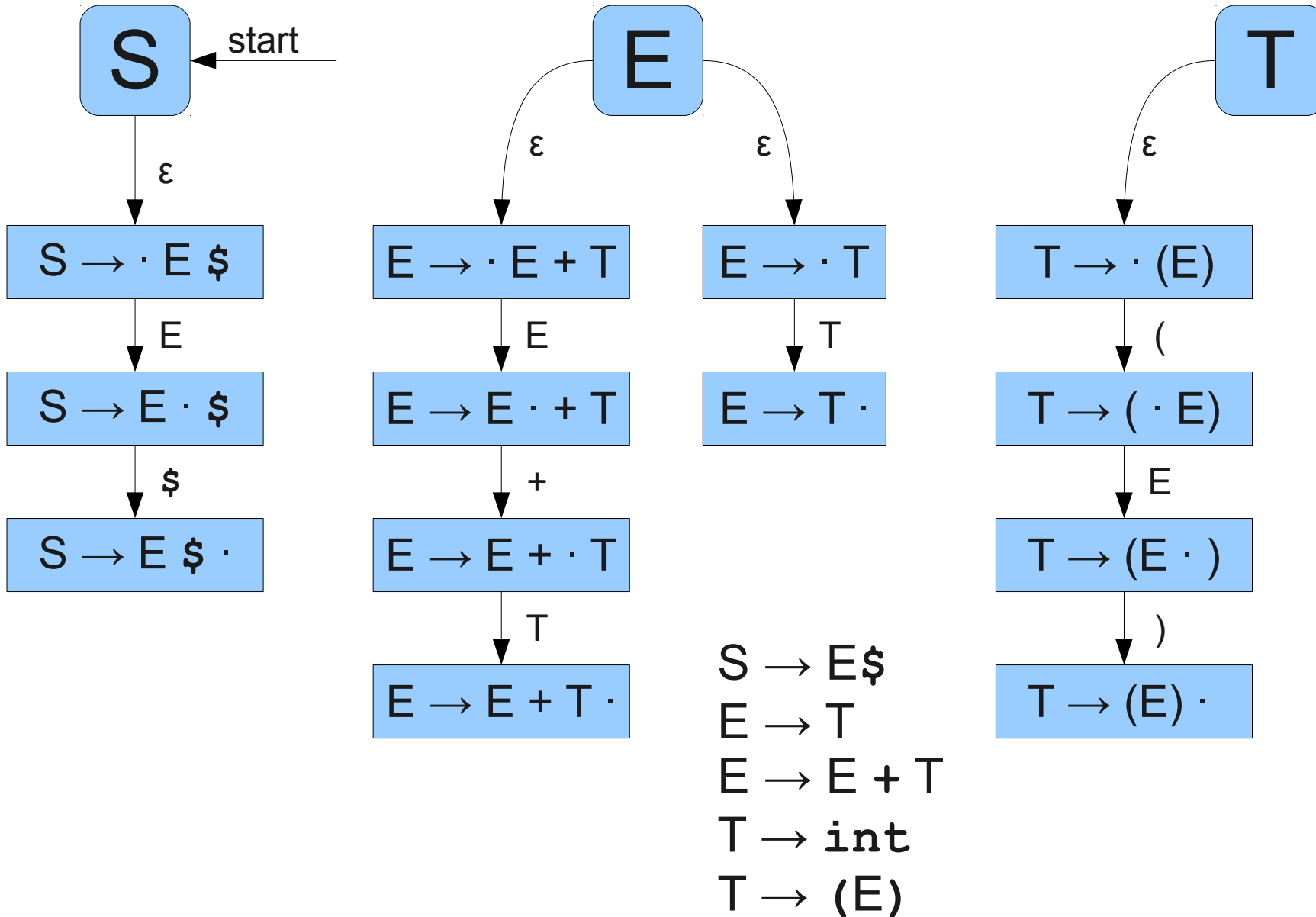
An Automaton for Viable Prefixes



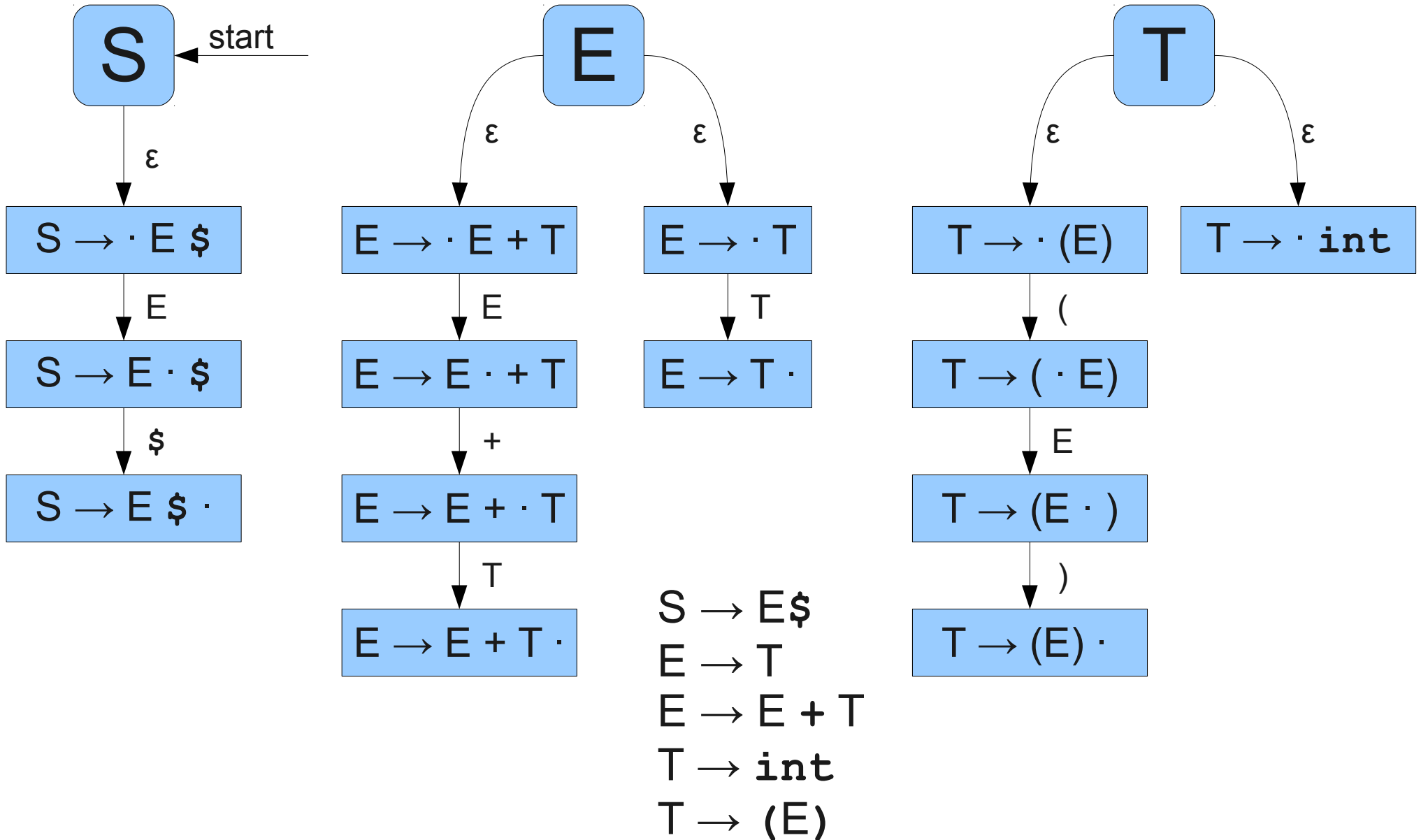
An Automaton for Viable Prefixes



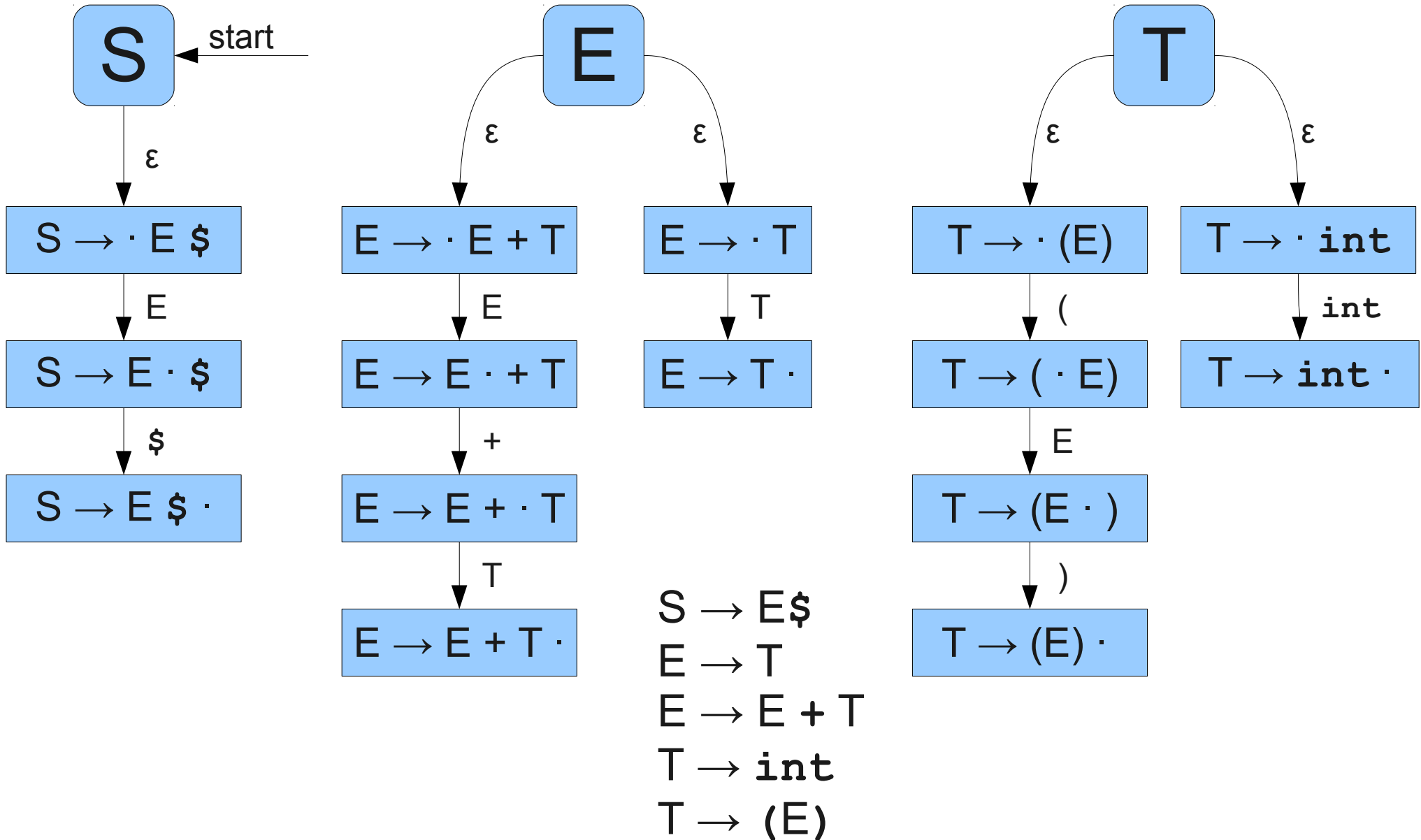
An Automaton for Viable Prefixes



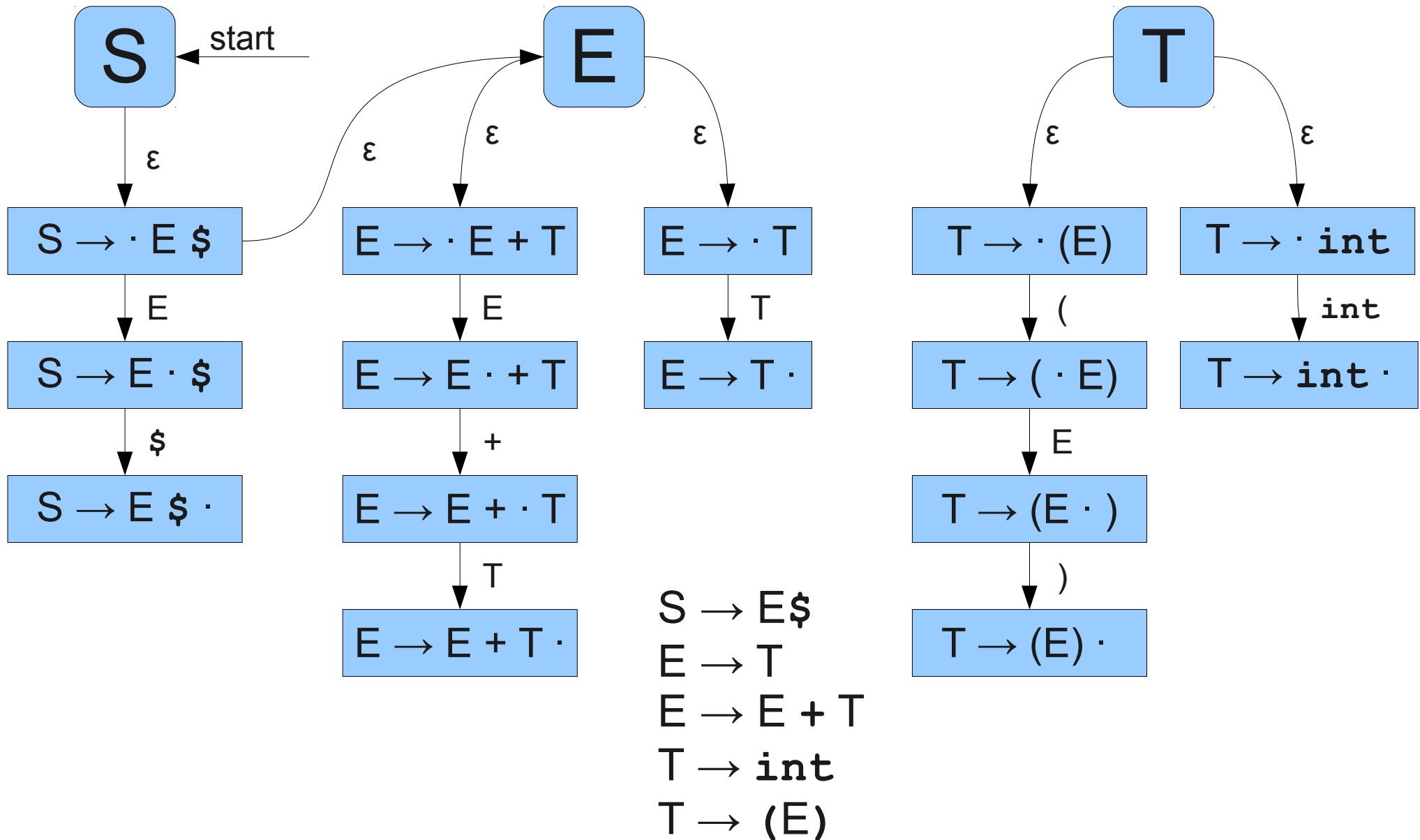
An Automaton for Viable Prefixes



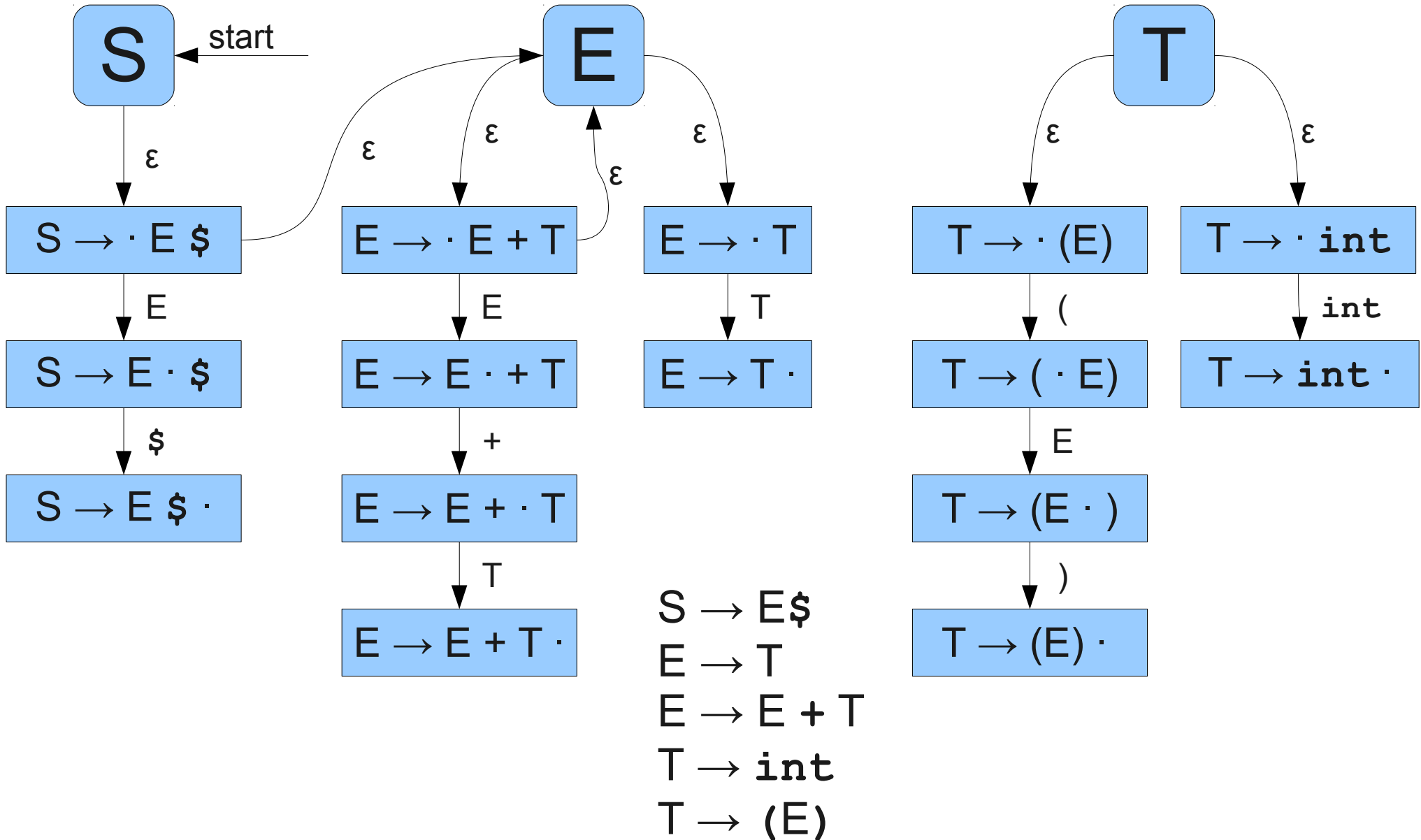
An Automaton for Viable Prefixes



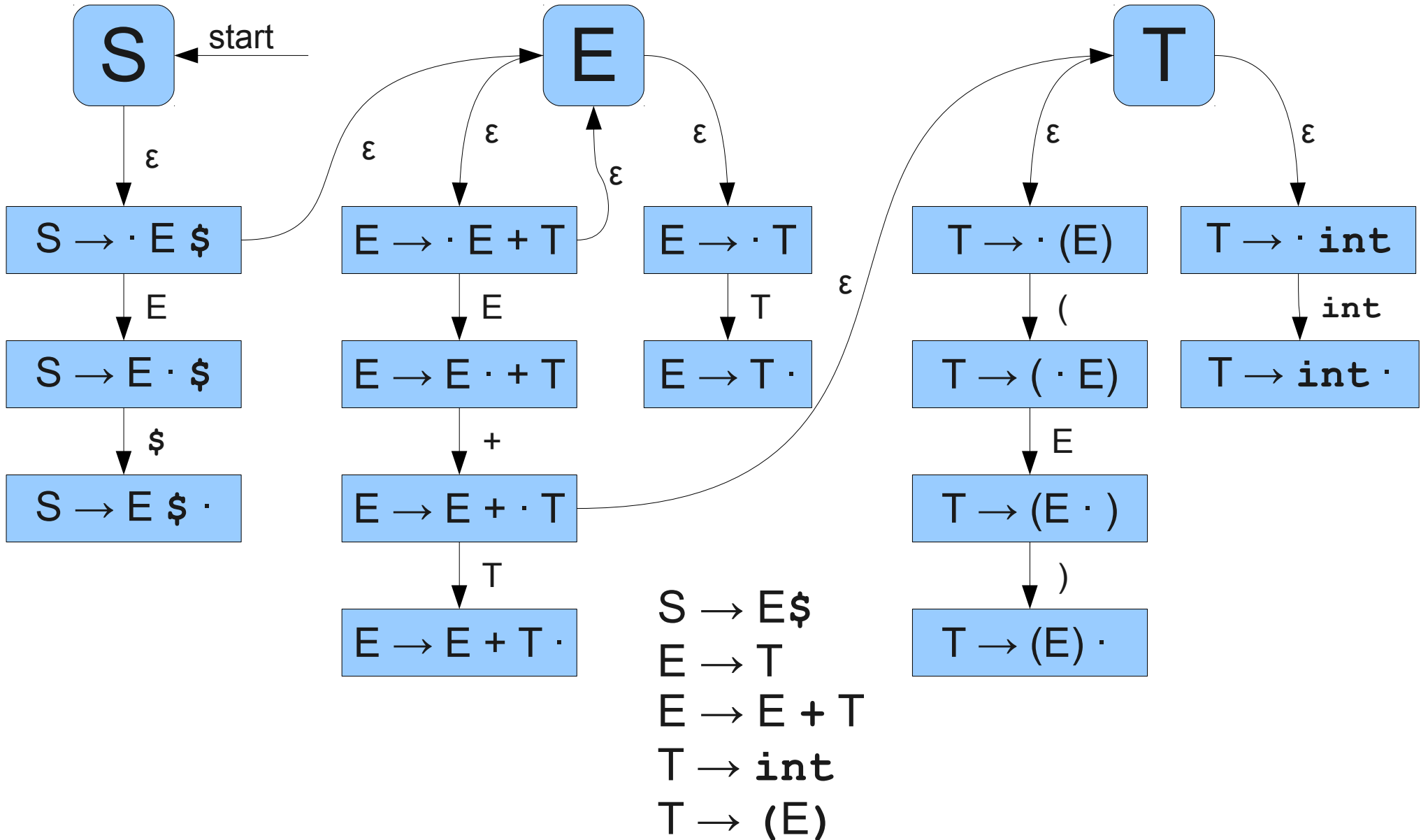
An Automaton for Viable Prefixes



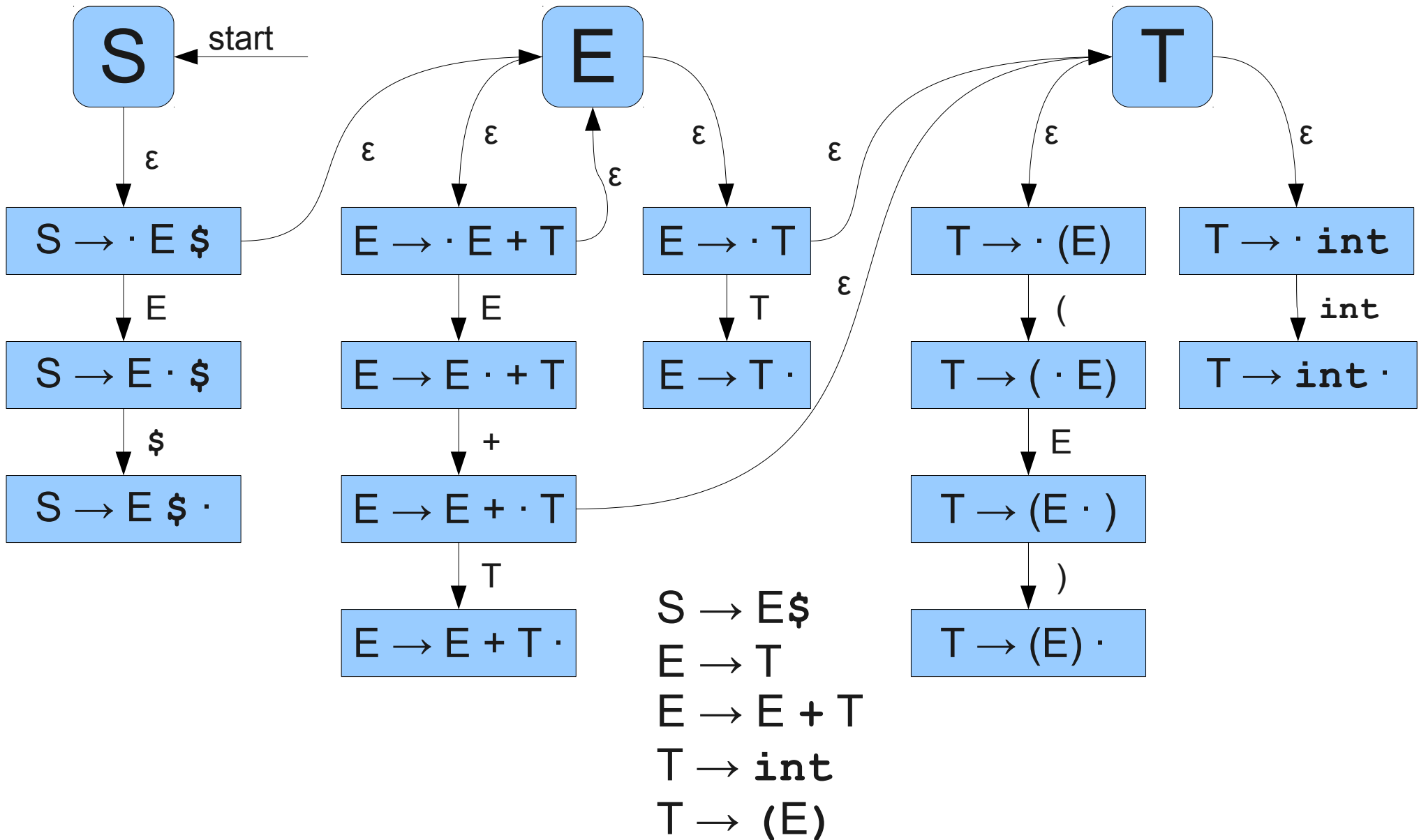
An Automaton for Viable Prefixes



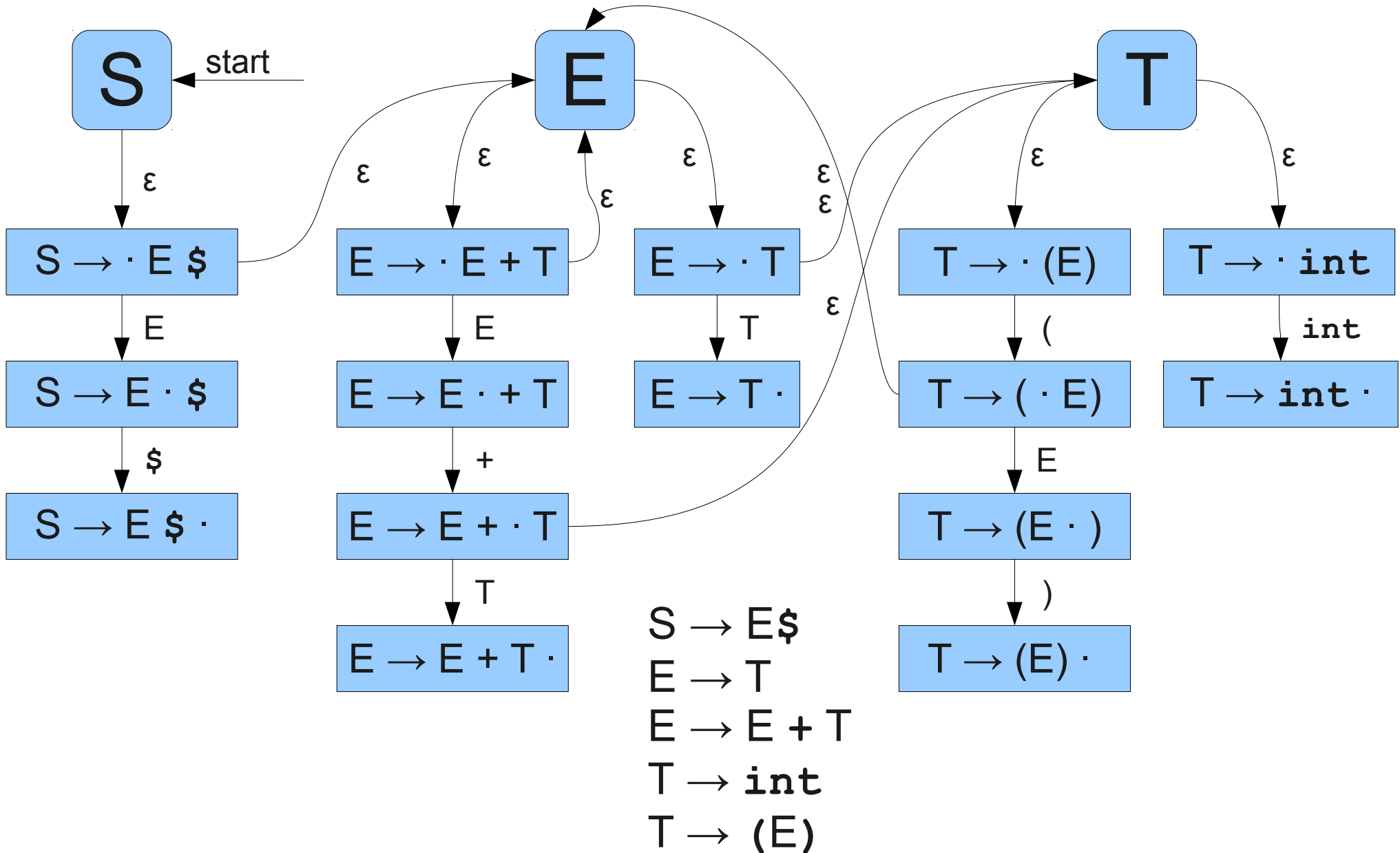
An Automaton for Viable Prefixes



An Automaton for Viable Prefixes



An Automaton for Viable Prefixes



Constructing the Automaton

- Create a state for each nonterminal.
- For each production $A \rightarrow a_1 a_2 \dots a_n$:
 - Construct states $A \rightarrow a_1 a_2 \dots a_i \cdot a_{i+1} \dots a_n$ for $0 \leq i \leq n$
 - Add transitions on x between $A \rightarrow u \cdot xw$ and $A \rightarrow ux \cdot w$
- For each state $A \rightarrow u \cdot Bv$ for nonterminal B , add an ε -transition from $A \rightarrow u \cdot Bv$ to B .

A Deterministic Automaton

$S \rightarrow E\$$

$E \rightarrow T$

$E \rightarrow E + T$

$T \rightarrow \mathbf{int}$

$T \rightarrow (E)$

A Deterministic Automaton

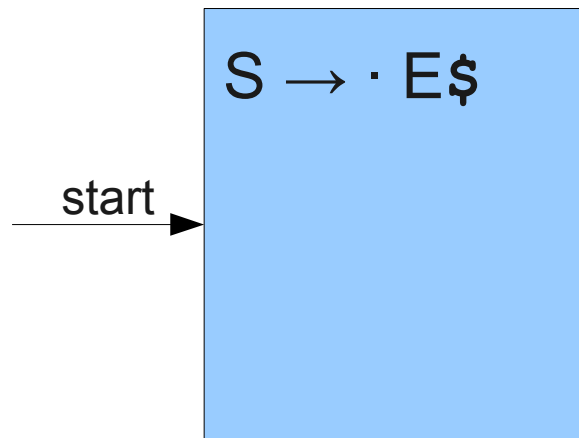
$S \rightarrow E\$$

$E \rightarrow T$

$E \rightarrow E + T$

$T \rightarrow \mathbf{int}$

$T \rightarrow (E)$



A Deterministic Automaton

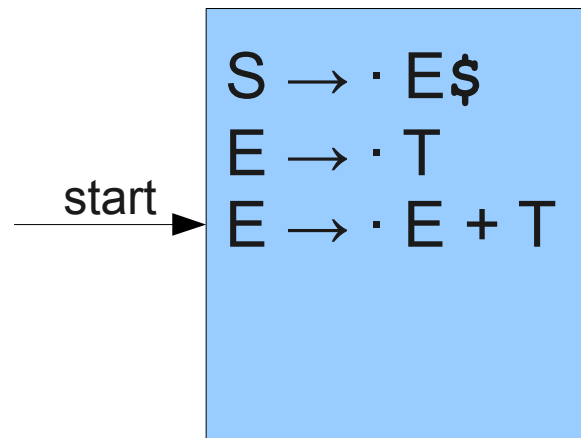
$S \rightarrow E\$$

$E \rightarrow T$

$E \rightarrow E + T$

$T \rightarrow \text{int}$

$T \rightarrow (E)$



A Deterministic Automaton

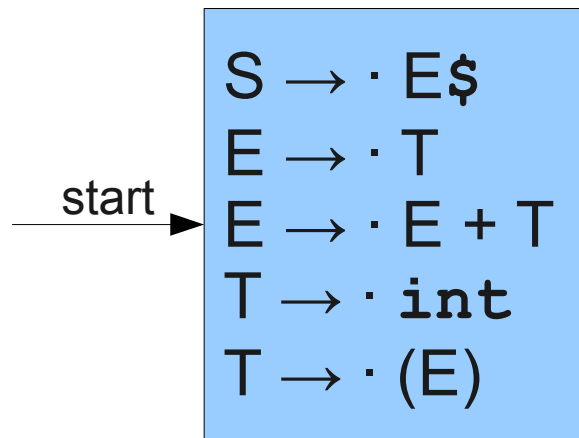
$S \rightarrow E\$$

$E \rightarrow T$

$E \rightarrow E + T$

$T \rightarrow \mathbf{int}$

$T \rightarrow (E)$



A Deterministic Automaton

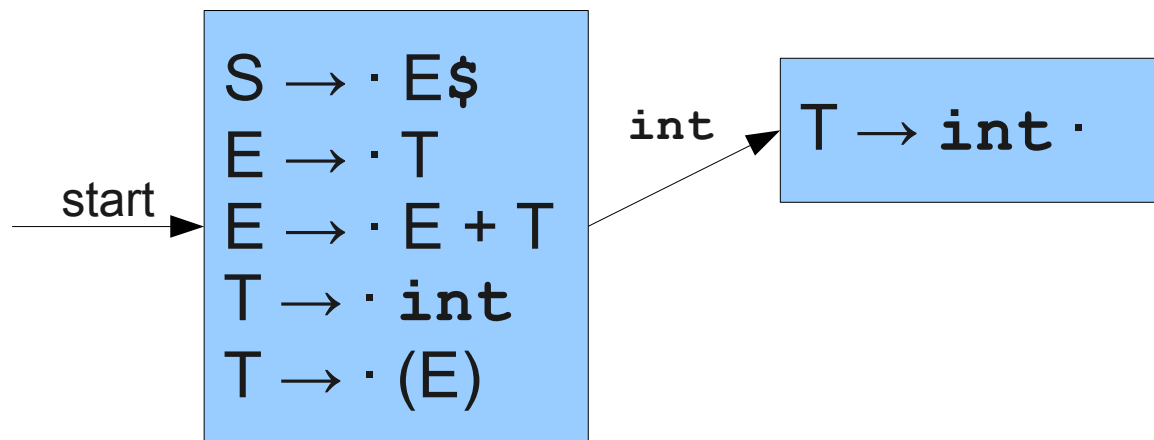
$S \rightarrow E\$$

$E \rightarrow T$

$E \rightarrow E + T$

$T \rightarrow \text{int}$

$T \rightarrow (E)$



A Deterministic Automaton

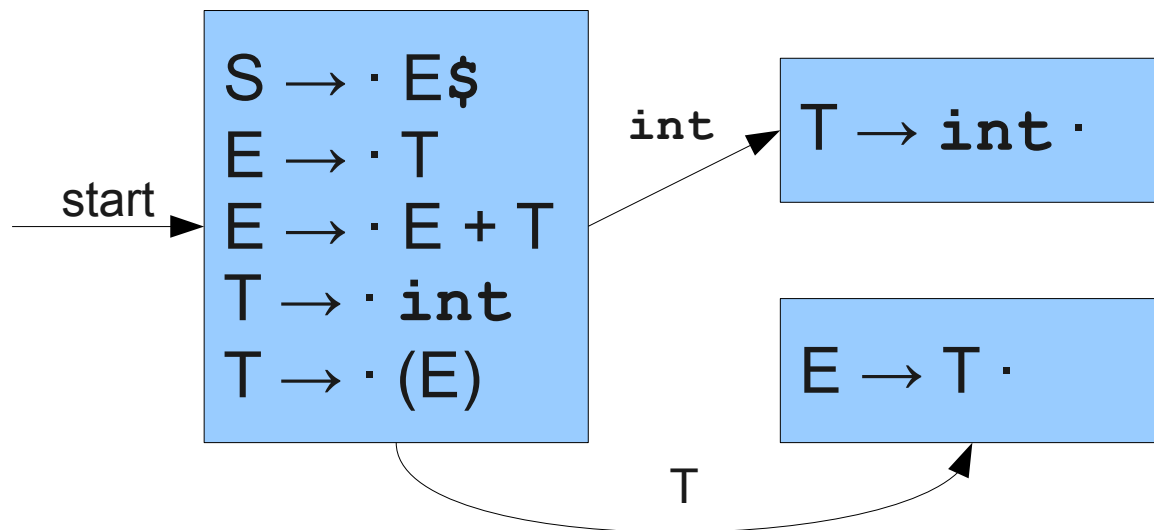
$S \rightarrow E\$$

$E \rightarrow T$

$E \rightarrow E + T$

$T \rightarrow \text{int}$

$T \rightarrow (E)$



A Deterministic Automaton

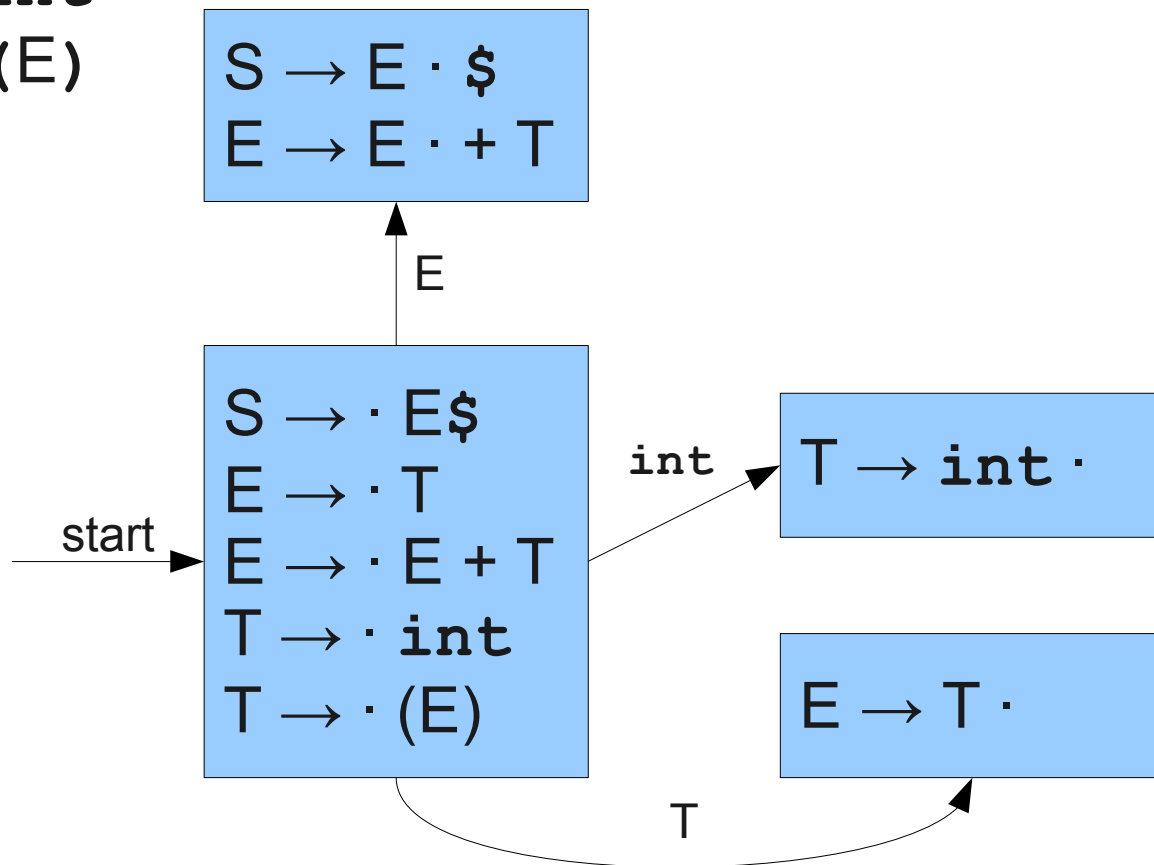
$S \rightarrow E\$$

$E \rightarrow T$

$E \rightarrow E + T$

$T \rightarrow \text{int}$

$T \rightarrow (E)$



A Deterministic Automaton

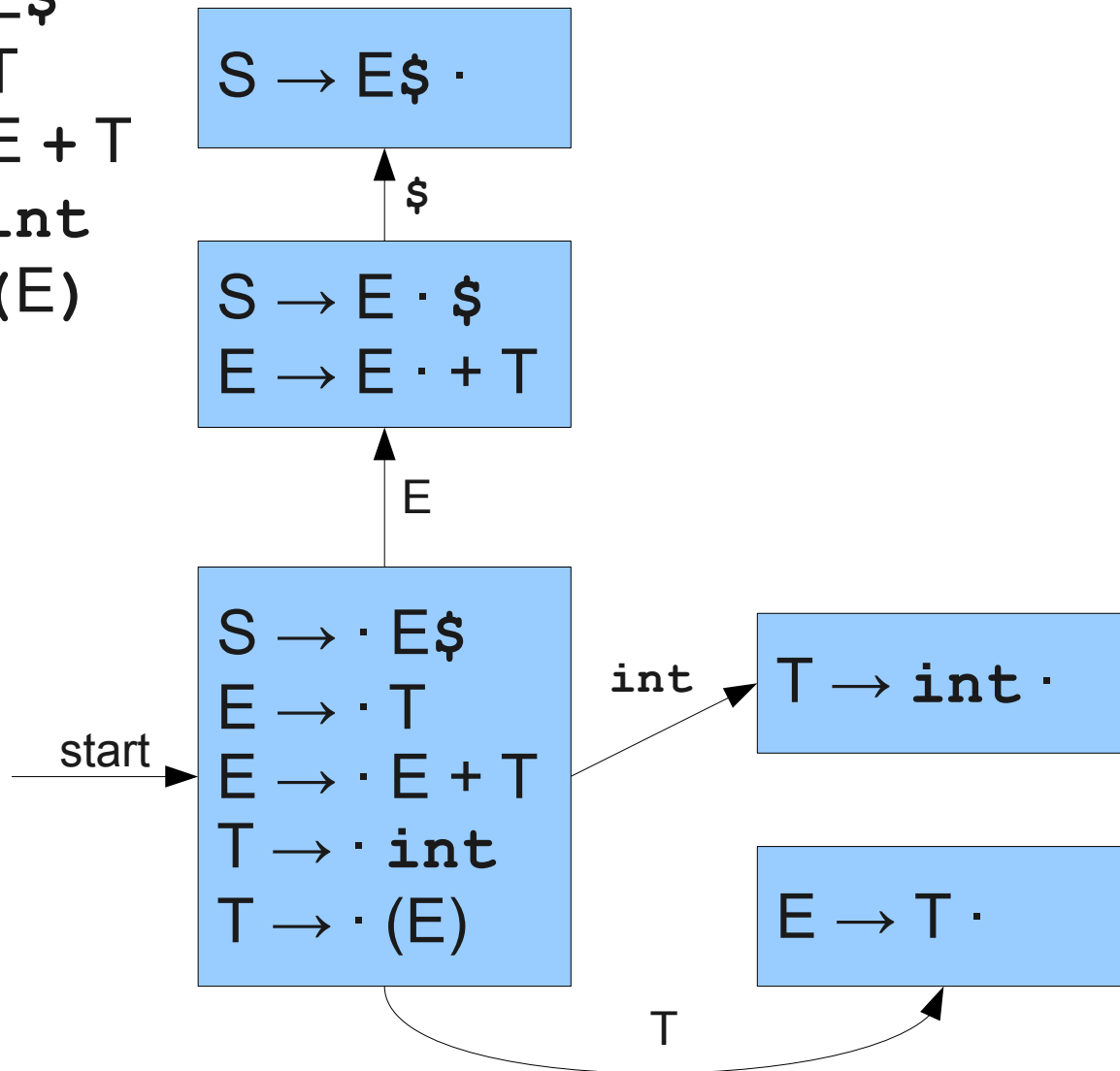
$S \rightarrow E\$$

$E \rightarrow T$

$E \rightarrow E + T$

$T \rightarrow \text{int}$

$T \rightarrow (E)$



A Deterministic Automaton

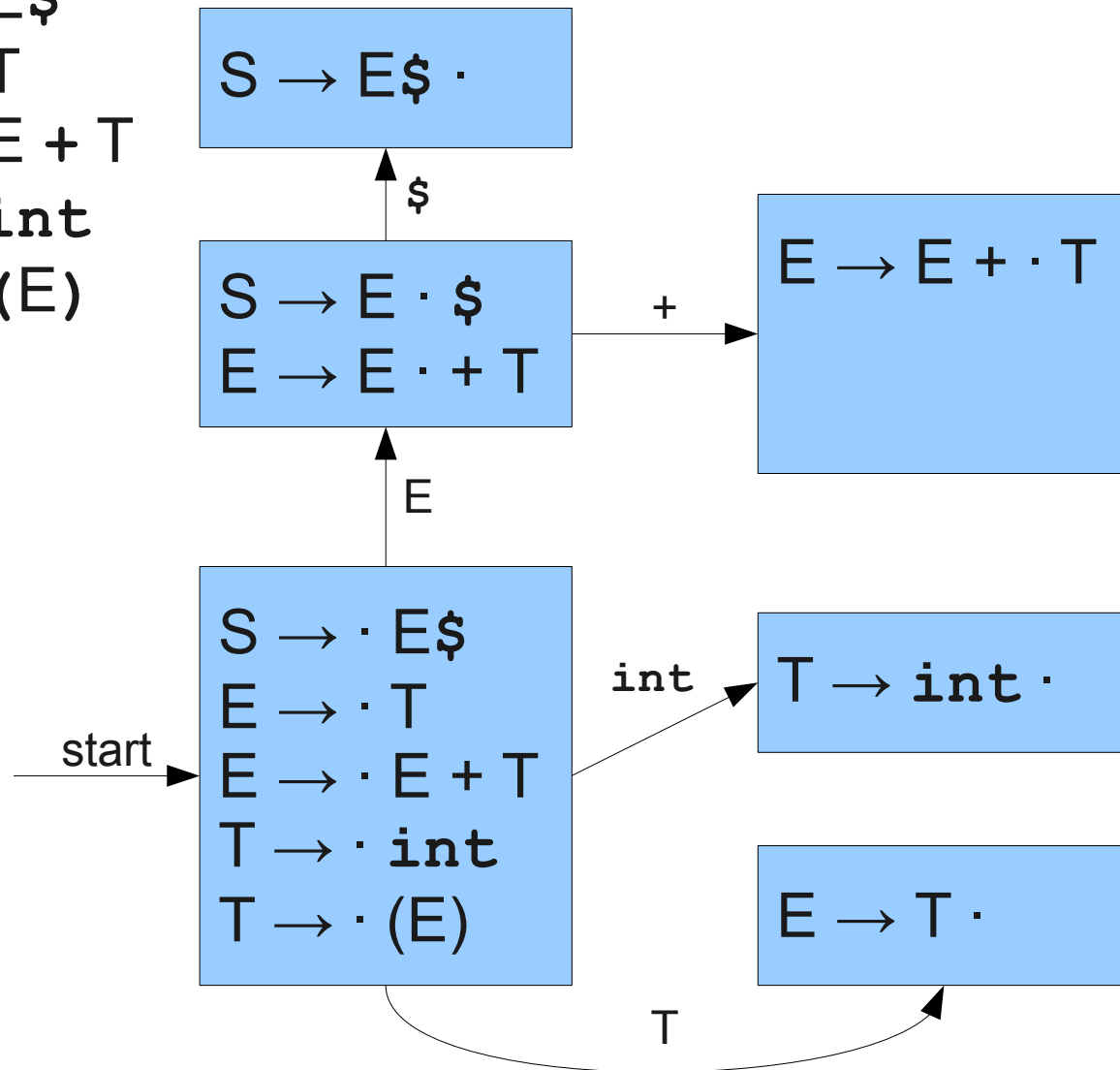
$S \rightarrow E\$$

$E \rightarrow T$

$E \rightarrow E + T$

$T \rightarrow \text{int}$

$T \rightarrow (E)$



A Deterministic Automaton

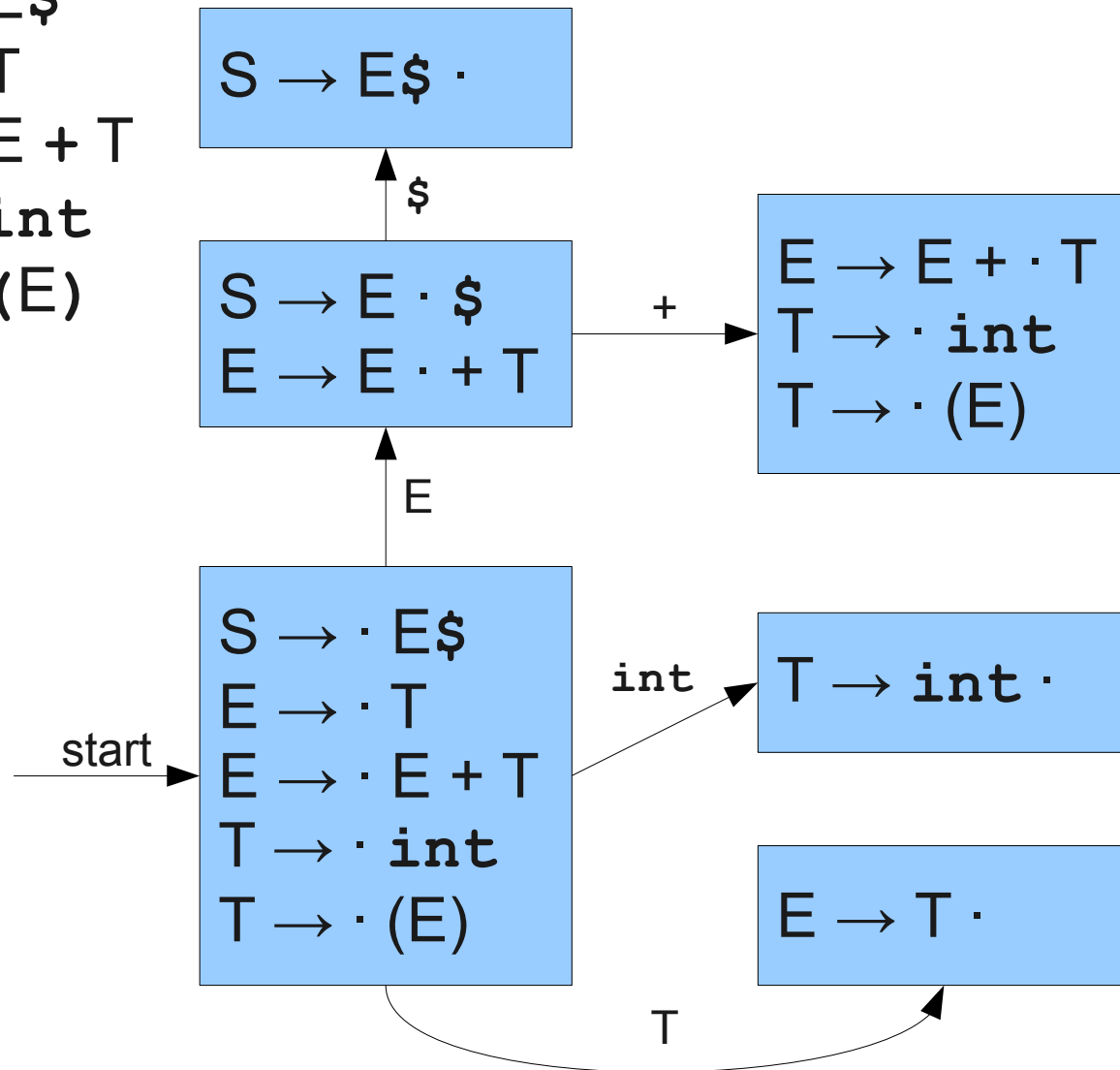
$S \rightarrow E\$$

$E \rightarrow T$

$E \rightarrow E + T$

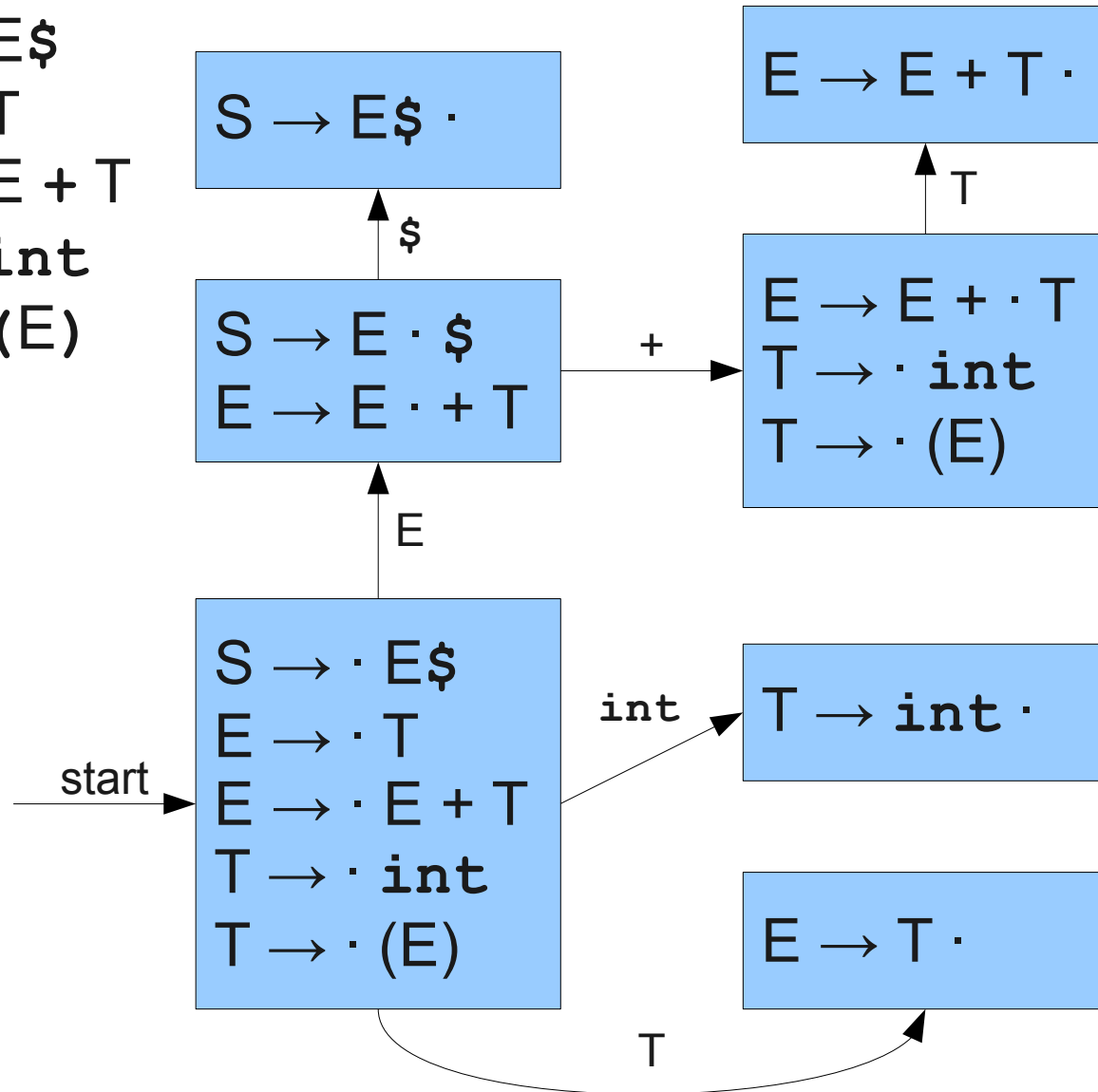
$T \rightarrow \text{int}$

$T \rightarrow (E)$



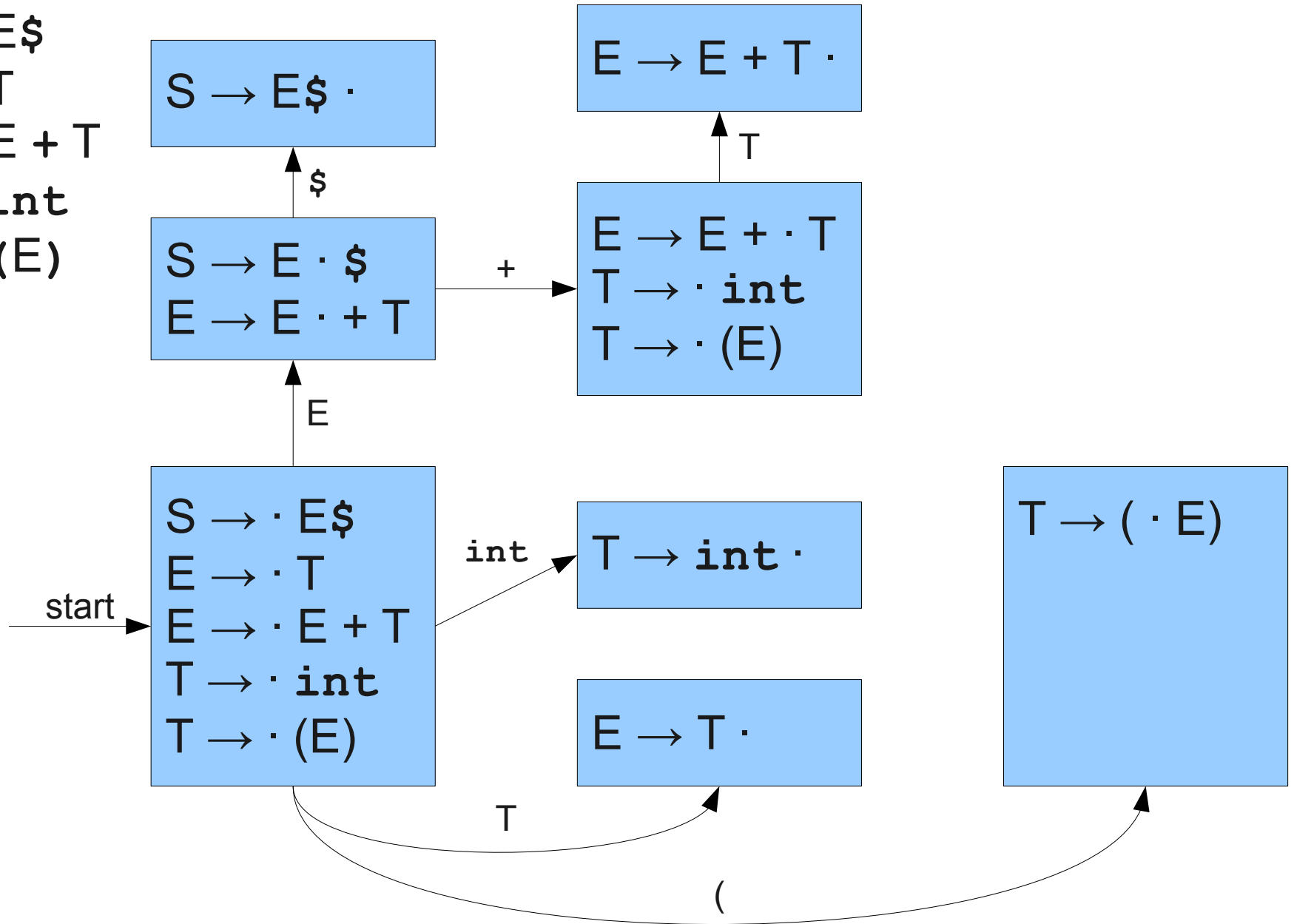
A Deterministic Automaton

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



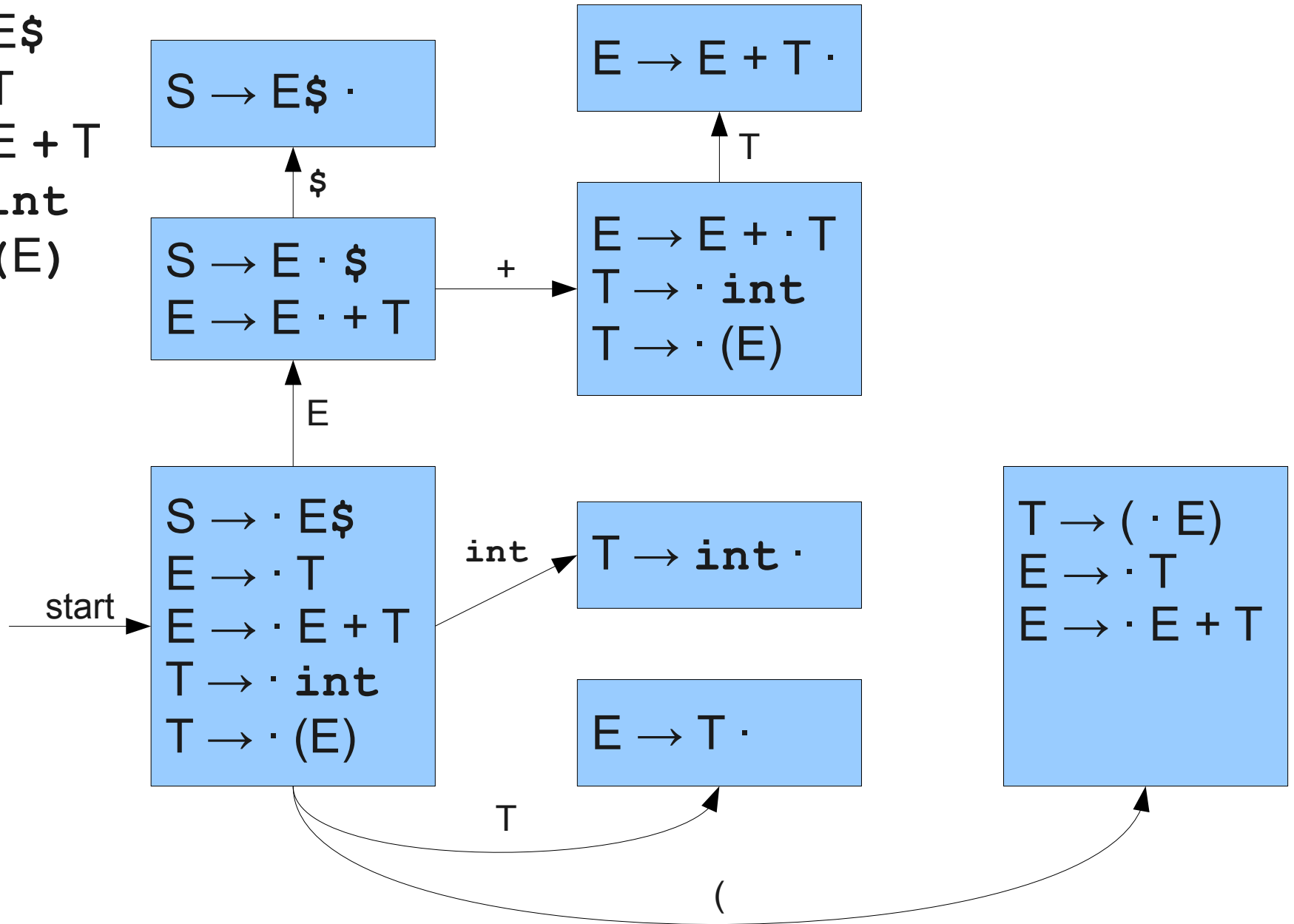
A Deterministic Automaton

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



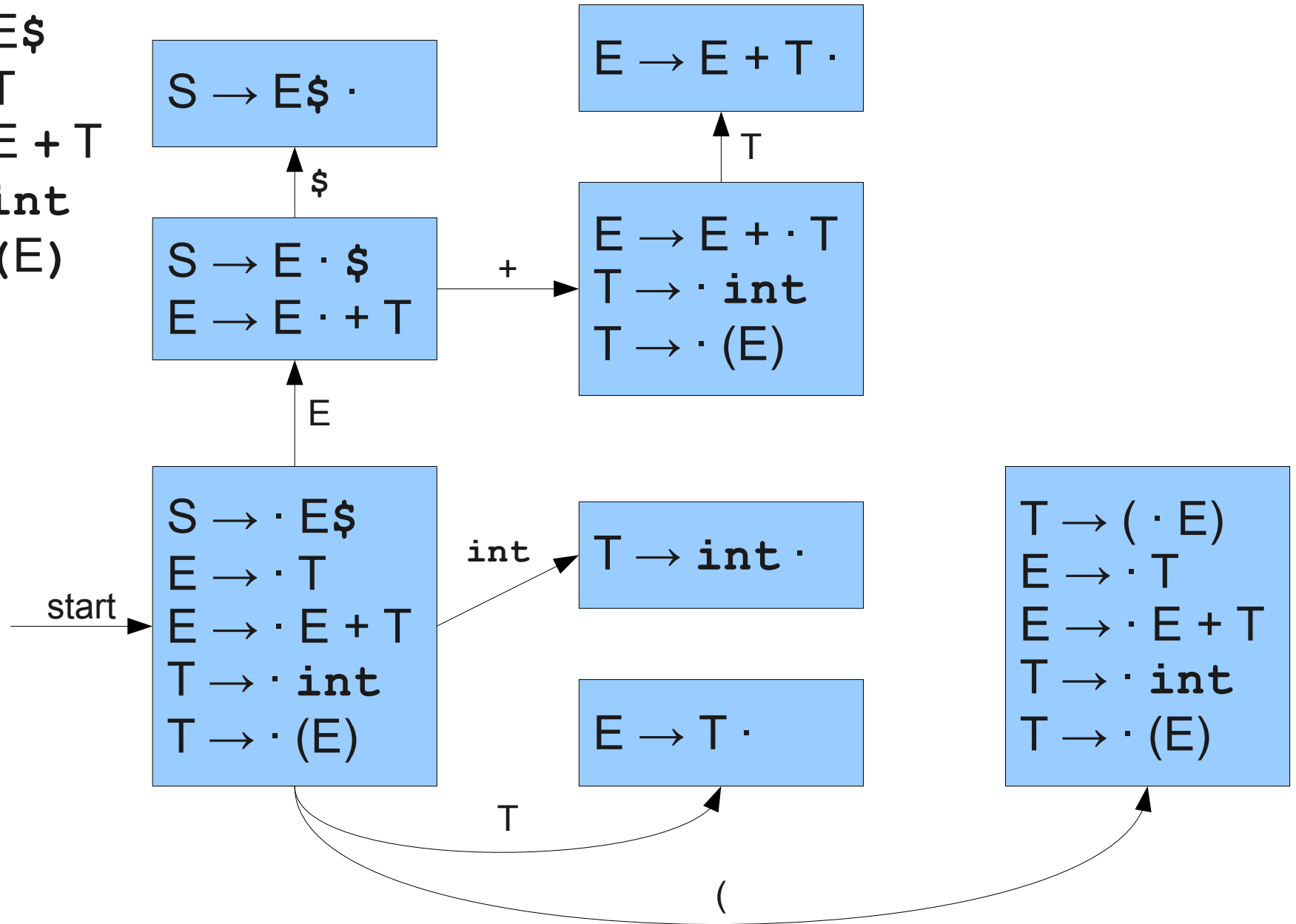
A Deterministic Automaton

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



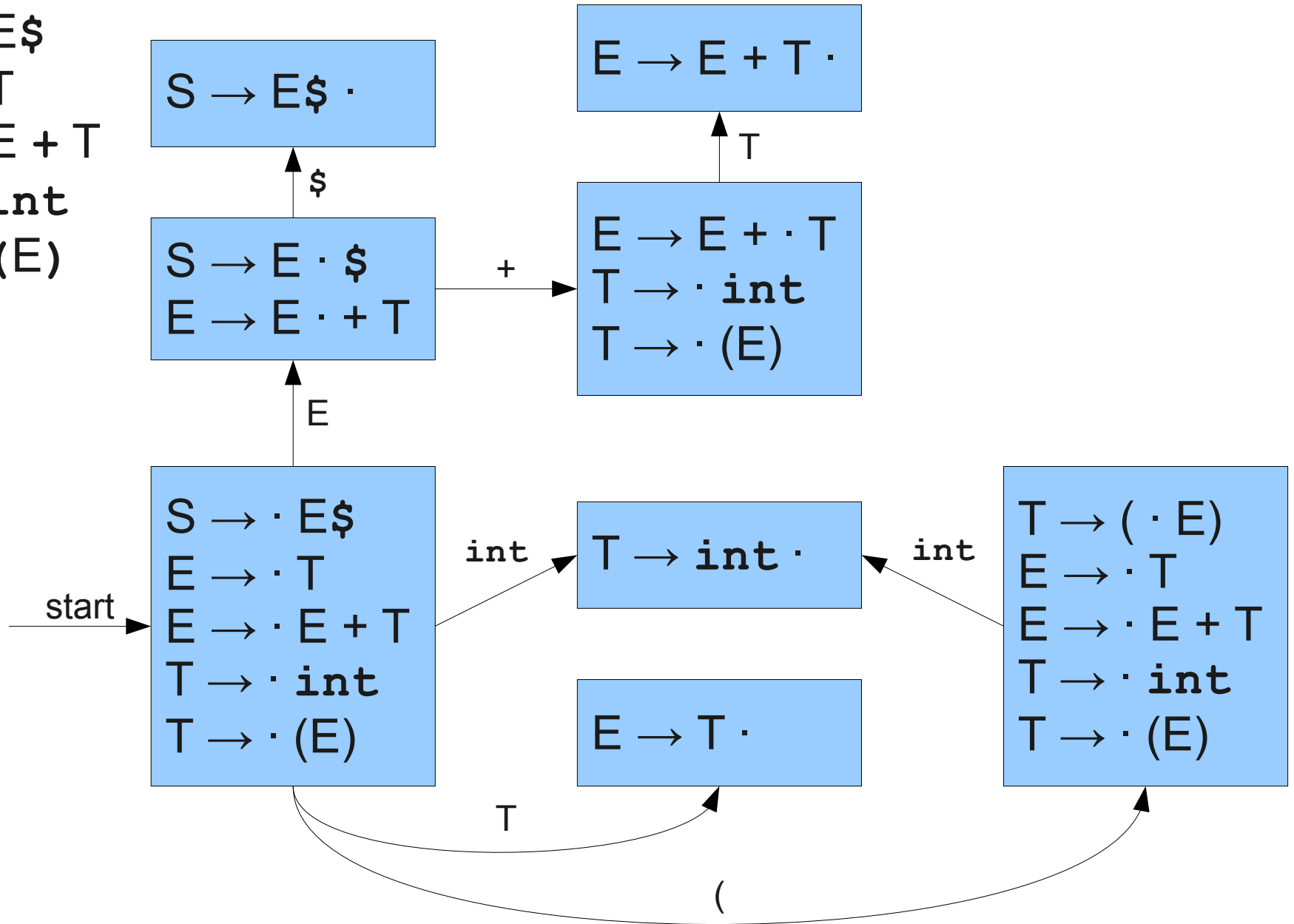
A Deterministic Automaton

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



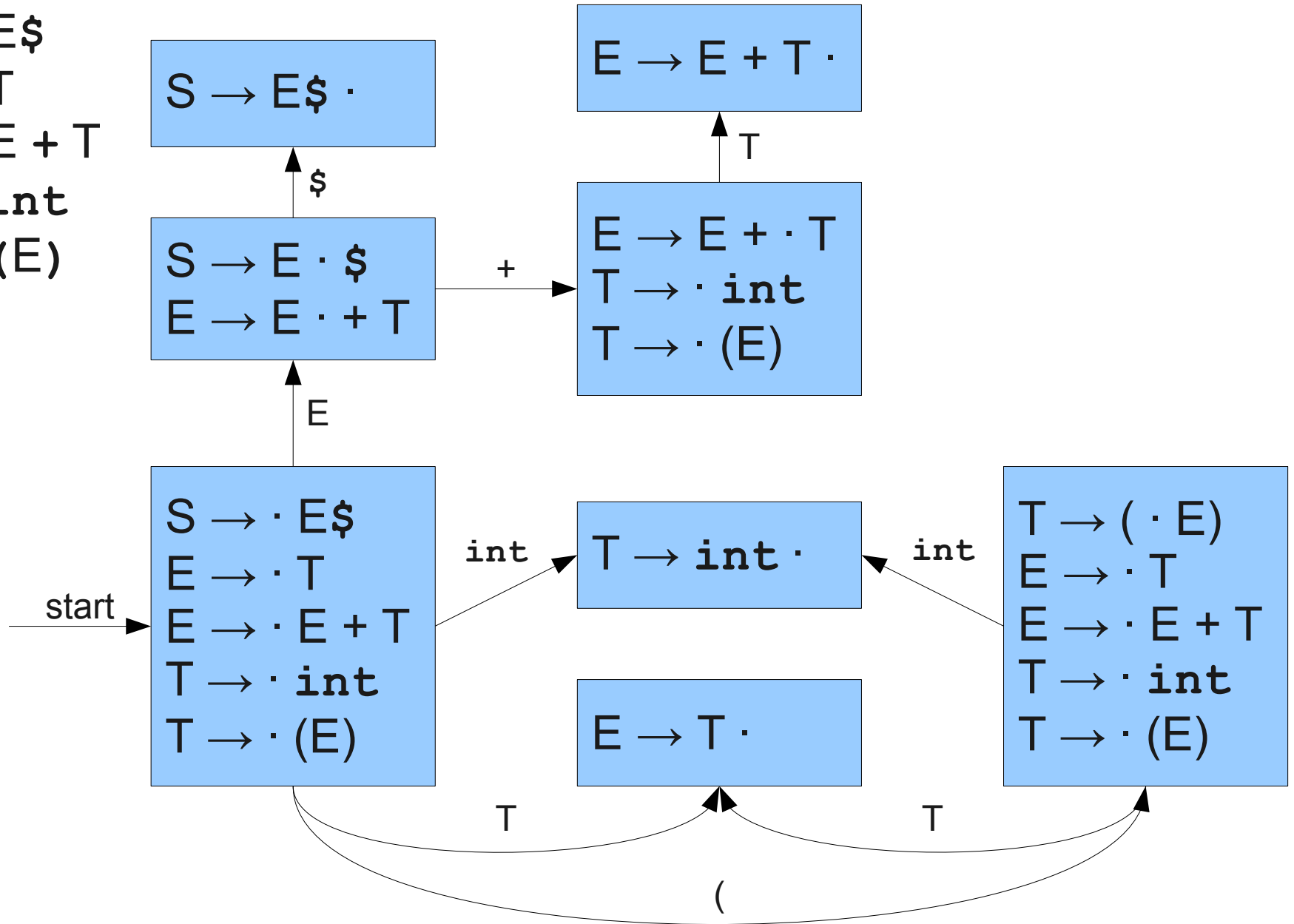
A Deterministic Automaton

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



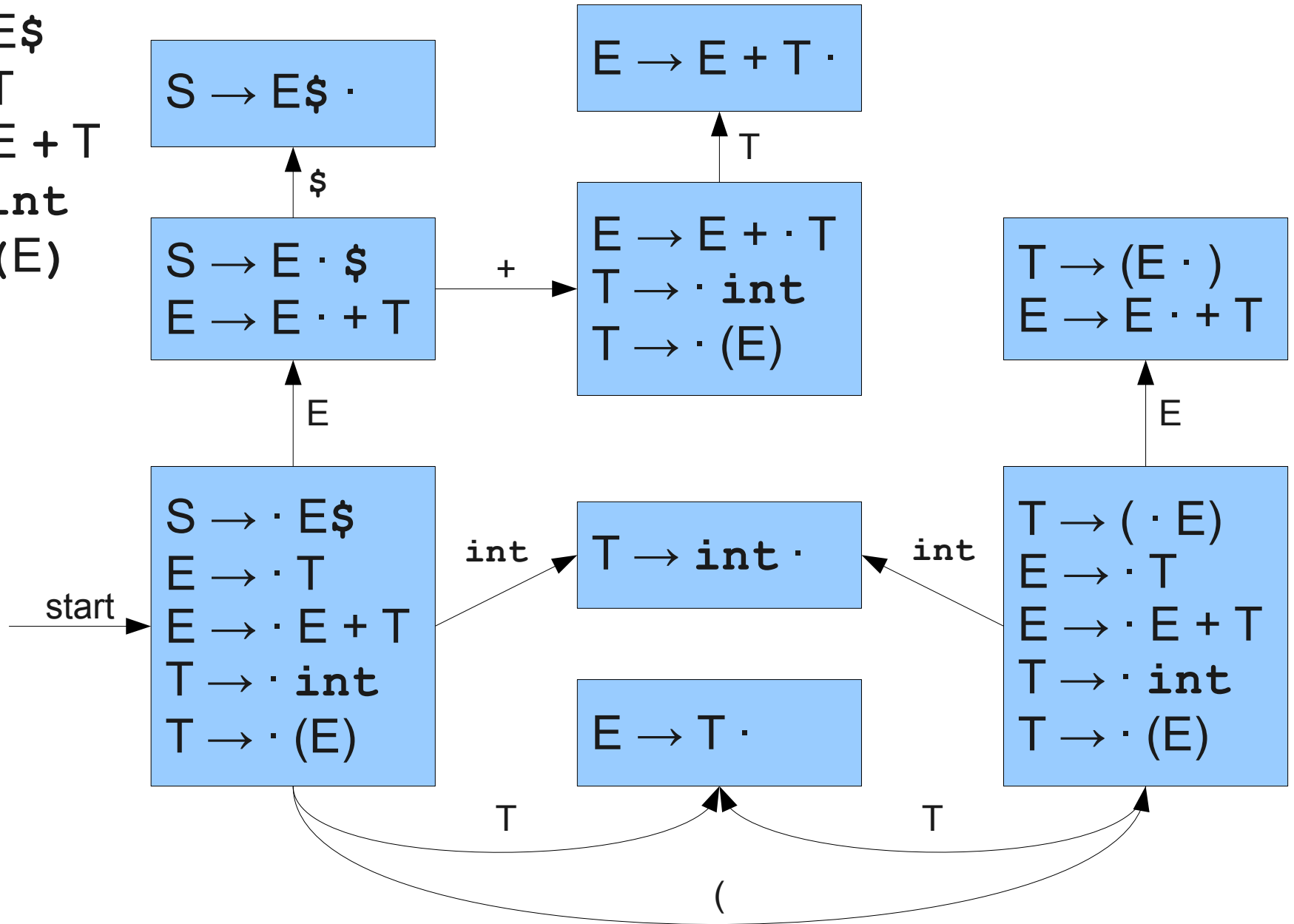
A Deterministic Automaton

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



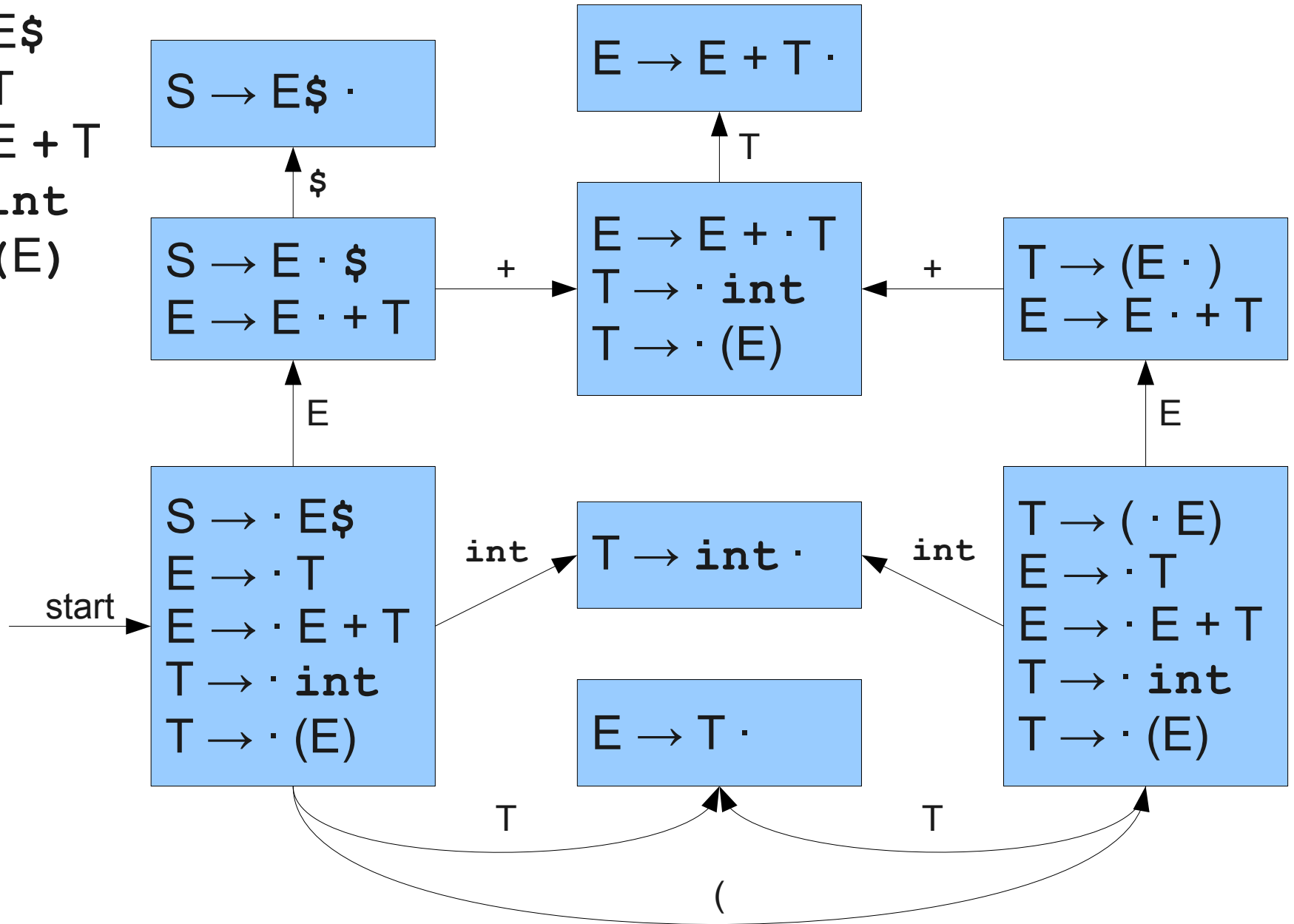
A Deterministic Automaton

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



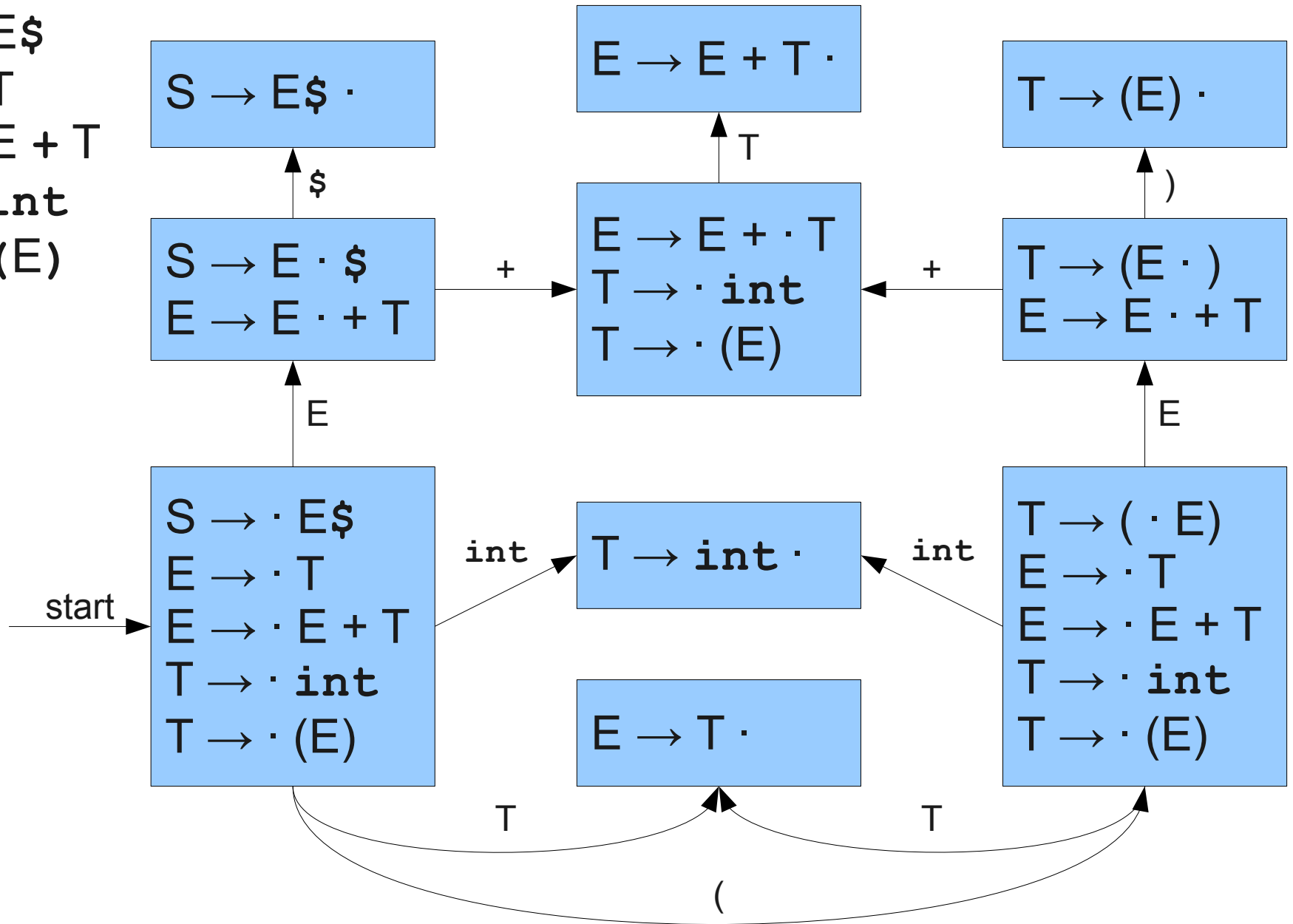
A Deterministic Automaton

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



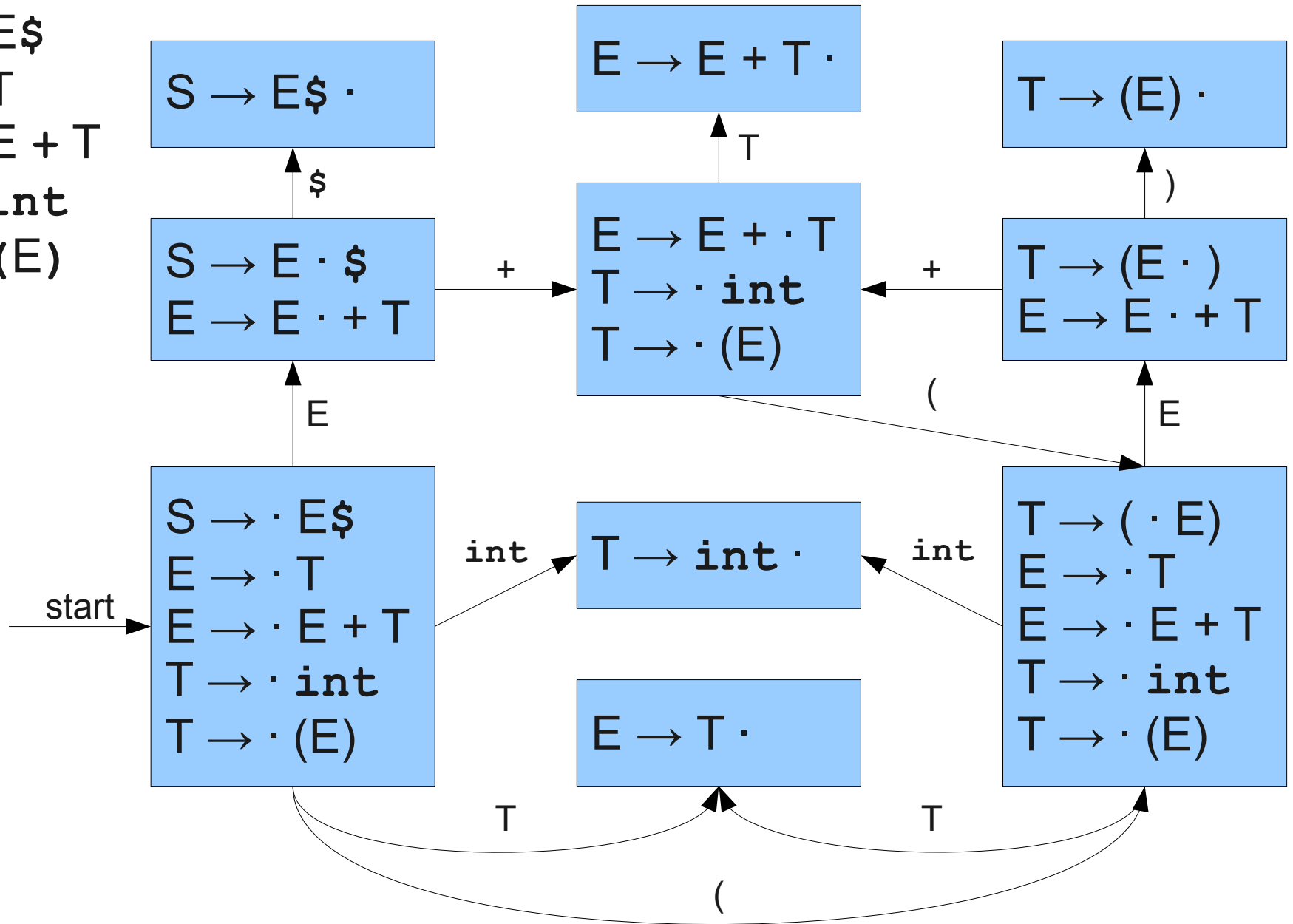
A Deterministic Automaton

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



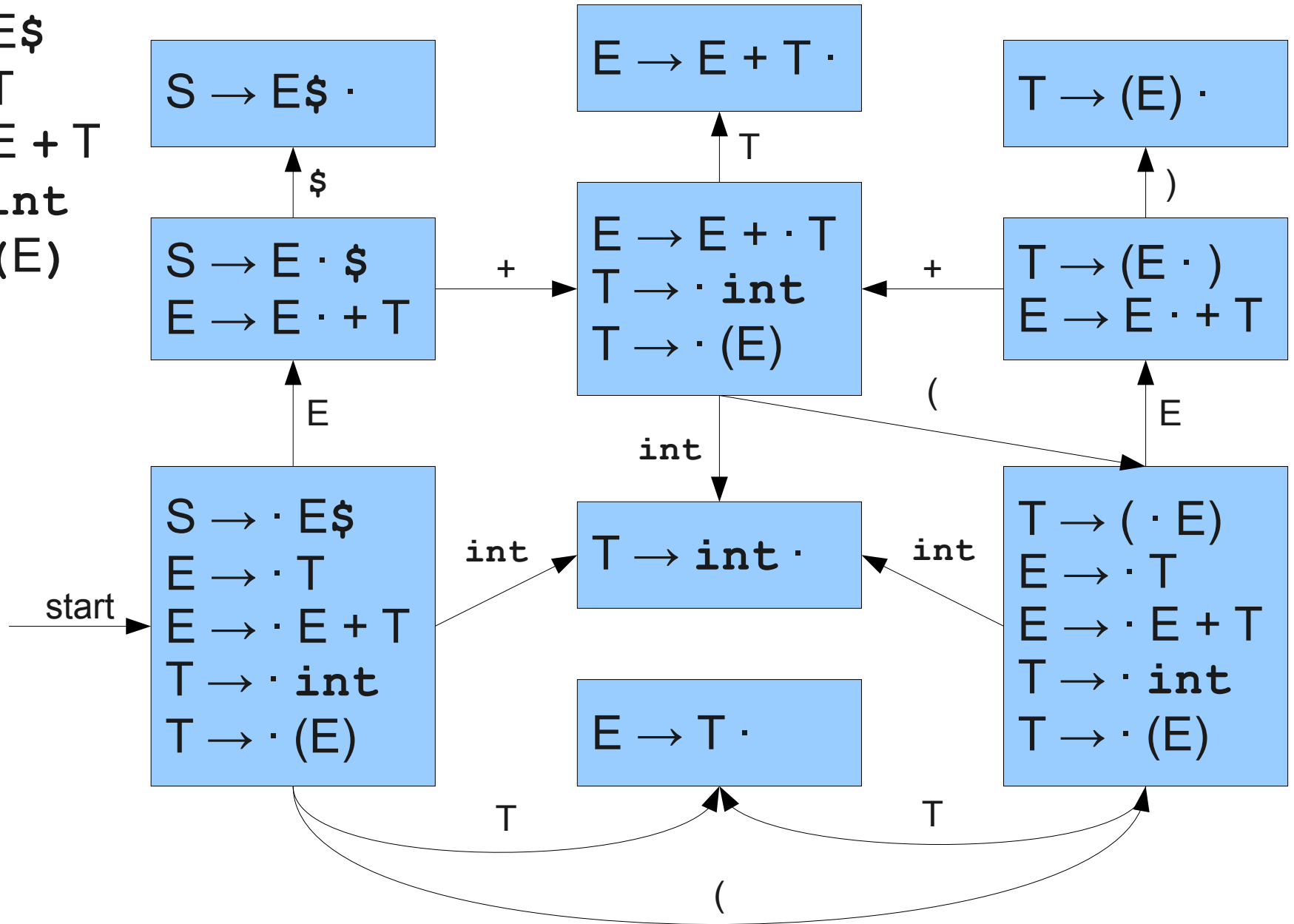
A Deterministic Automaton

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



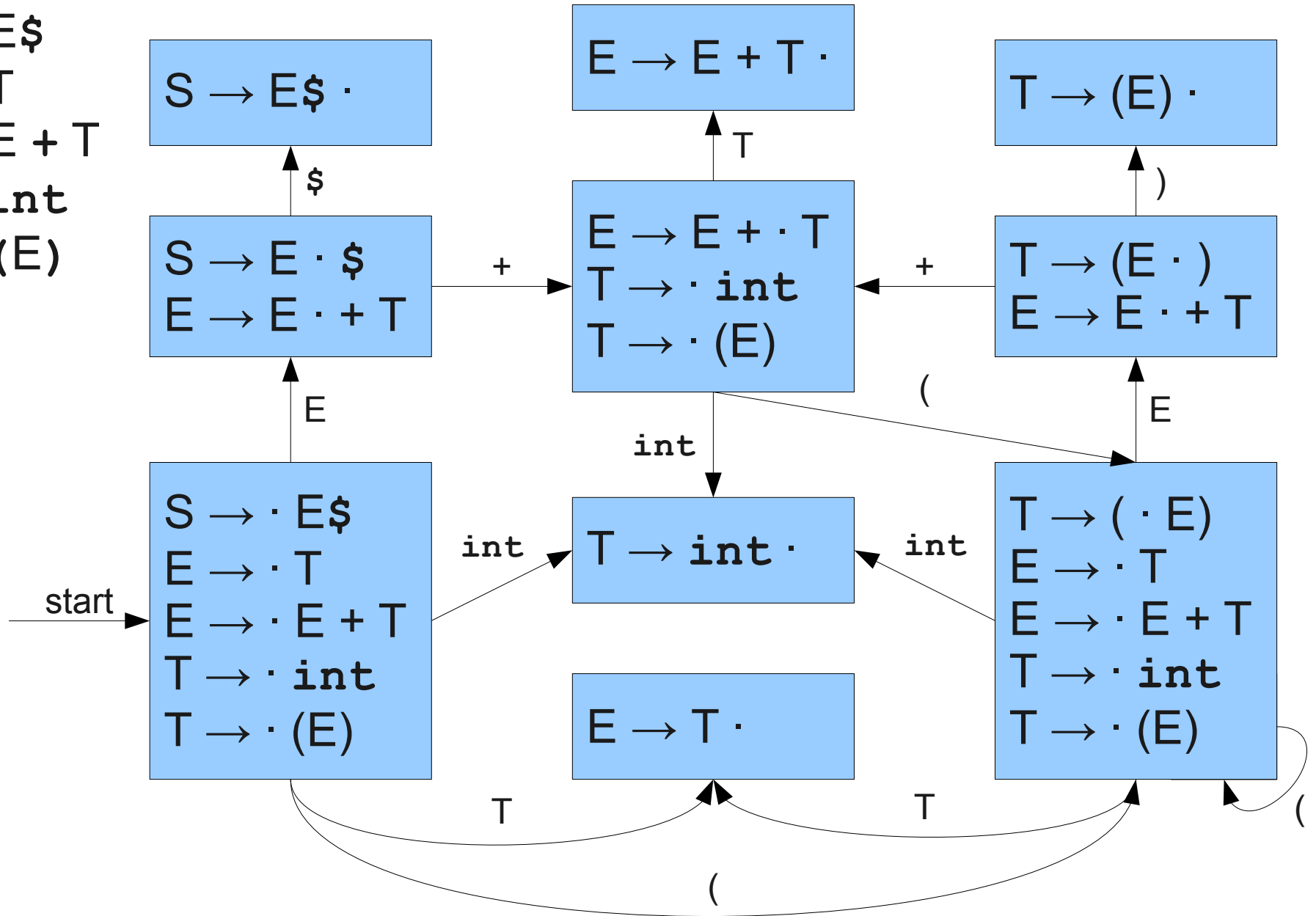
A Deterministic Automaton

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



A Deterministic Automaton

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



Constructing the Automaton II

- Begin in a state containing $S \rightarrow \cdot E$, where S is the start symbol.
- Compute the **closure** of the state:
 - If $A \rightarrow v \cdot Bw$ is in the state, add $B \rightarrow \cdot x$ to the state for each production $B \rightarrow x$.
 - Yet another **fixed-point iteration!**
- Repeat until no new states are added:
 - If a state contains a production $A \rightarrow v \cdot \mathbf{x}w$ for symbol \mathbf{x} , add a transition on \mathbf{x} from that state to the state containing the closure of $A \rightarrow v\mathbf{x} \cdot w$
- Interestingly, this is **equivalent** to doing a subset construction on the NFA!

A Note on Handle-Finding Automata

- Handling-finding automata can be very large.
- NFA has states proportional to the size of the grammar, so DFA can have size **exponential** in the size of the grammar.
 - There are grammars that can exhibit this worst-case.
- Automata are almost always generated by tools like **bison**.

Finding Handles

- Where do we look for handles?
 - **At the top of the stack.**
- How do we search for handles?
 - **Build a handle-finding automaton.**
- How do we recognize handles?
 - Once we've found a possible handle, how do we confirm that it's correct?

Question Three:

How do we recognize handles?

Handle Recognition

- The general problem of recognizing handles is hard.
- We'll thus use **predictive bottom-up parsing**:
 - Have a deterministic procedure for guessing where handles are.
- There are many predictive algorithms, each of which recognize different grammars.

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

$S \rightarrow E\$$

$E \rightarrow T$

$E \rightarrow E + T$

$T \rightarrow \text{int}$

$T \rightarrow (E)$

int	+	(int	+	int	+	int)	\$
-----	---	---	-----	---	-----	---	-----	---	----

LR(0) Parsing


$S \rightarrow E\$$

$E \rightarrow T$

$E \rightarrow E + T$

$T \rightarrow \text{int}$

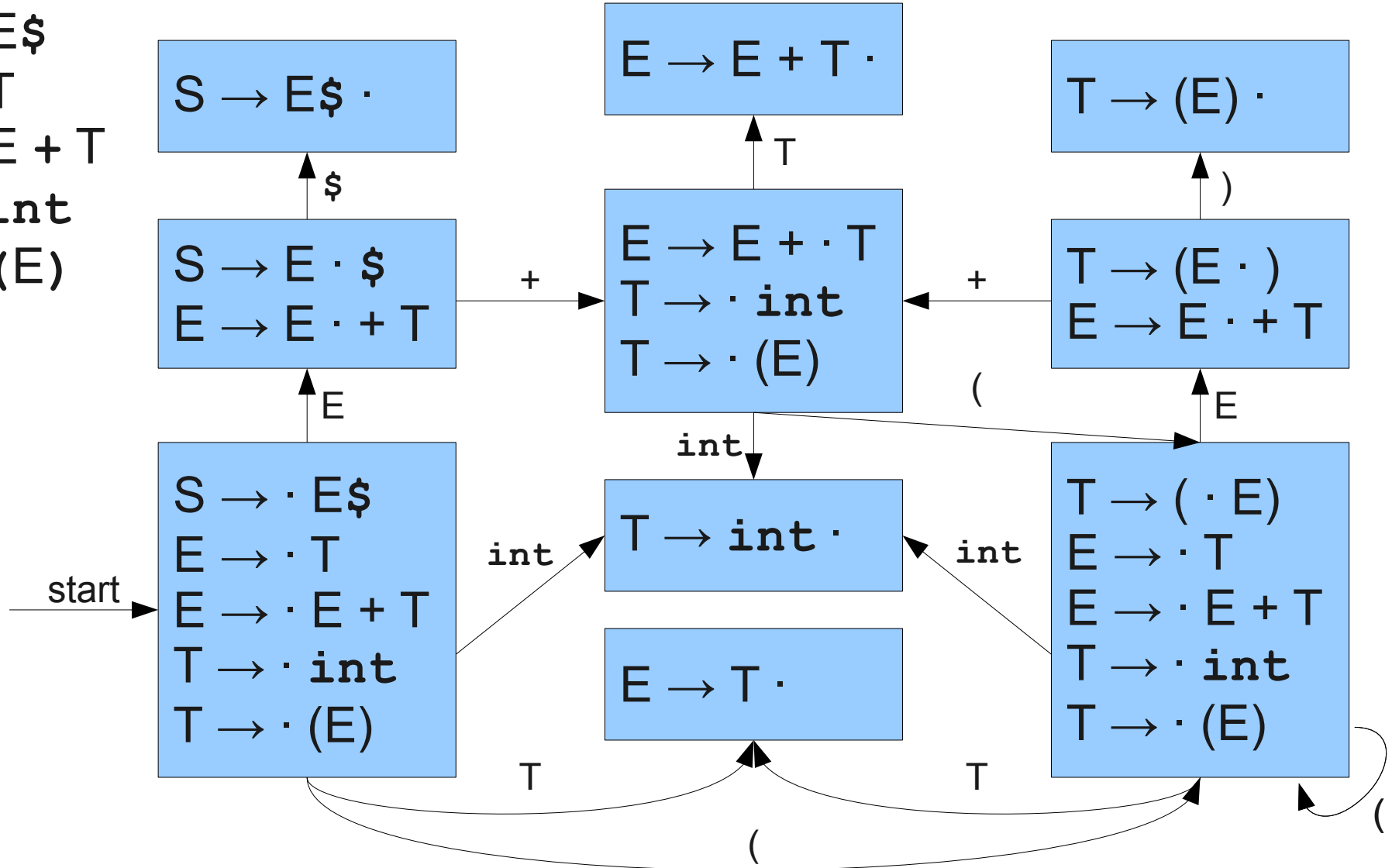
$T \rightarrow (E)$



int	+	(int	+	int	+	int)	\$
-----	---	---	-----	---	-----	---	-----	---	----

LR(0) Parsing

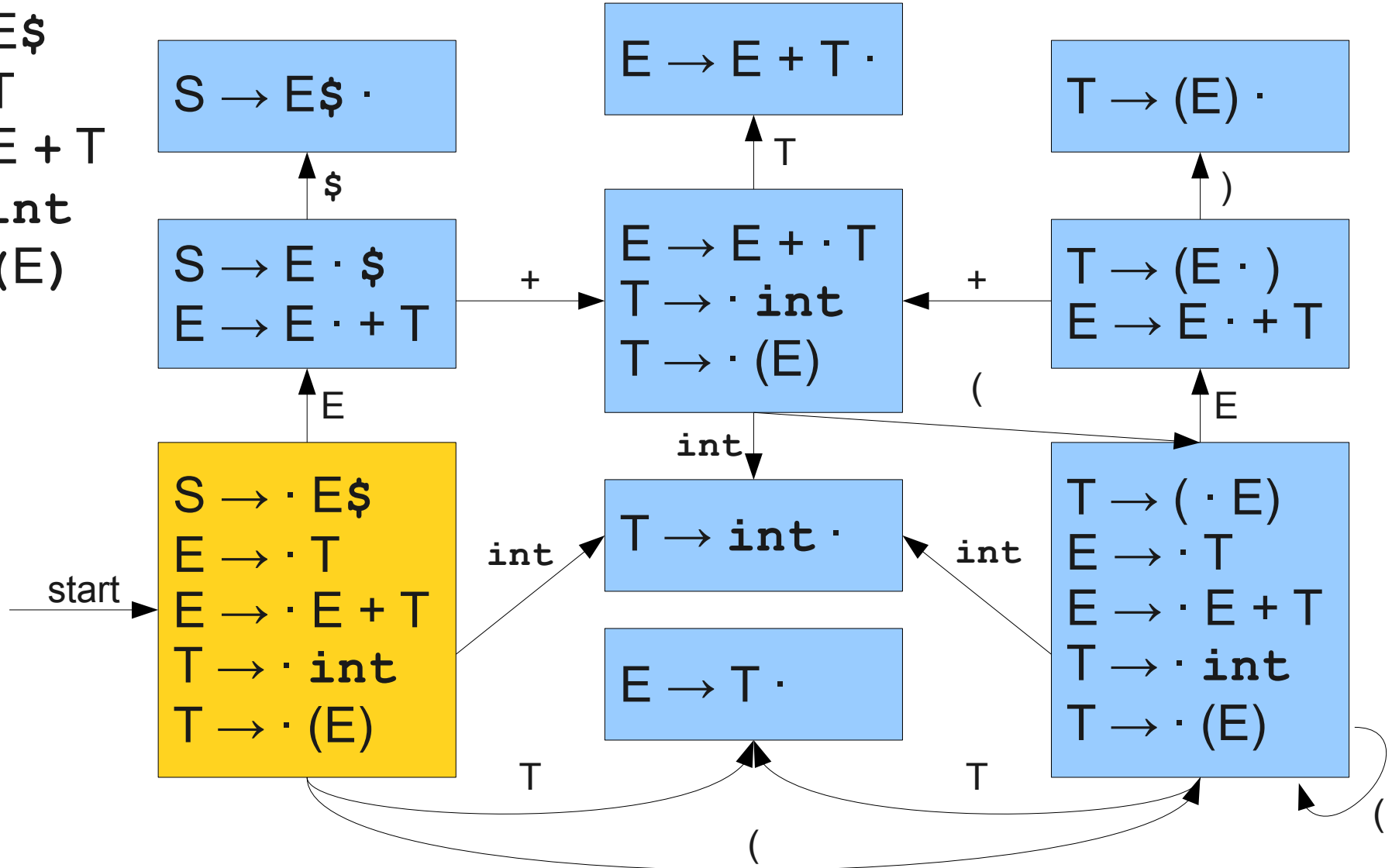
$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



int	+	(int	+	int	+	int)	\$
-----	---	---	-----	---	-----	---	-----	---	----

LR(0) Parsing

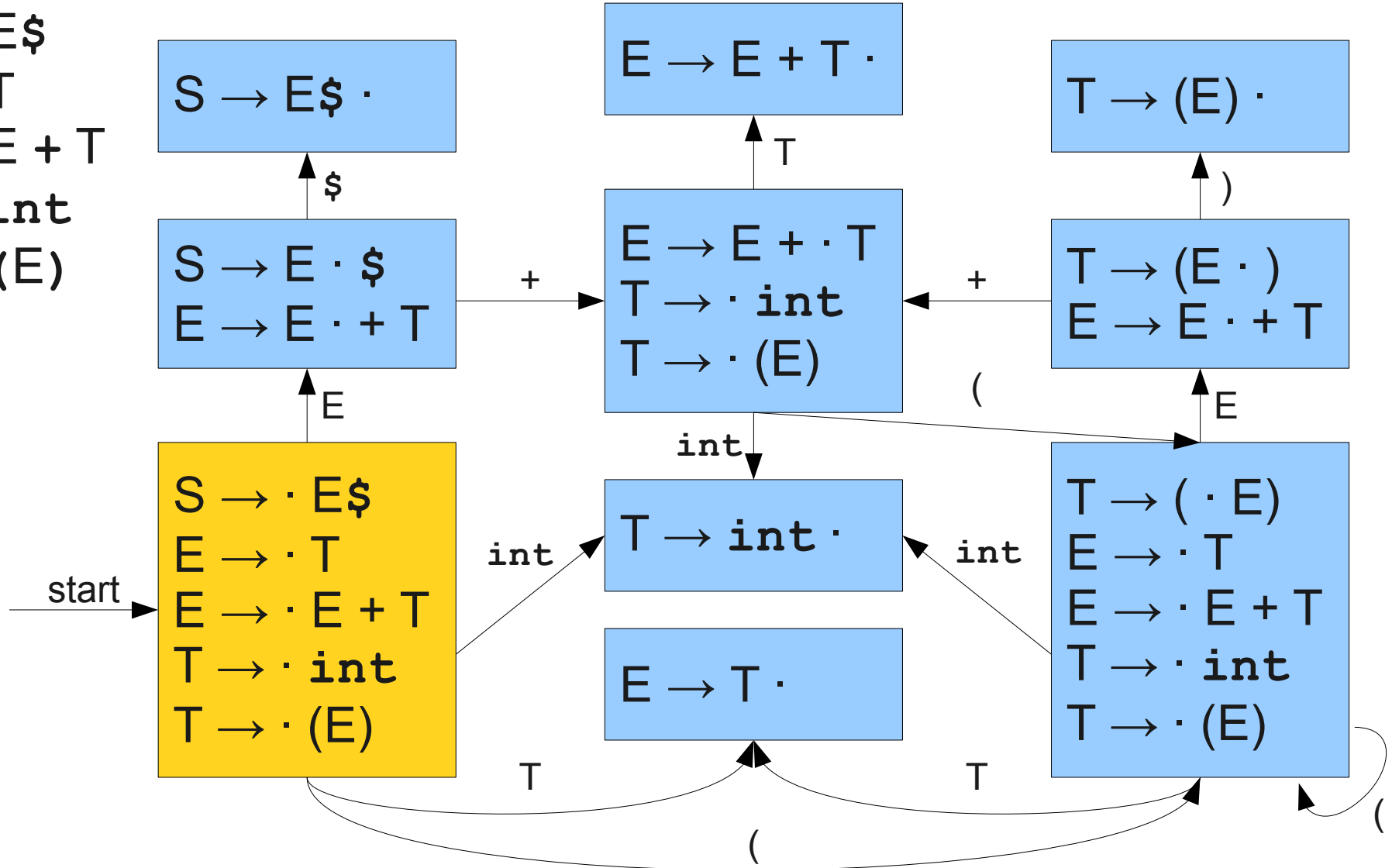
$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



int	+	(int	+	int	+	int)	\$
-----	---	---	-----	---	-----	---	-----	---	----

LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$

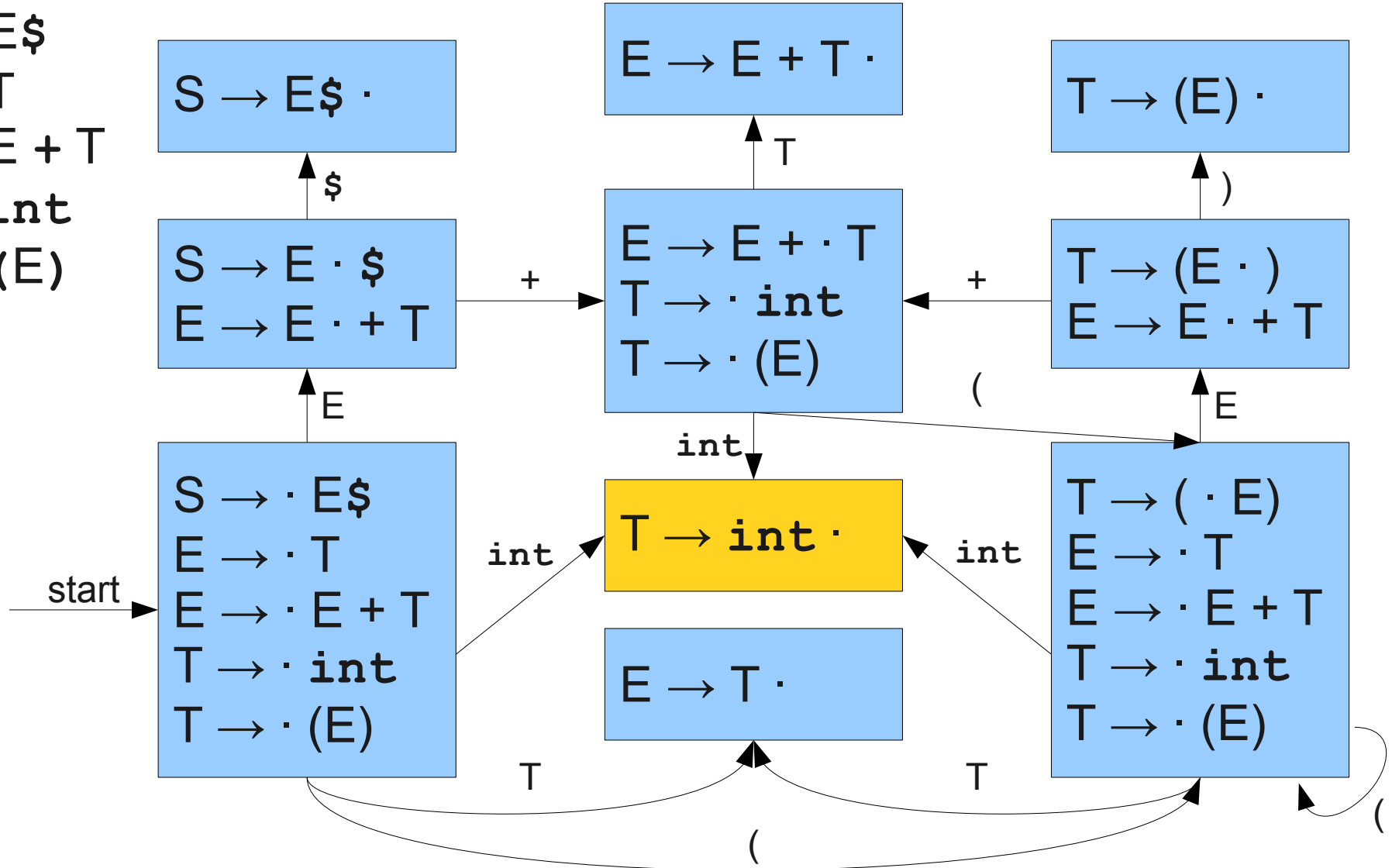


int

+	(int	+	int	+	int)	\$
---	---	-----	---	-----	---	-----	---	----

LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



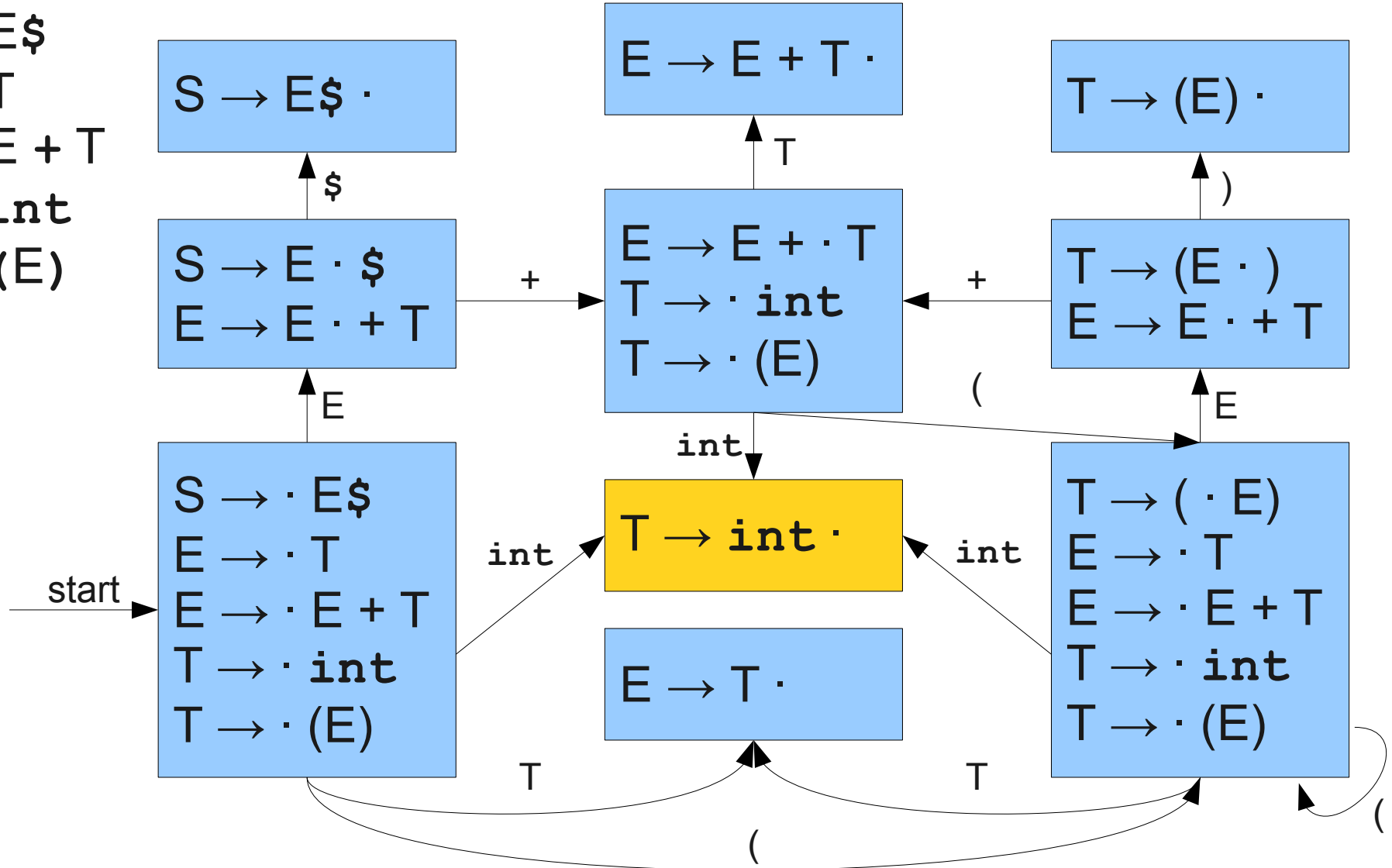
int



+	(int	+	int	+	int)	\$
---	---	-----	---	-----	---	-----	---	----

LR(0) Parsing

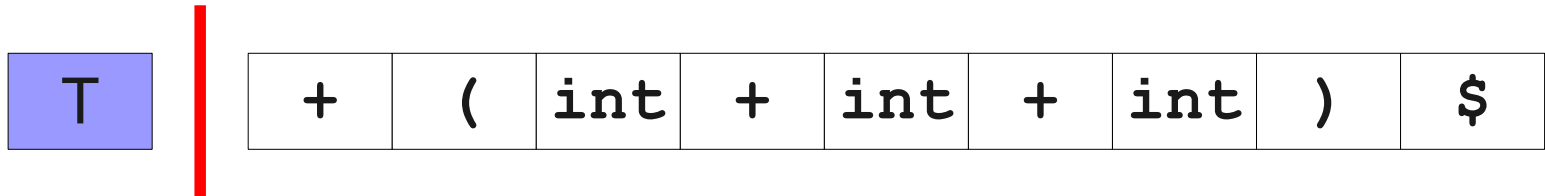
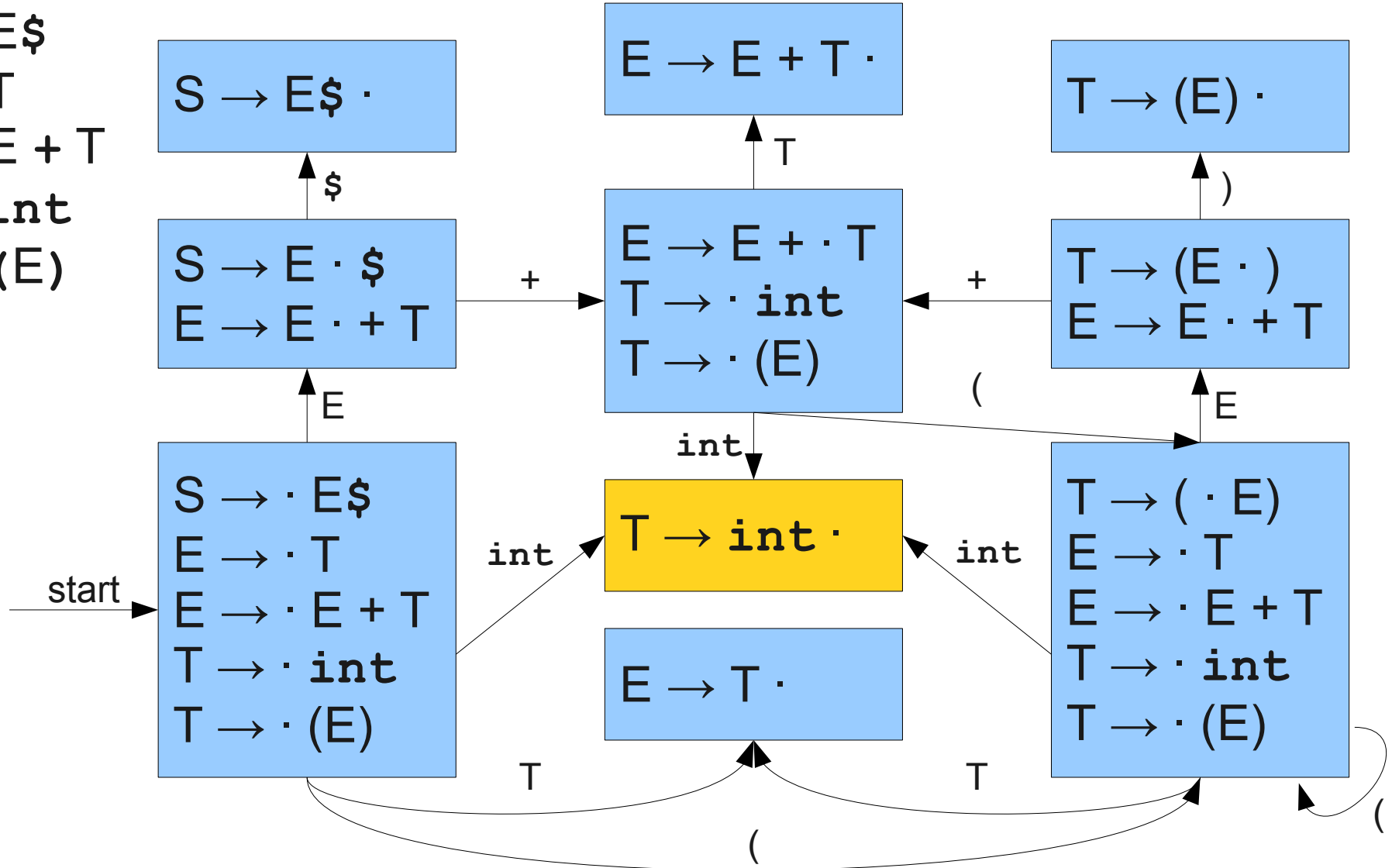
$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



+	(int	+	int	+	int)	\$
---	---	-----	---	-----	---	-----	---	----

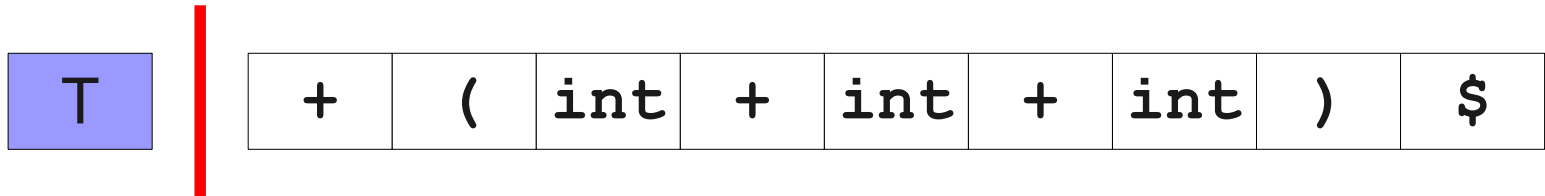
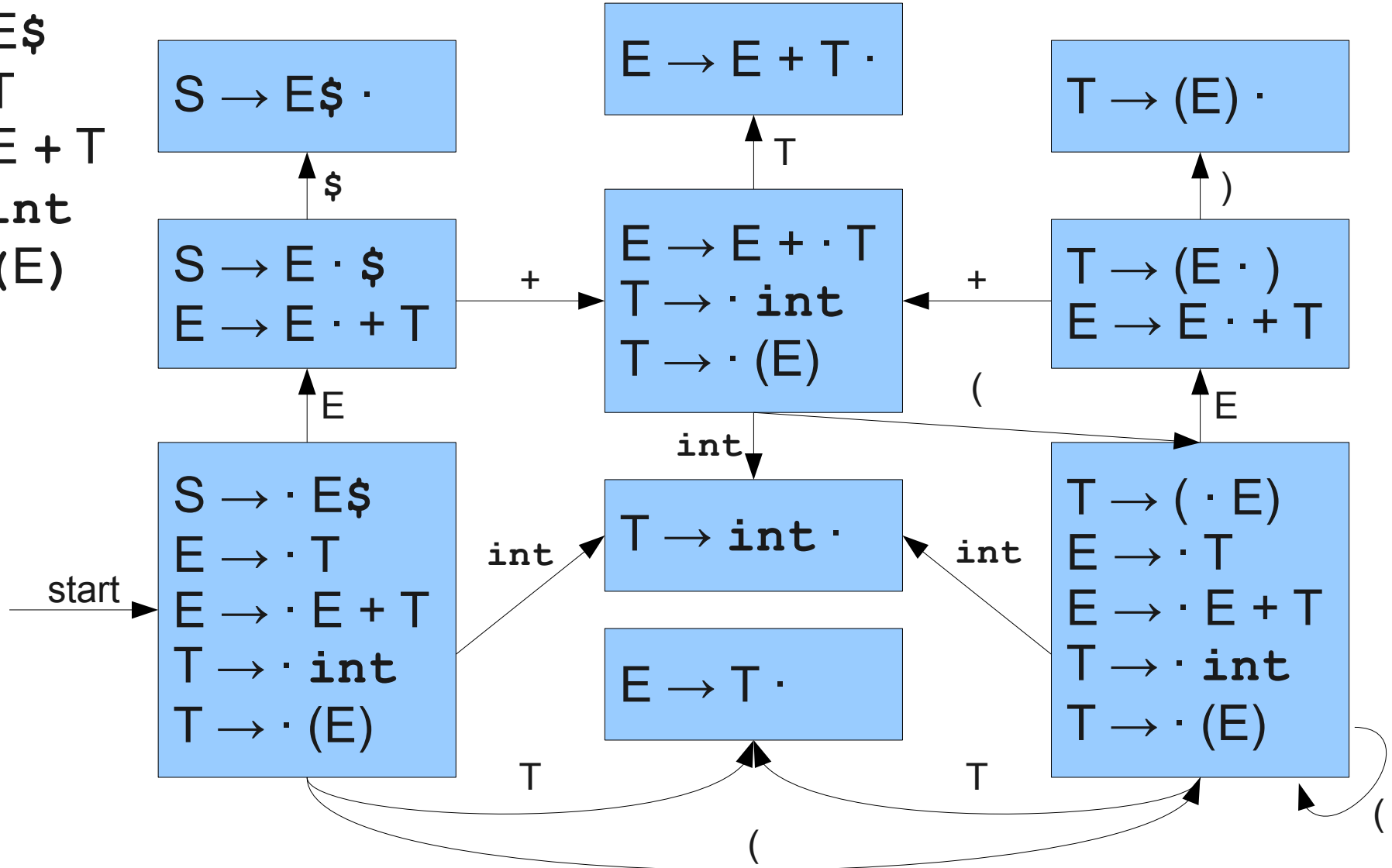
LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



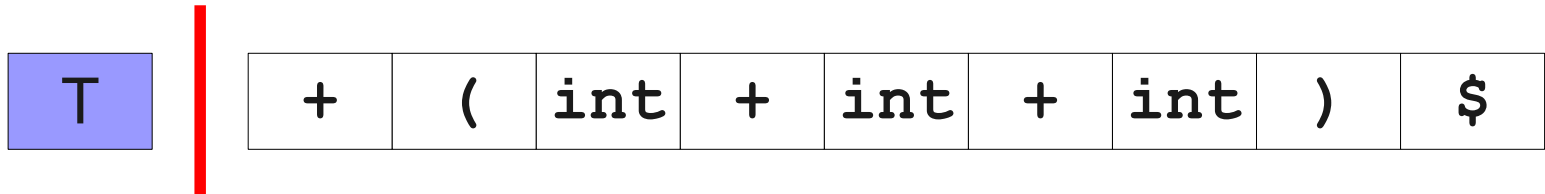
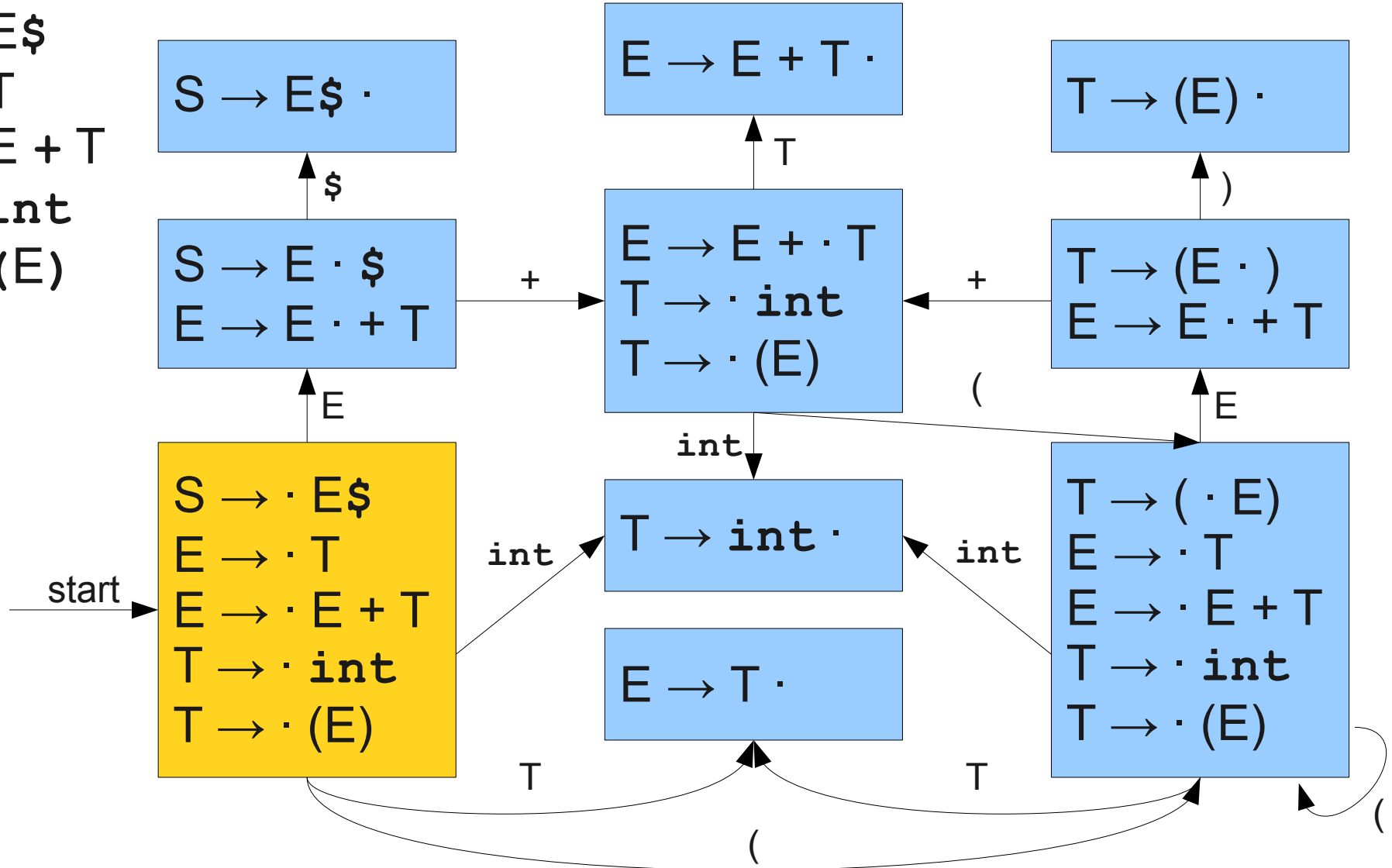
LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



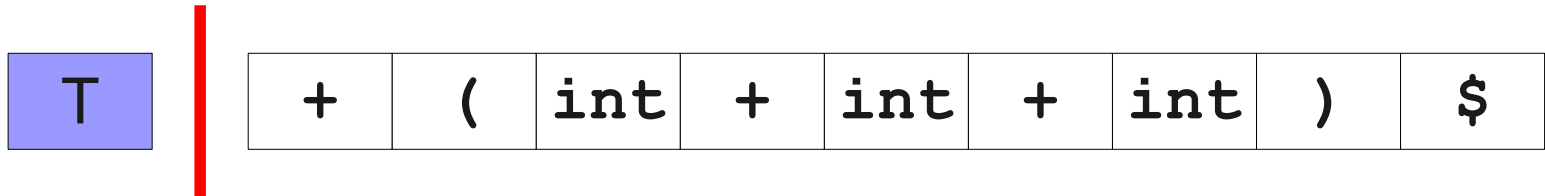
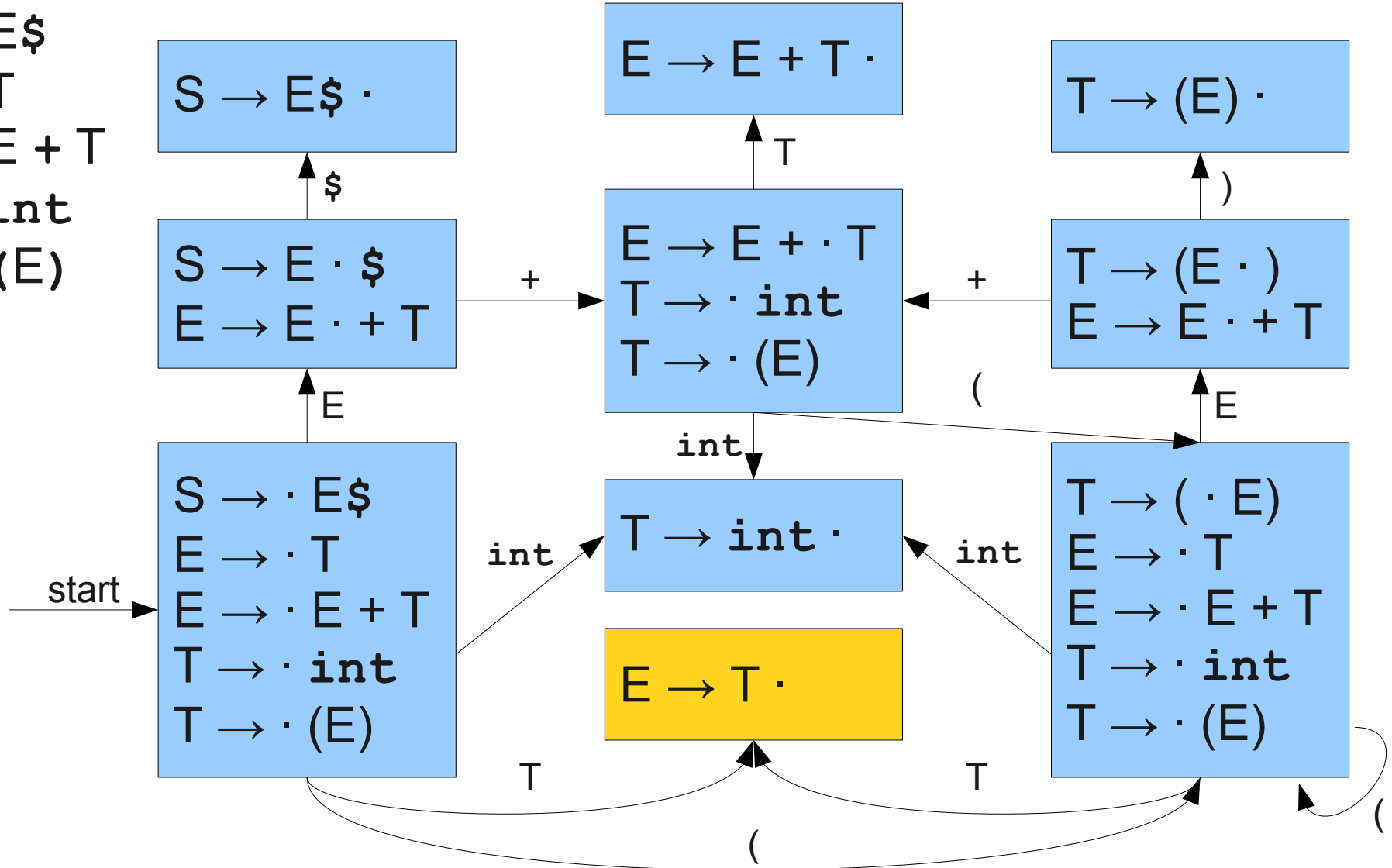
LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



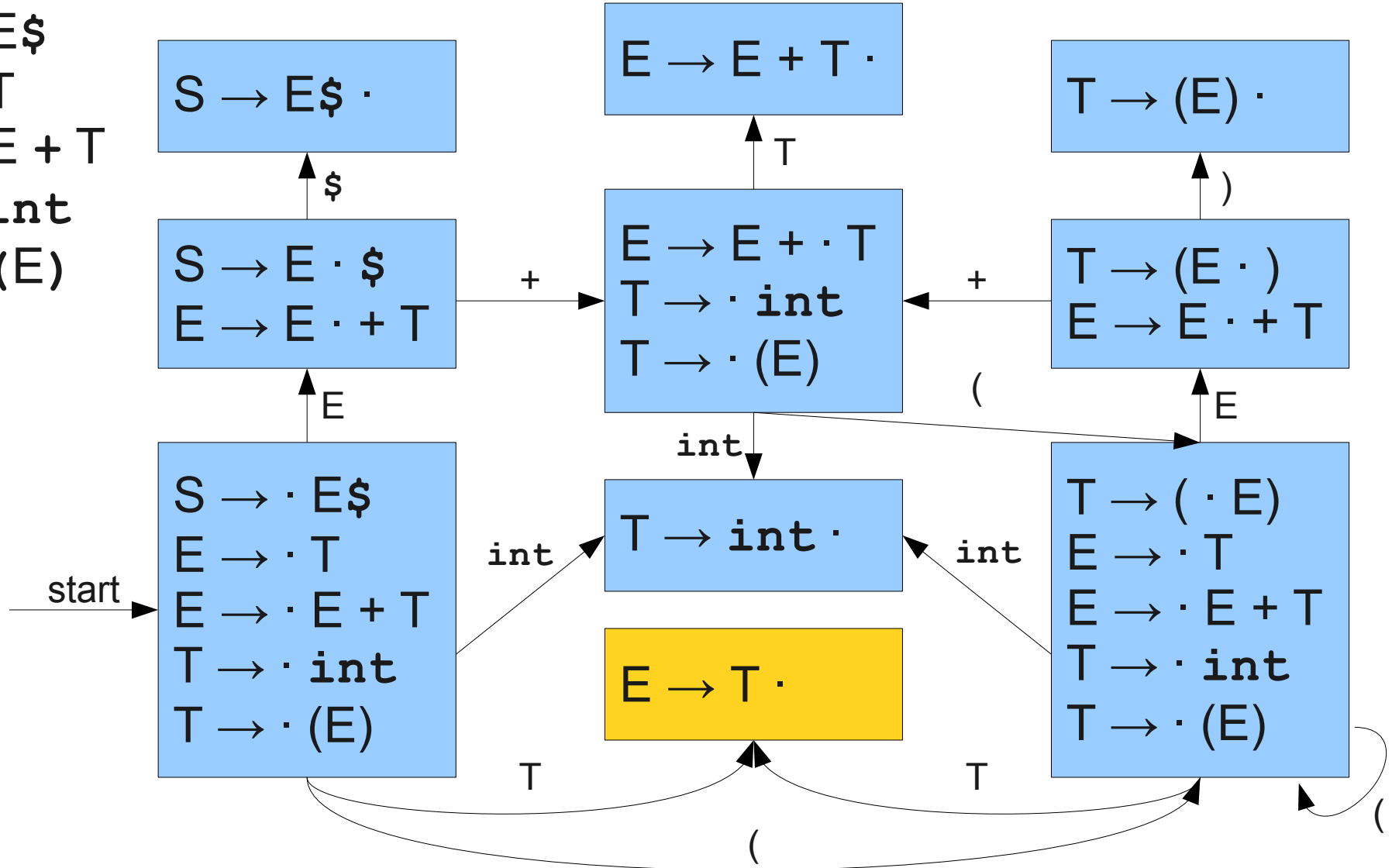
LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



LR(0) Parsing

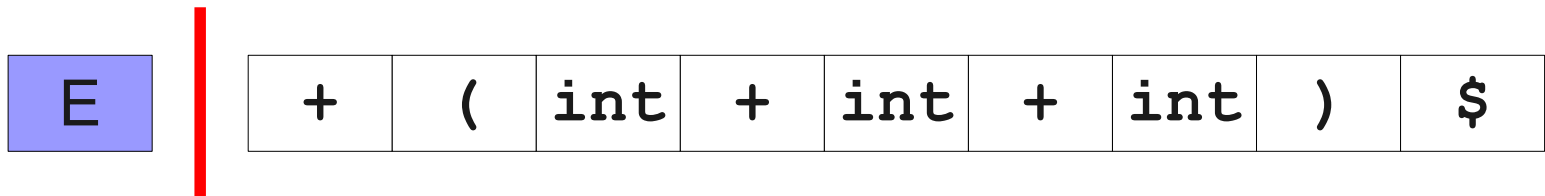
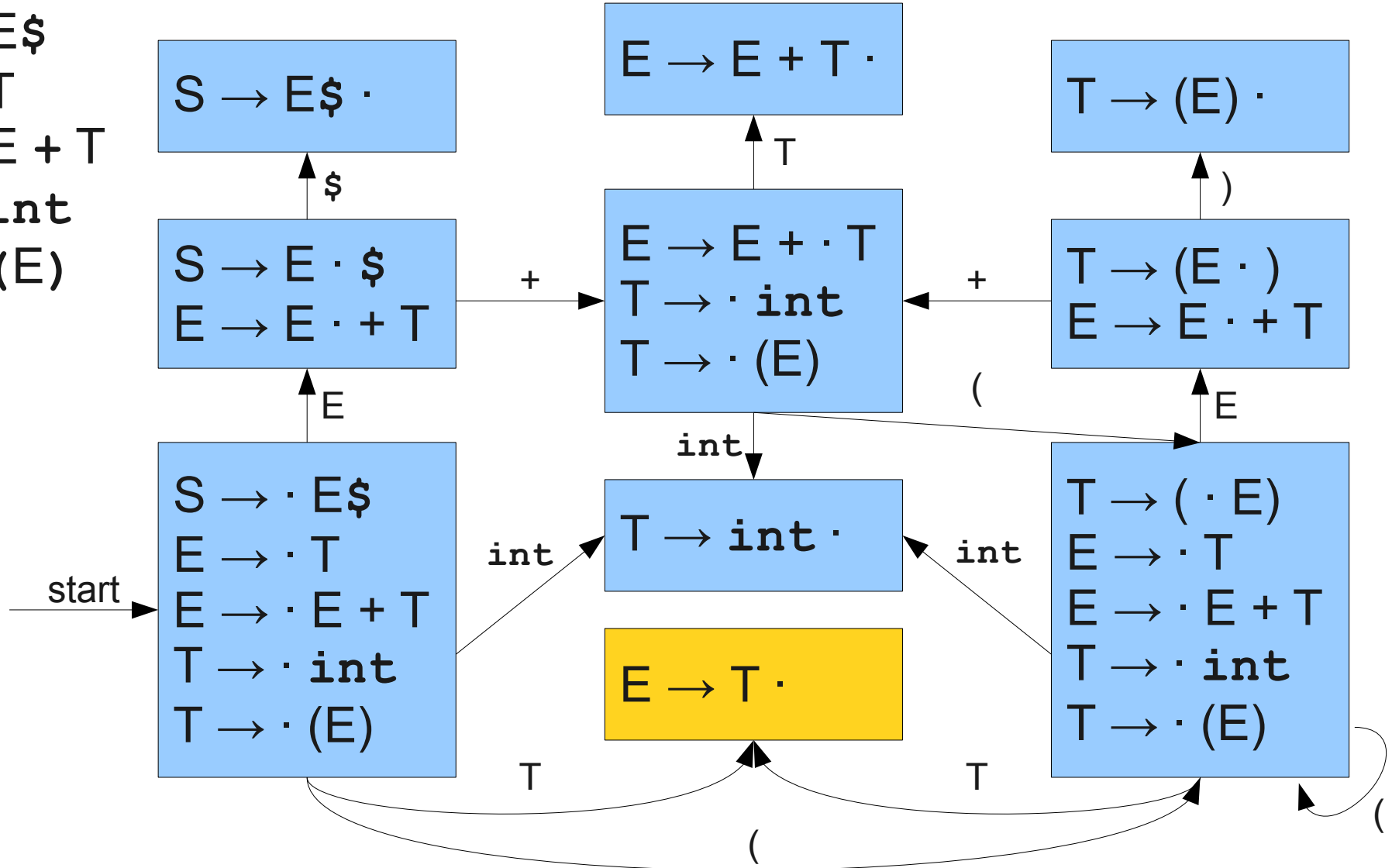
$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



		+	(int	+	int	+	int)	\$
--	--	---	---	-----	---	-----	---	-----	---	----

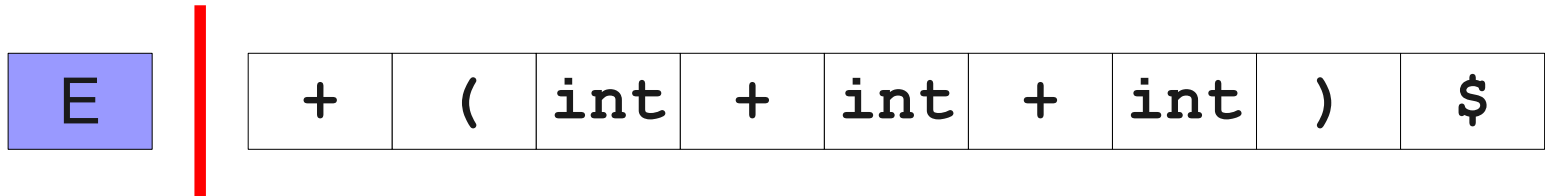
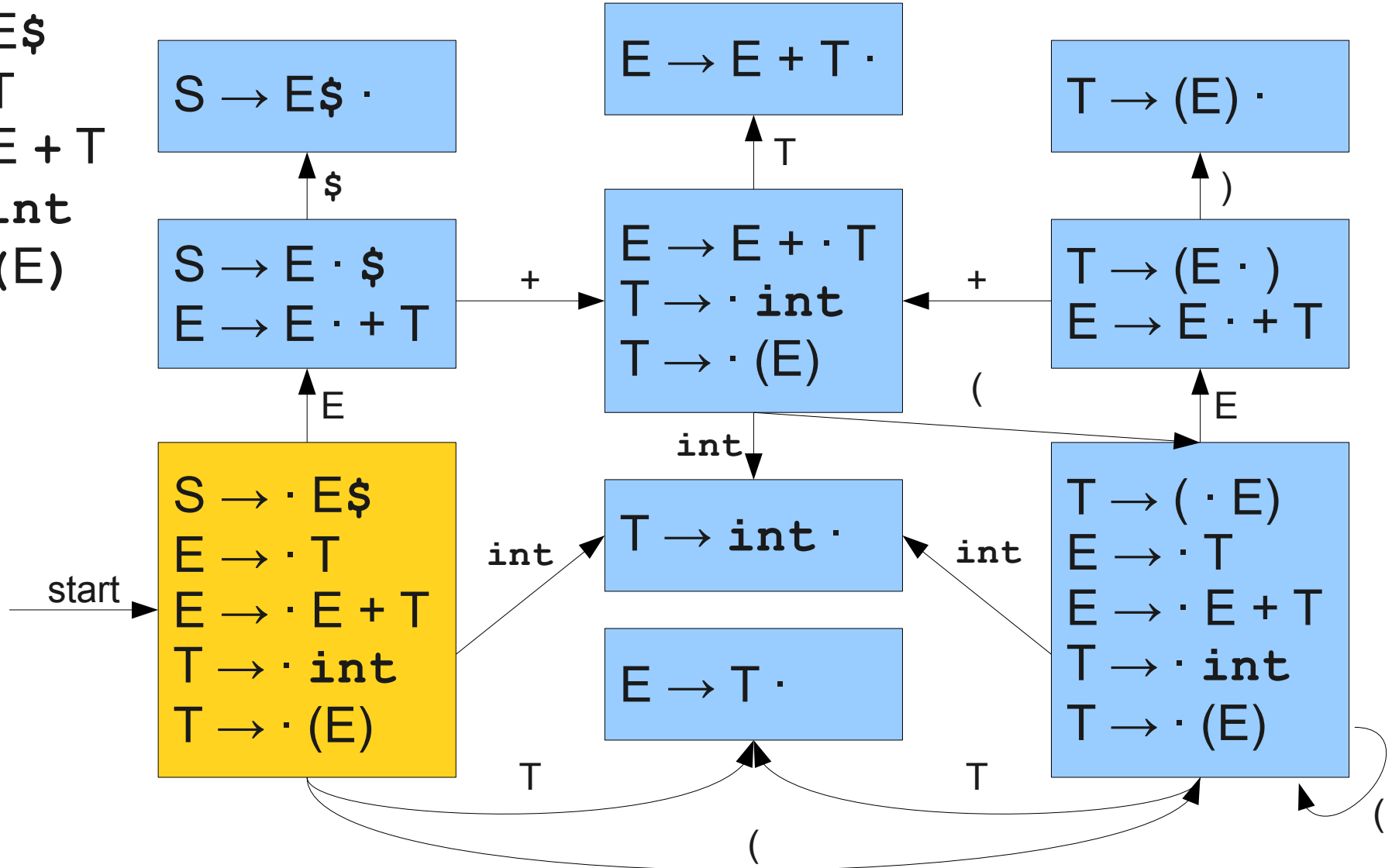
LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



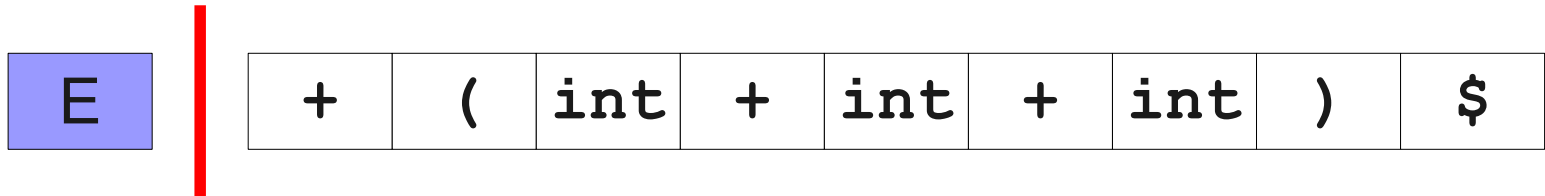
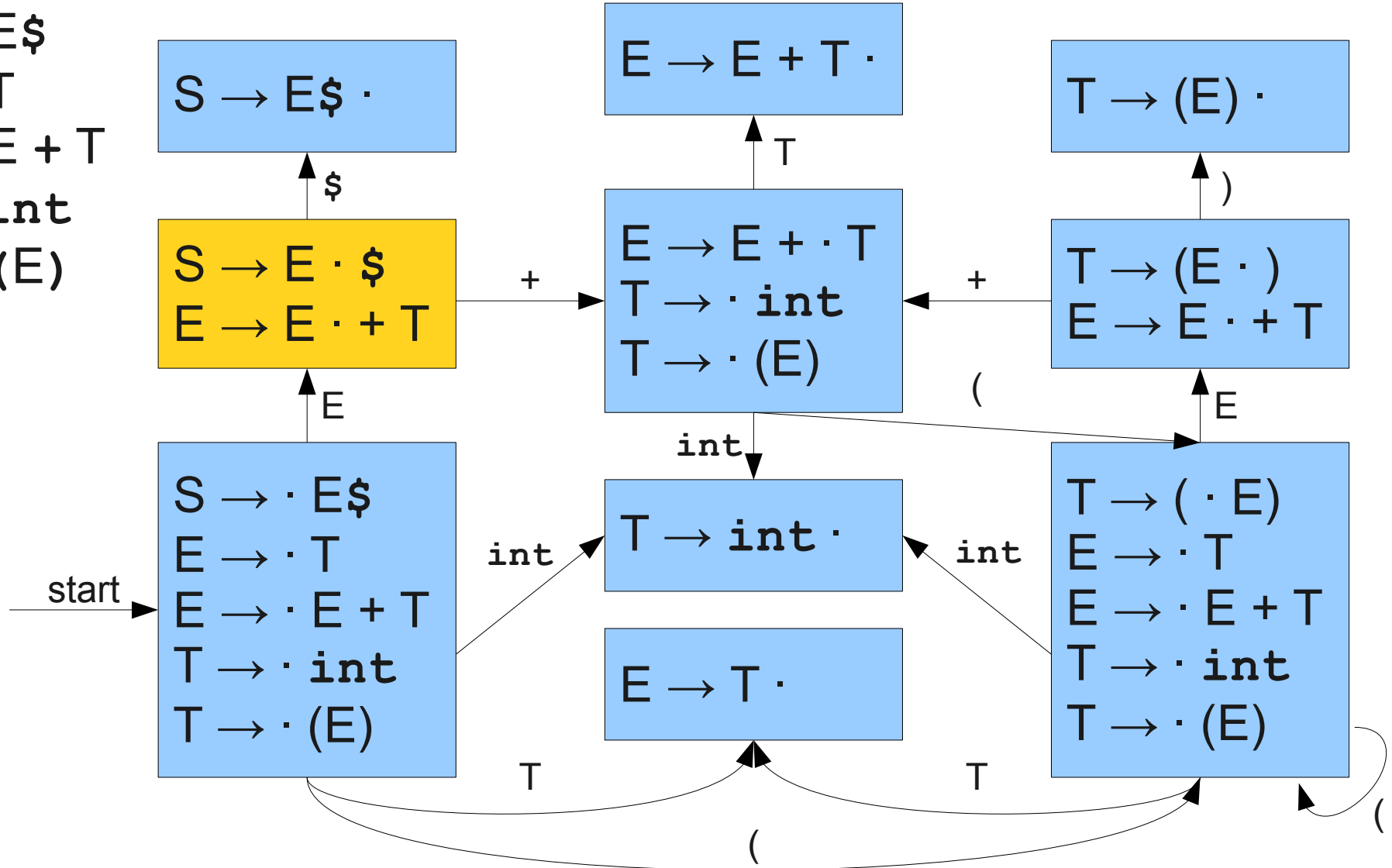
LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



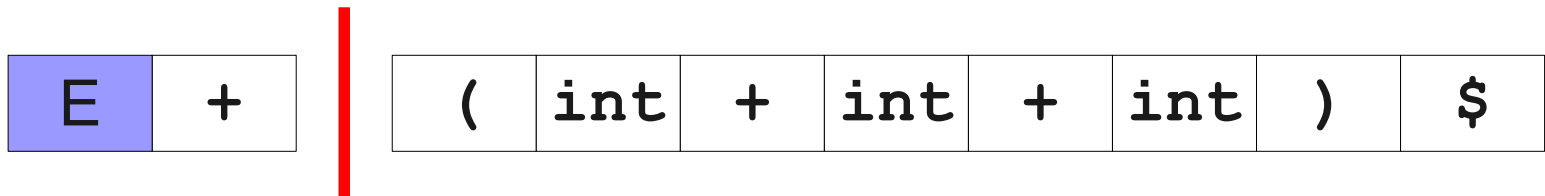
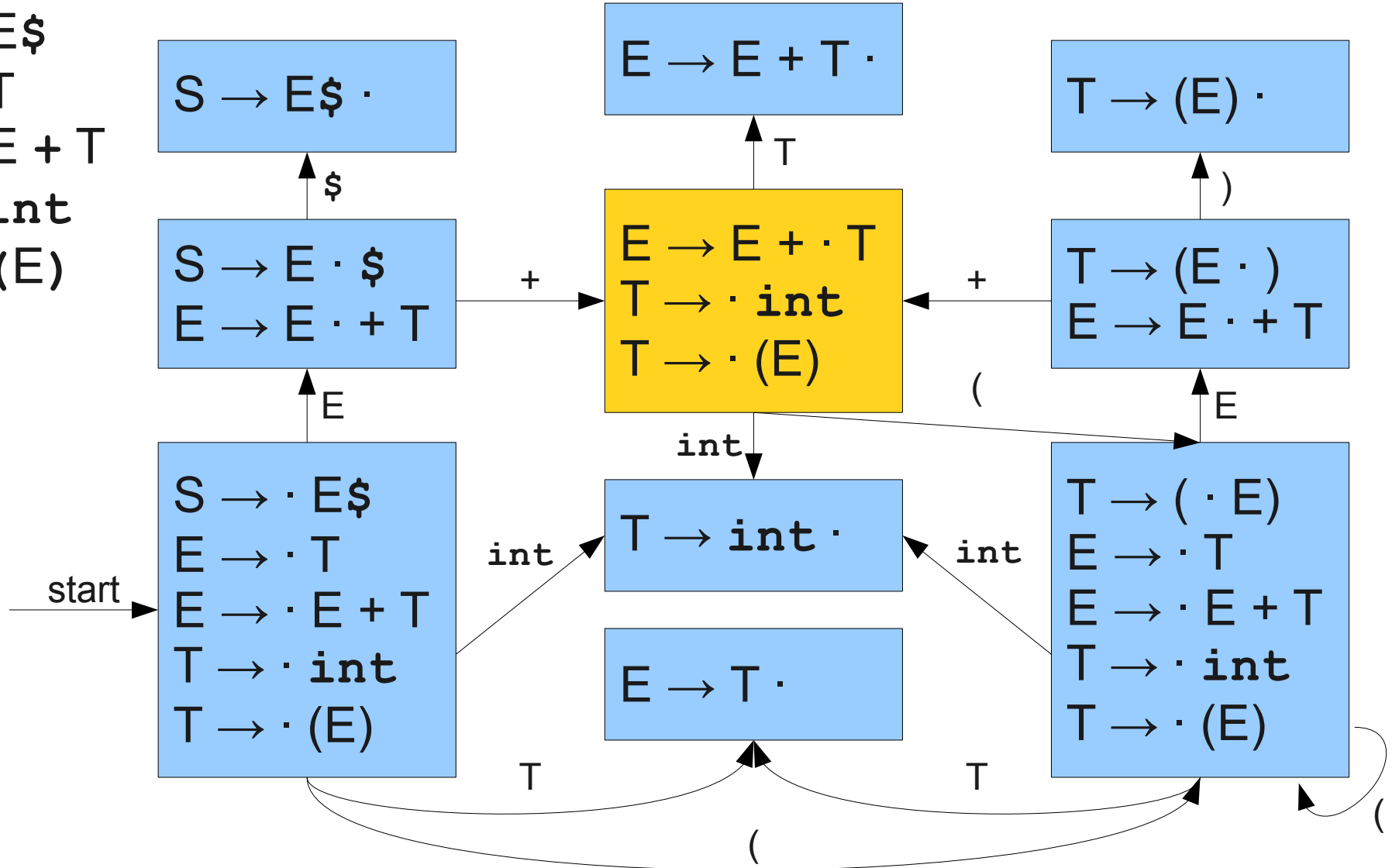
LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



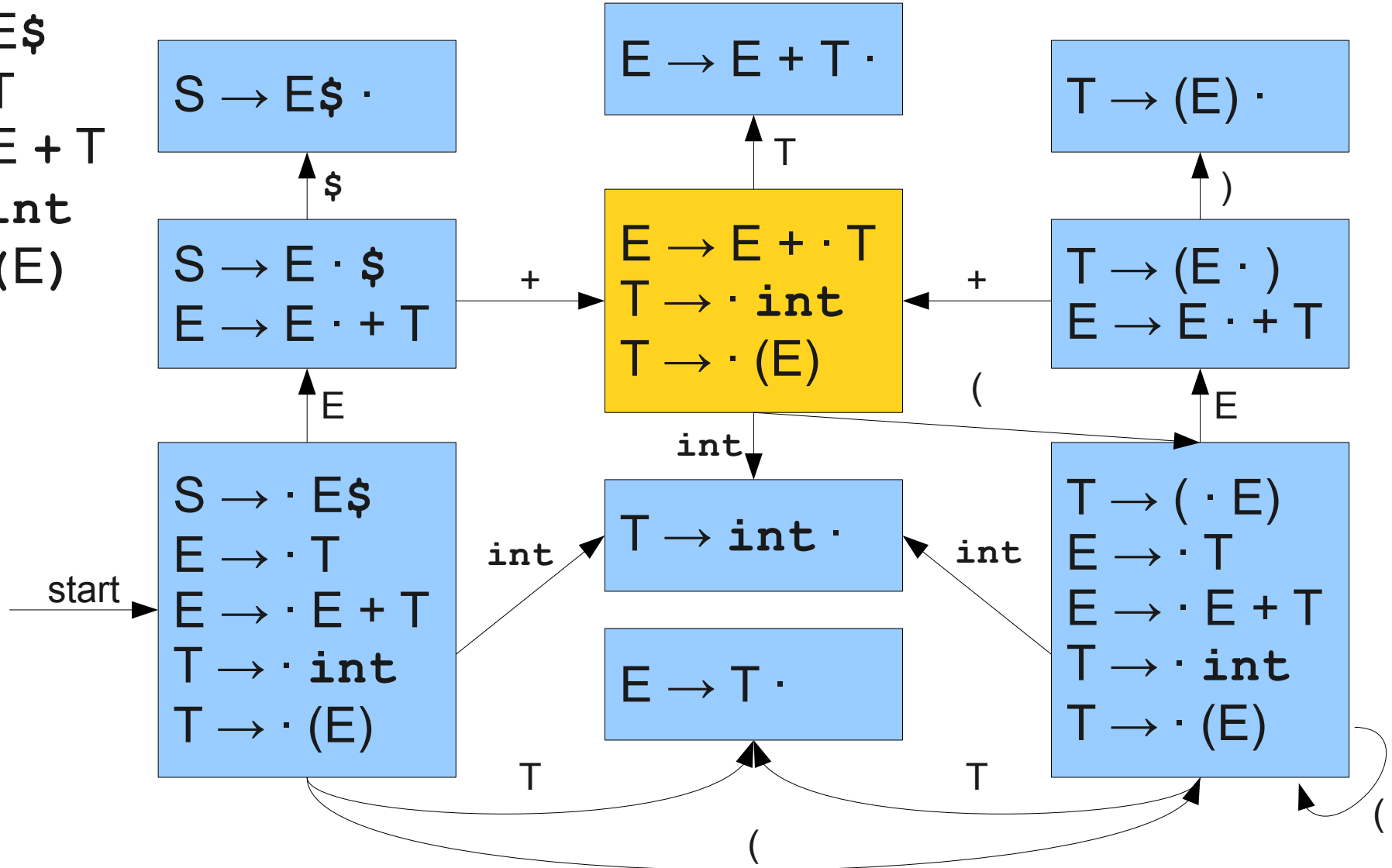
LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$

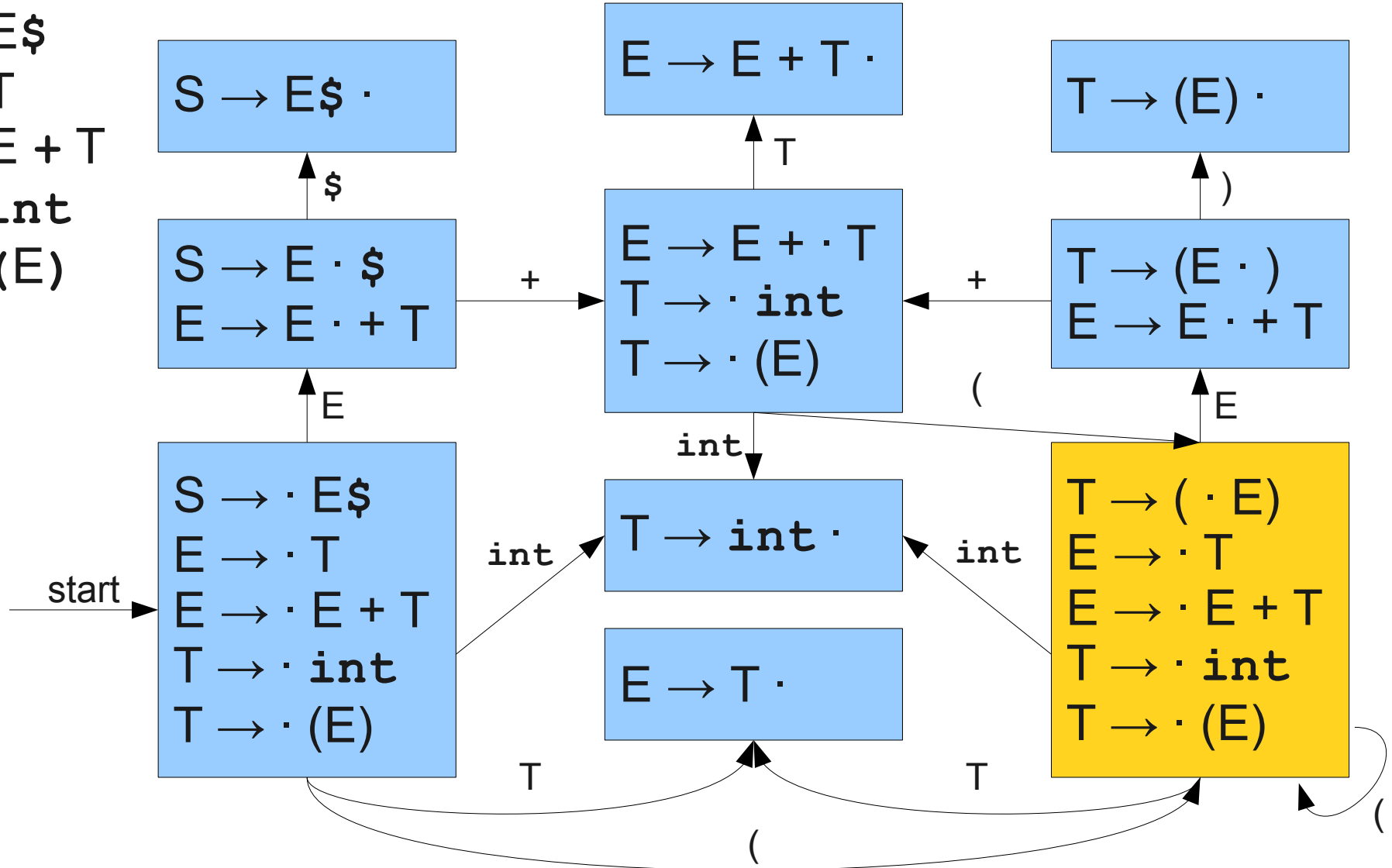


E	+	(
---	---	---

int	+	int	+	int)	\$
-----	---	-----	---	-----	---	----

LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$

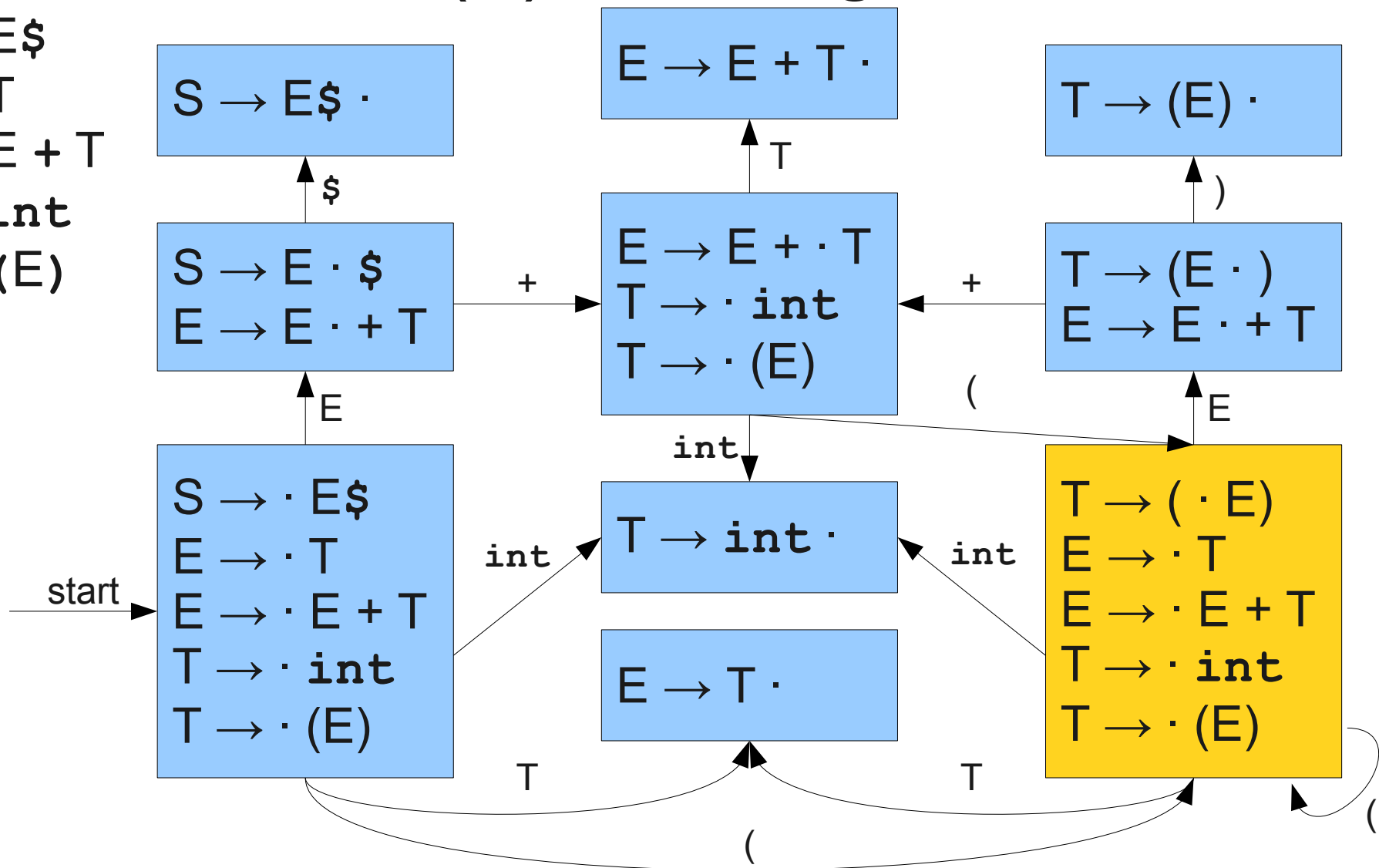


E	+	(
---	---	---

int	+	int	+	int)	\$
-----	---	-----	---	-----	---	----

LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$

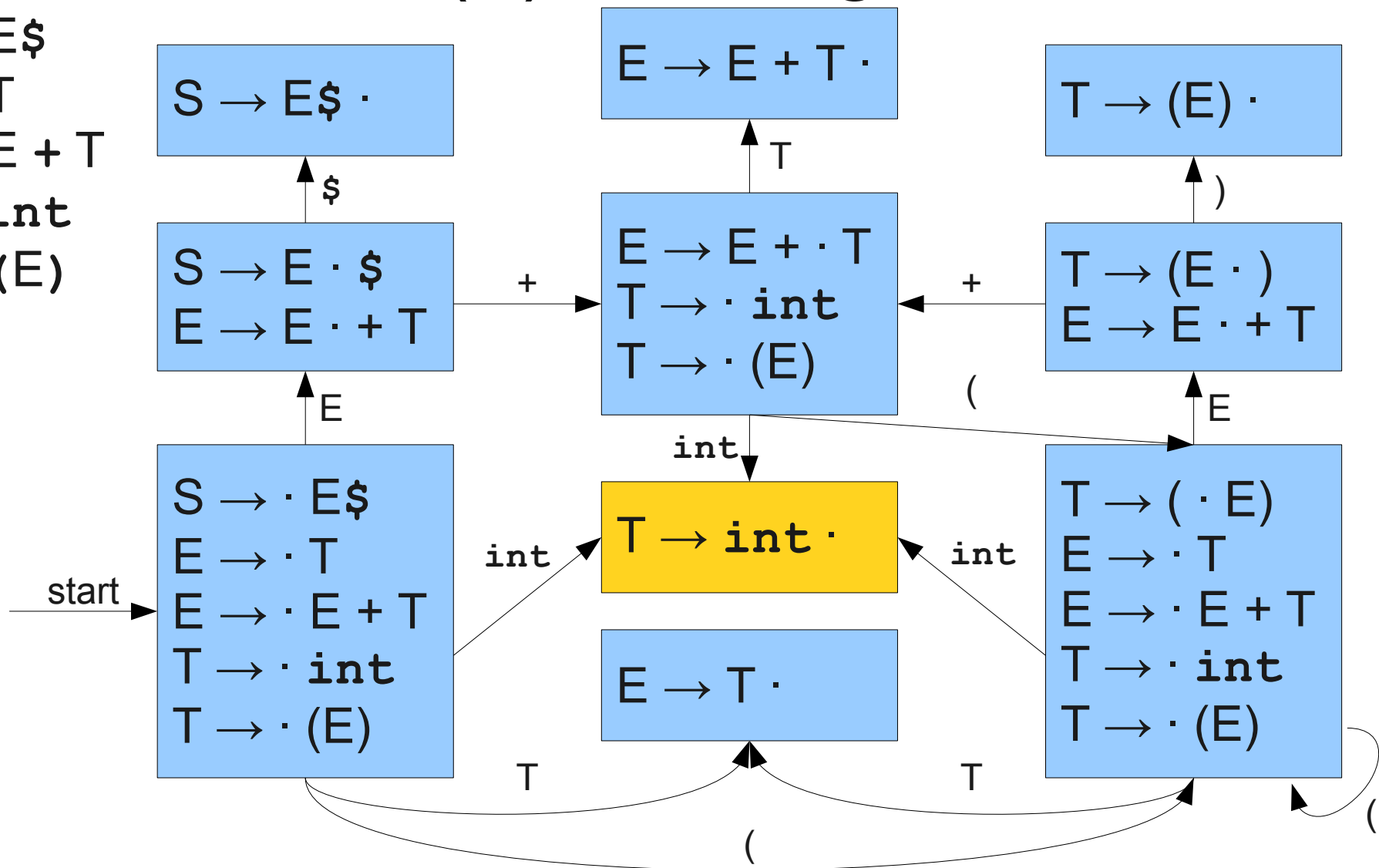


E	+	(int
---	---	---	-----

+	int	+	int)	\$
---	-----	---	-----	---	----

LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$

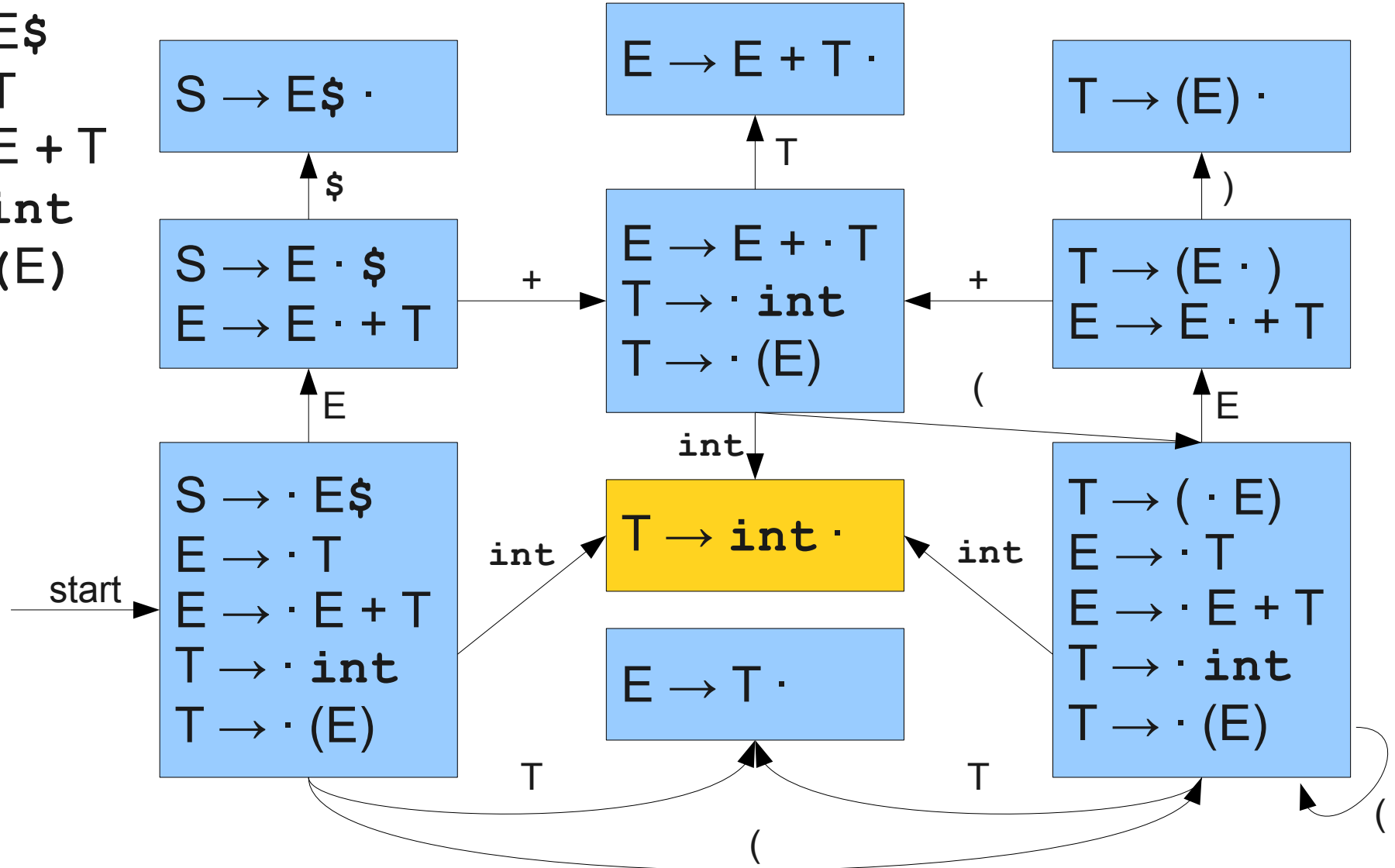


E + (int

+ int + int) \$

LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$

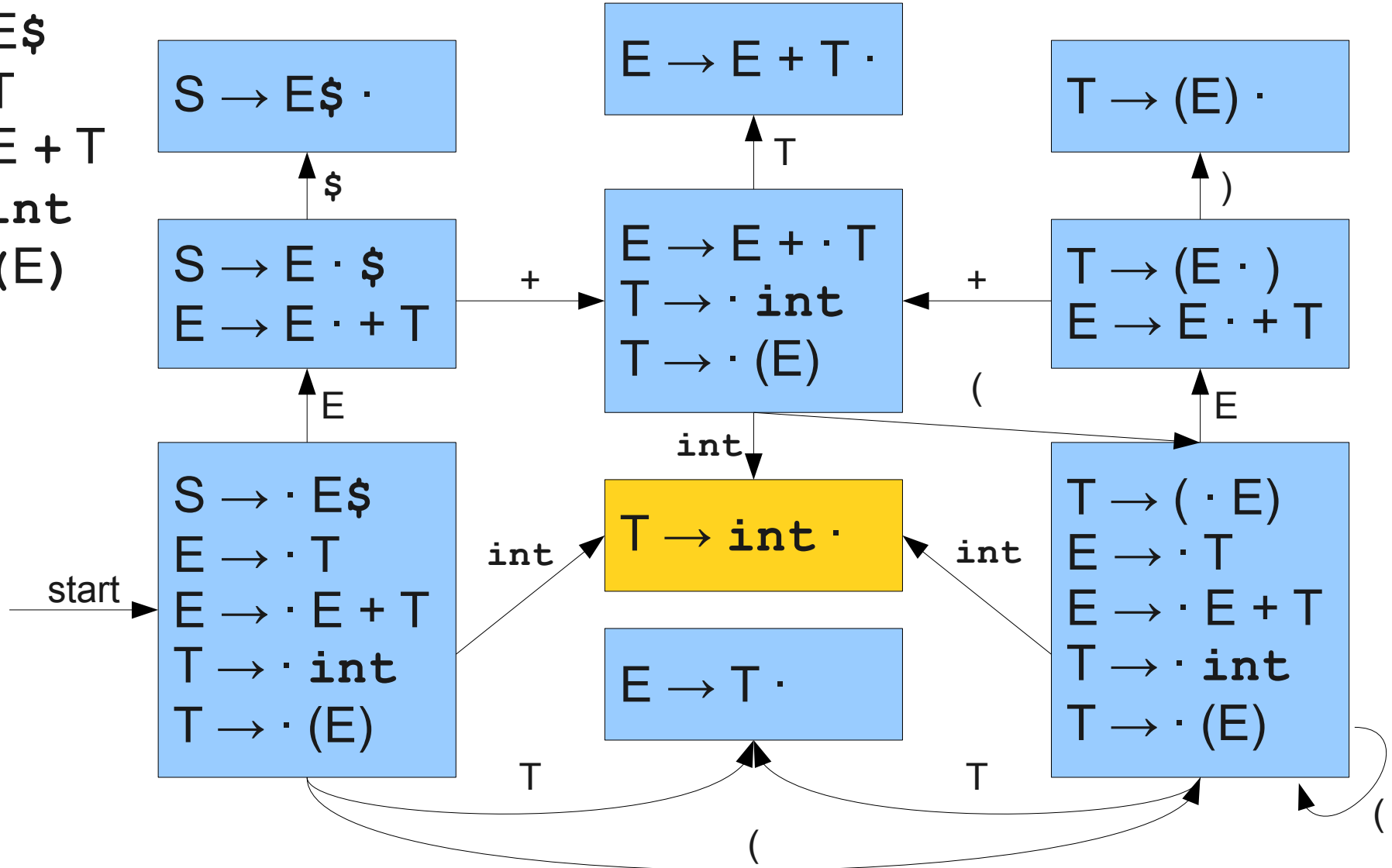


E	+	(
---	---	---

+	int	+	int)	\$
---	-----	---	-----	---	----

LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$

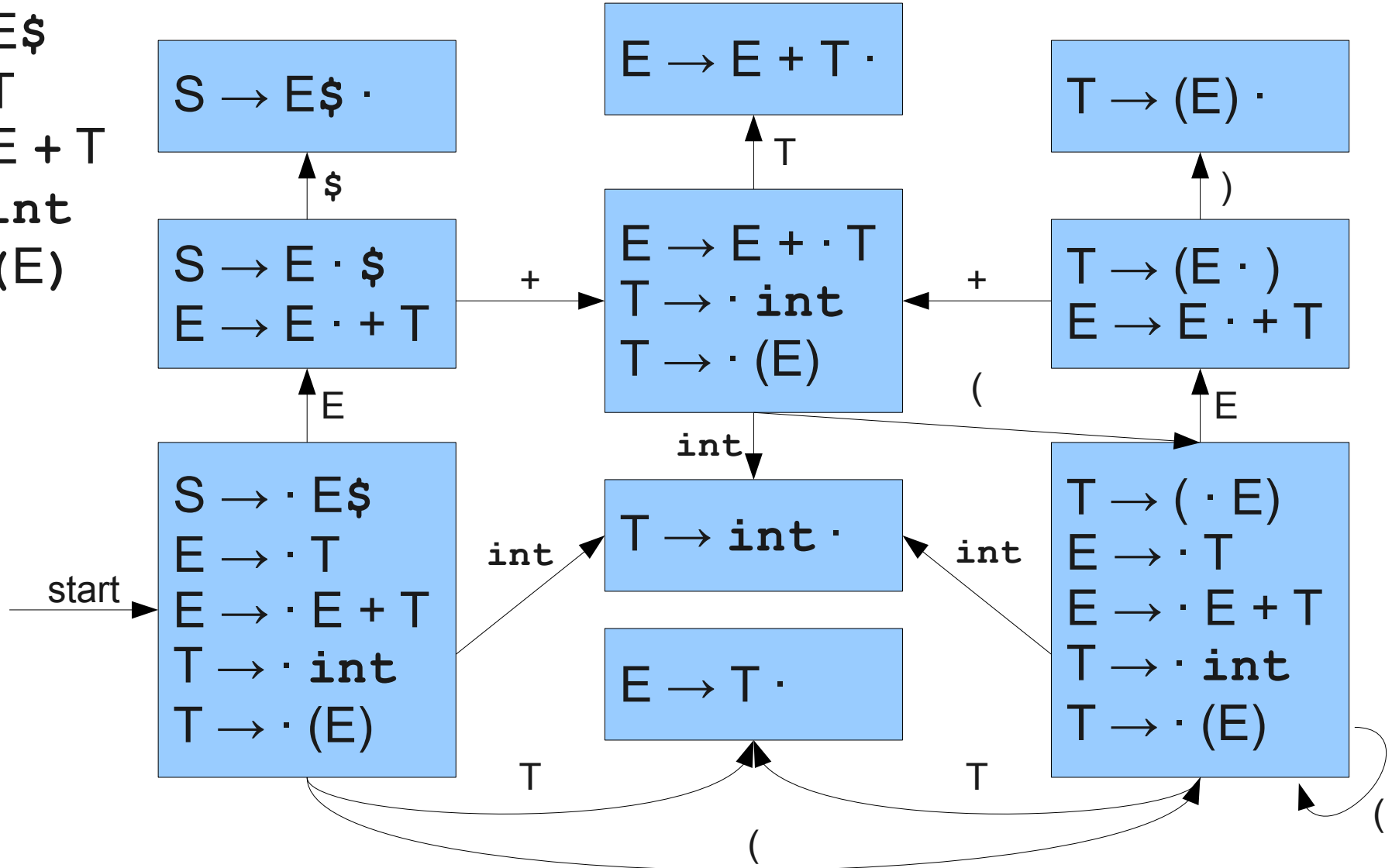


E	+	(T
---	---	---	---

+	int	+	int)	\$
---	-----	---	-----	---	----

LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$

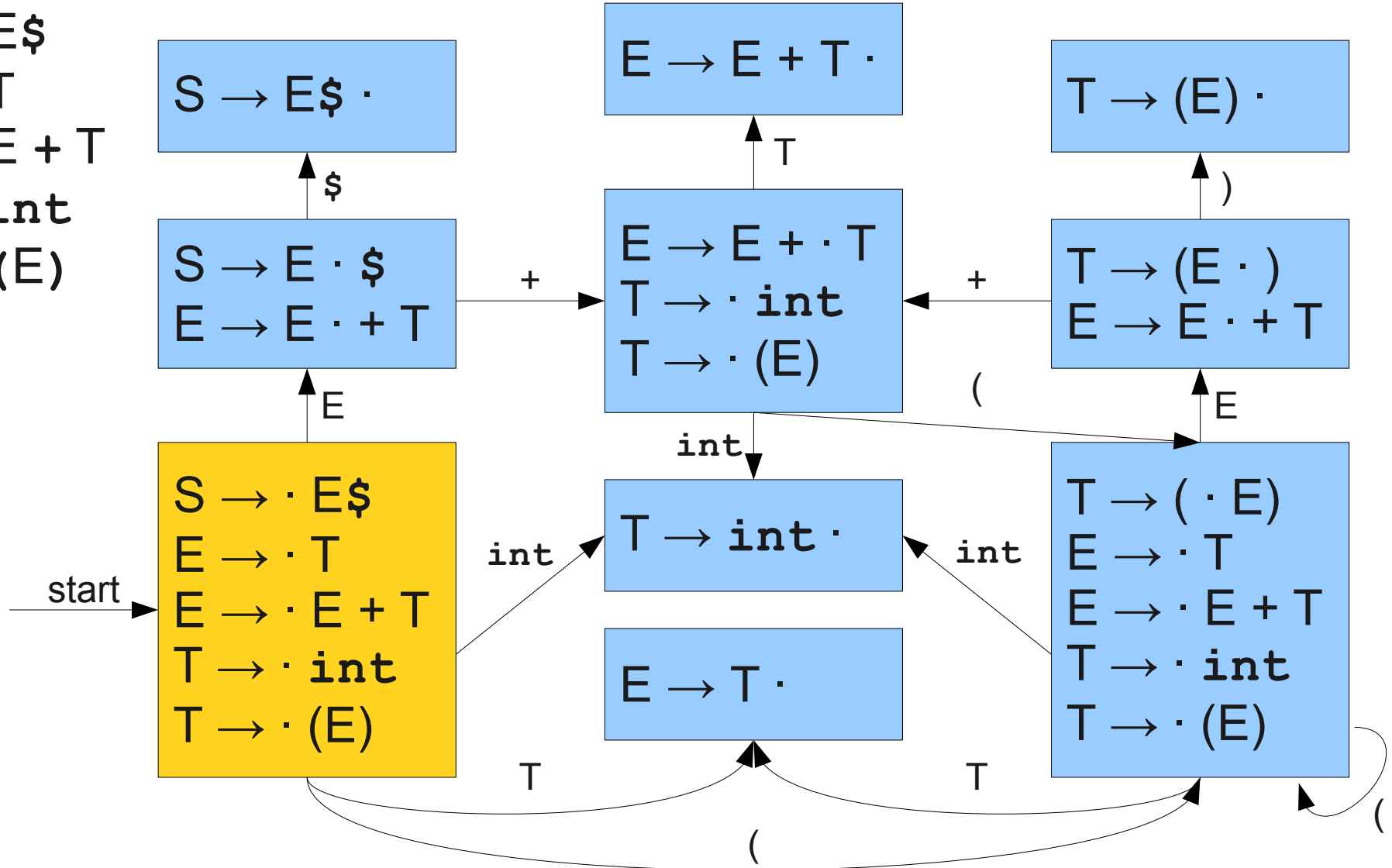


E	+	(T
---	---	---	---

+	int	+	int)	\$
---	-----	---	-----	---	----

LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$

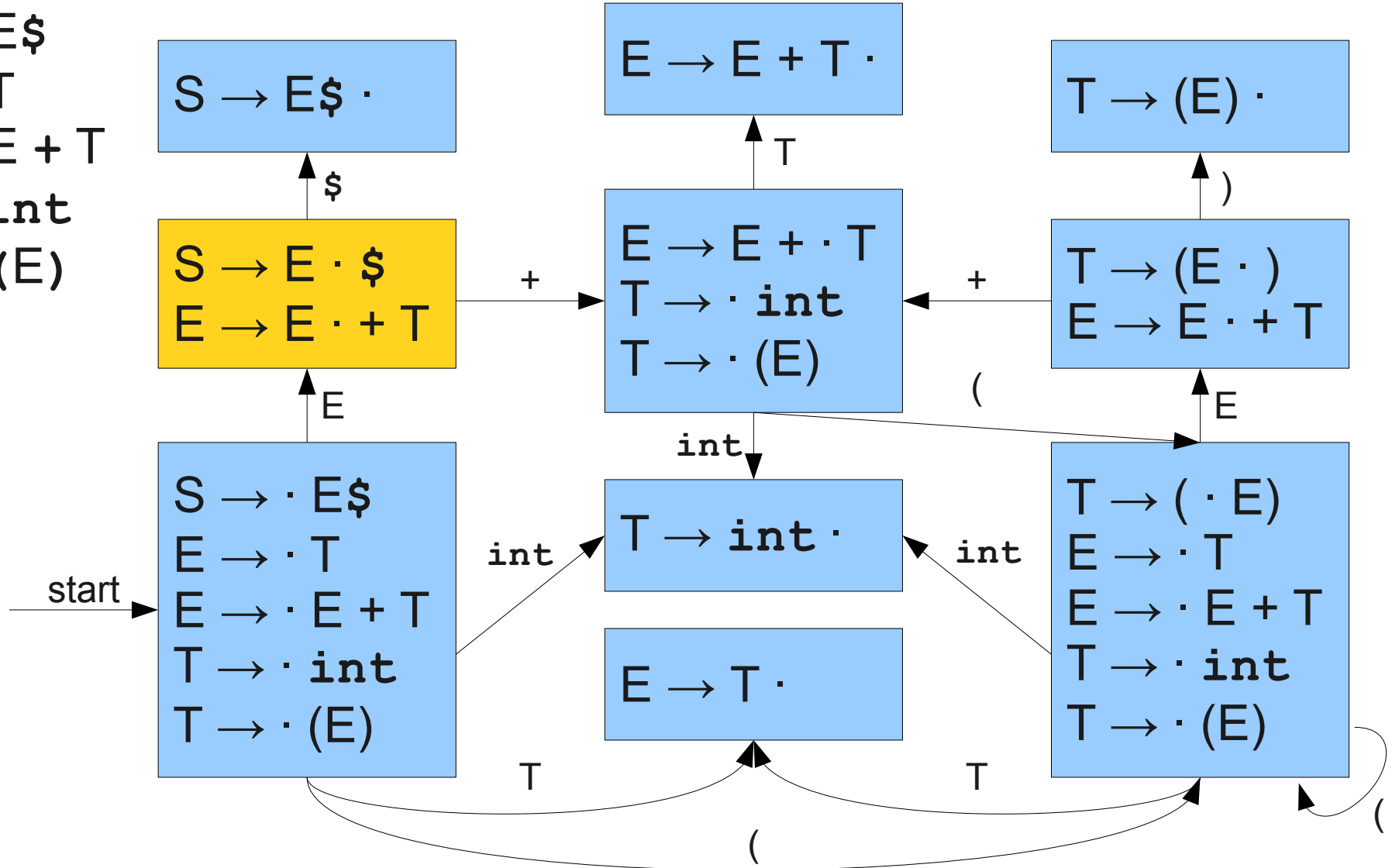


E + (T

+ int + int) \$

LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$

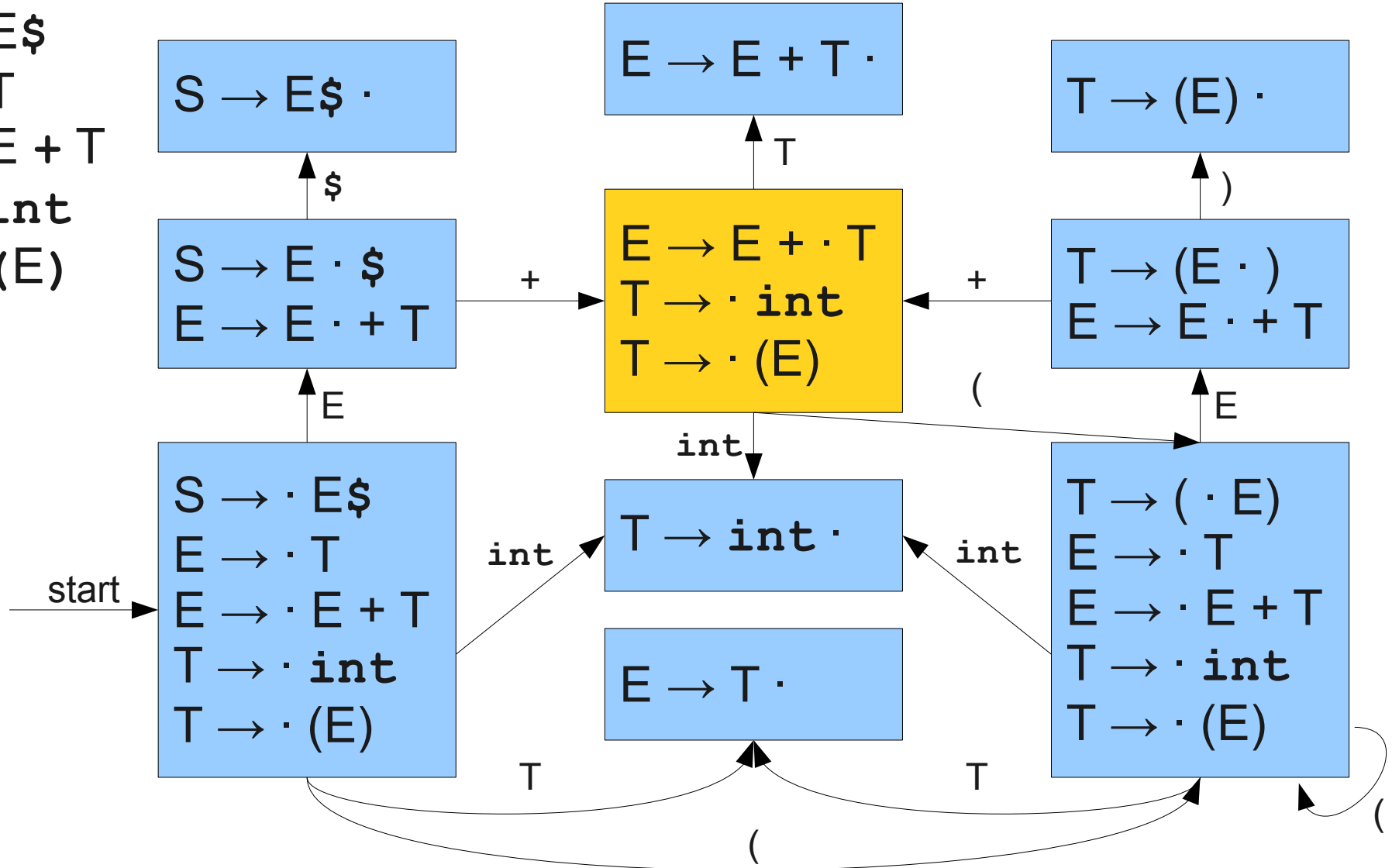


E + (T

+ int + int) \$

LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$

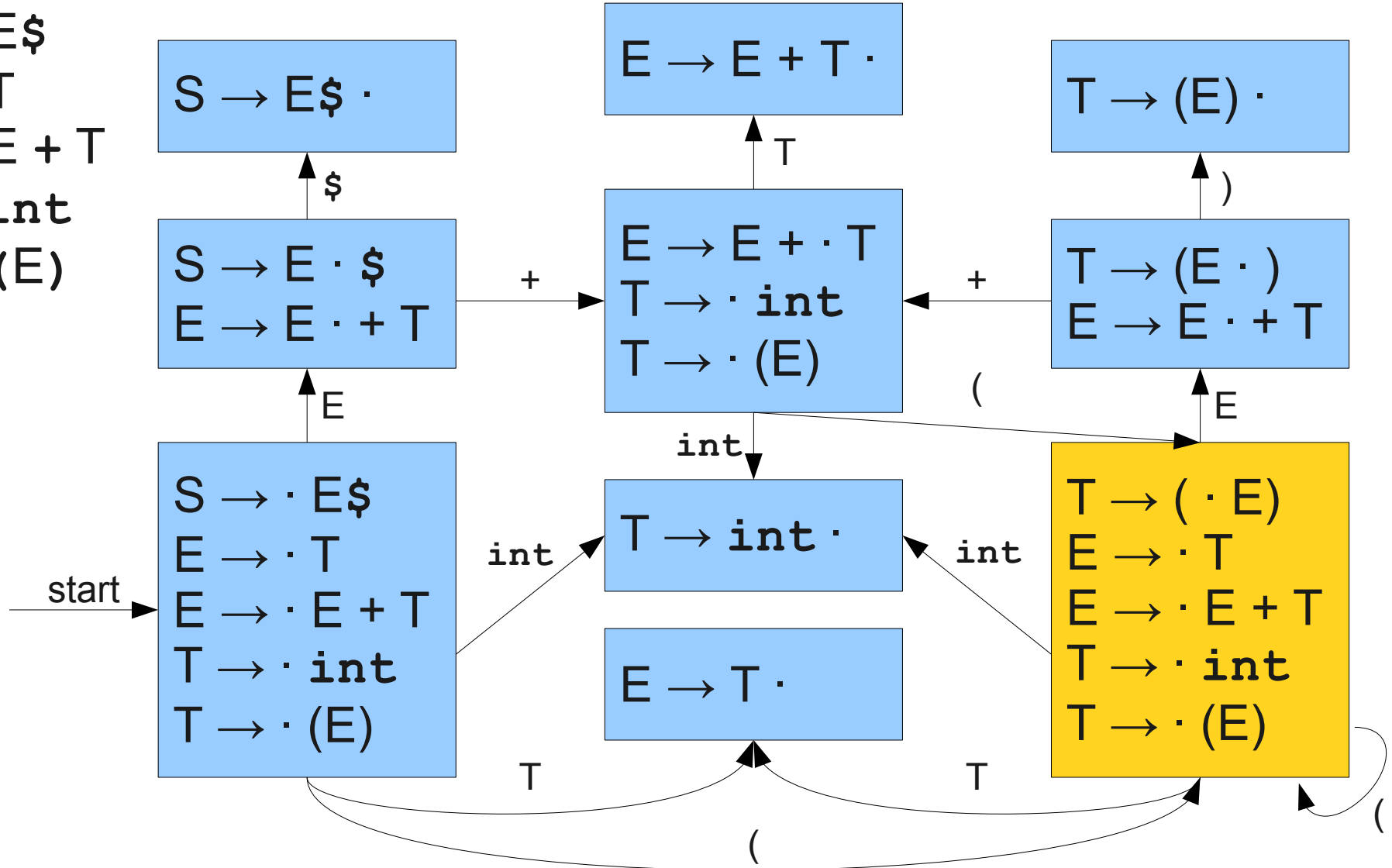


E	+	(T
---	---	---	---

+	int	+	int)	\$
---	-----	---	-----	---	----

LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$

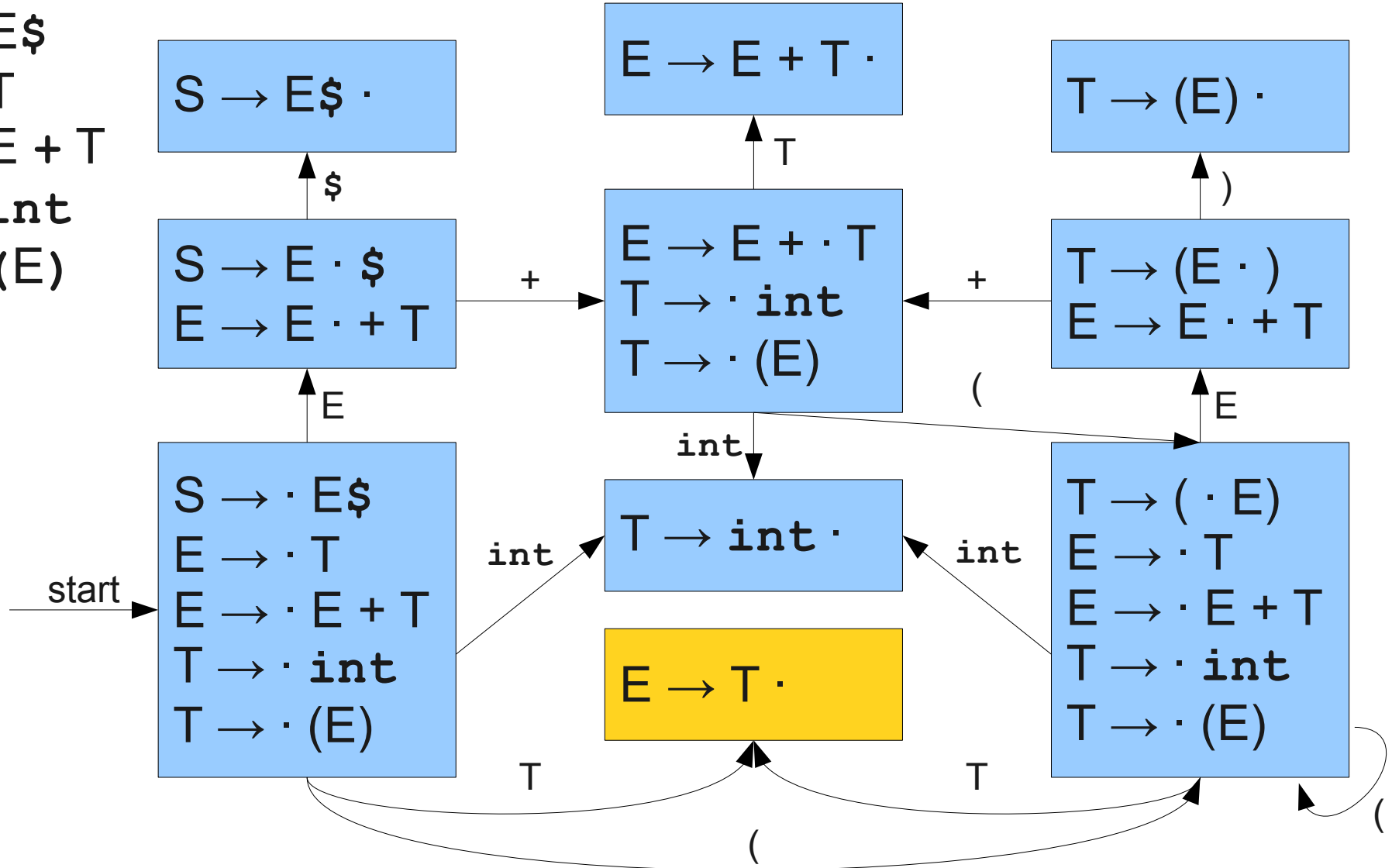


E + (T

+ int + int) \$

LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$

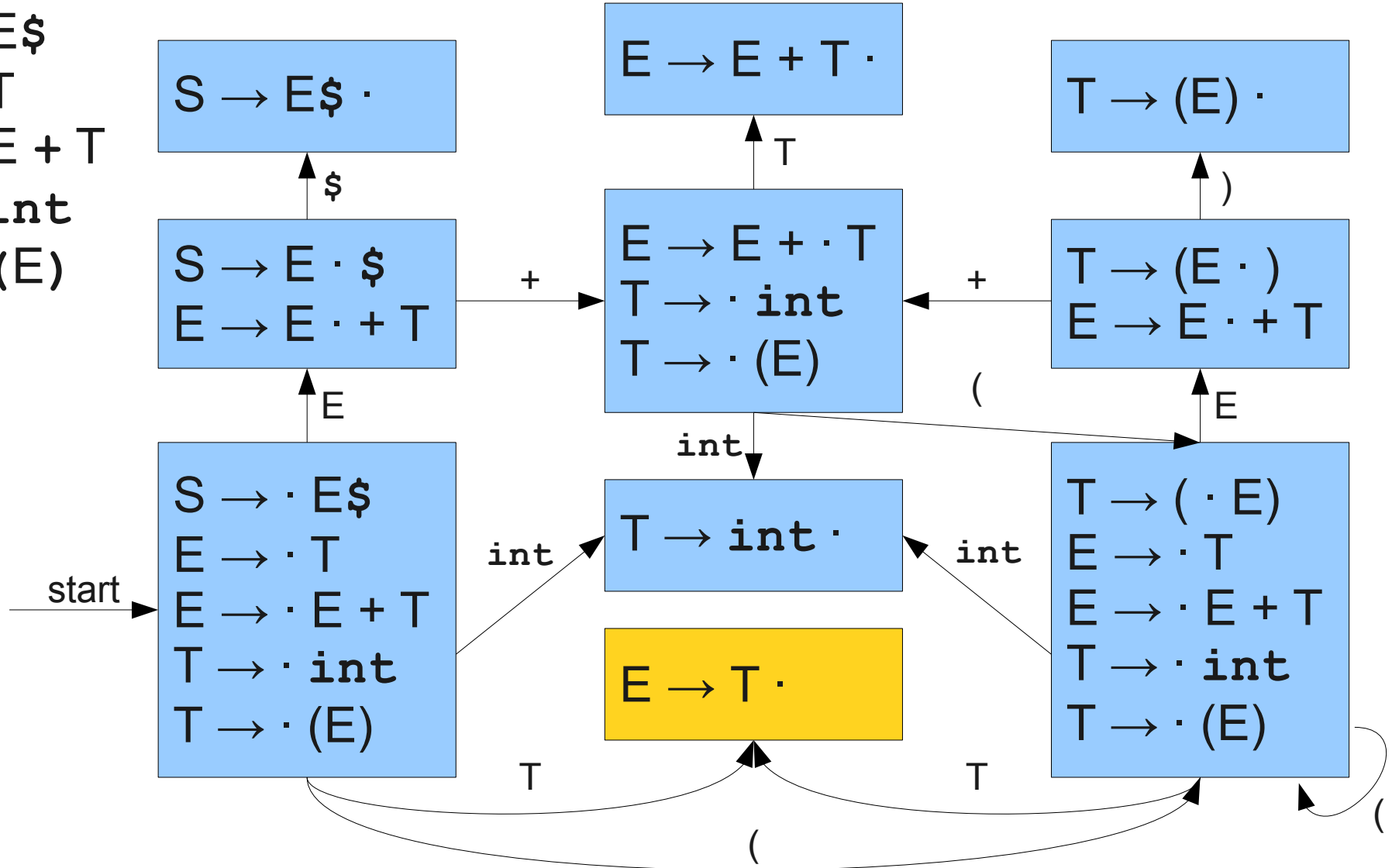


E + (T

+ int + int) \$

LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$

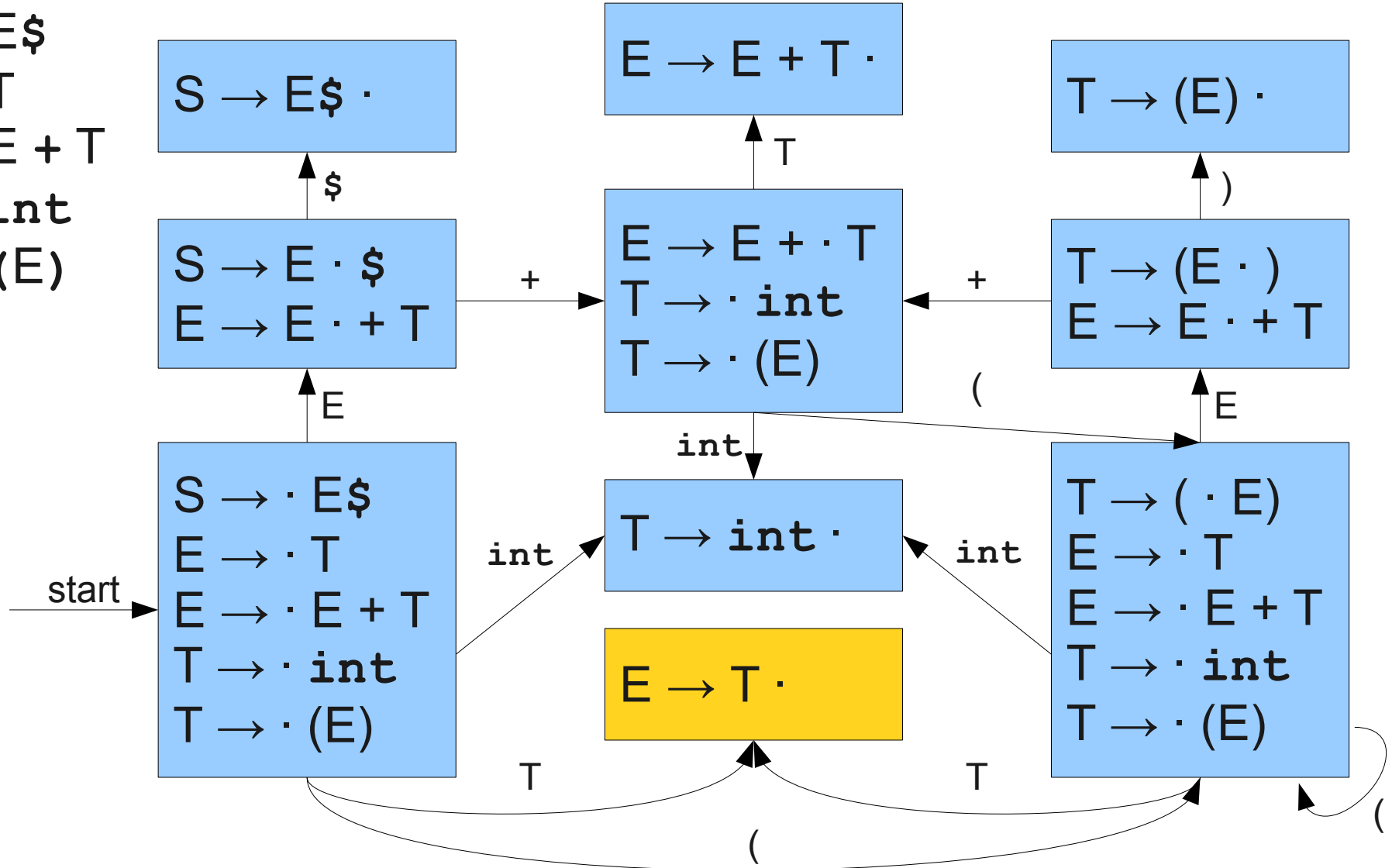


E	+	(
---	---	---

+	int	+	int)	\$
---	-----	---	-----	---	----

LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$

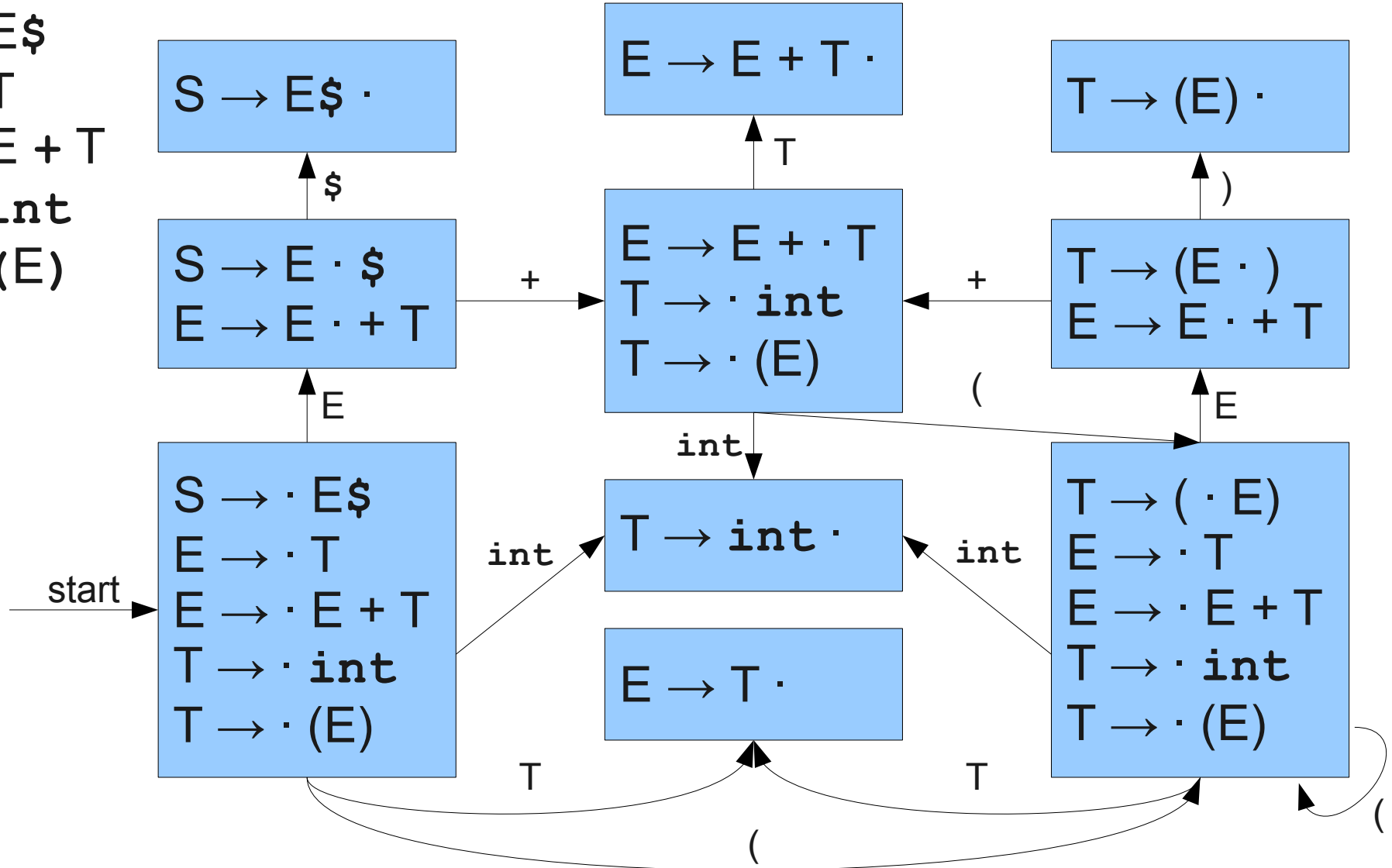


E + (E

+ int + int) \$

LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$

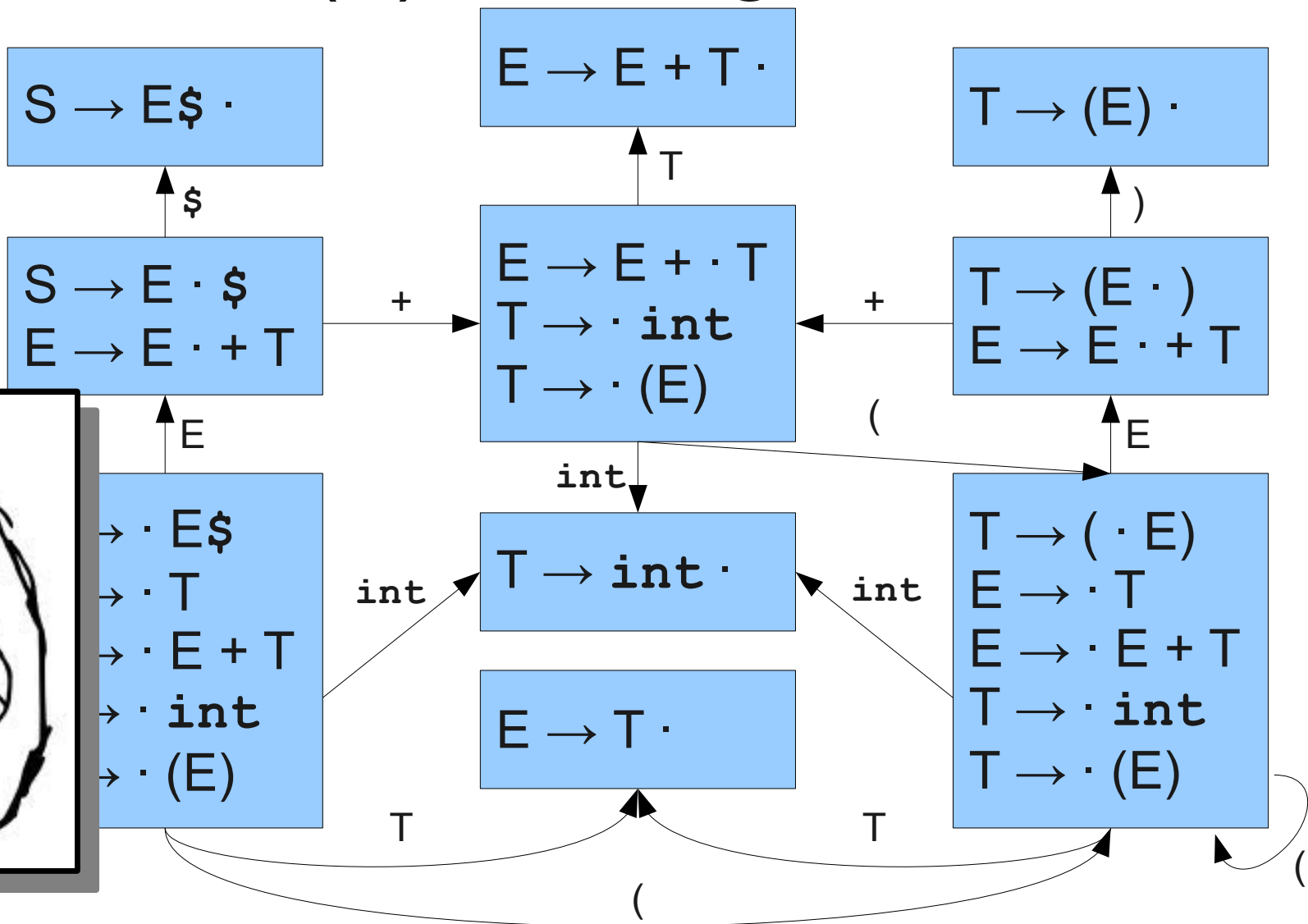


E + (**E**

+ int + int) \$

LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



E	+	(E
---	---	---	---

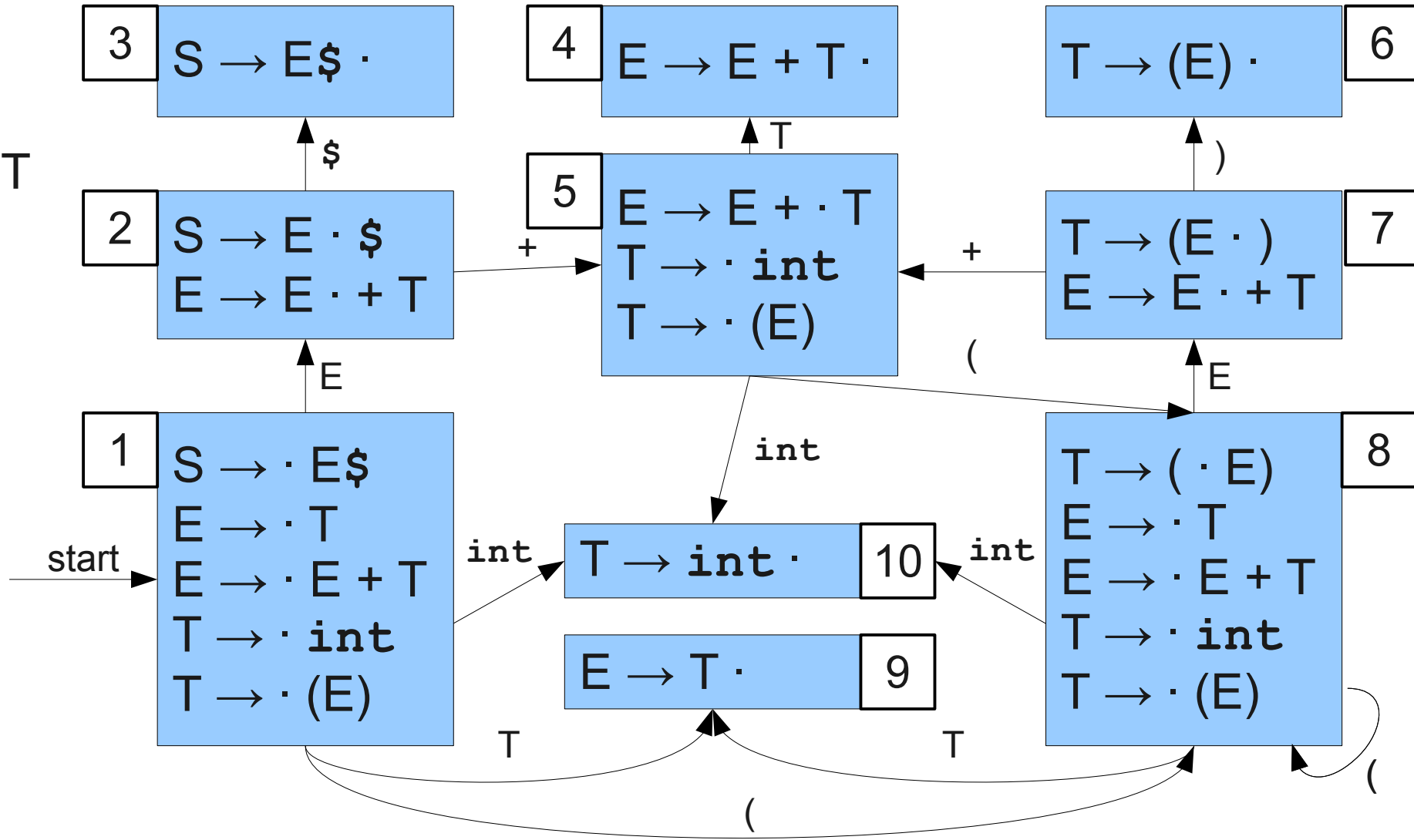
+	int	+	int)	\$
---	-----	---	-----	---	----

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

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$

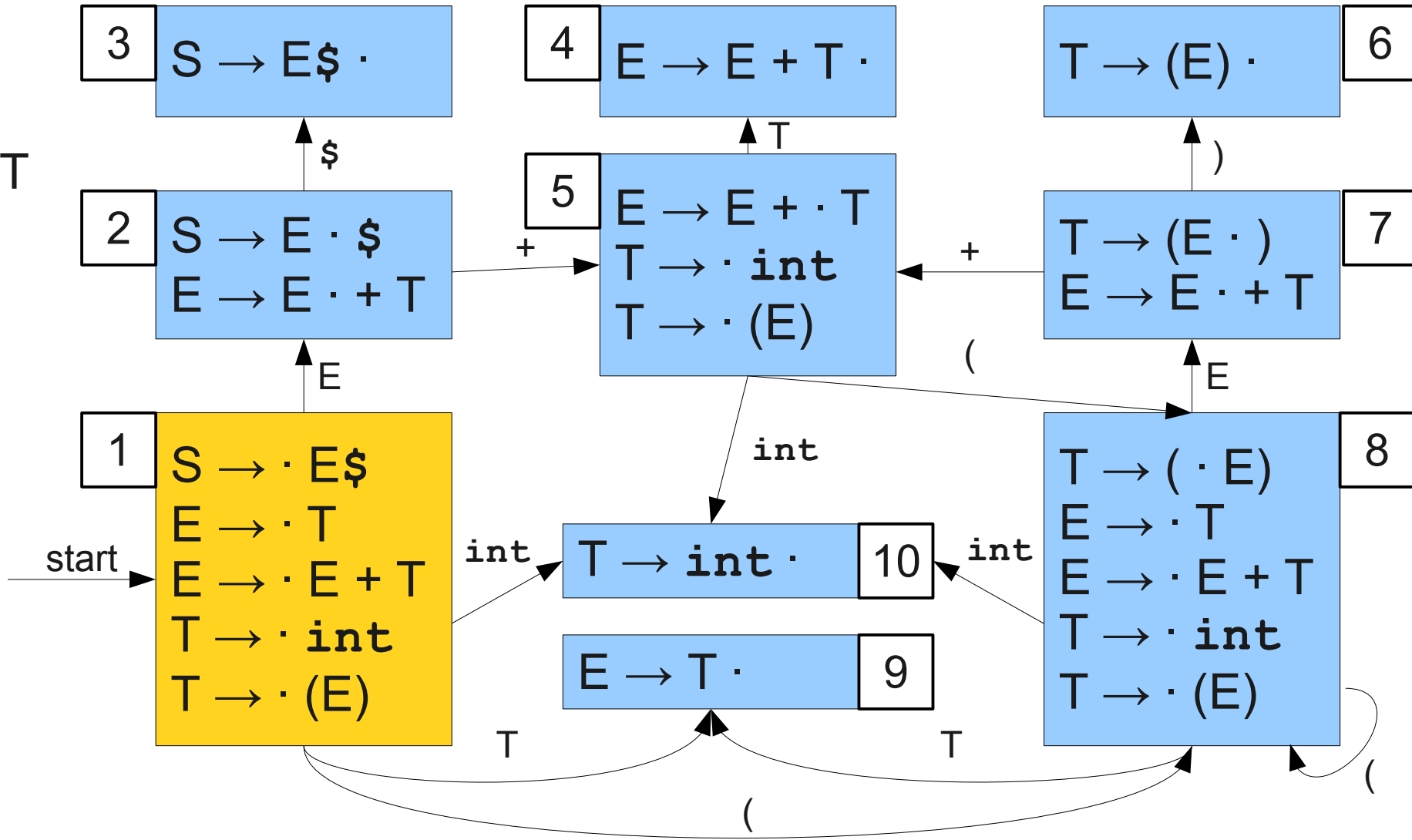


int	+	(int	+	int	+	int)	\$
-----	---	---	-----	---	-----	---	-----	---	----

1

LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$

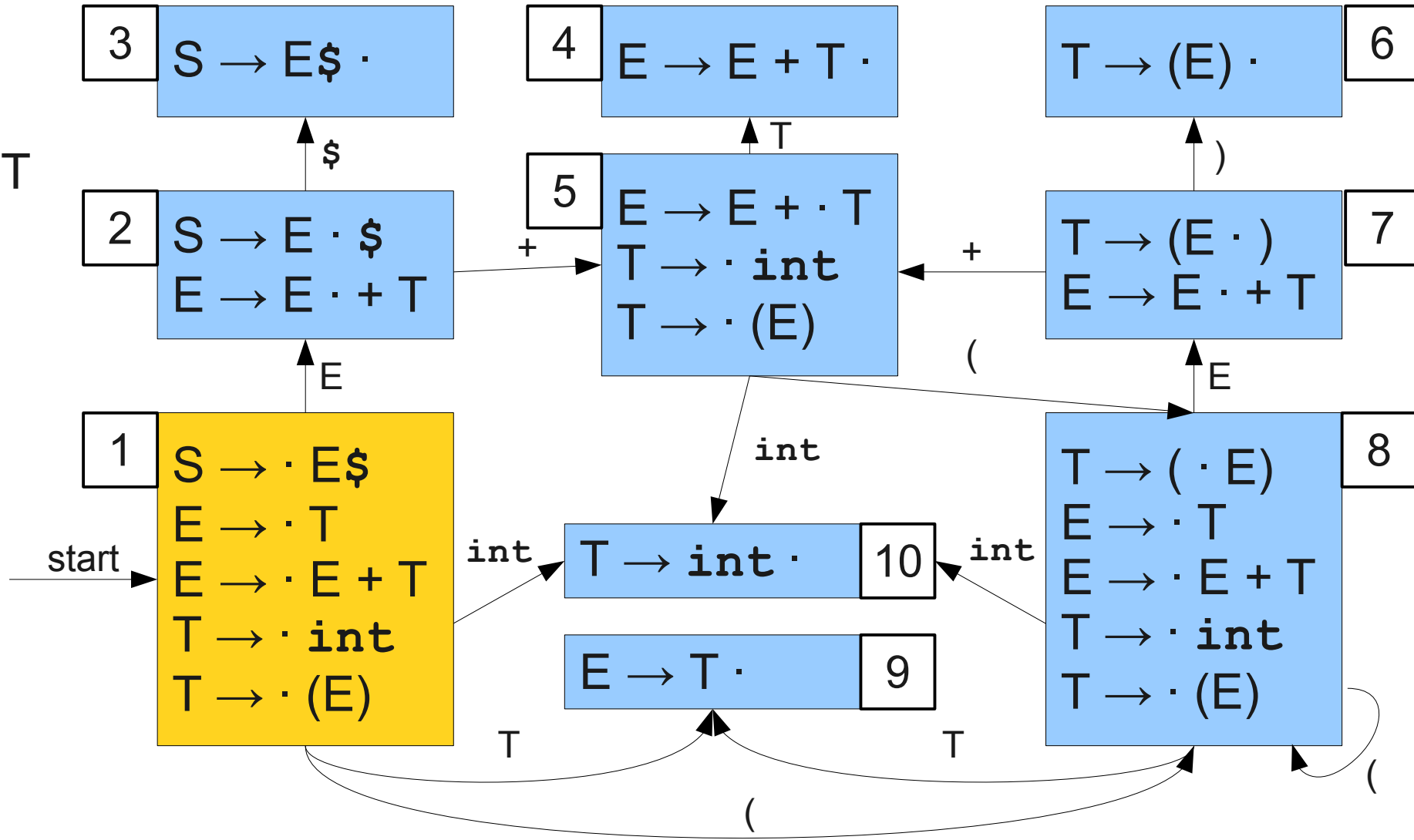


int	+	(int	+	int	+	int)	\$
-----	---	---	-----	---	-----	---	-----	---	----

1

LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



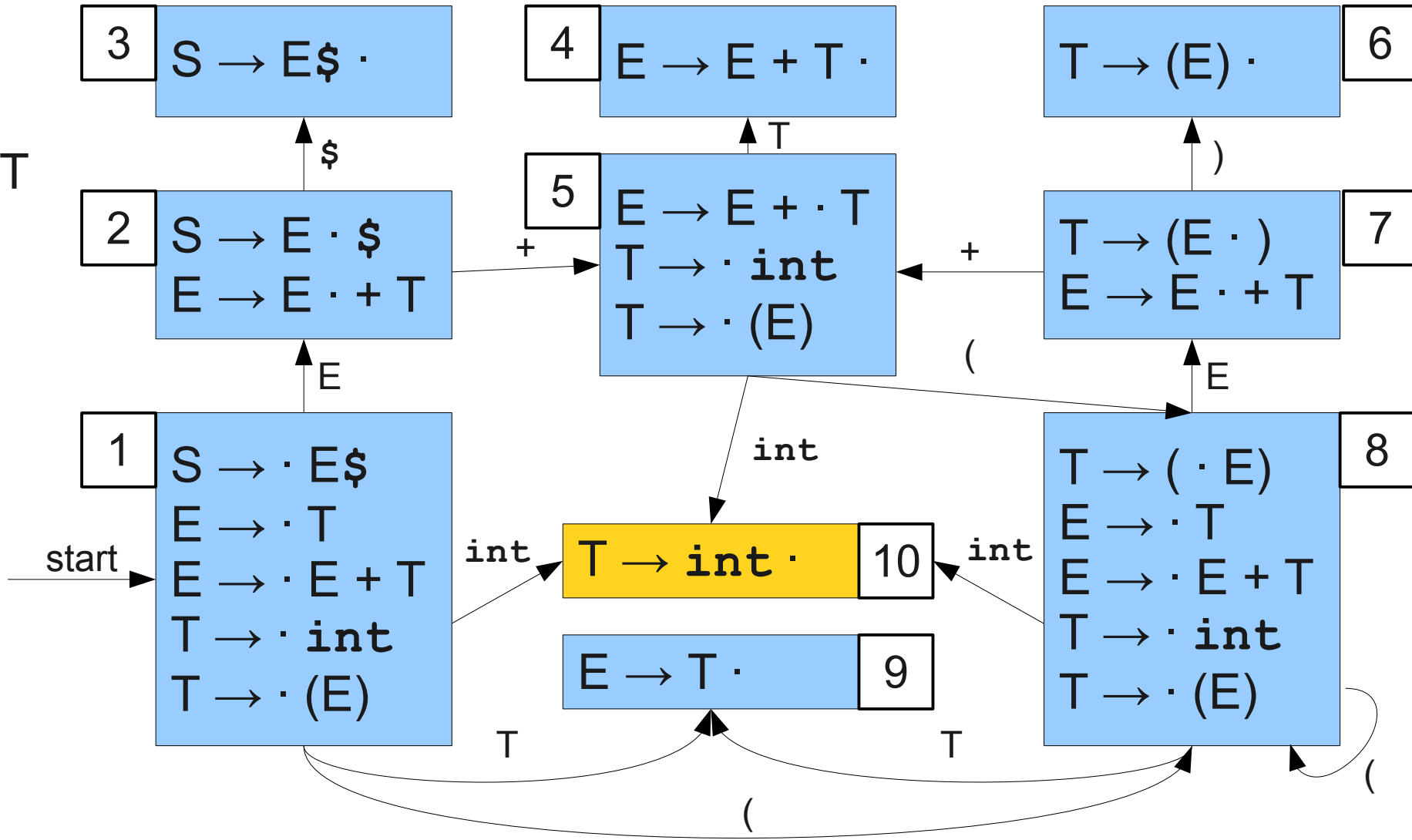
int

+ (int + int + int) \$

1

LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



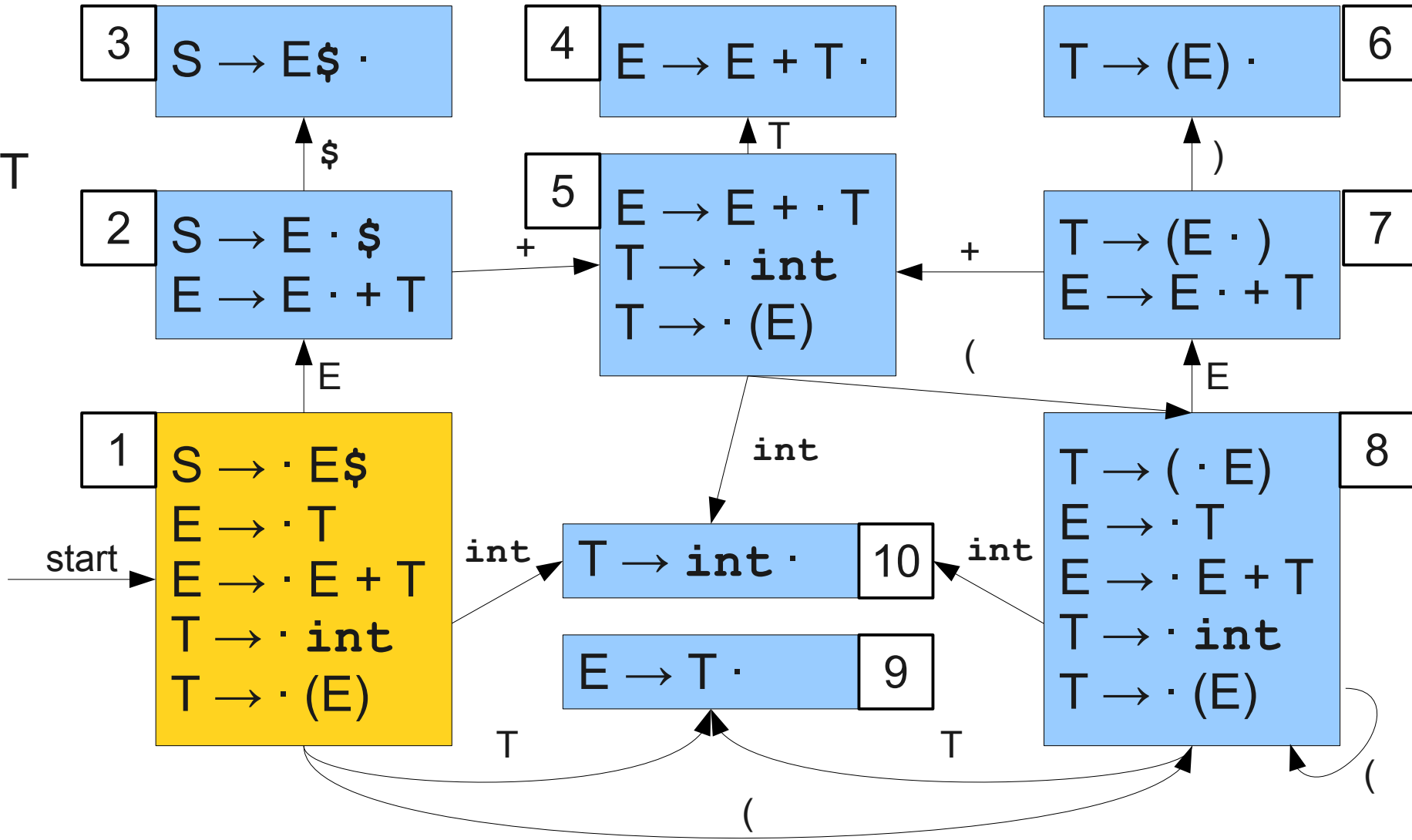
int

+ (int + int + int) \$

1 10

LR(0) Parsing

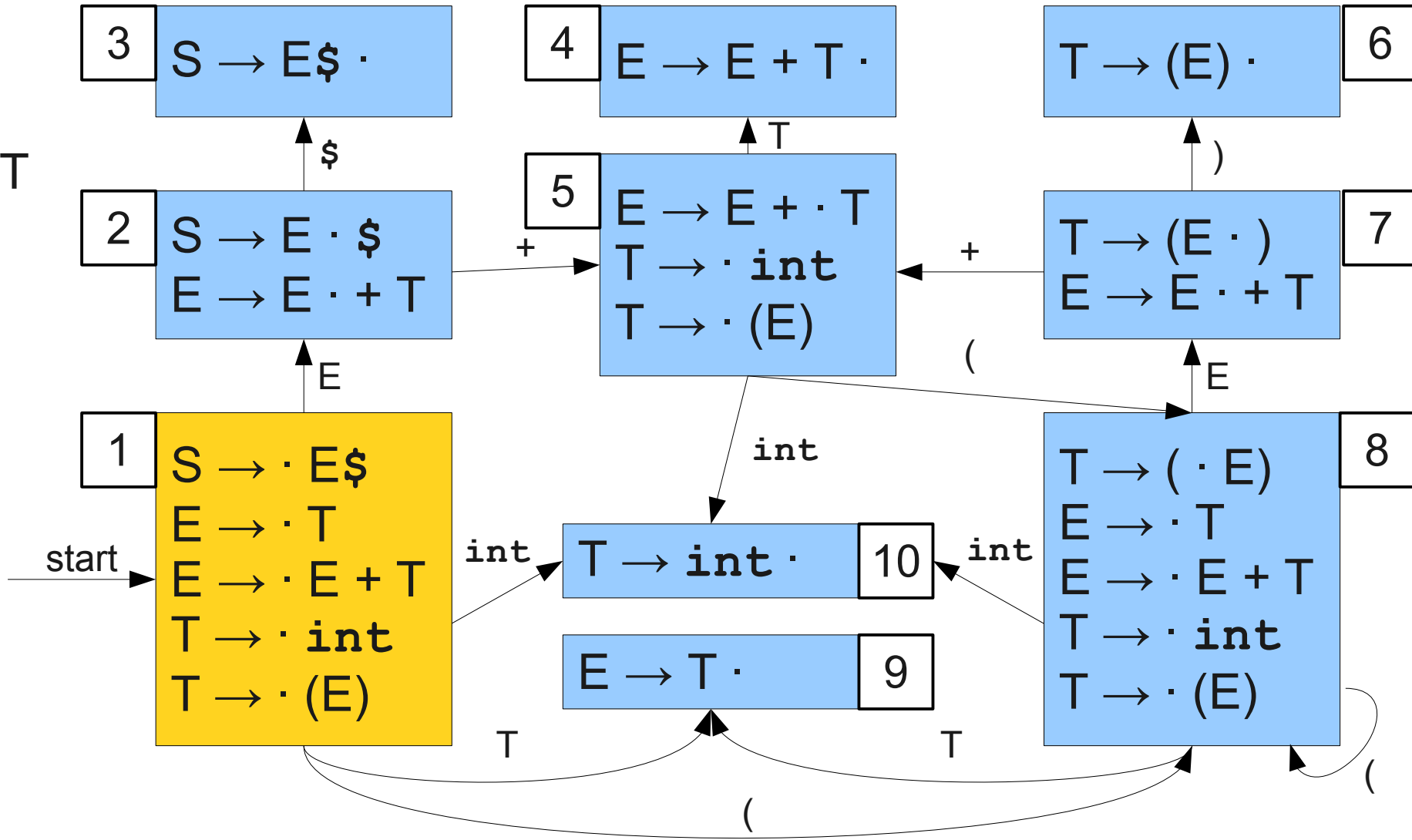
$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



+	(int	+	int	+	int)	\$
---	---	-----	---	-----	---	-----	---	----

LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



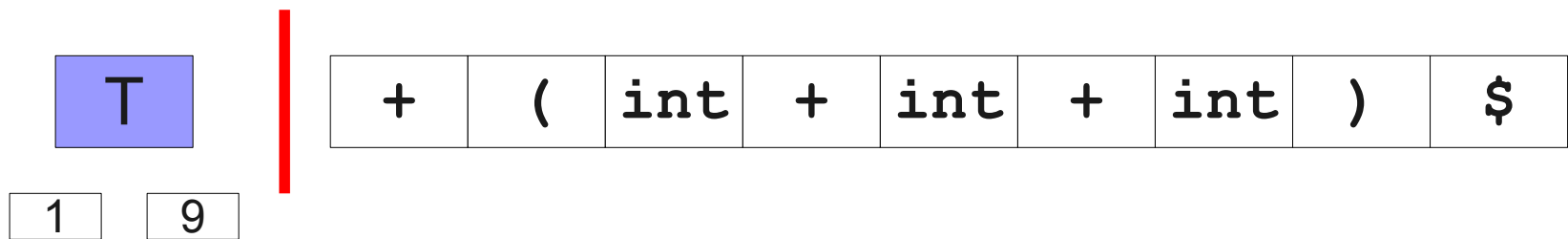
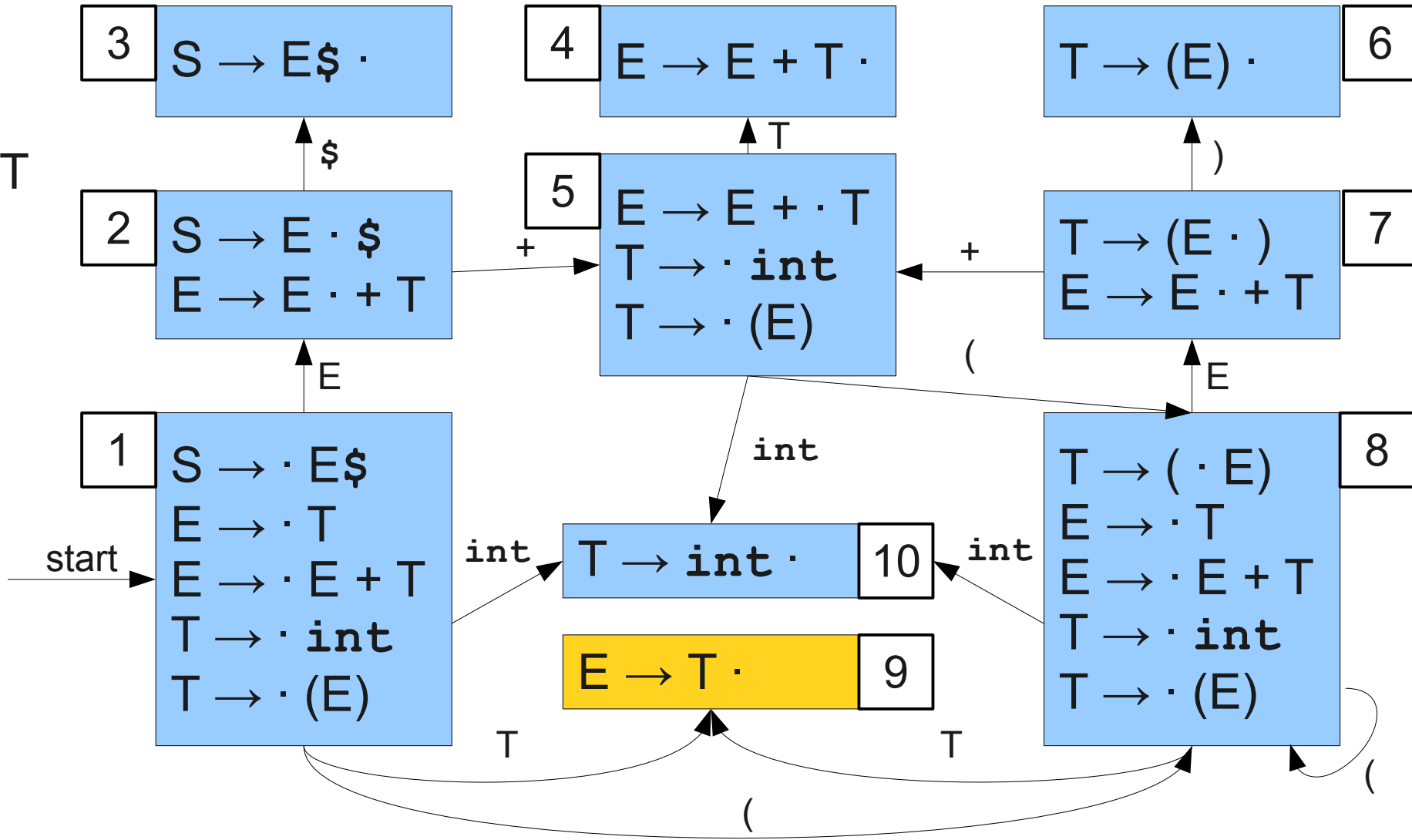
T

+ (int + int + int) \$

1

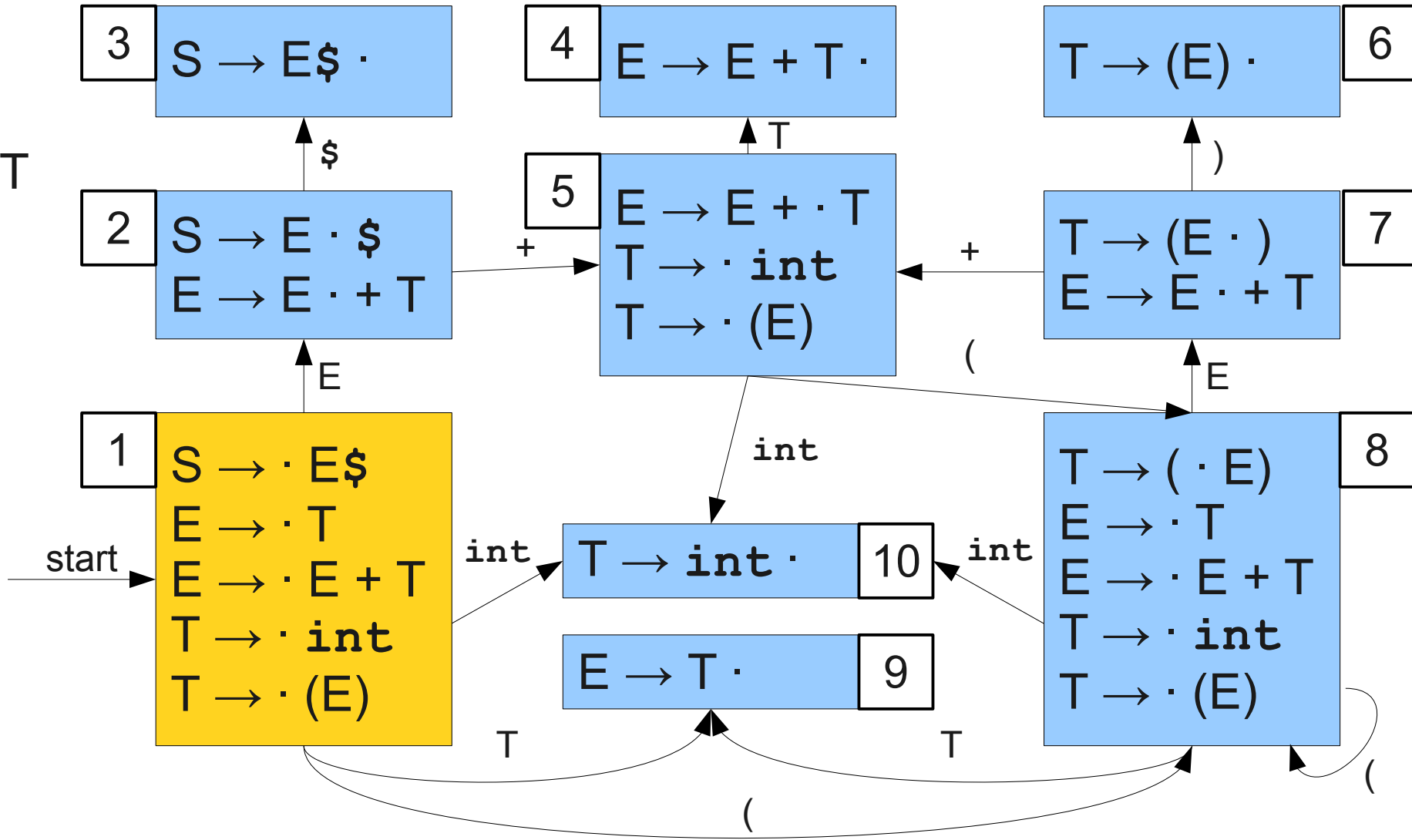
LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



LR(0) Parsing

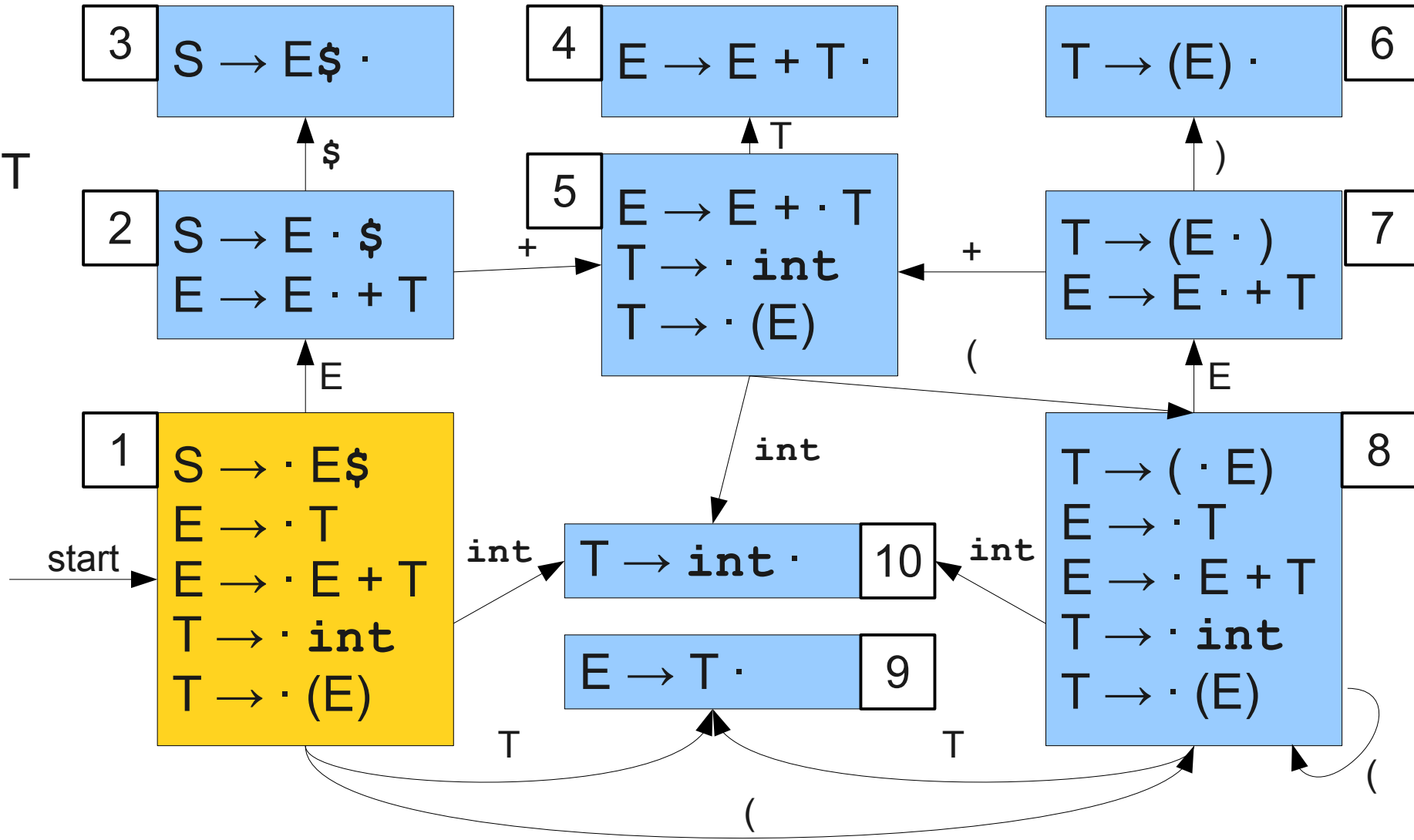
$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



+	(int	+	int	+	int)	\$
---	---	-----	---	-----	---	-----	---	----

LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



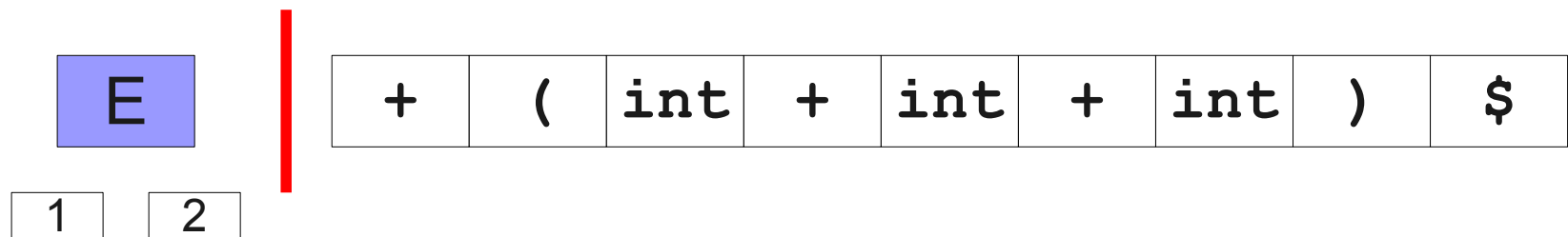
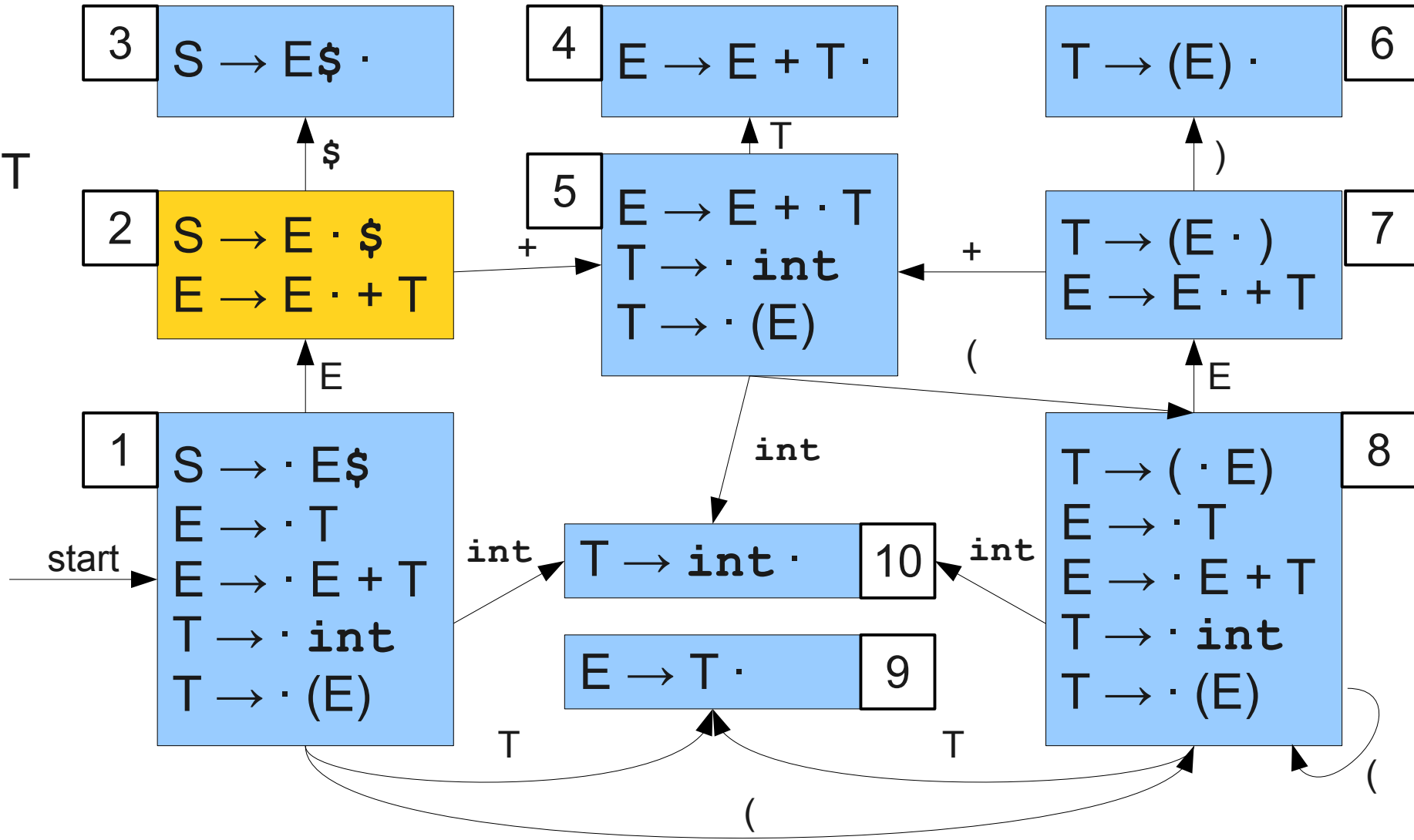
E

+	(int	+	int	+	int)	\$
---	---	-----	---	-----	---	-----	---	----

1

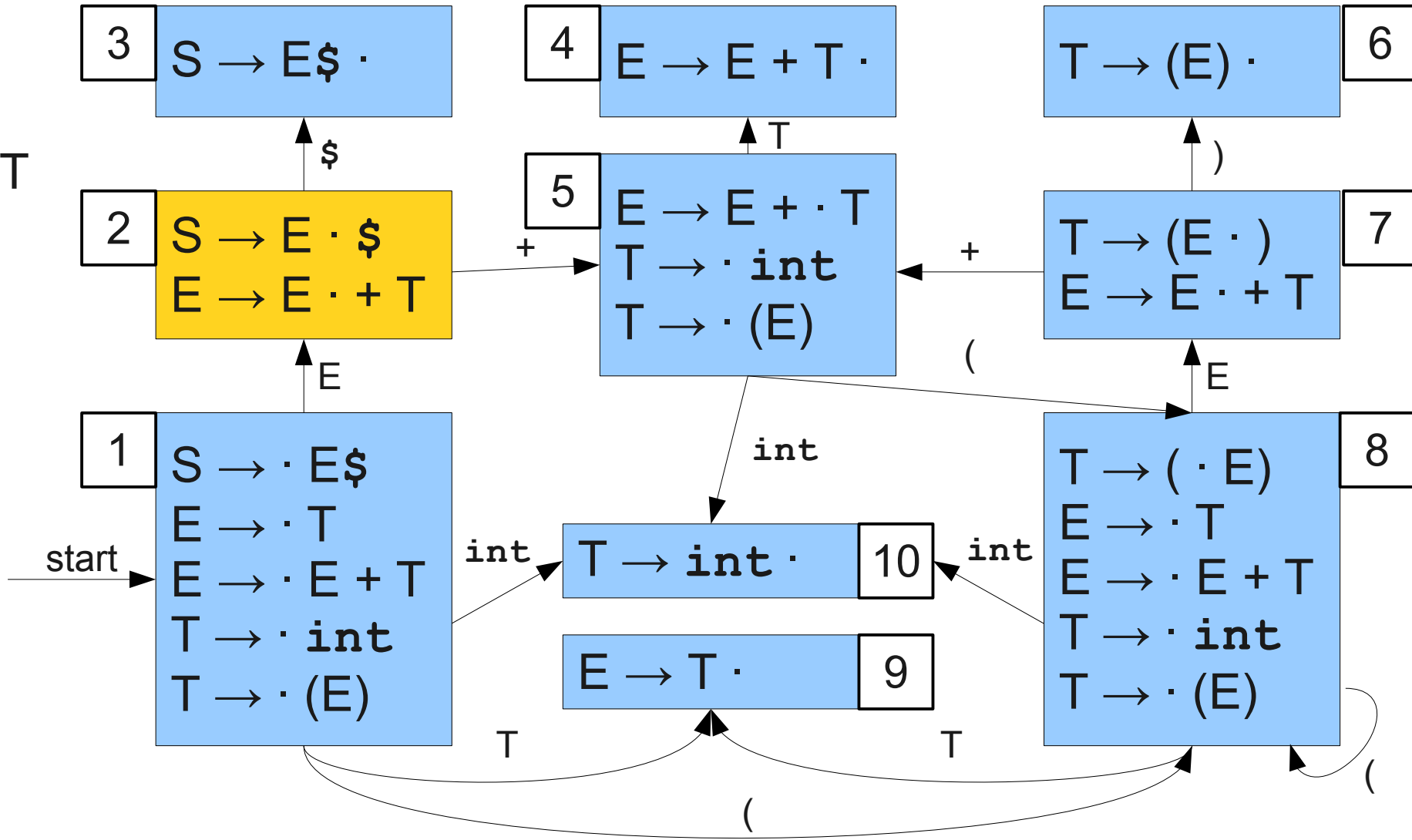
LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



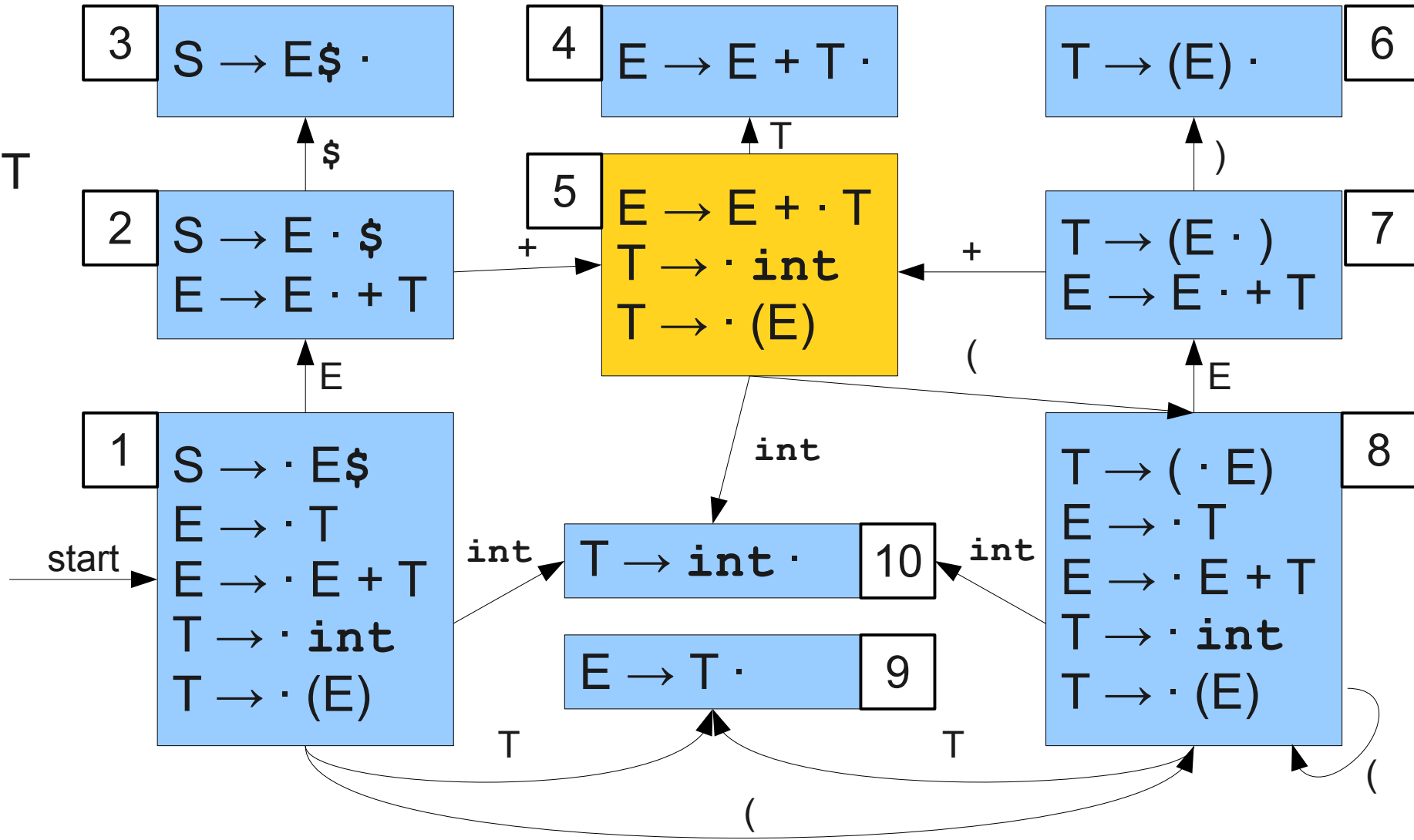
E	+
---	---

(int	+	int	+	int)	\$
---	-----	---	-----	---	-----	---	----

1	2
---	---

LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



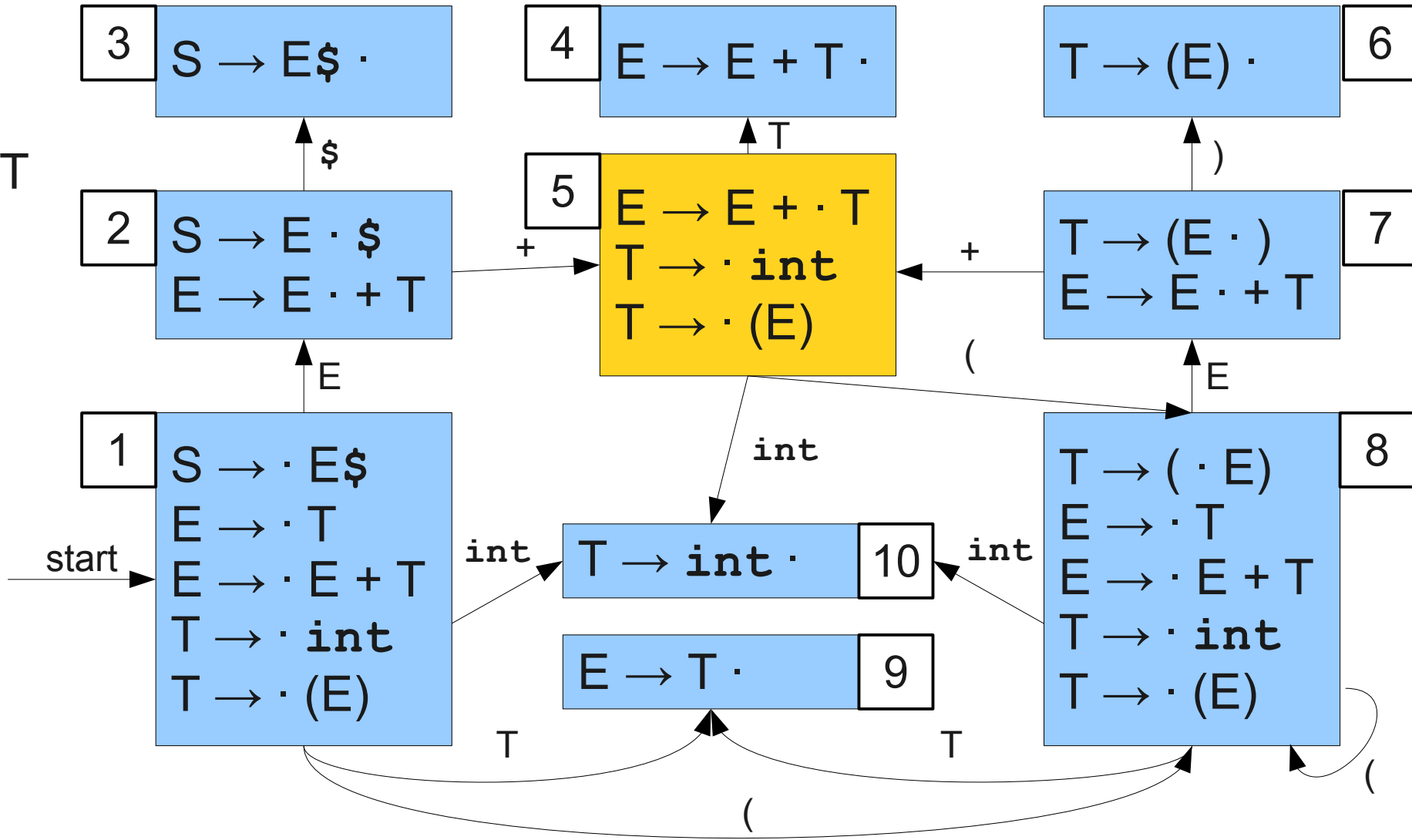
E +

(int + int + int) \$

1 2 5

LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



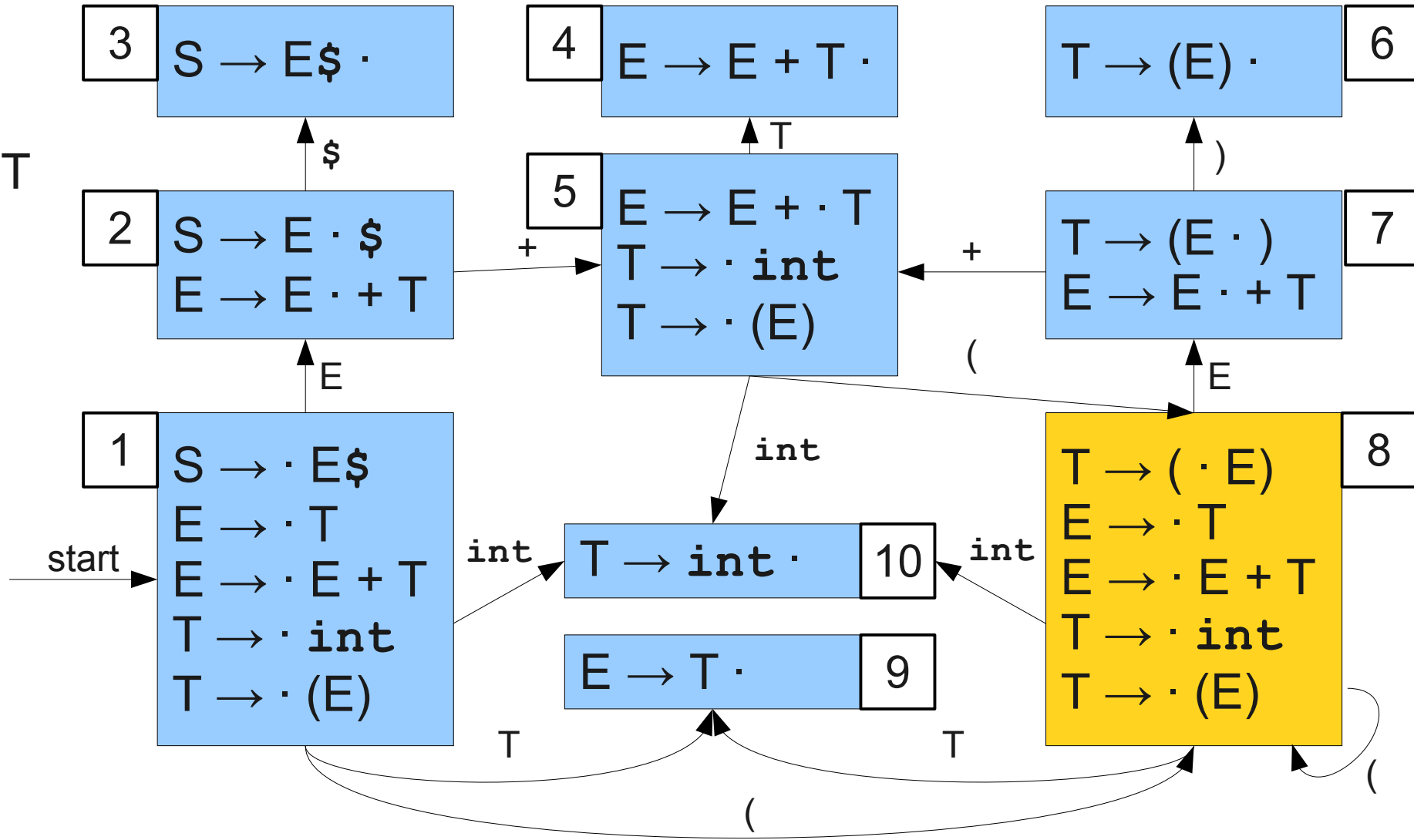
E	+	(
---	---	---

int	+	int	+	int)	\$
-----	---	-----	---	-----	---	----

1	2	5
---	---	---

LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



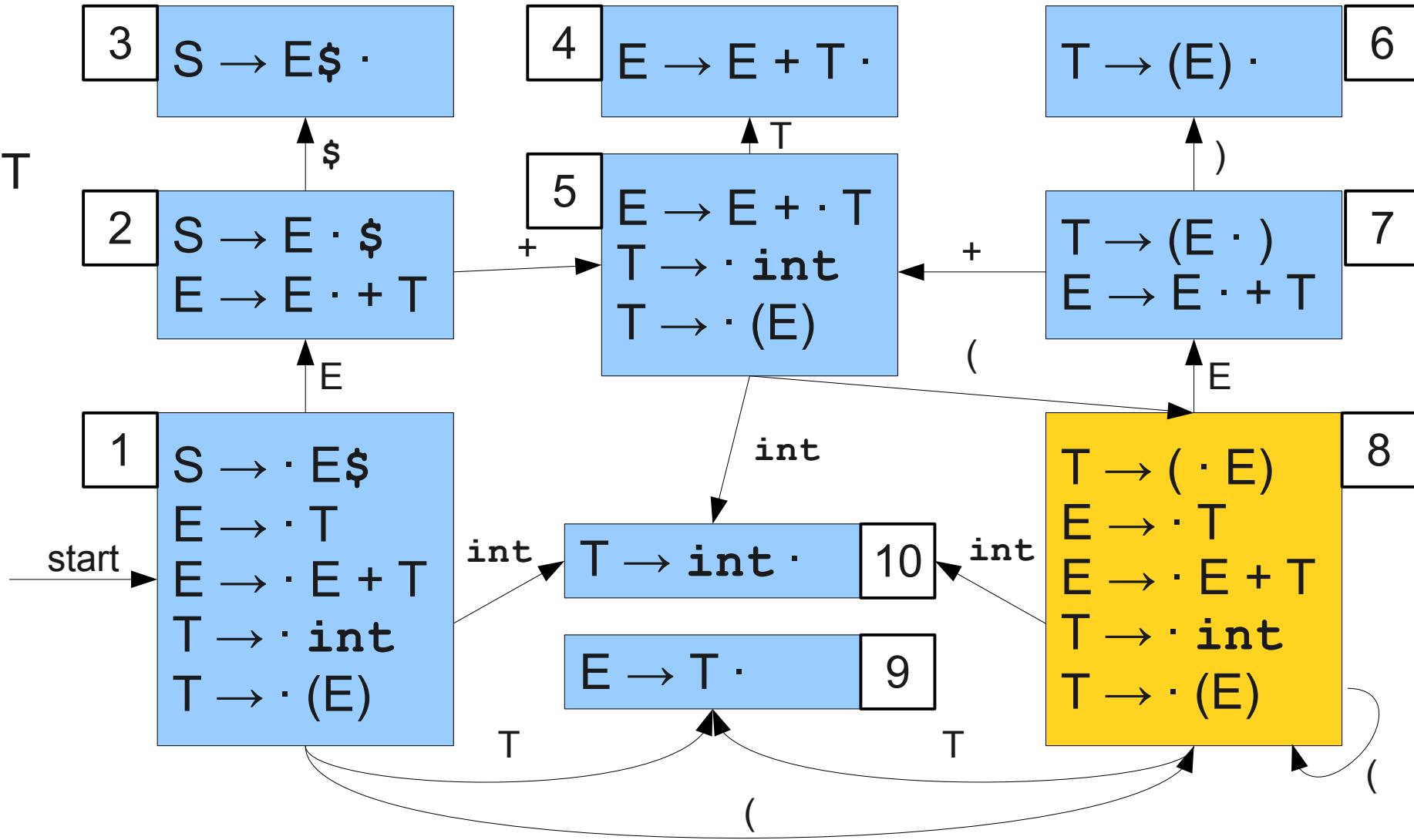
E	+	(
---	---	---

int	+	int	+	int)	\$
-----	---	-----	---	-----	---	----

1	2	5	8
---	---	---	---

LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



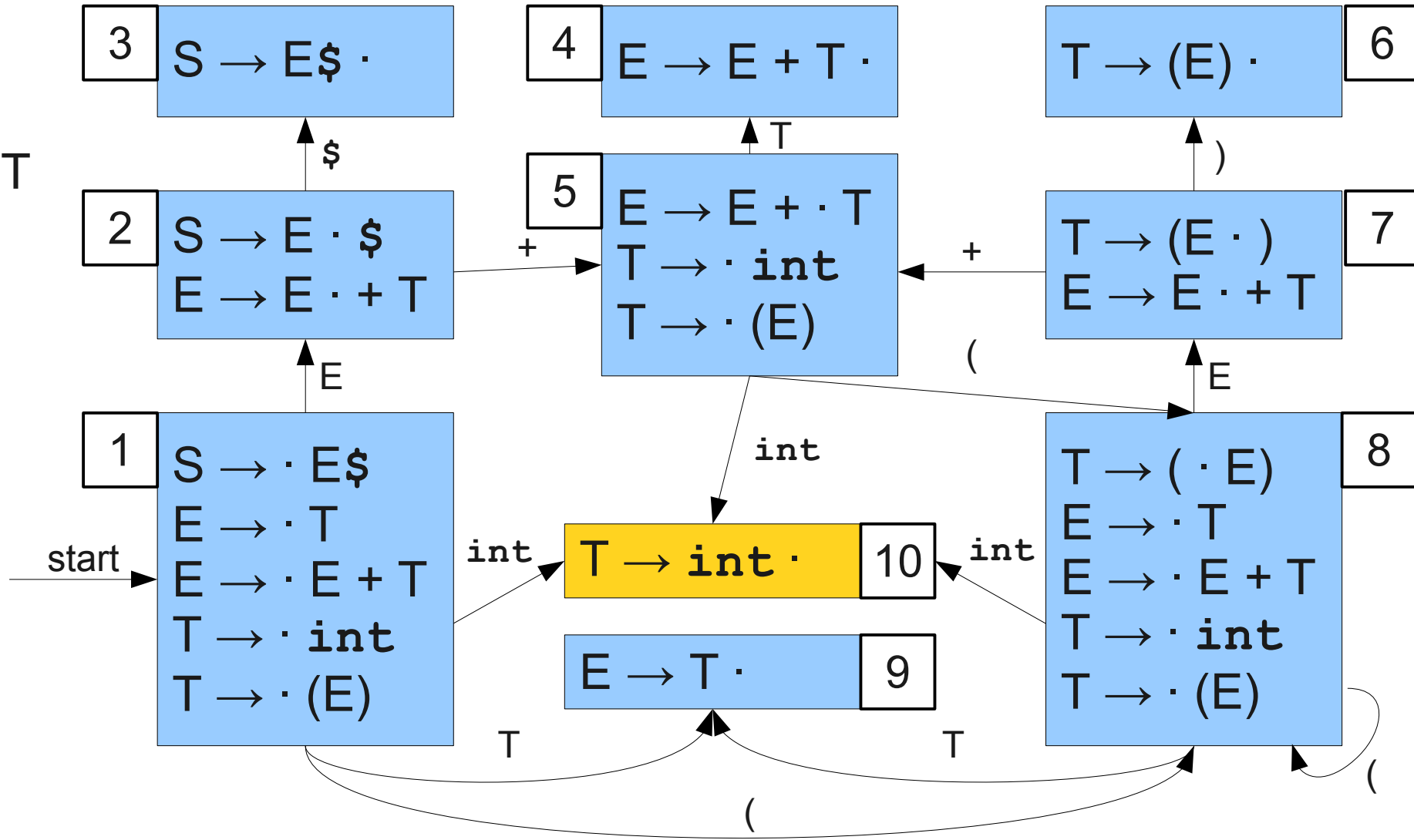
E	+	(int
---	---	---	-----

+	int	+	int)	\$
---	-----	---	-----	---	----

1	2	5	8
---	---	---	---

LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



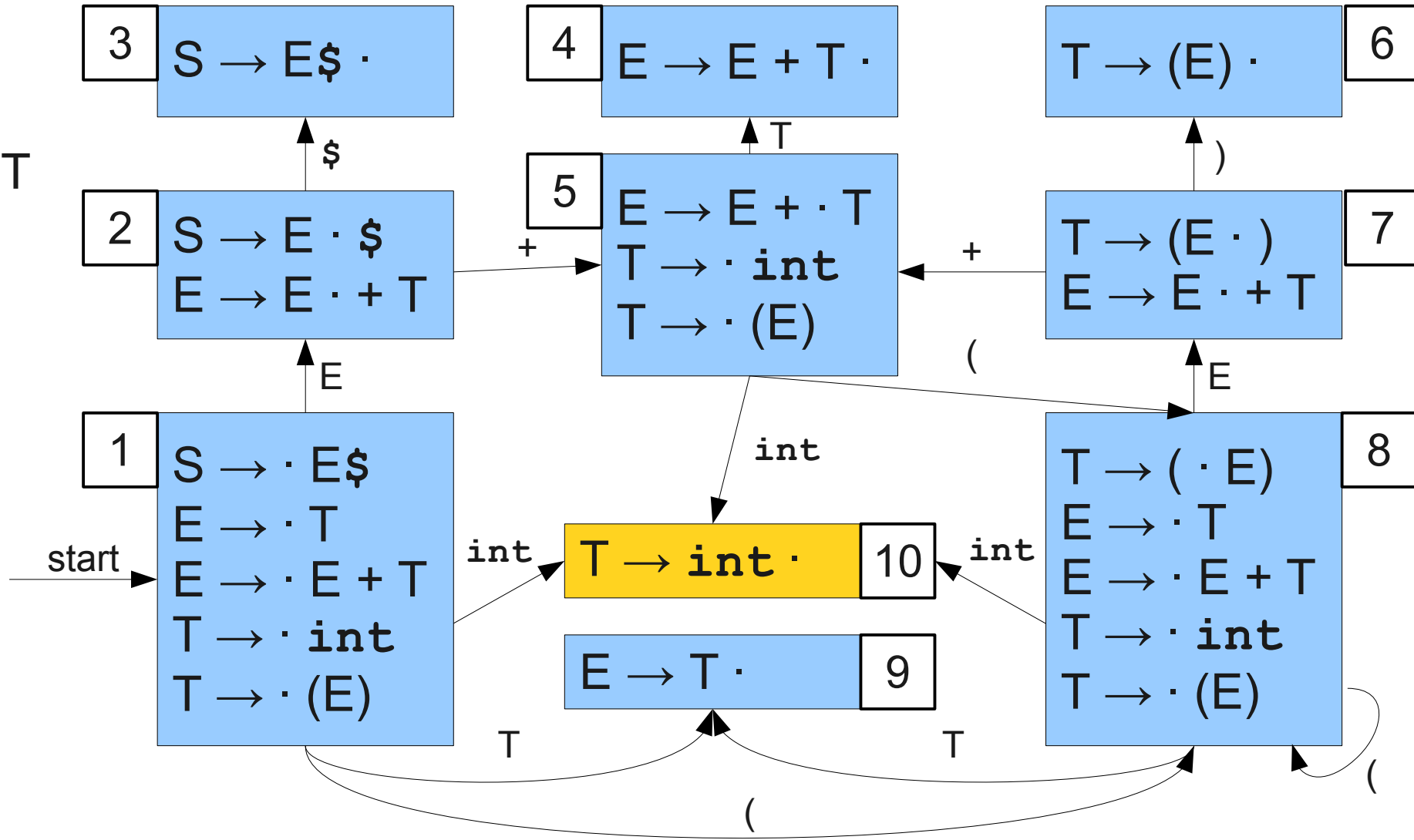
E	+	(int
---	---	---	-----

+	int	+	int)	\$
---	-----	---	-----	---	----

1	2	5	8
---	---	---	---

LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



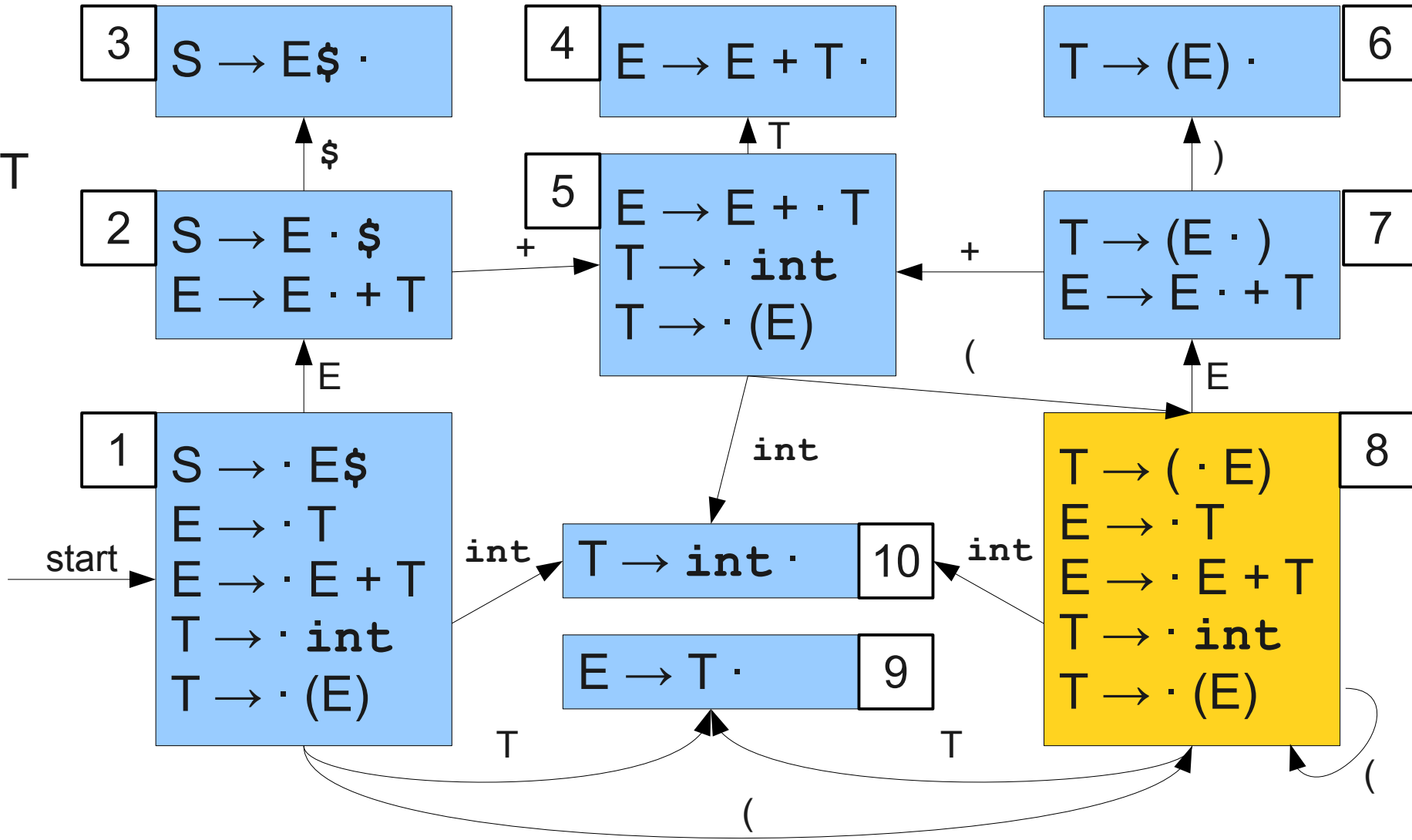
E	+	(int
---	---	---	-----

+	int	+	int)	\$
---	-----	---	-----	---	----

1	2	5	8	10
---	---	---	---	----

LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



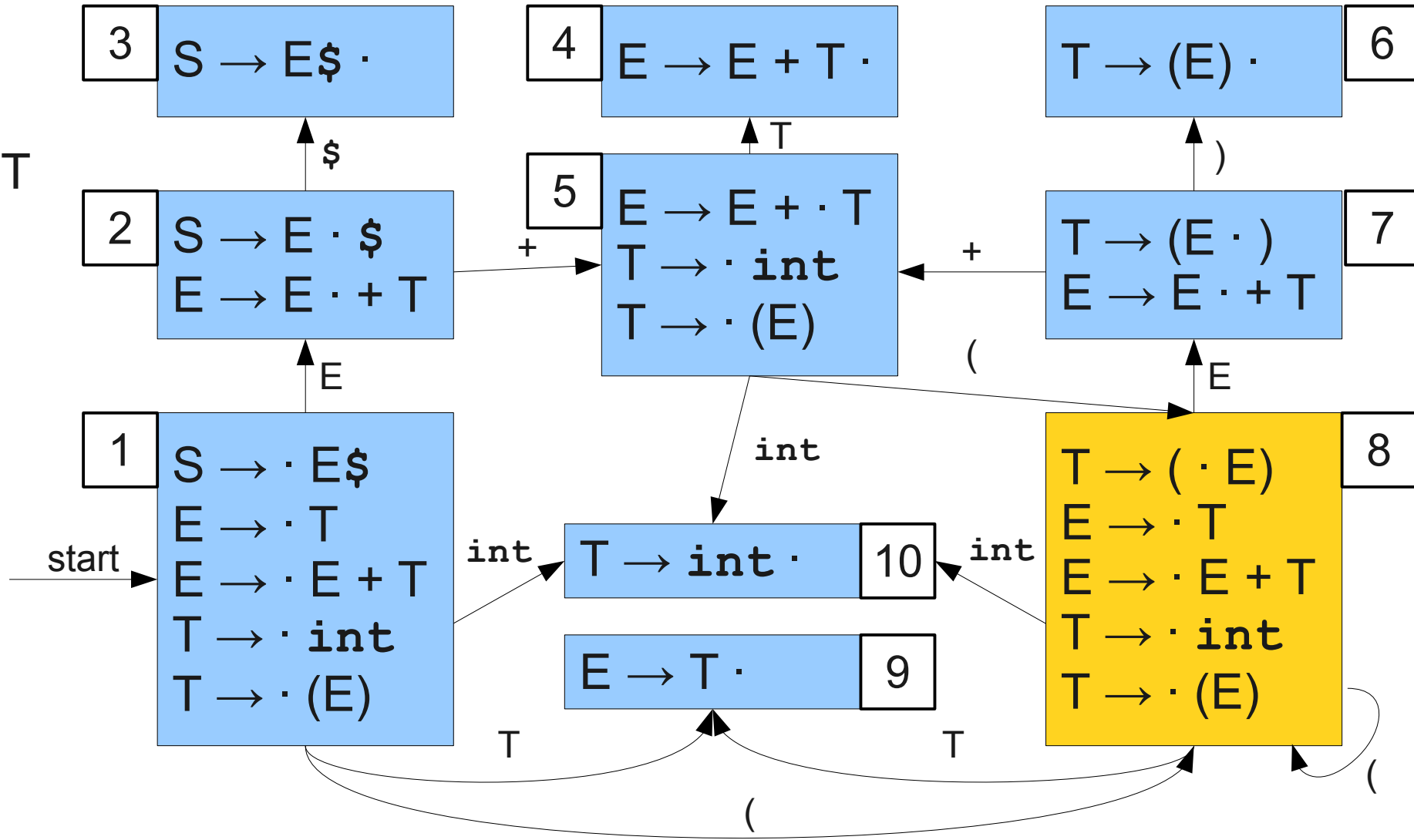
E	+	(
---	---	---

+	int	+	int)	\$
---	-----	---	-----	---	----

1	2	5	8
---	---	---	---

LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



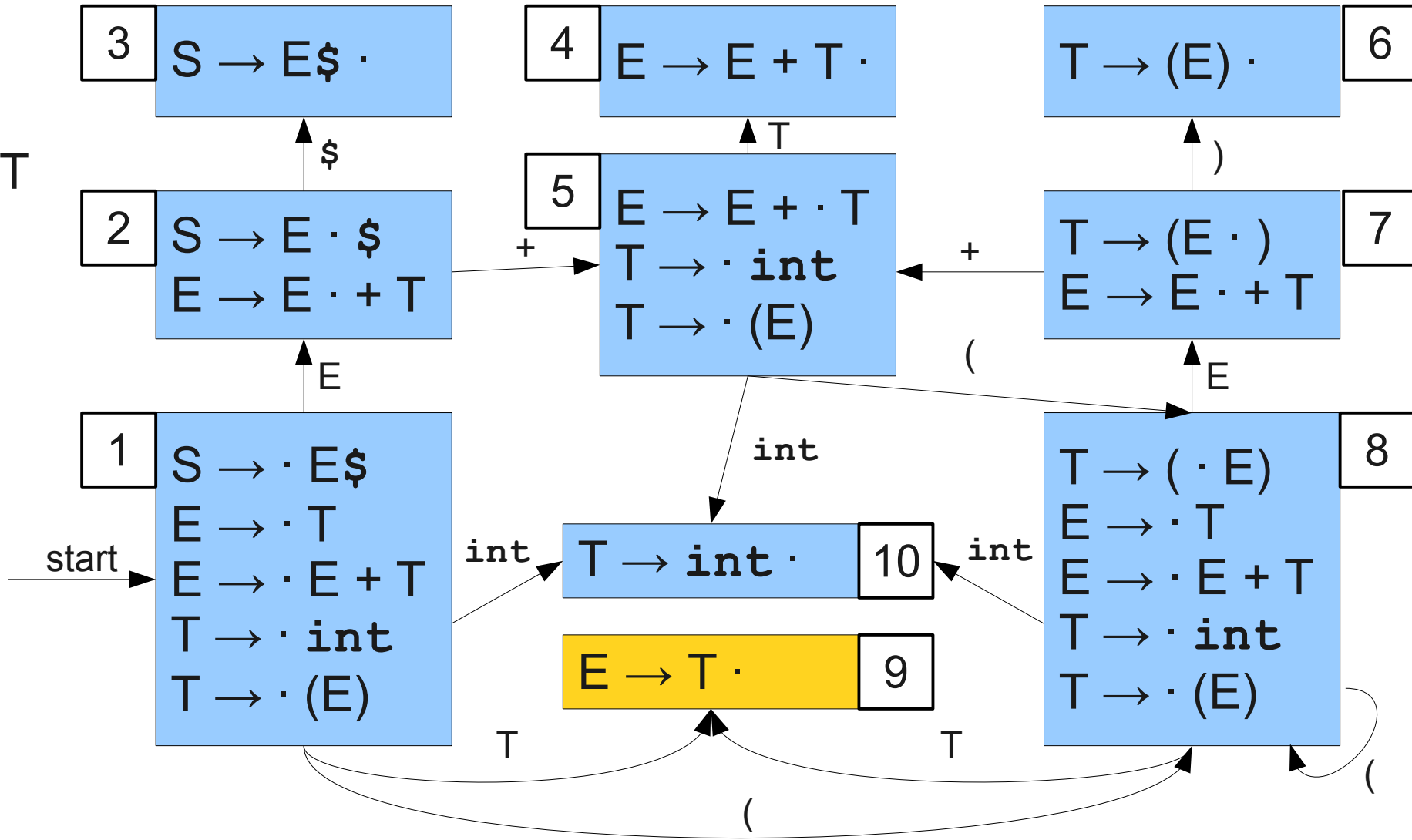
E + (**T**

+ int + int) \$

1 2 5 8

LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



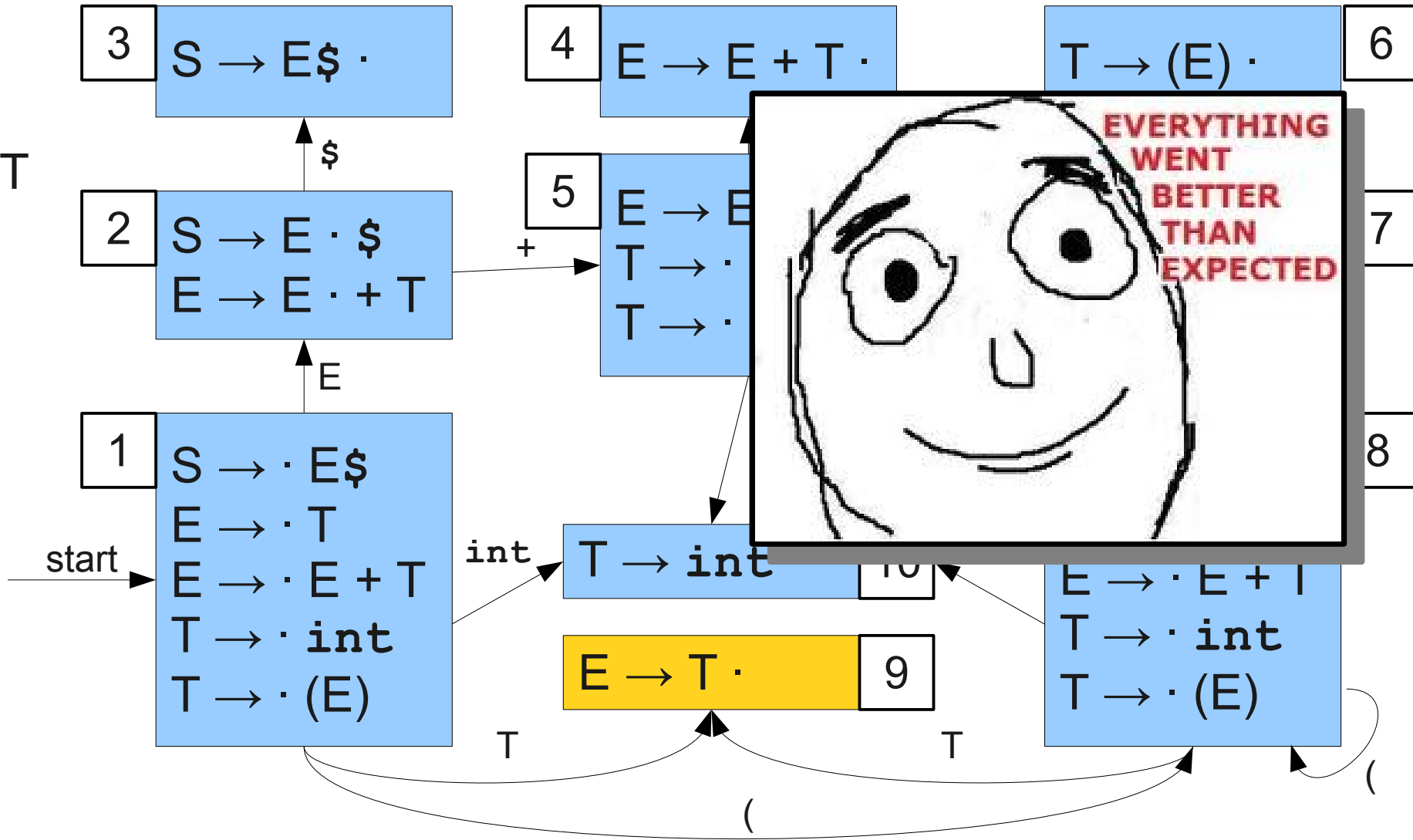
E + (**T**

+ int + int) \$

1 2 5 8 9

LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



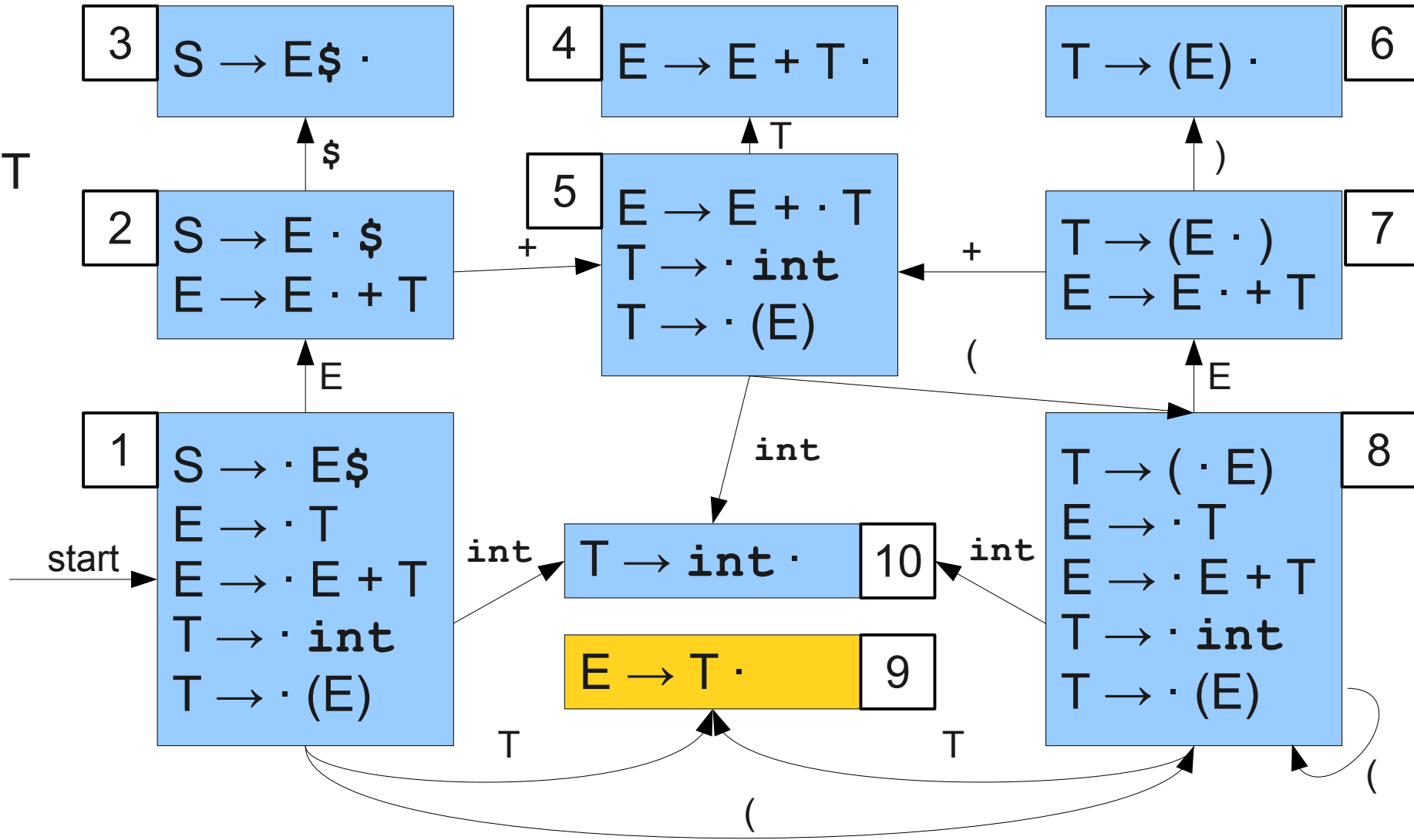
E + (T

+ int + int) \$

1 2 5 8 9

LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



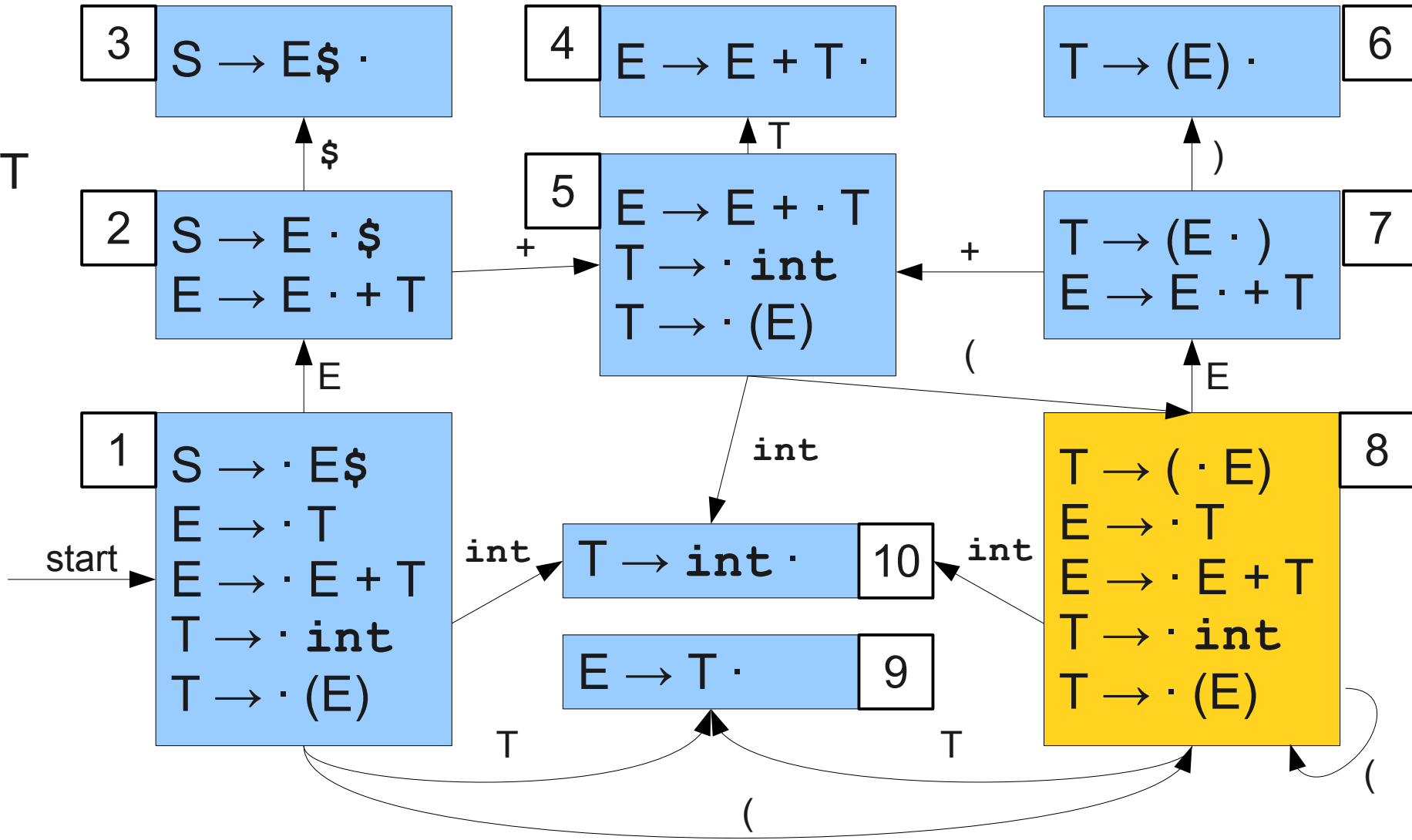
E + (**T**

+ int + int) \$

1 2 5 8 9

LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



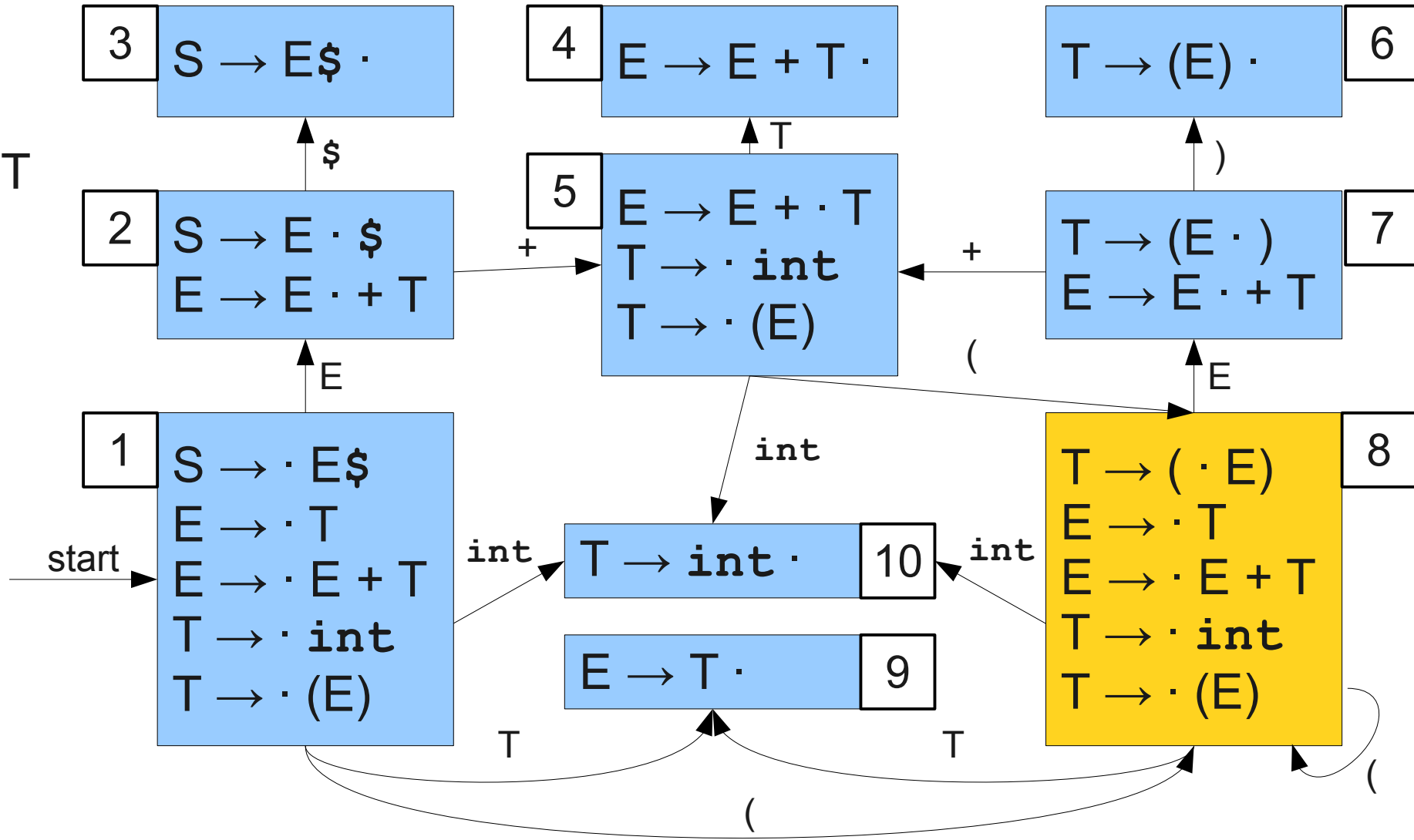
E	+	(
---	---	---

+	int	+	int)	\$
---	-----	---	-----	---	----

1	2	5	8
---	---	---	---

LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



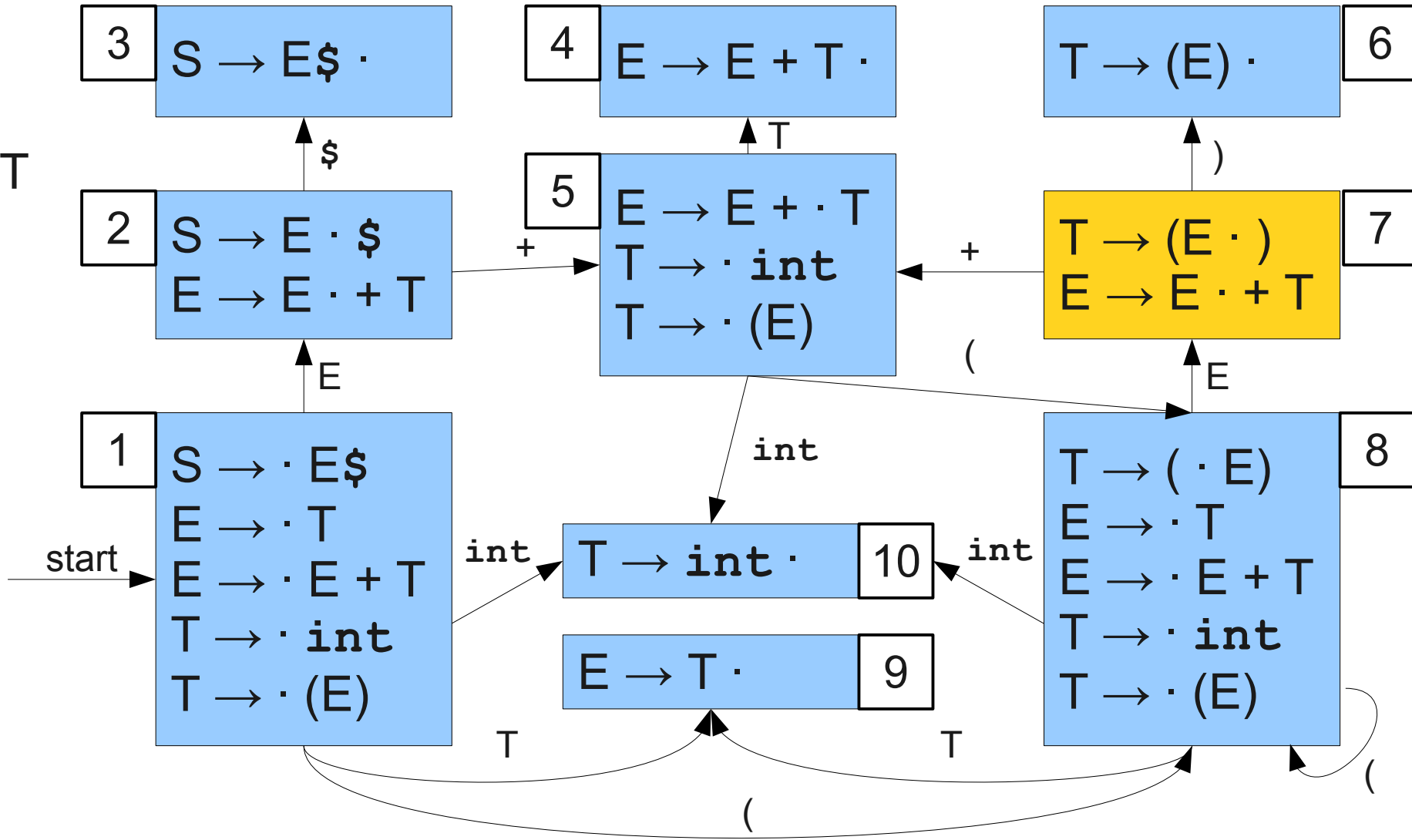
E + (**E**

+ int + int) \$

1 2 5 8

LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



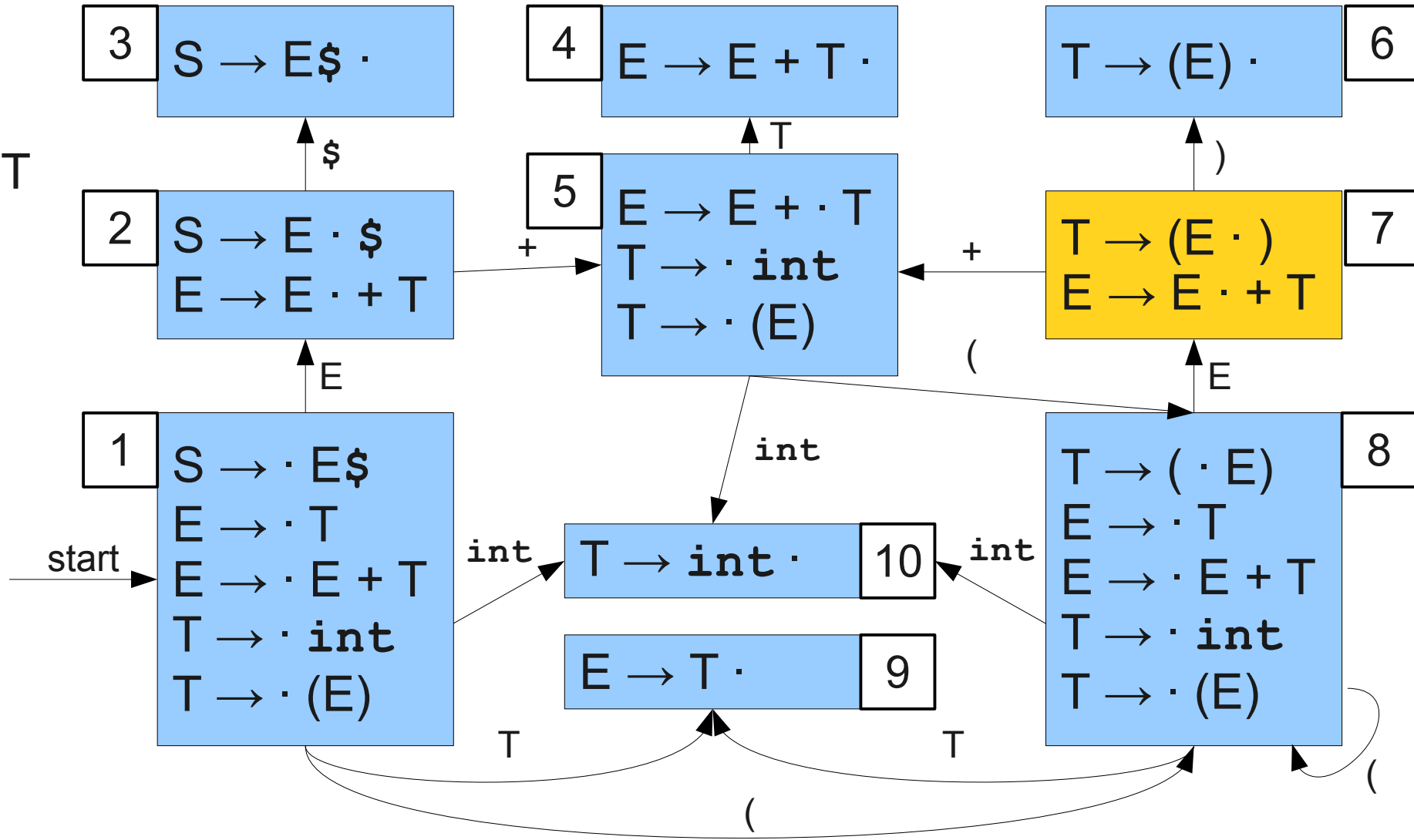
E + (**E**

+ int + int) \$

1 2 5 8 7

LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



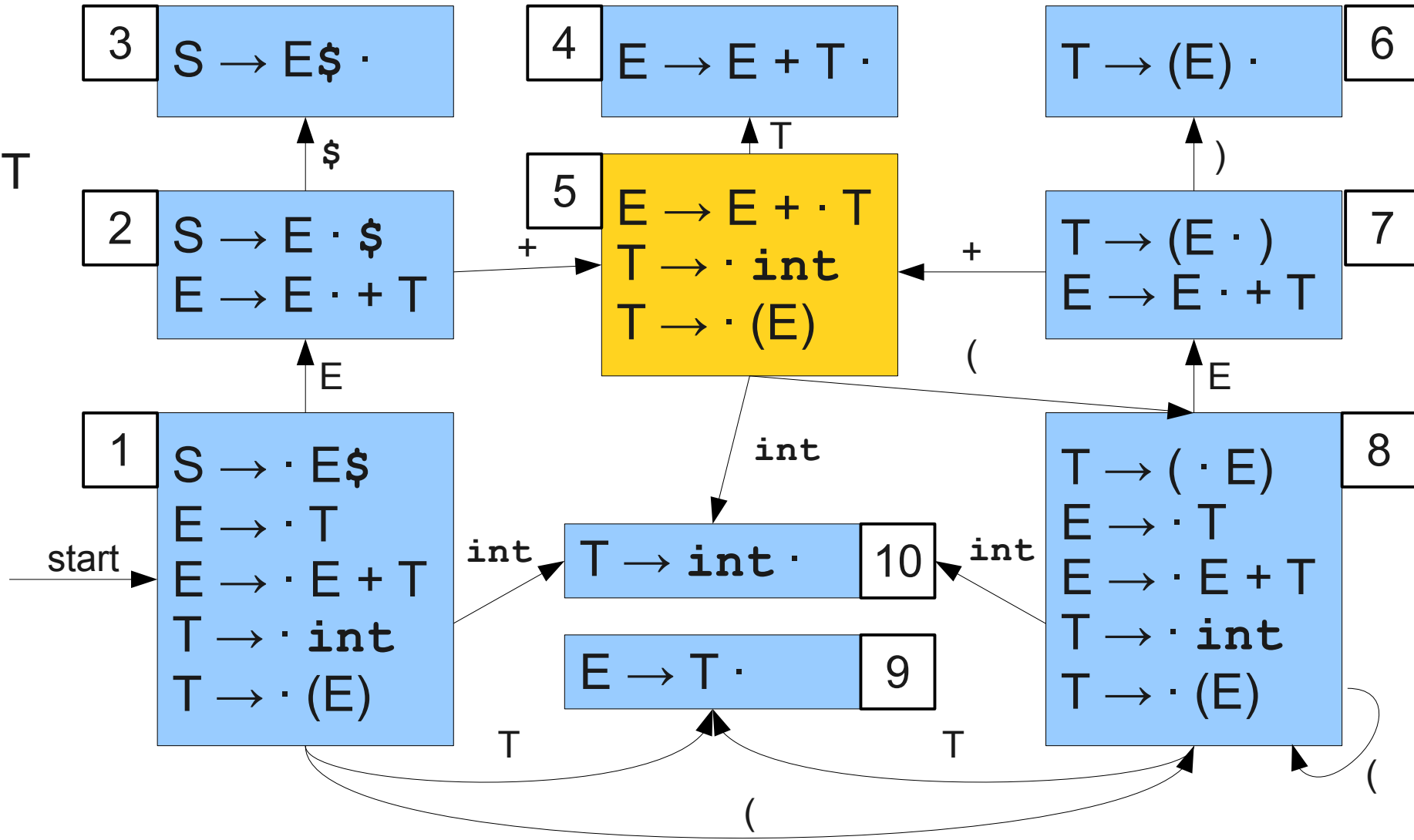
E	+	(E	+
---	---	---	---	---

int	+	int)	\$
-----	---	-----	---	----

1	2	5	8	7
---	---	---	---	---

LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



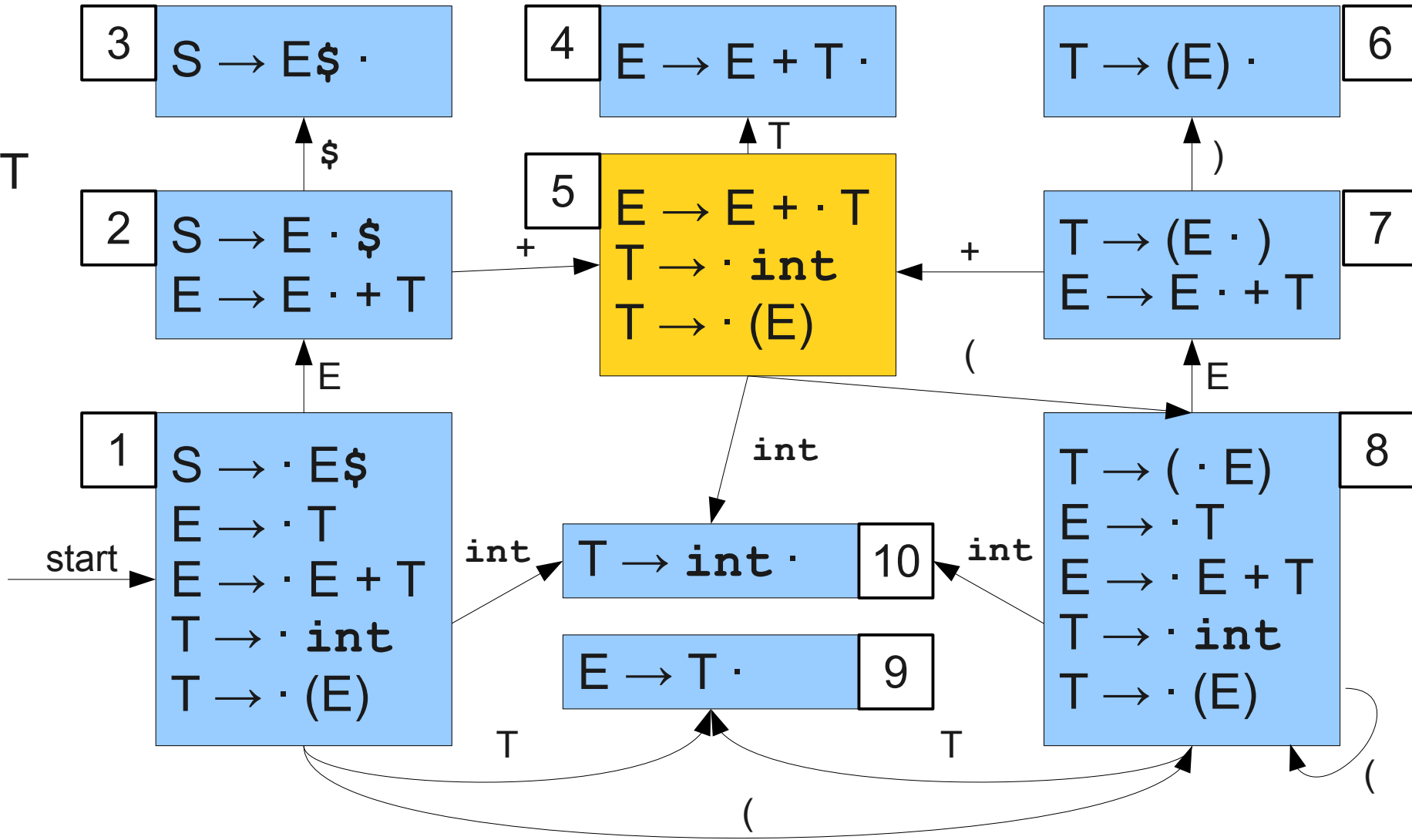
E	+	(E	+
---	---	---	---	---

int	+	int)	\$
-----	---	-----	---	----

1	2	5	8	7	5
---	---	---	---	---	---

LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



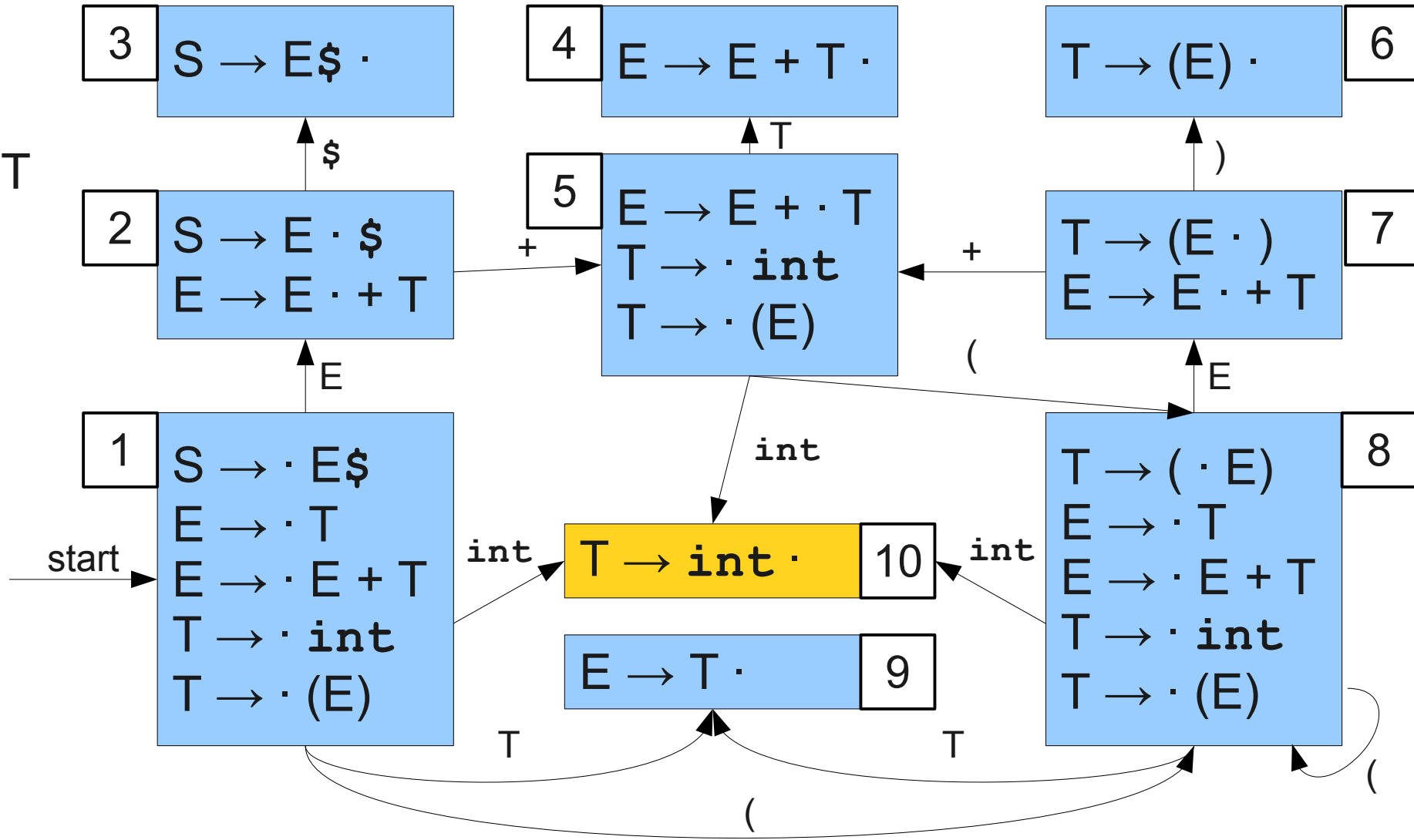
E + (**E** + int

+ int) \$

1 2 5 8 7 5

LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



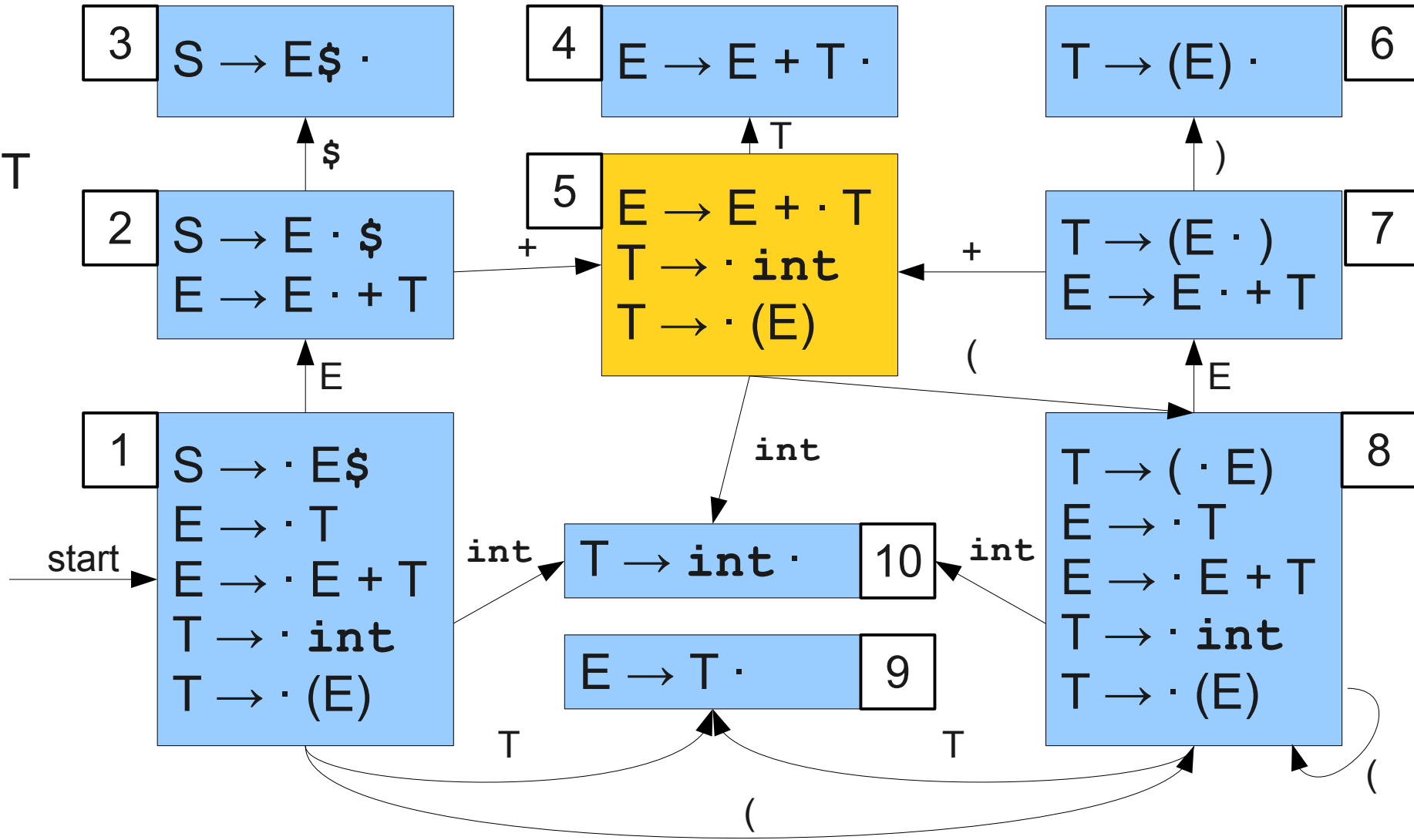
E + (**E** + int

+ int) \$

1 2 5 8 7 5 10

LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



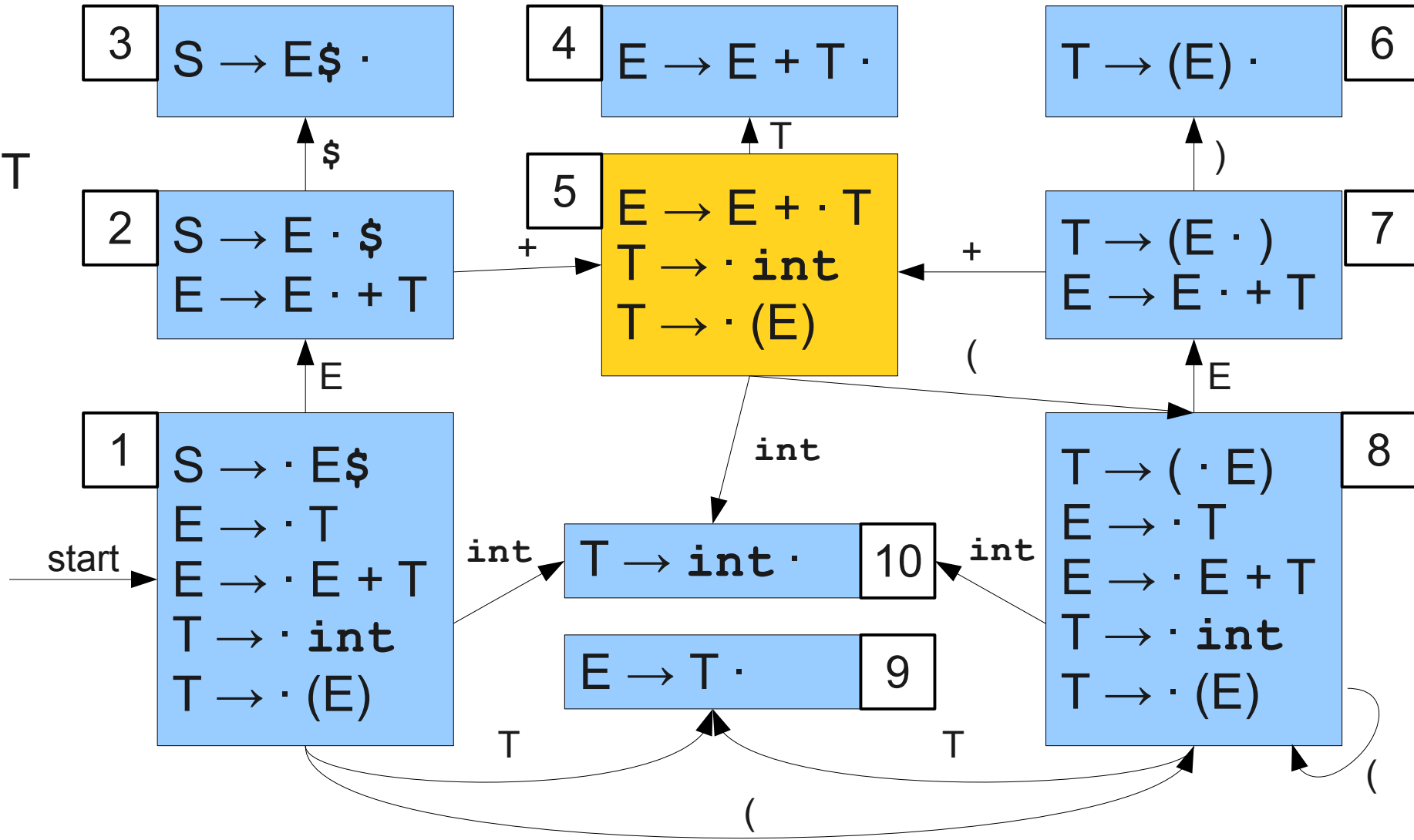
E	+	(E	+
---	---	---	---	---

+	int)	\$
---	-----	---	----

1	2	5	8	7	5
---	---	---	---	---	---

LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



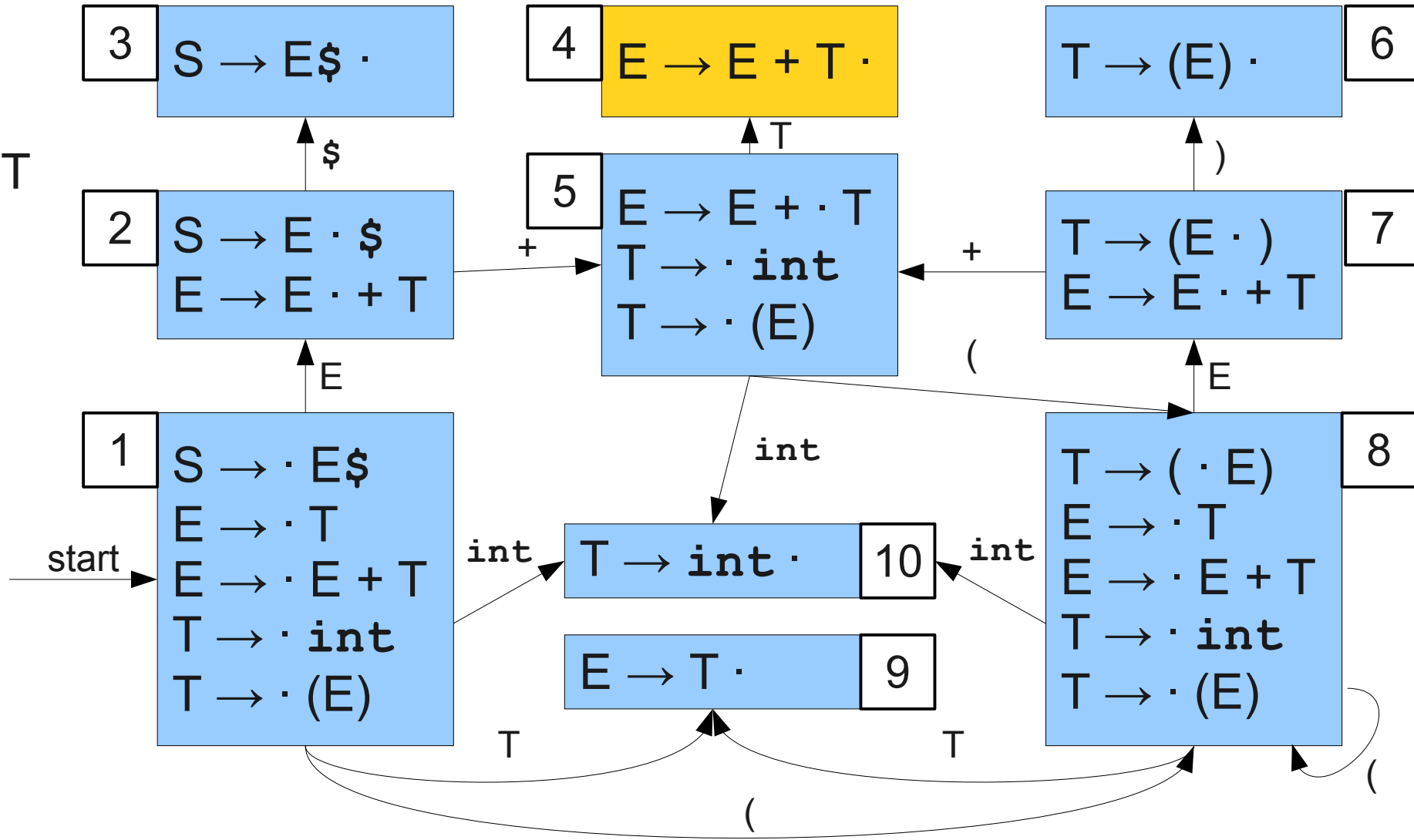
E + (E + T

+ int) \$

1 2 5 8 7 5

LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



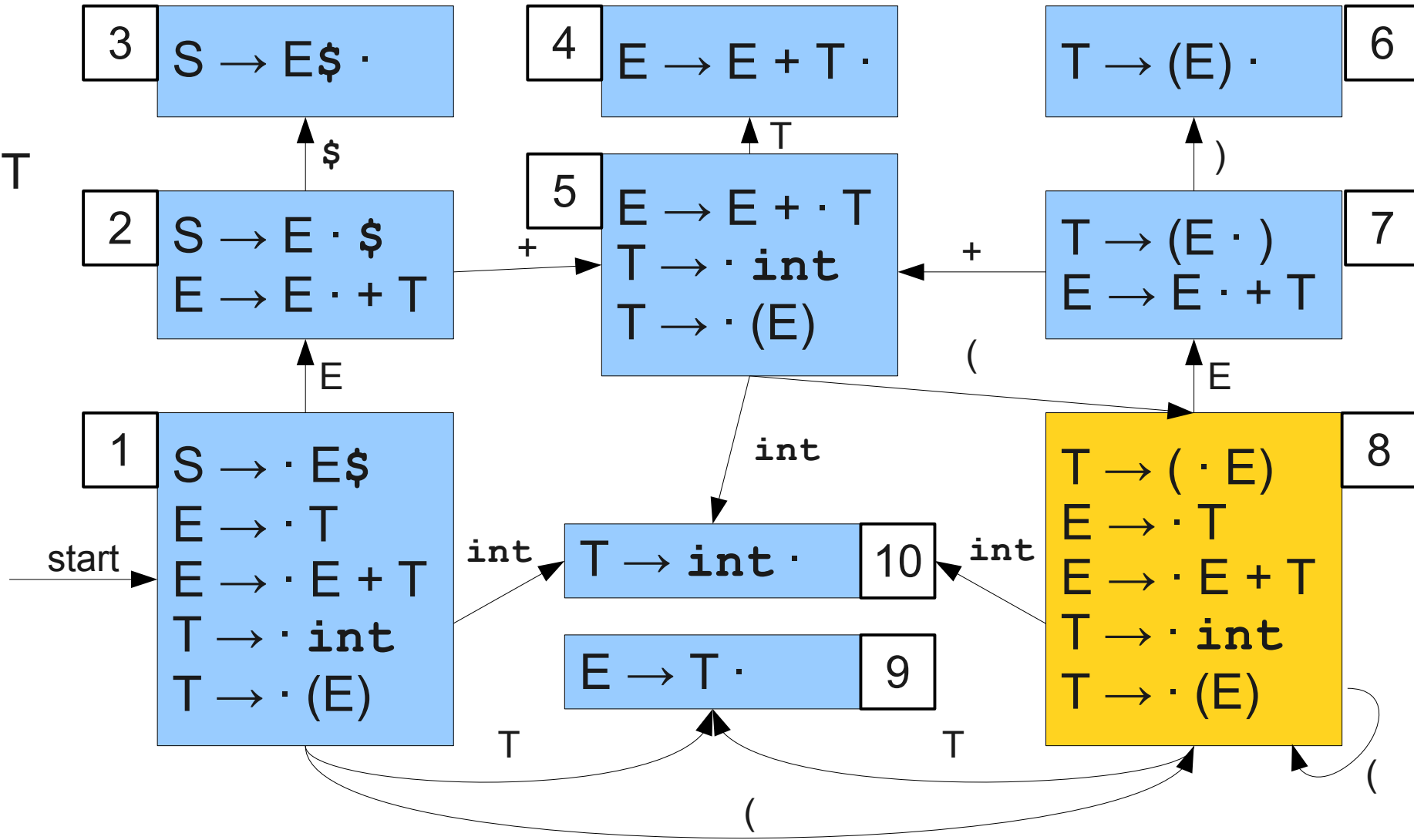
E + (E + T

+ int) \$

1 2 5 8 7 5 4

LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



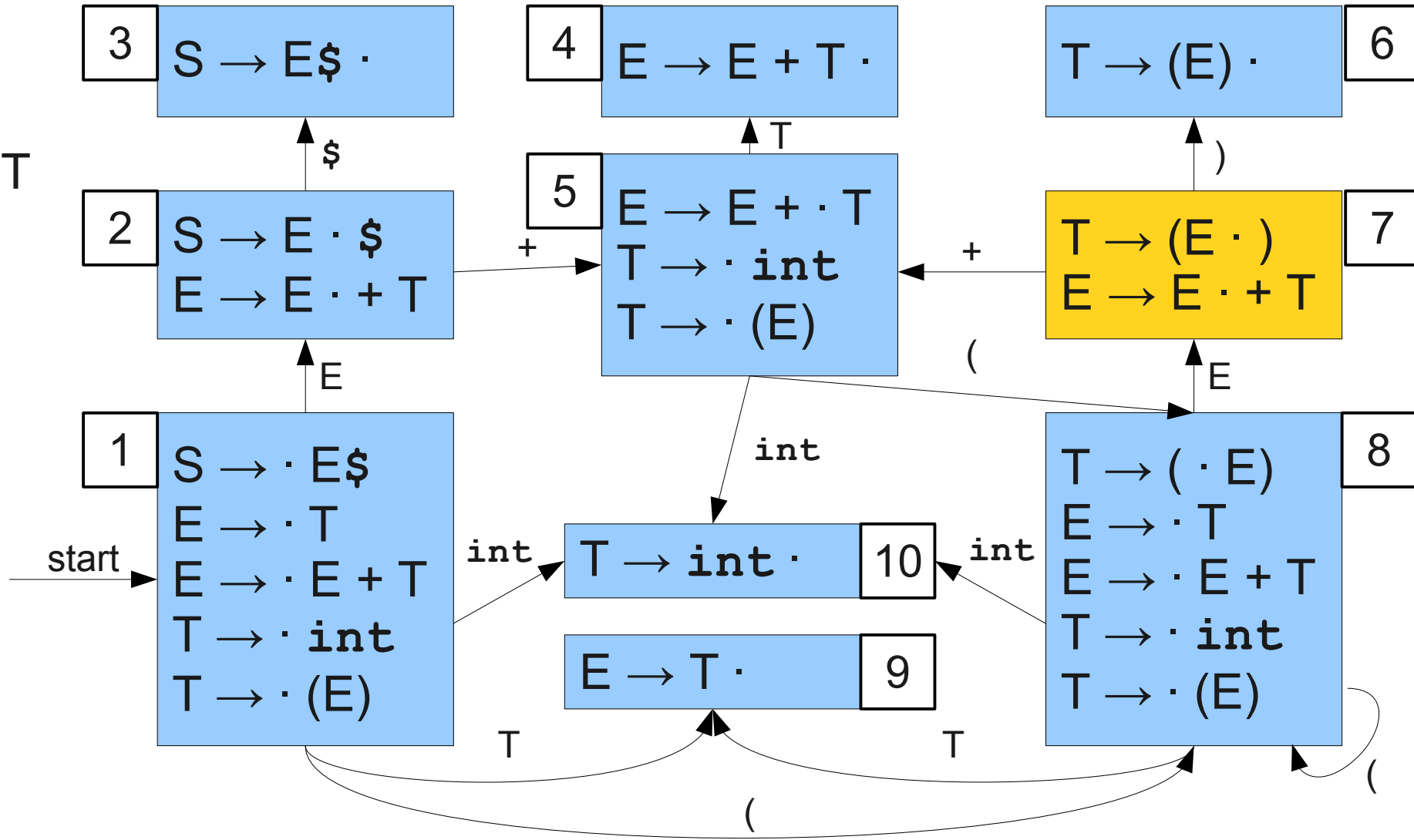
E + (E

+ int) \$

1 2 5 8

LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



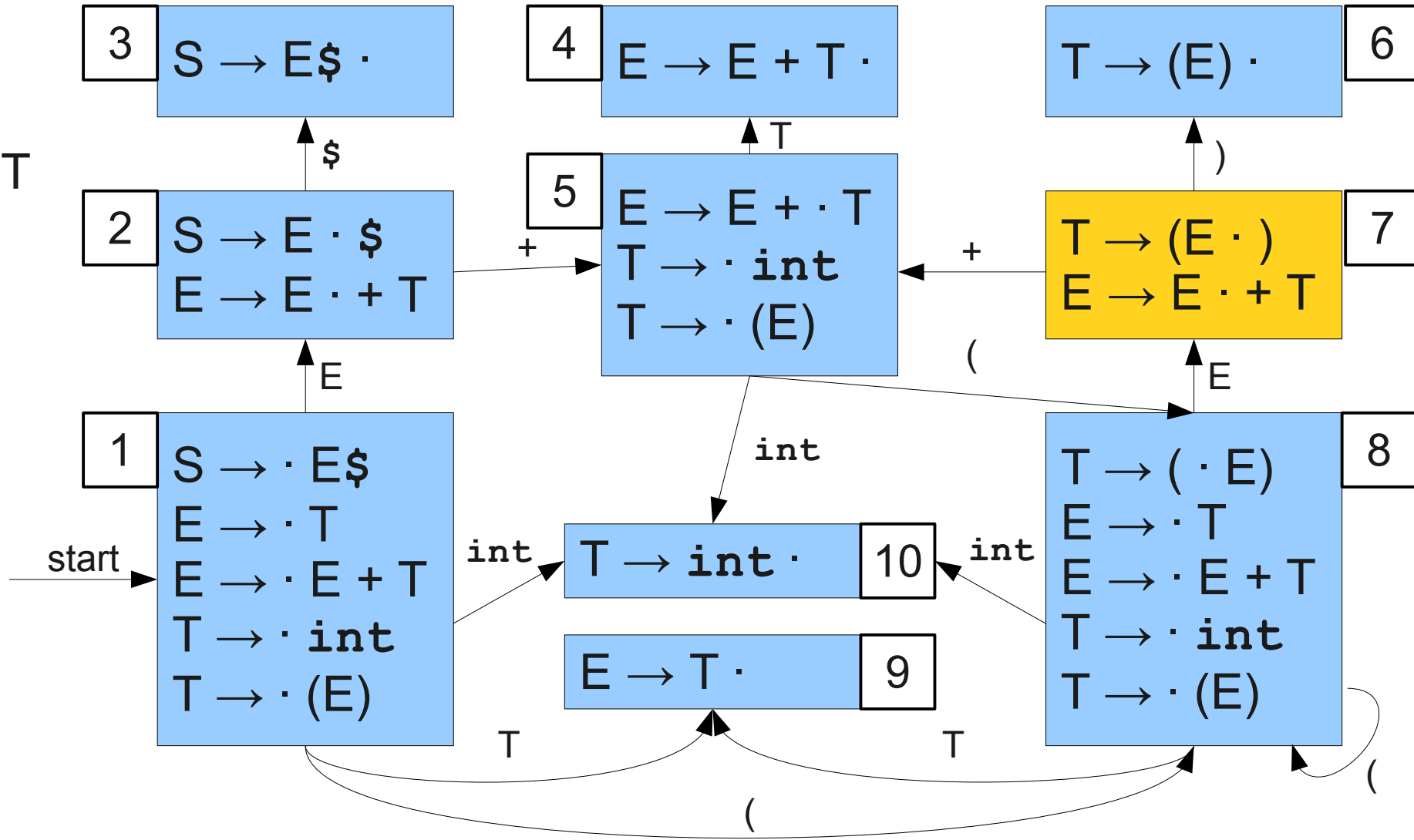
E + (**E**

+ int) \$

1 2 5 8 7

LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



E
+
(
E
+

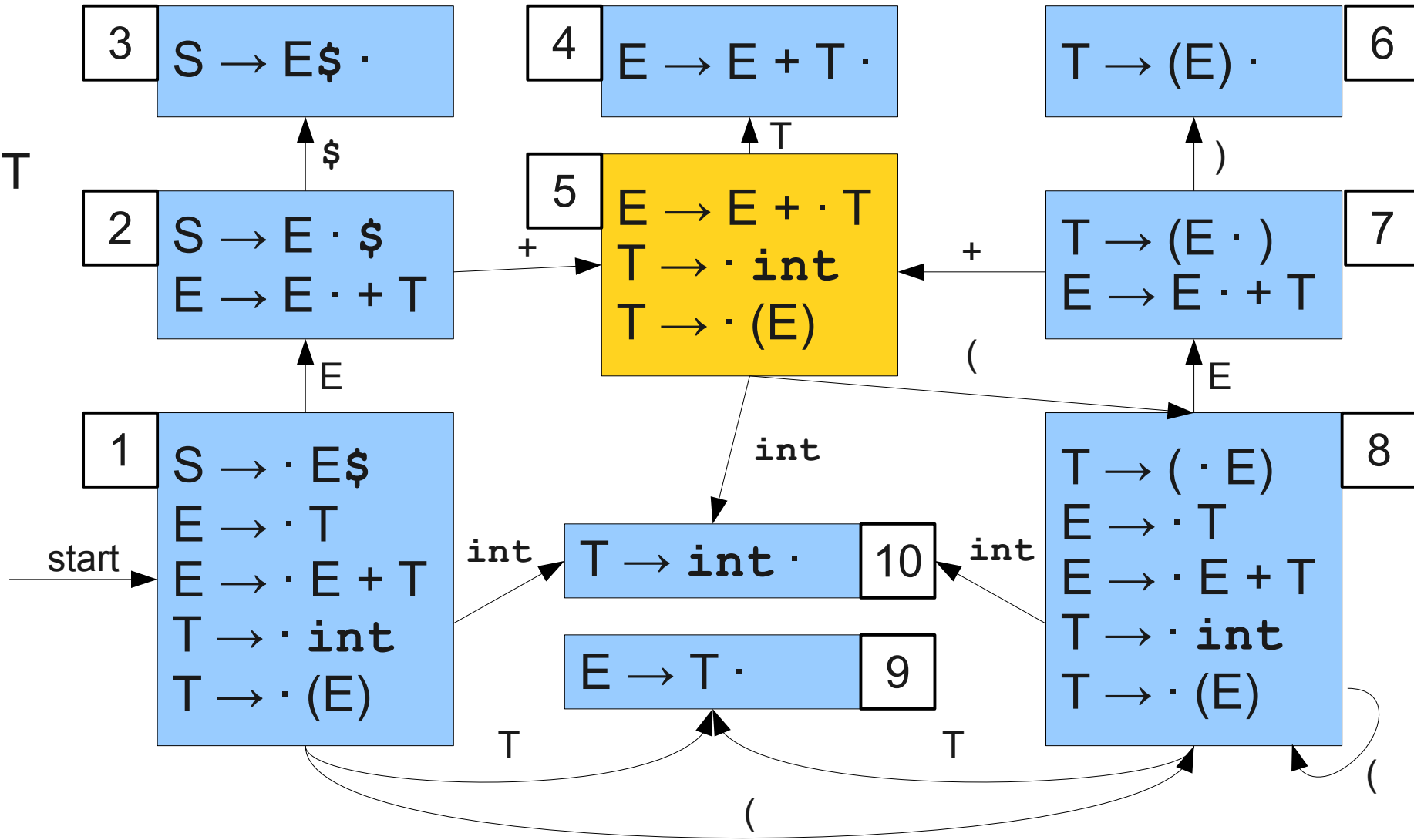
int
)
\$

1
2
5
8

7

LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



E + (E +

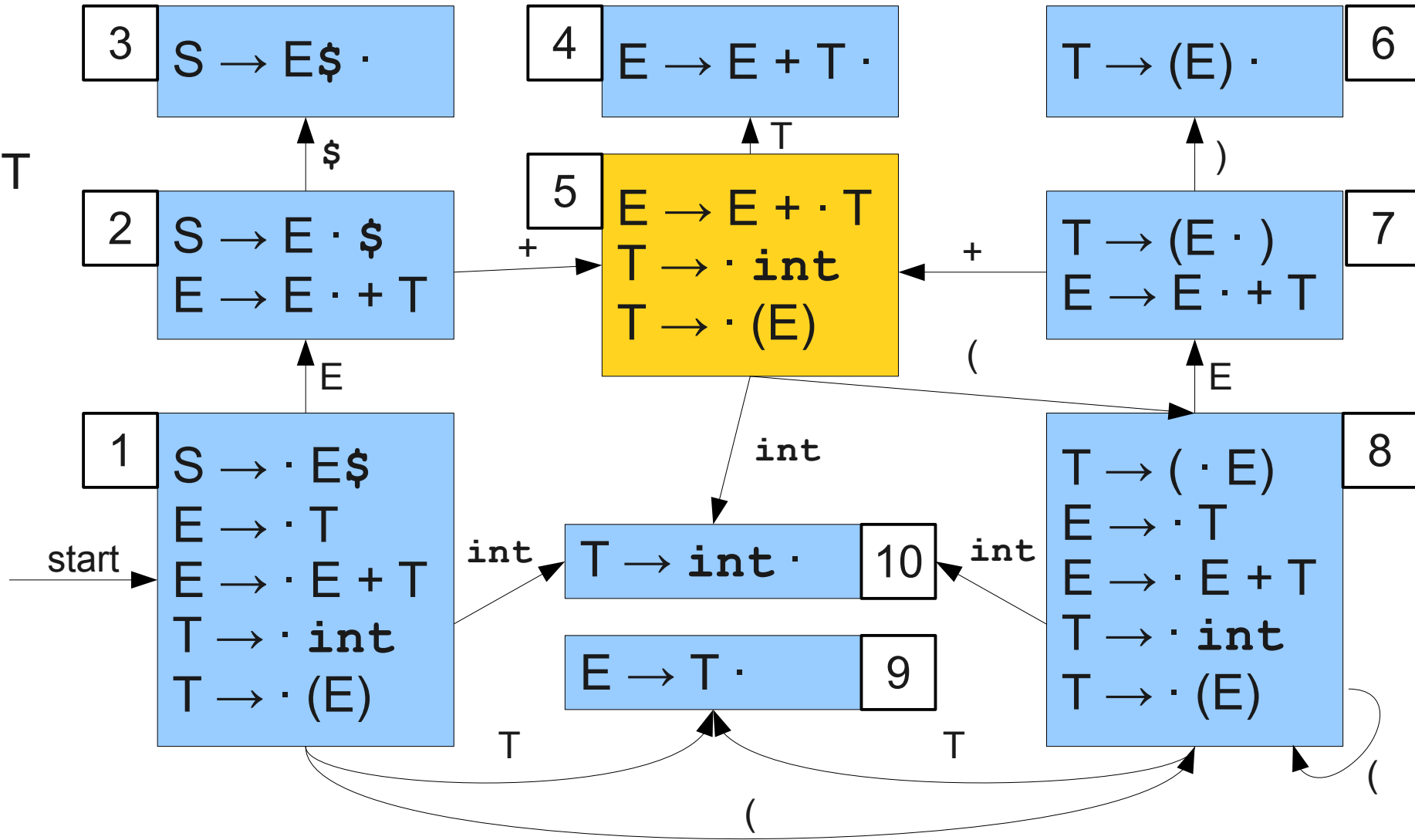
int) \$

1 2 5 8

7 5

LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



E + (E + int

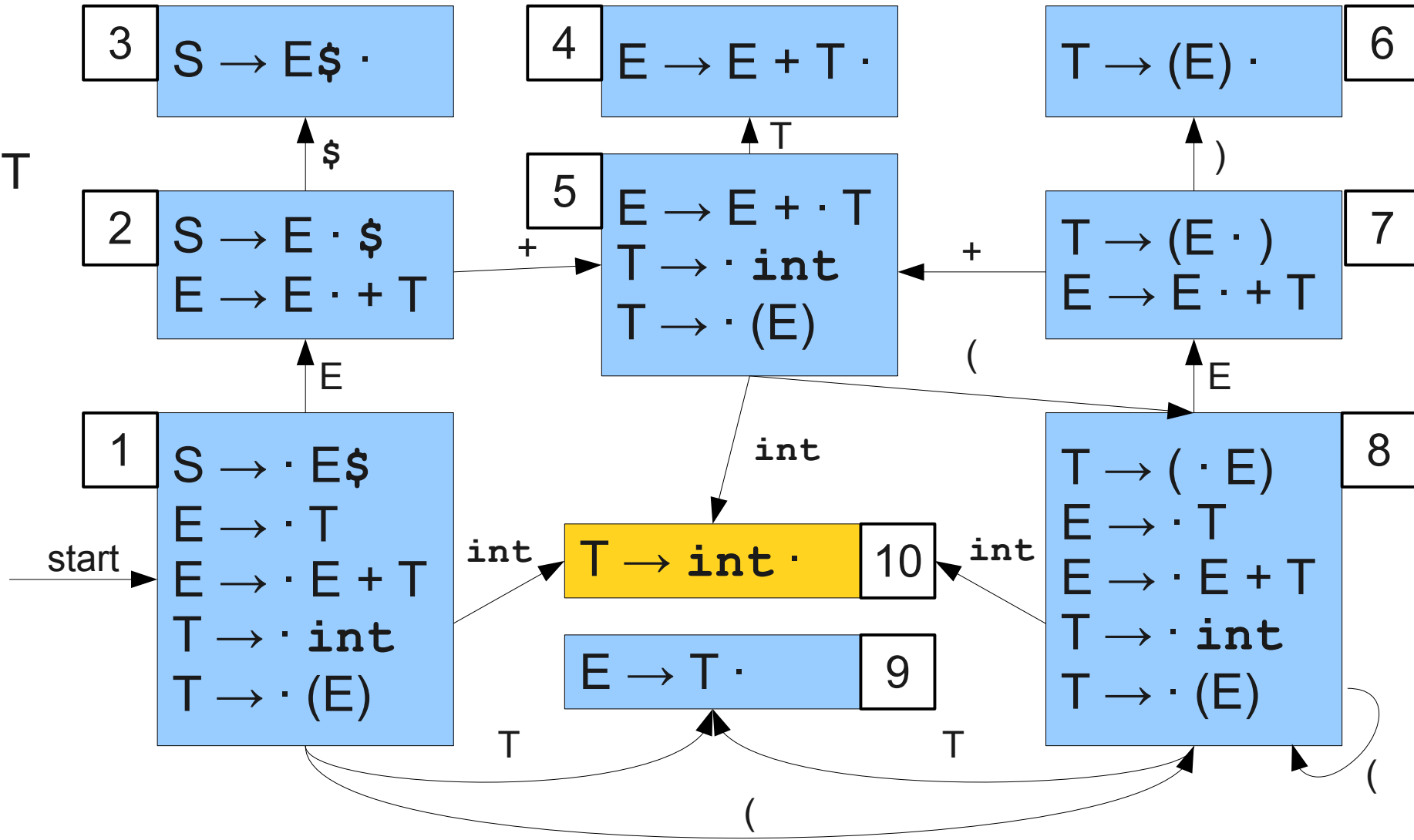
) \$

1 2 5 8

7 5

LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



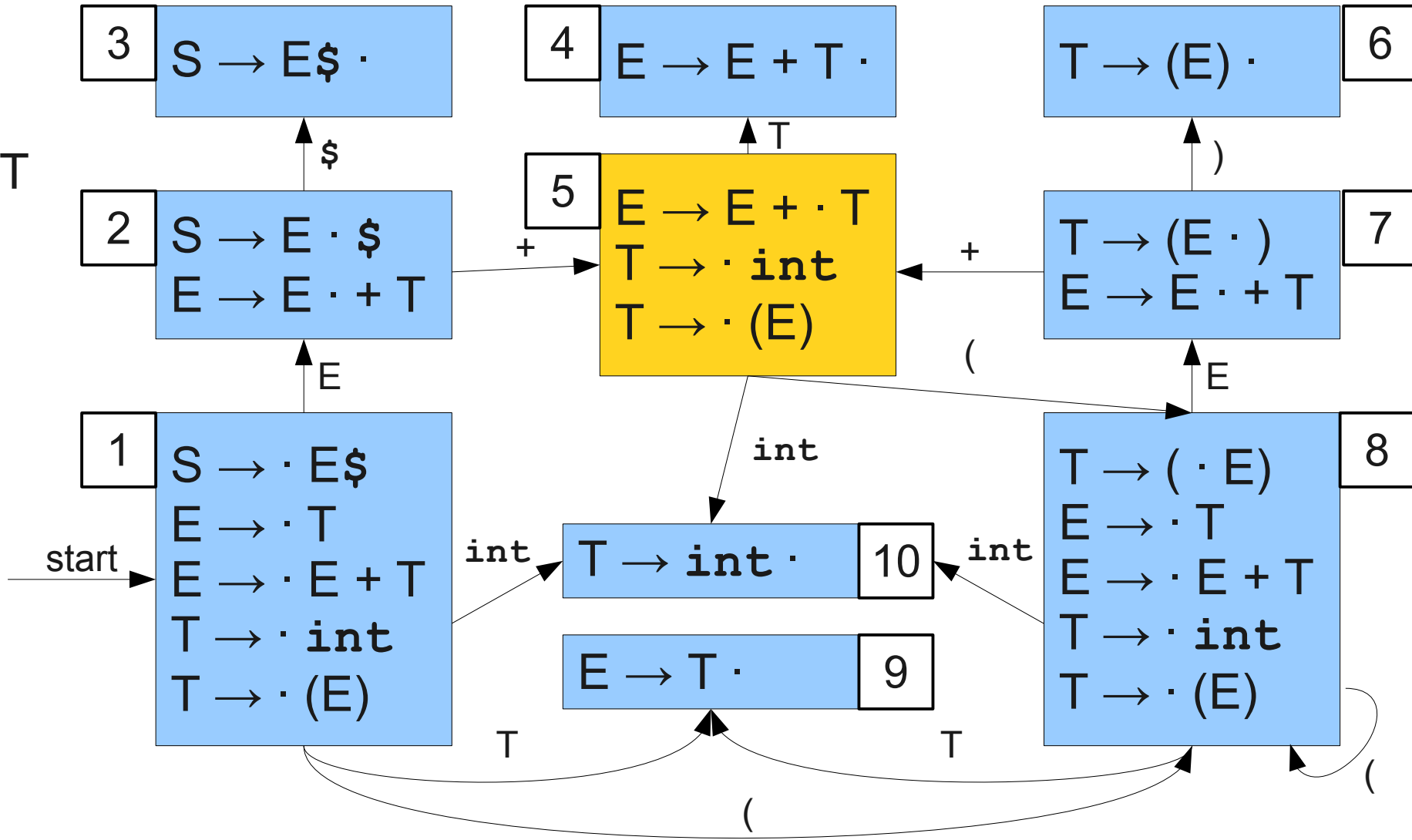
E
+
(
E
+
int
)
\$

1
2
5
8

7
5
10

LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



E + (E +) \$

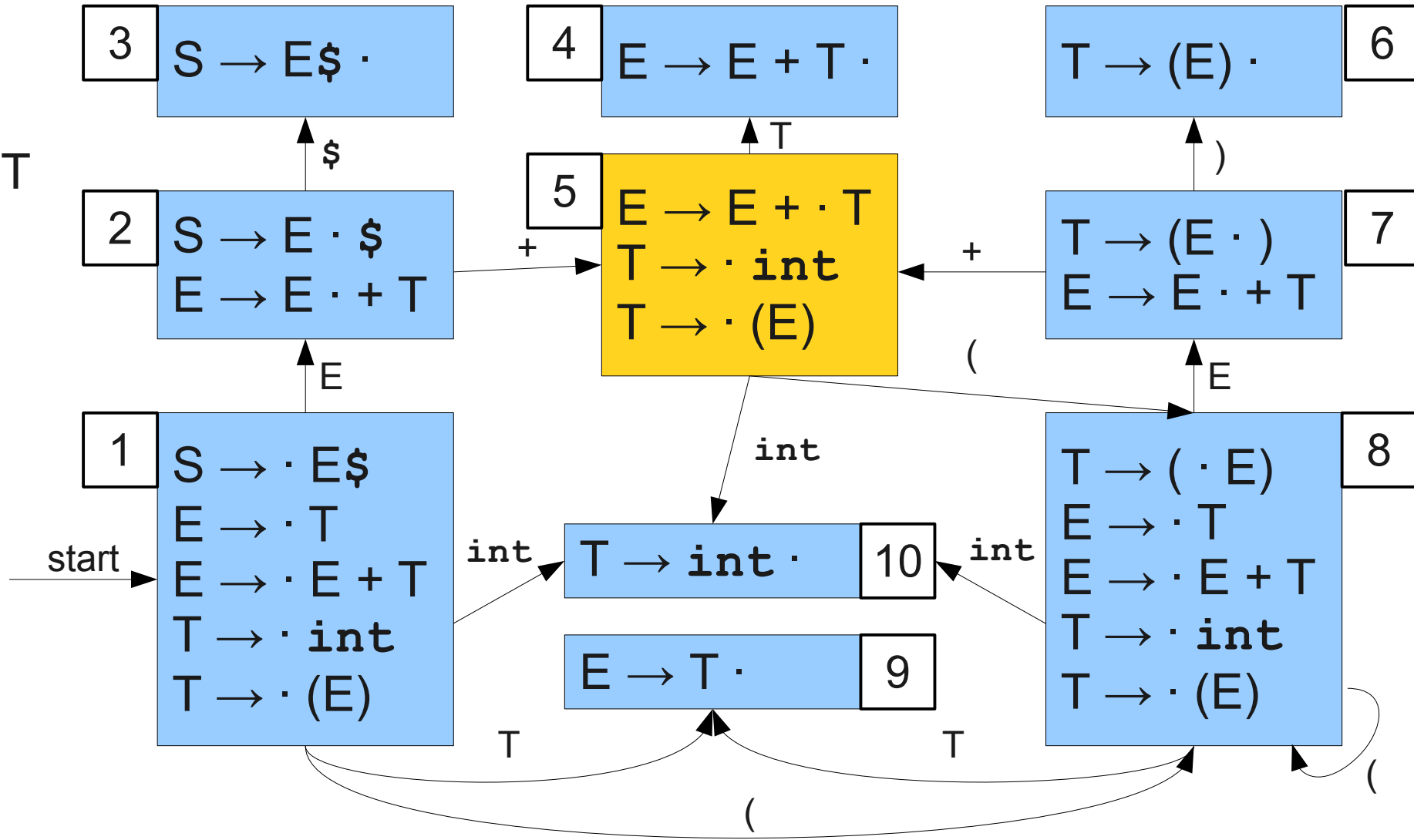
1 2 5 8

7 5

) \$

LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



E + (E + T

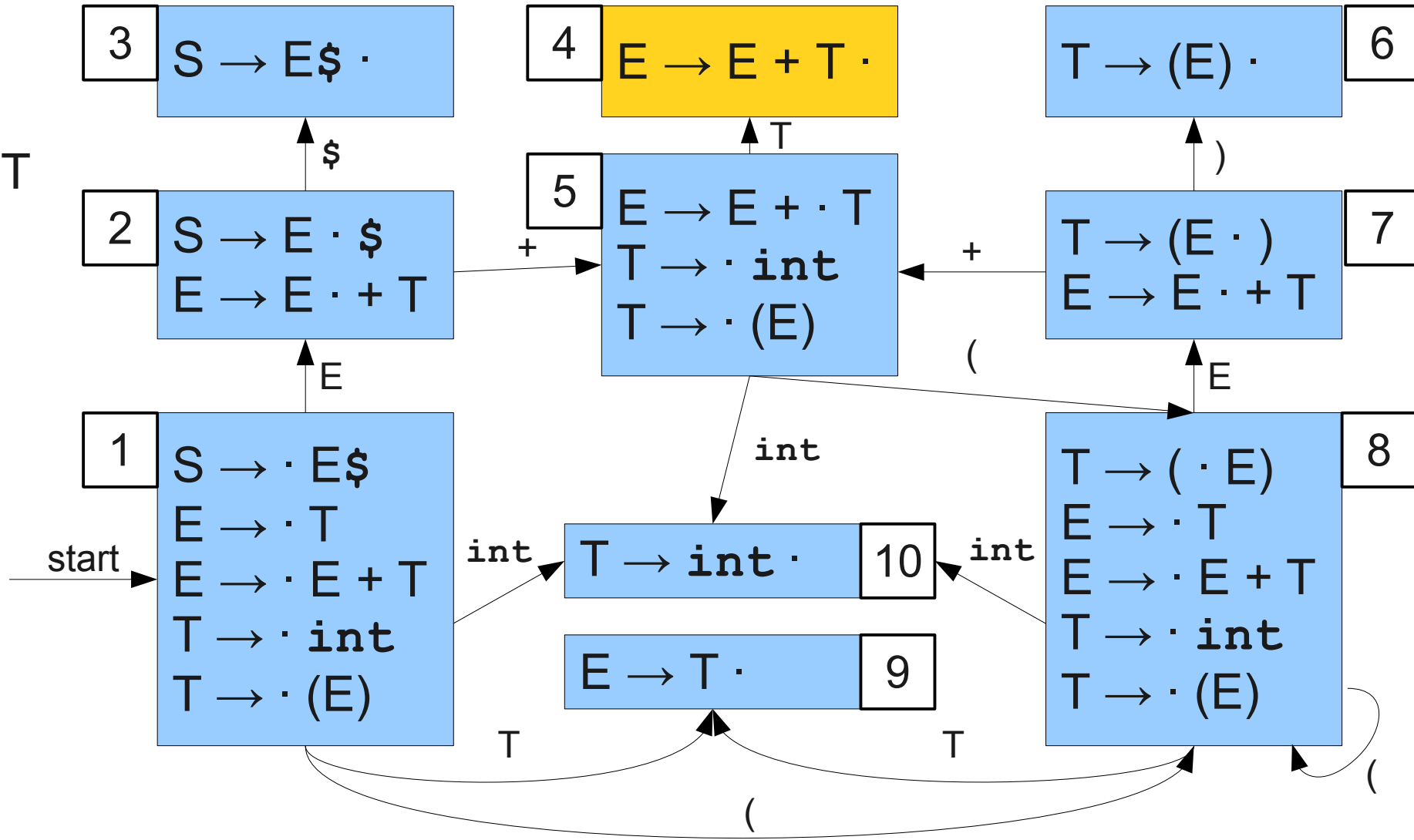
) \$

1 2 5 8

7 5

LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



E + (E + T

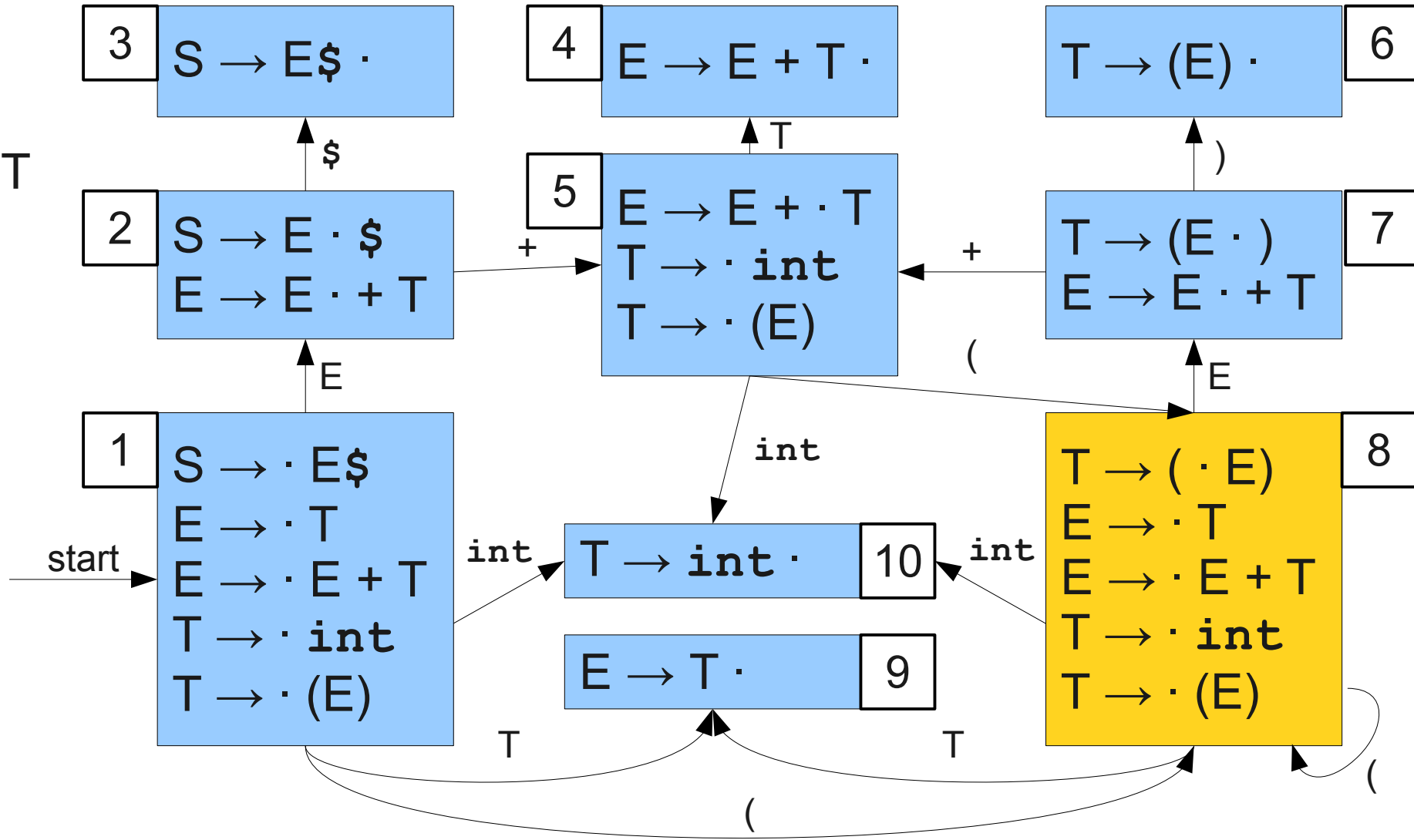
) \$

1 2 5 8

7 5 4

LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



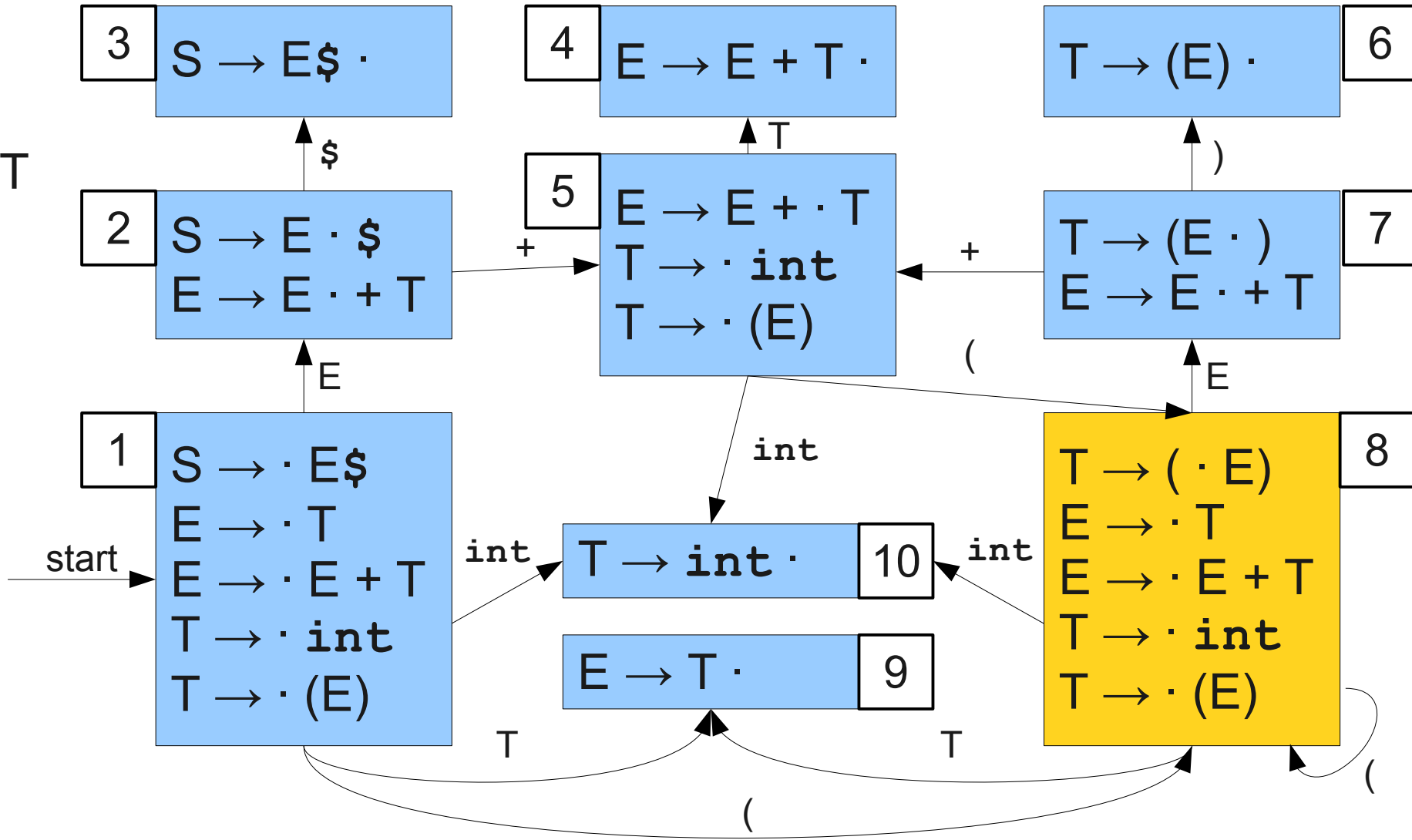
E + (

) \$

1 2 5 8

LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



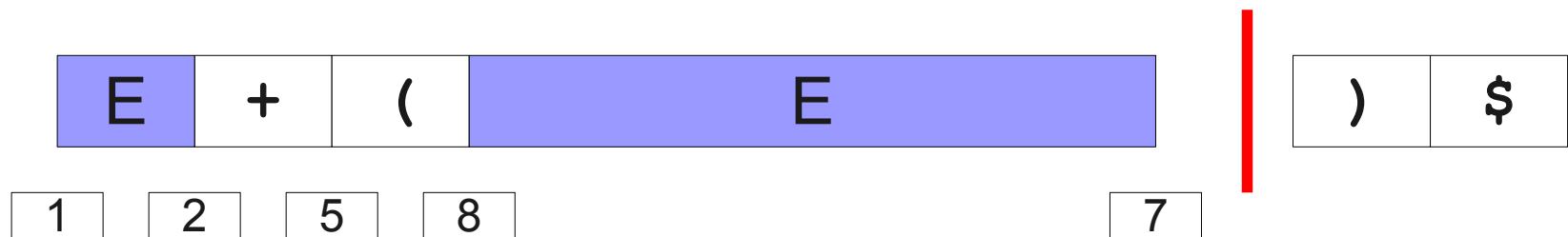
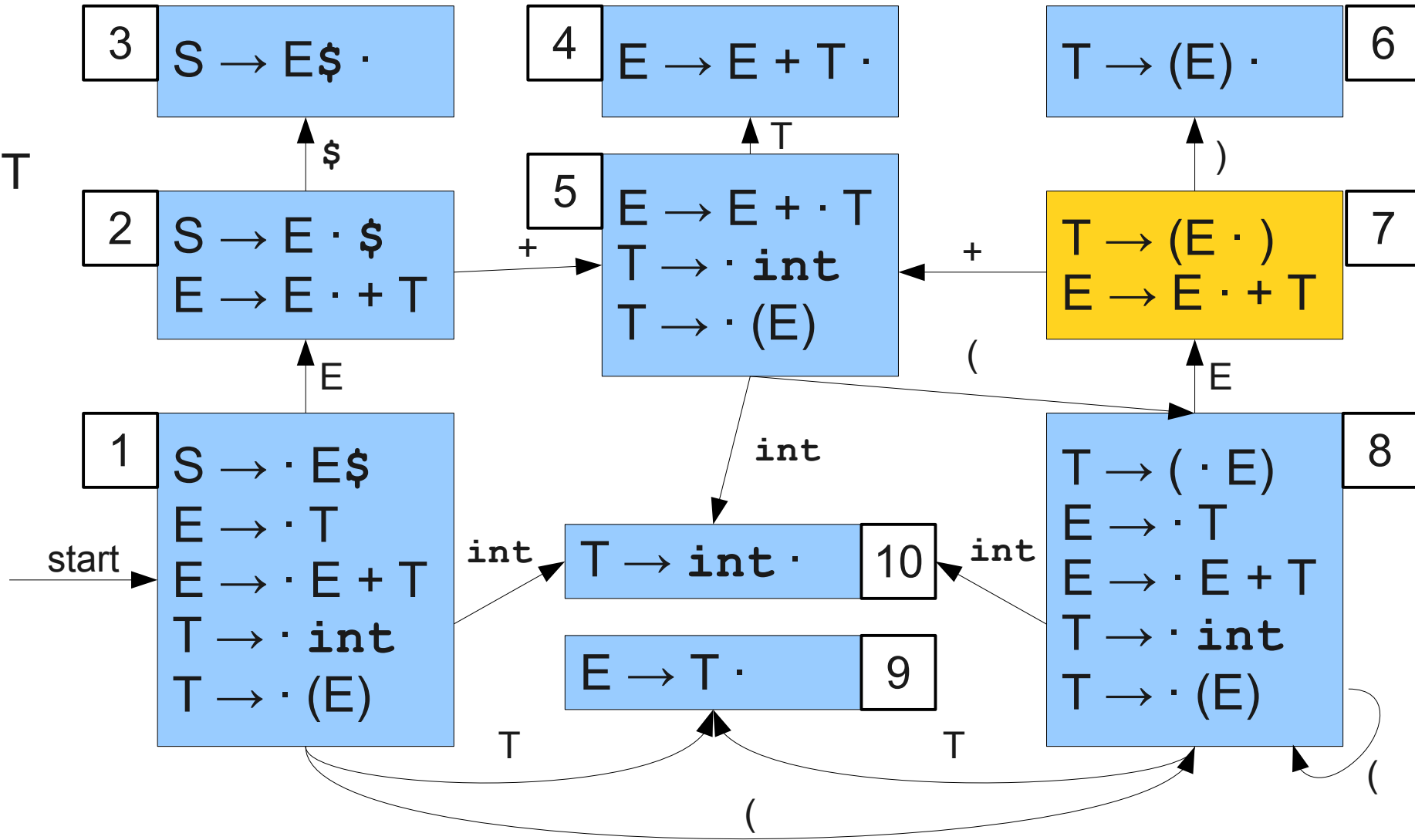
E + (E

) \$

1 2 5 8

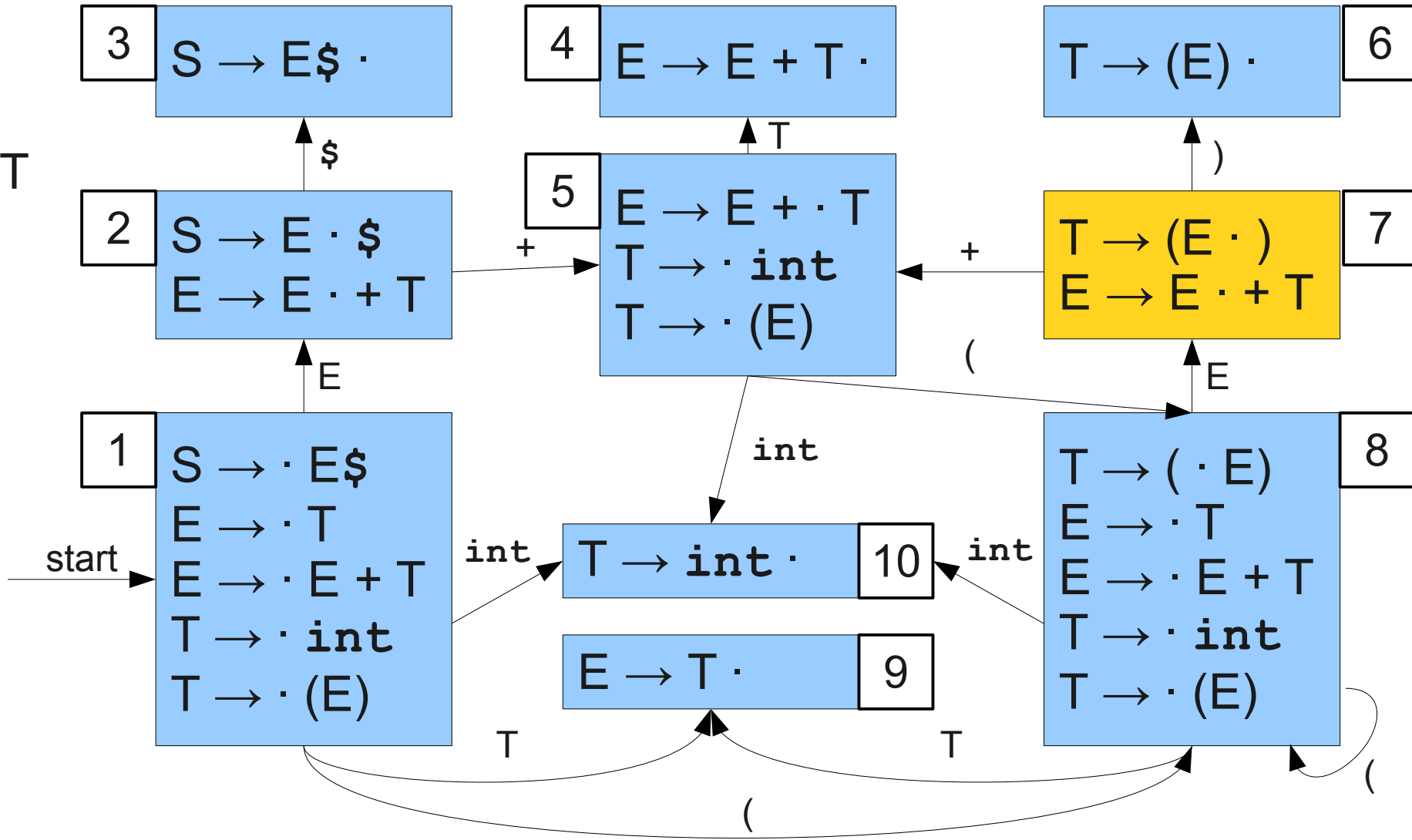
LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



E + (E E)

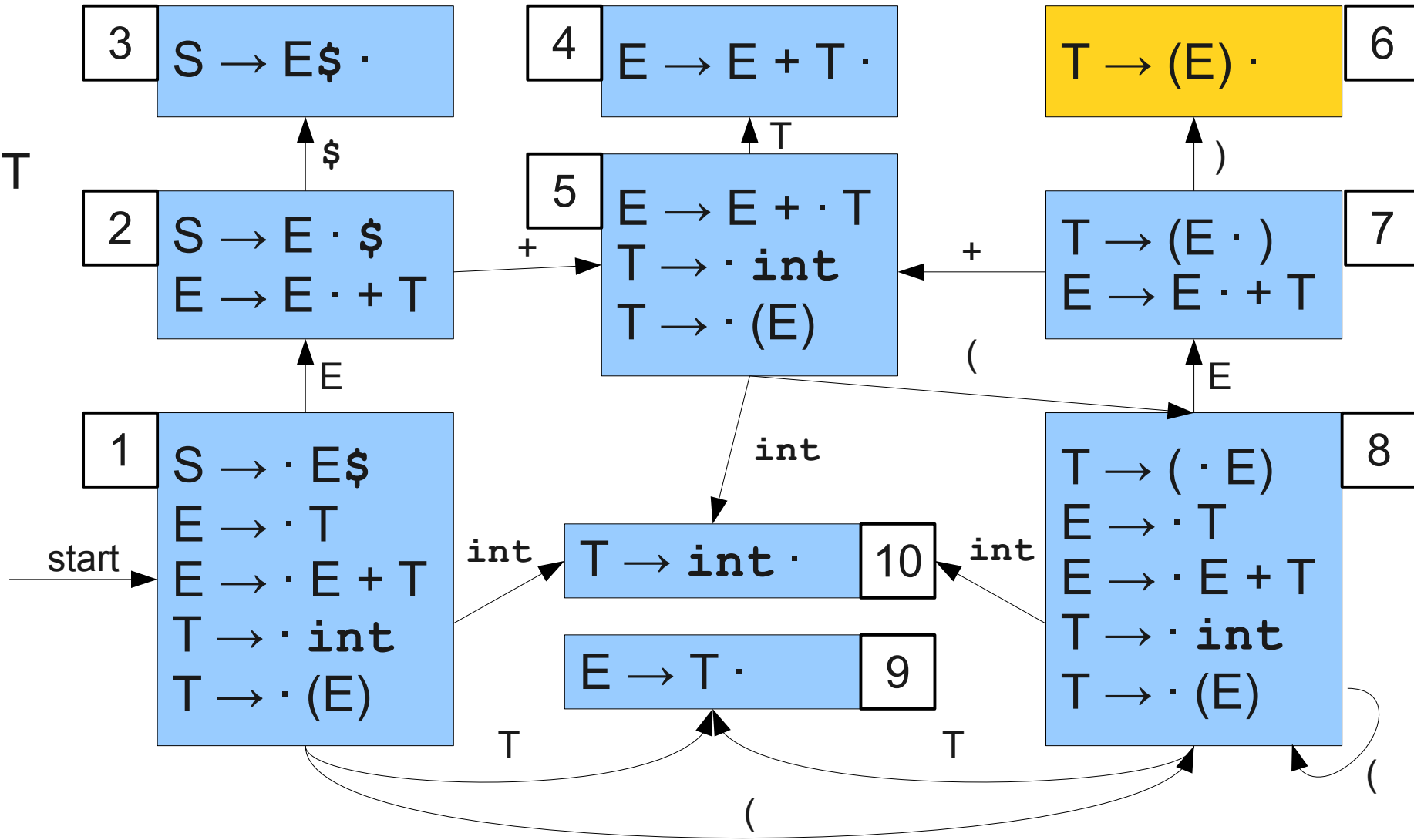
\$

1 2 5 8

7

LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



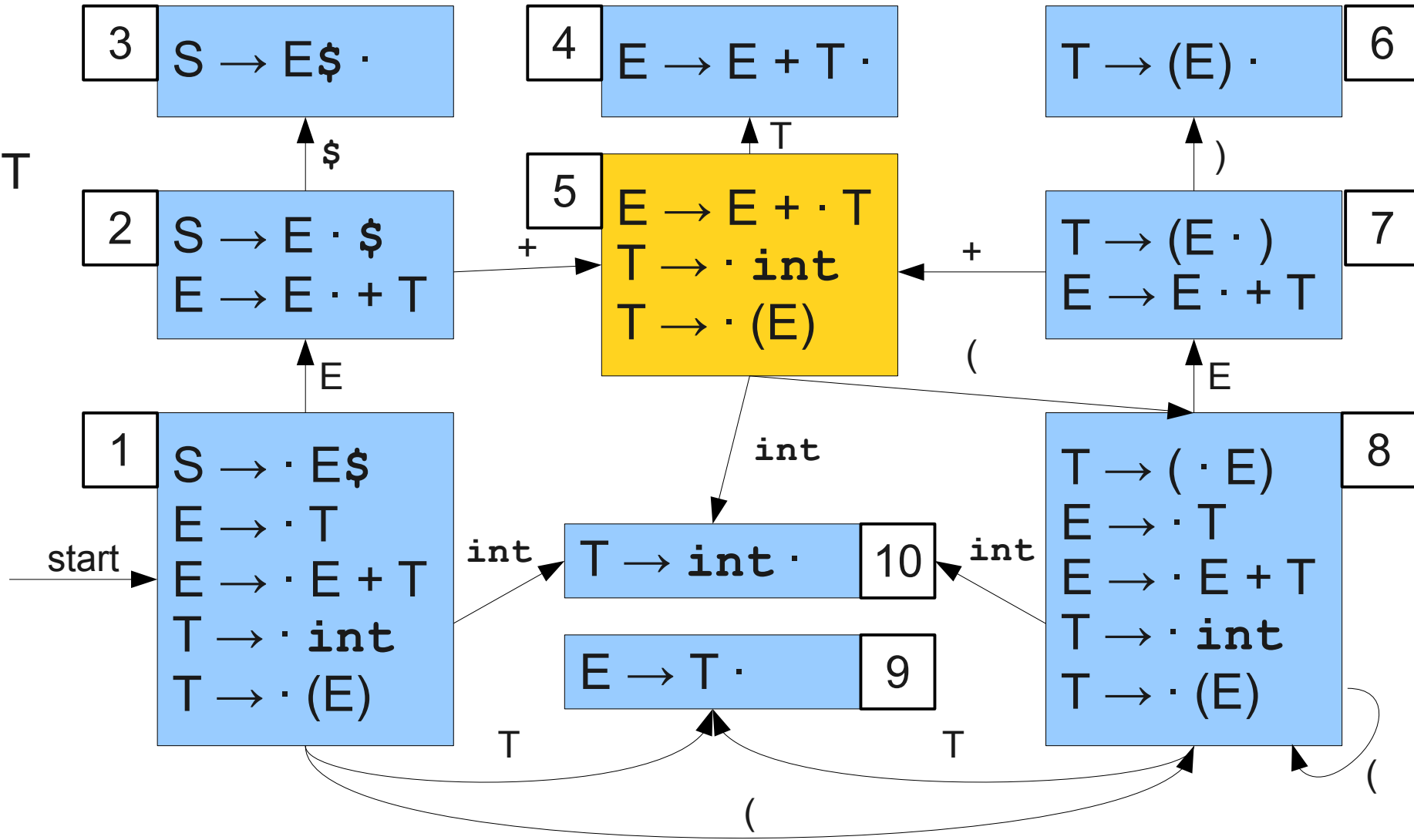
E	+	(E)	\$
---	---	---	---	---	----

1
2
5
8

7
6

LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



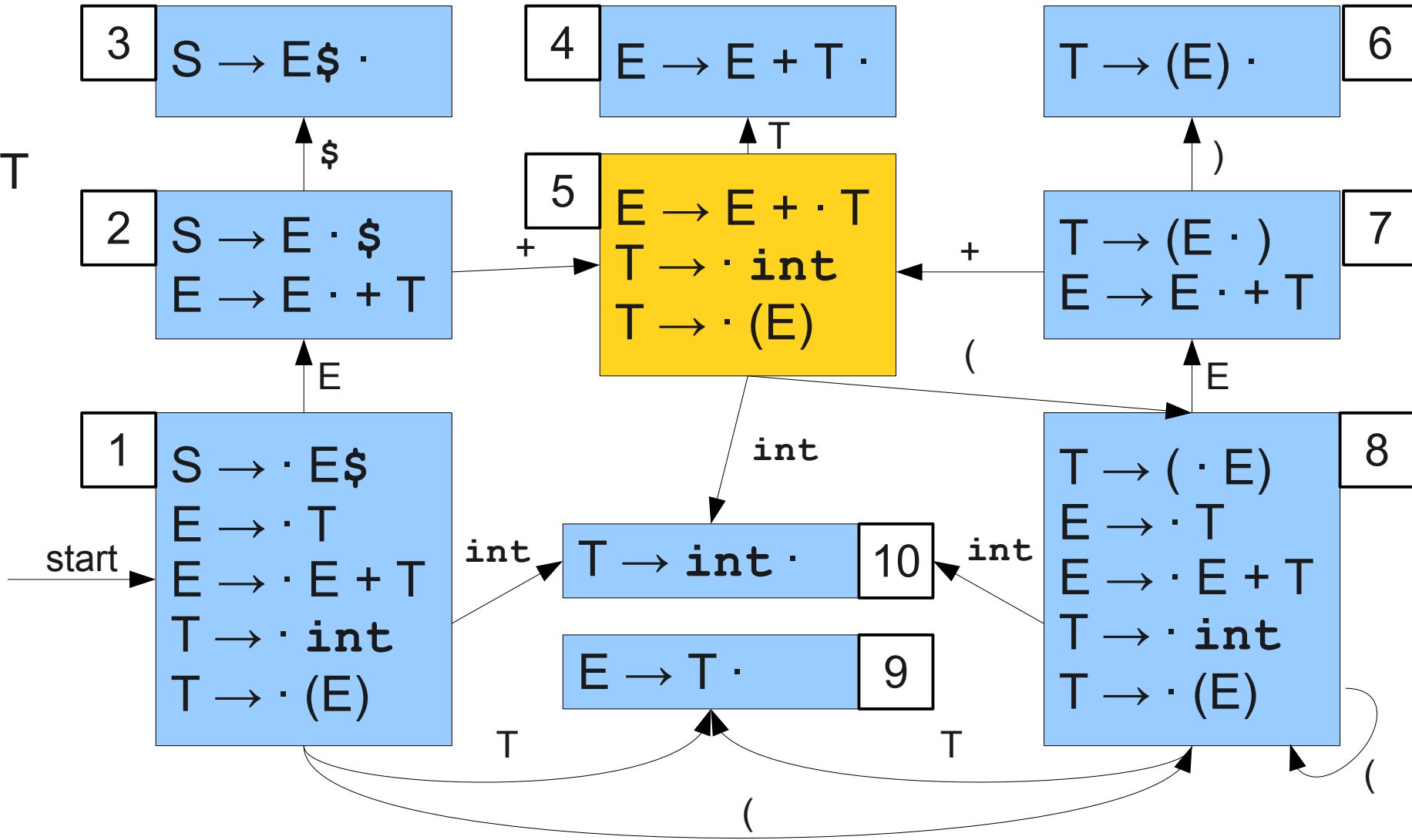
E +

| \$

1 2 5

LR(0) Parsing

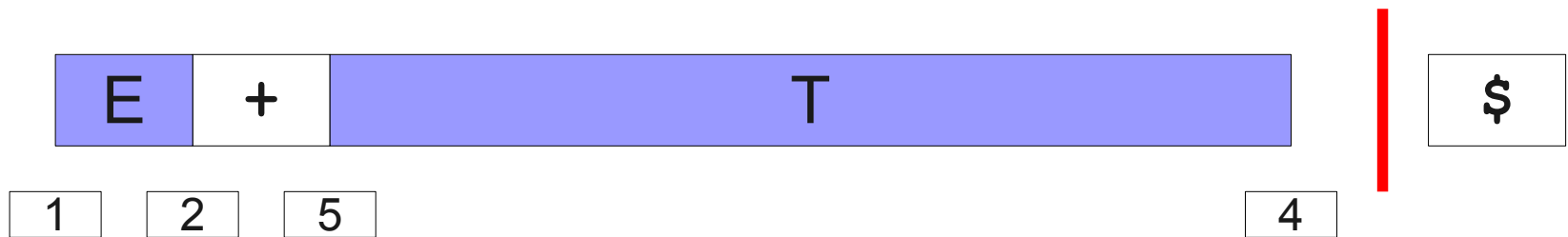
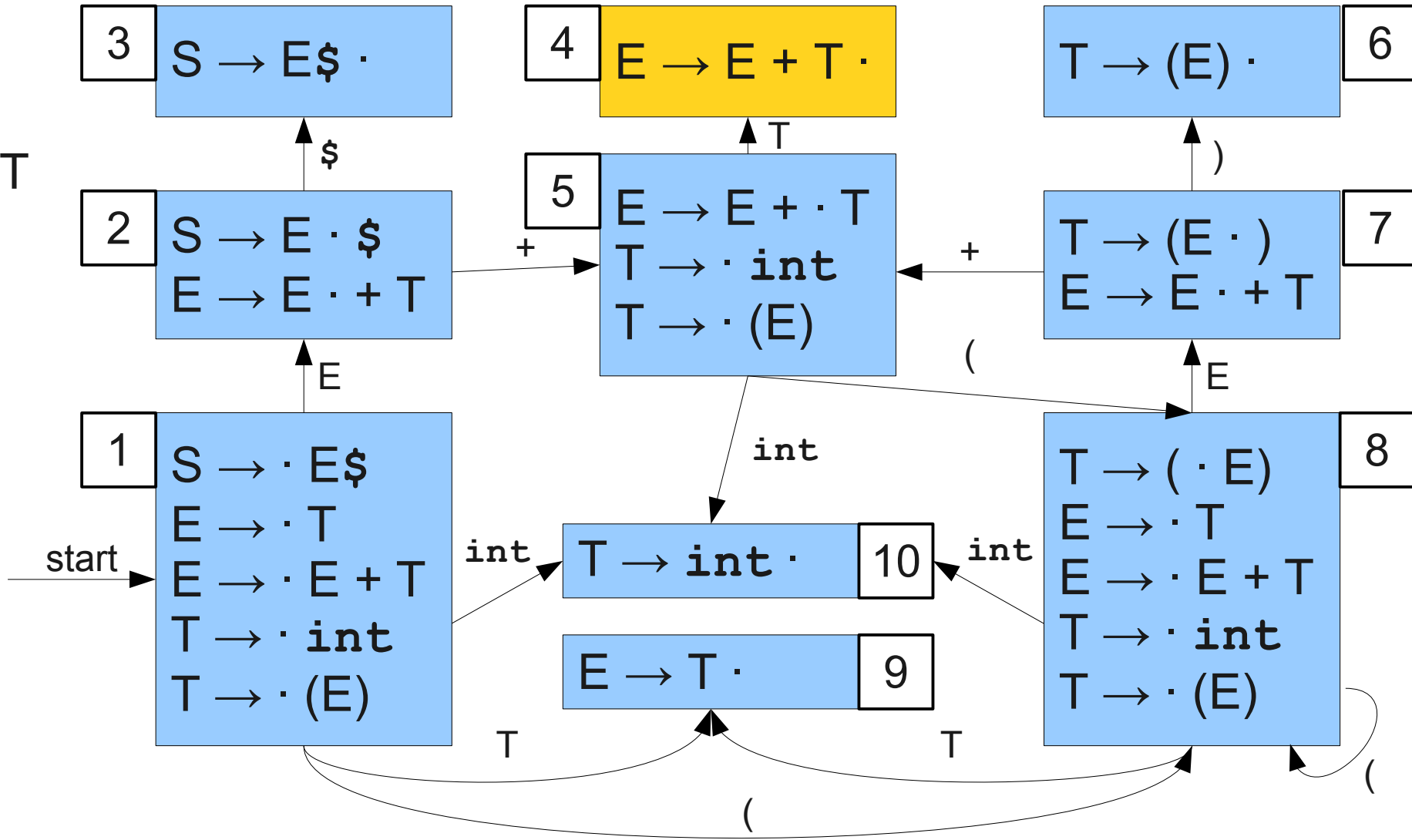
$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



1 2 5

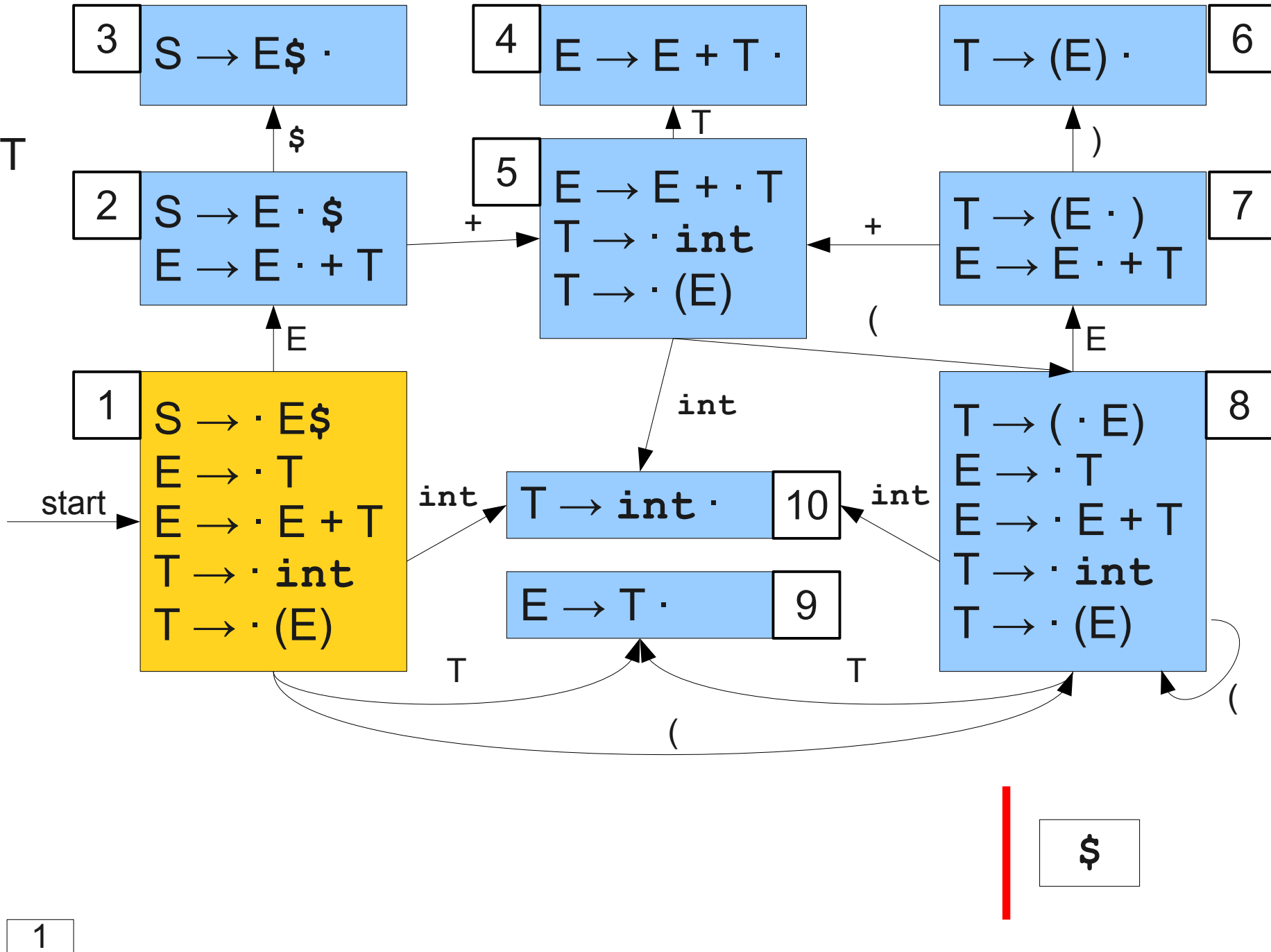
LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



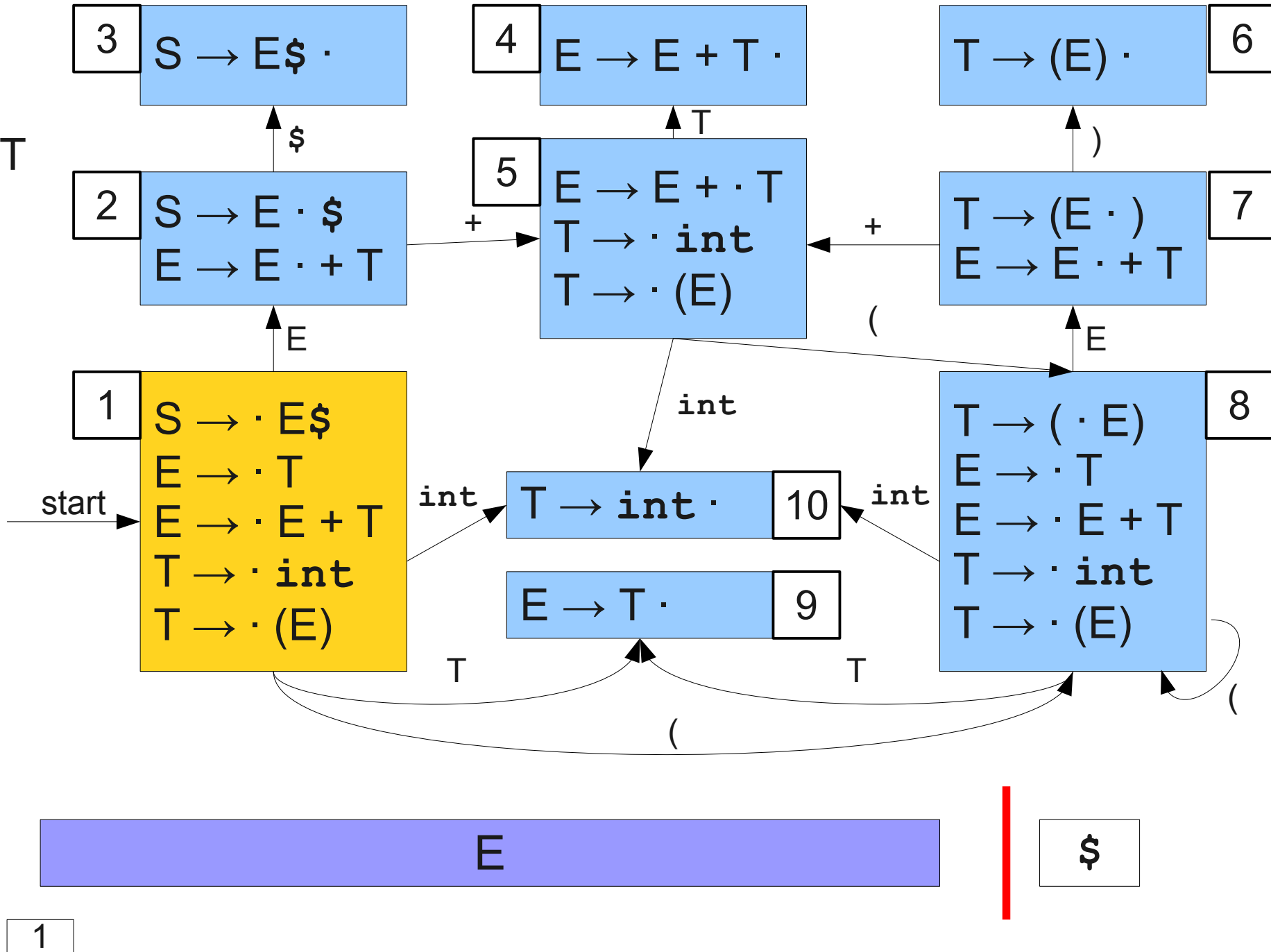
LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



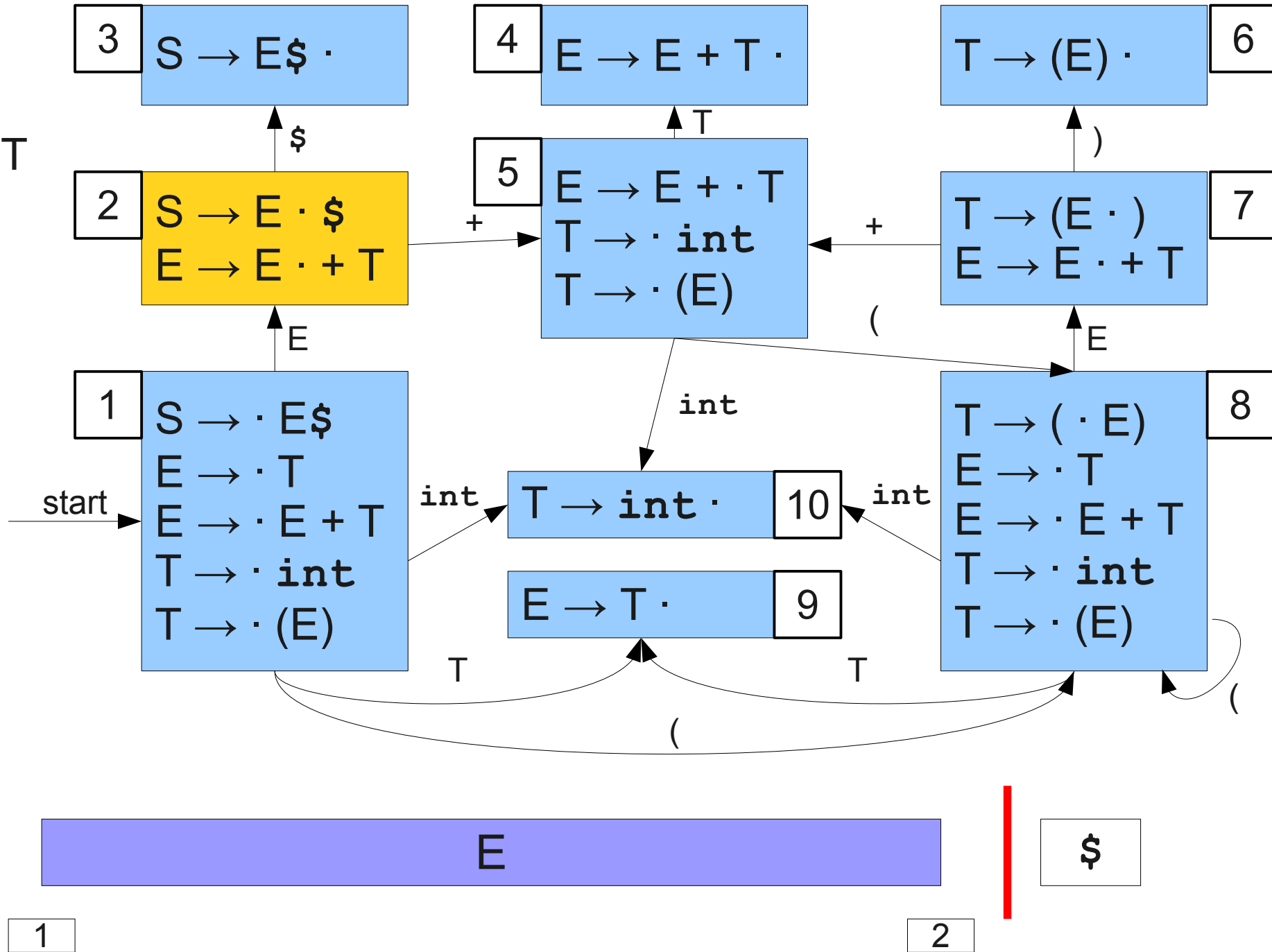
LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



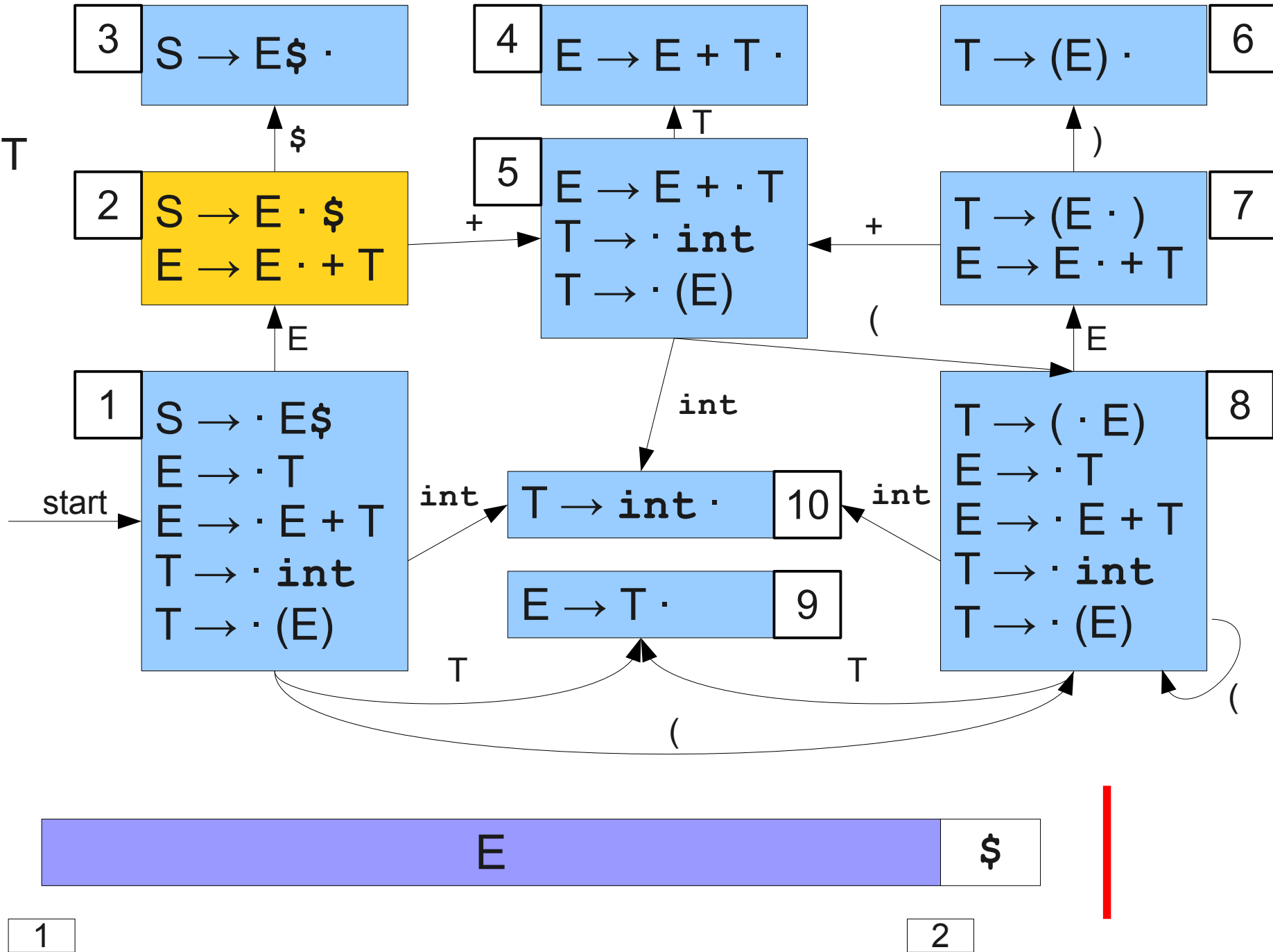
LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



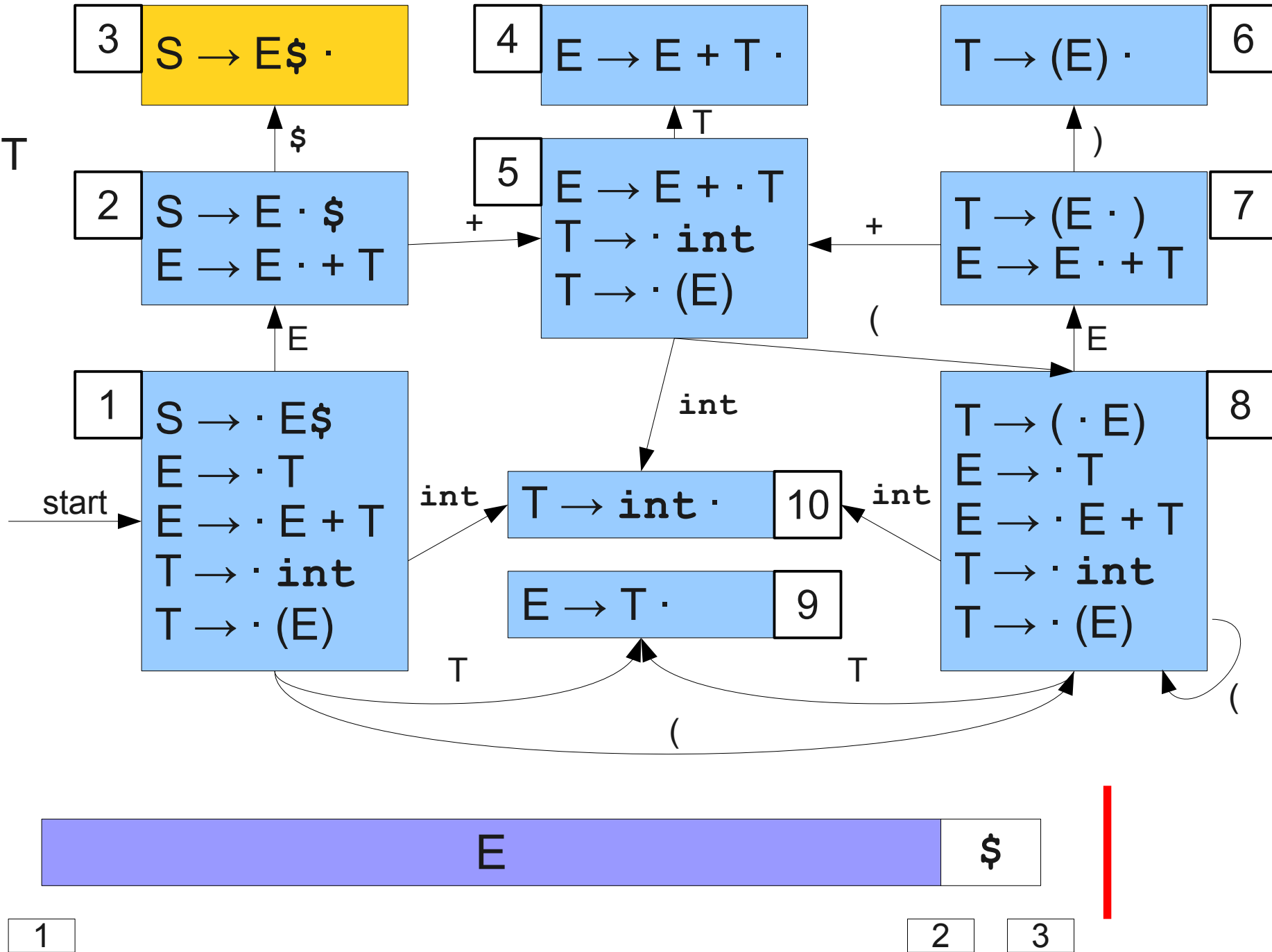
LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



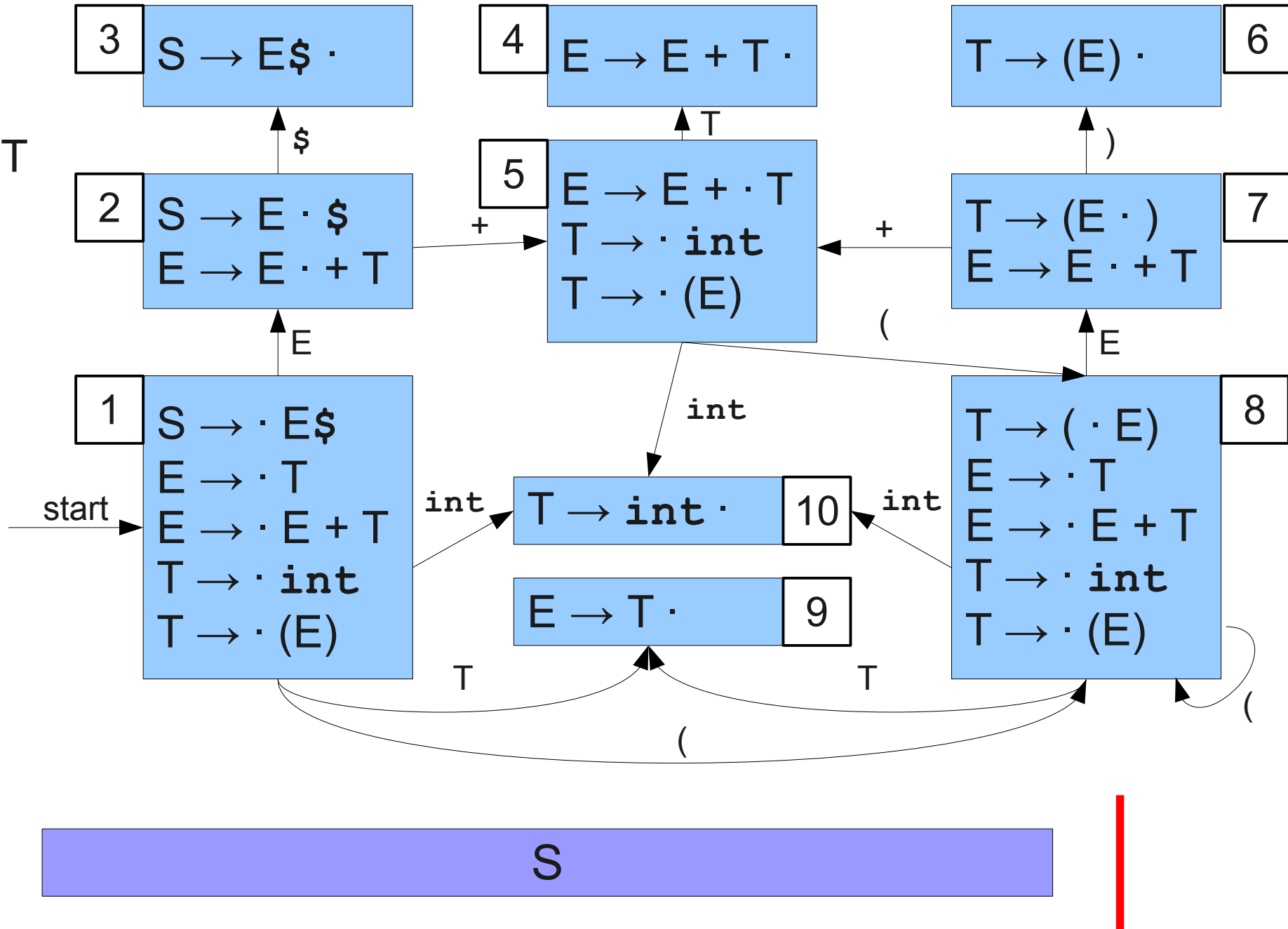
LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



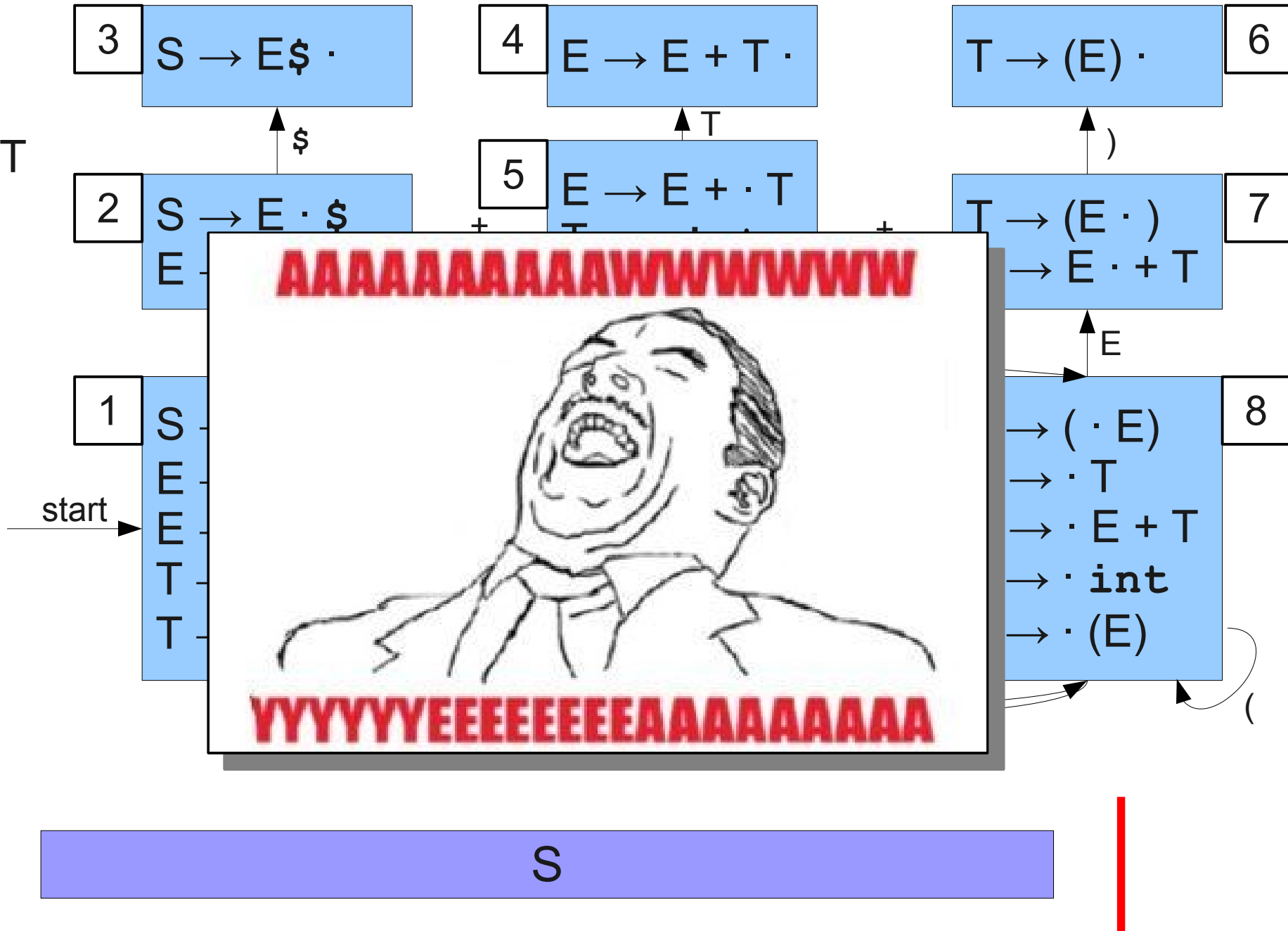
LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



LR(0) Parsing

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



Representing the Automaton

- LR(0) parsers are usually represented via an **action** and **goto** table.
- The **action** table maps each state to an action:
 - **shift**, which shifts the next terminal, and
 - **reduce N** , which performs reduction **N** .
 - Any state of the form $A \rightarrow v \cdot$ 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.

LR(0) Tables

	int	+	()	\$	S	E	T
1	10		8				2	9
2		5			3			
3								
4								
5	10		8					4
6								
7		5		6				
8	10		8				7	9
9								
10								

Action
Shift
Shift
Accept
Reduce $E \rightarrow E + T$
Shift
Reduce $T \rightarrow (E)$
Shift
Shift
Reduce $T \rightarrow \text{int}$
Reduce $E \rightarrow T$

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** $A \rightarrow v$:
 - Remove $|v|$ symbols from the top of the stack.
 - Let **top-state** be the state on top of the stack.
 - Push (A , **goto[A, top-state]**) atop the stack.
 - Otherwise, report an error.

The Limits of LR(0)

A Non-LR(0) Grammar

$S \rightarrow E\$$

$E \rightarrow T$

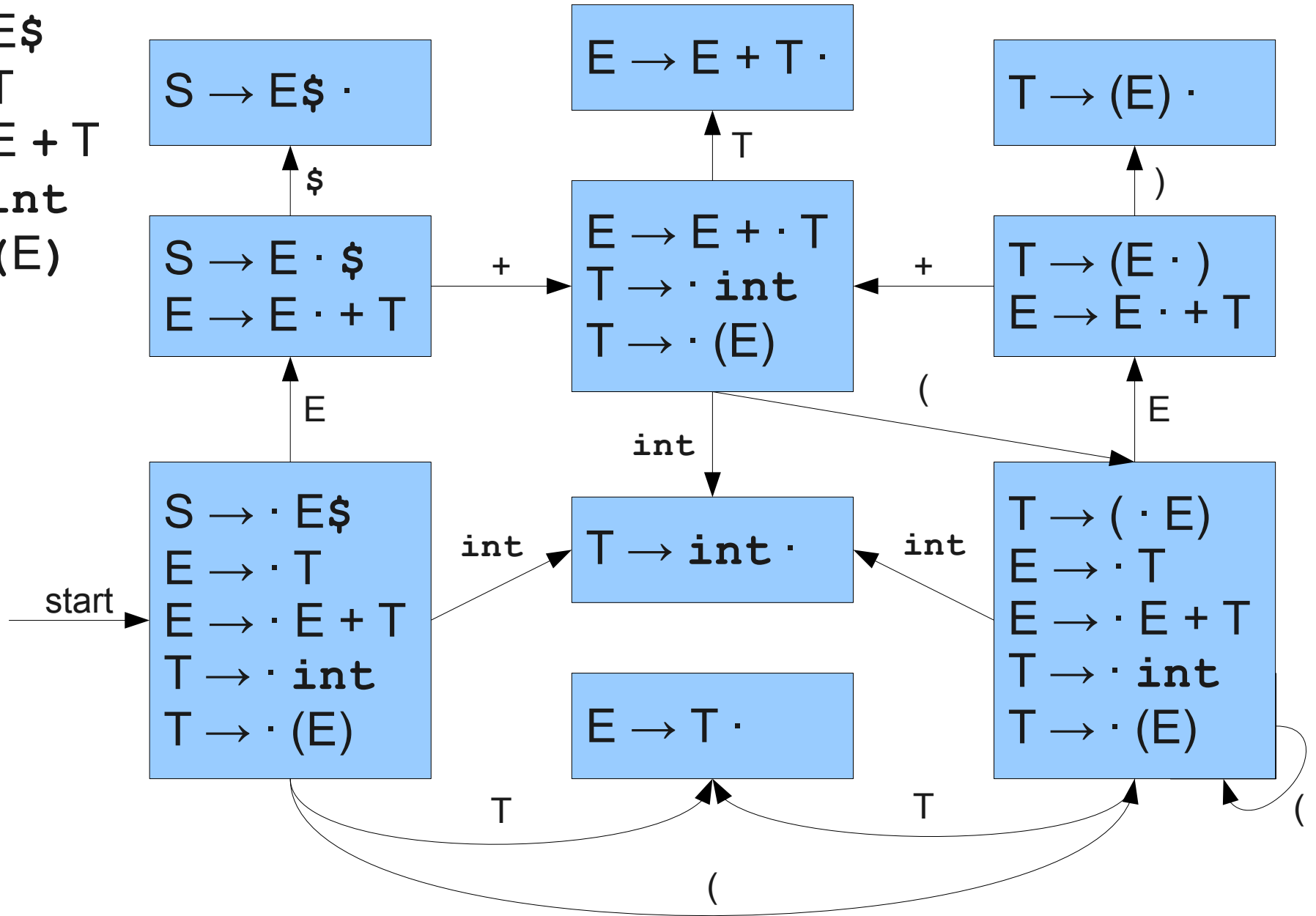
$E \rightarrow E + T$

$T \rightarrow \mathbf{int}$

$T \rightarrow (E)$

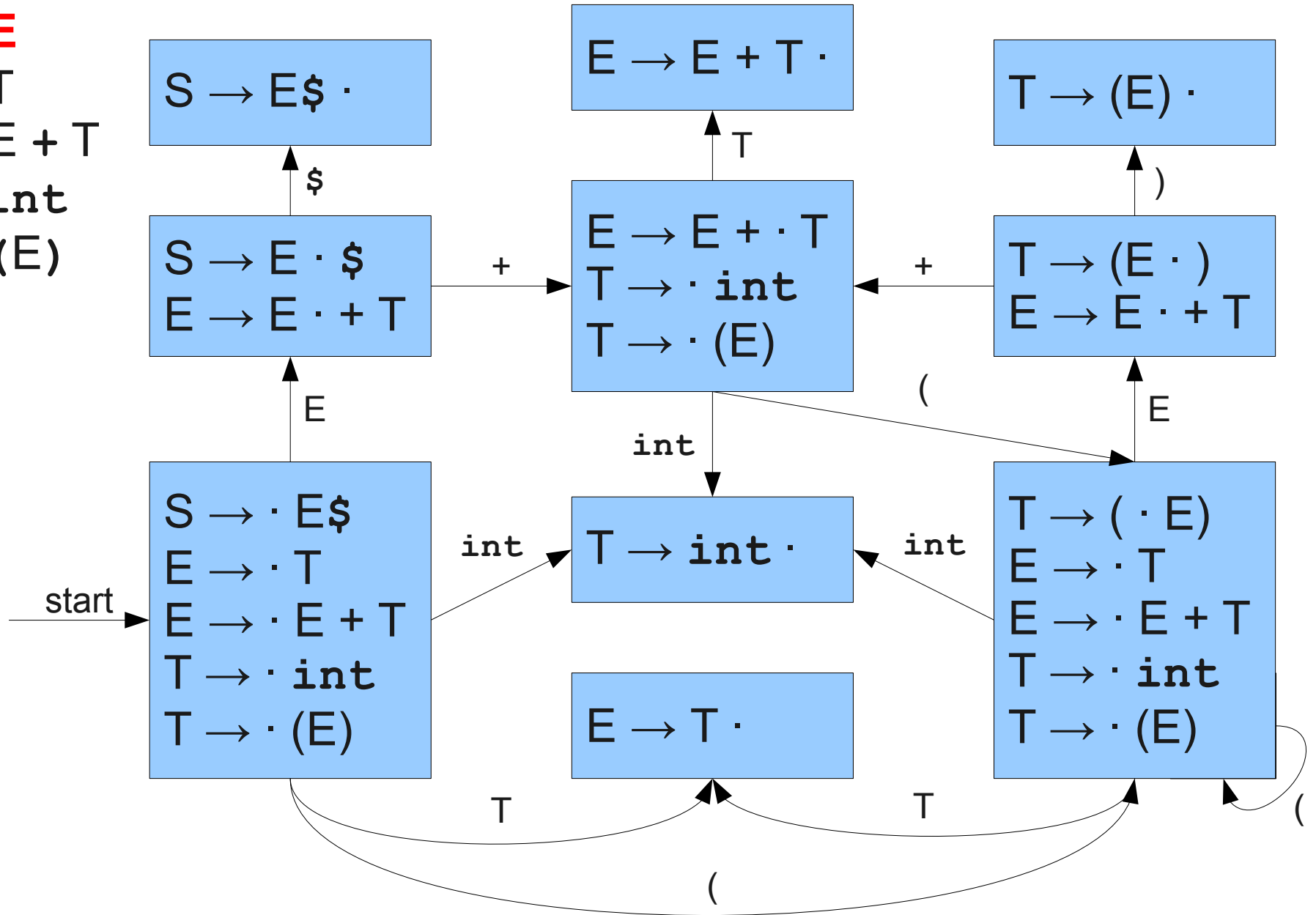
A Non-LR(0) Grammar

$S \rightarrow E\$$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



A Non-LR(0) Grammar

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



A Non-LR(0) Grammar

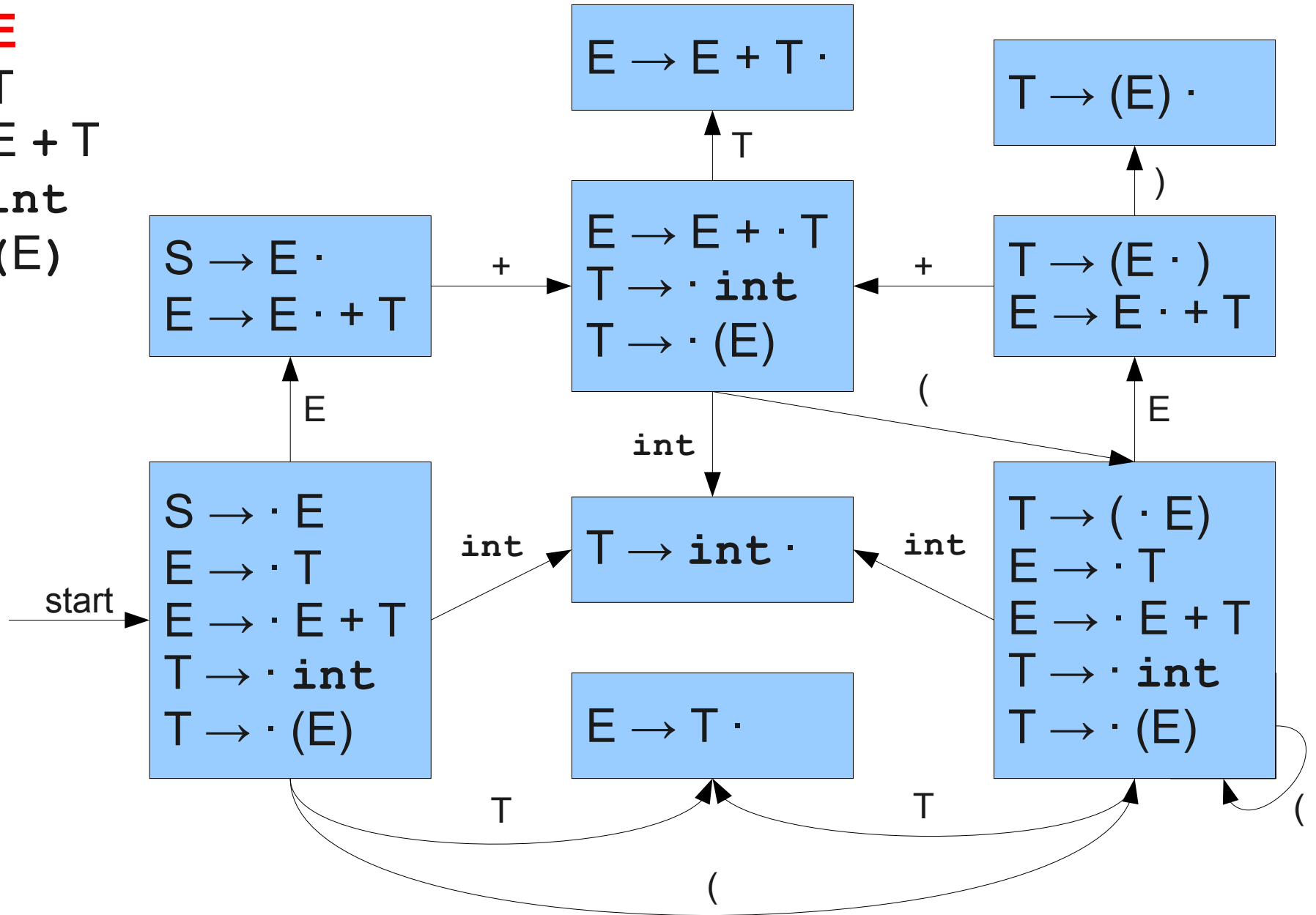
S → **E**

E → T

E → E + T

T → int

T → (E)



A Non-LR(0) Grammar

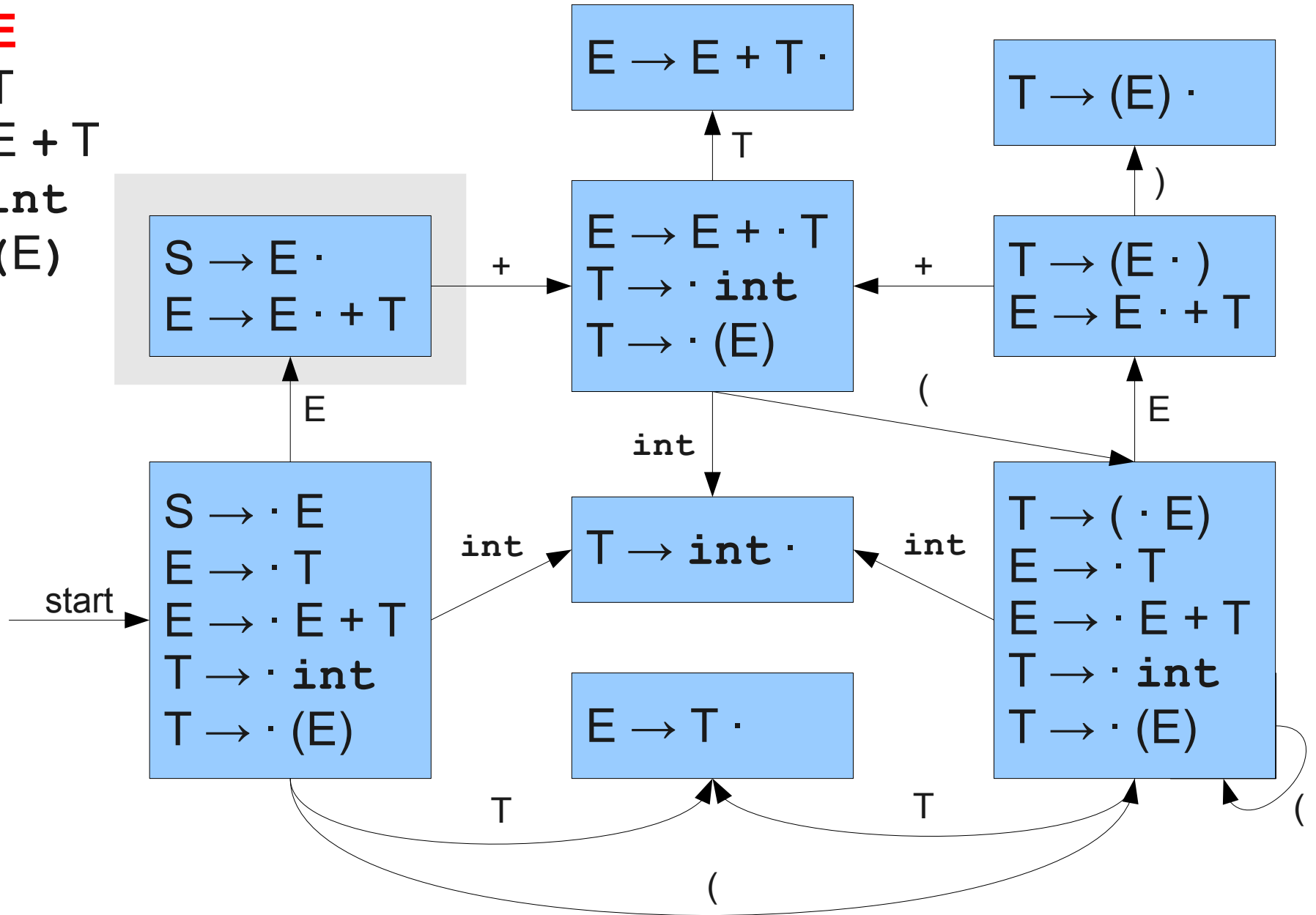
S → **E**

E → T

E → E + T

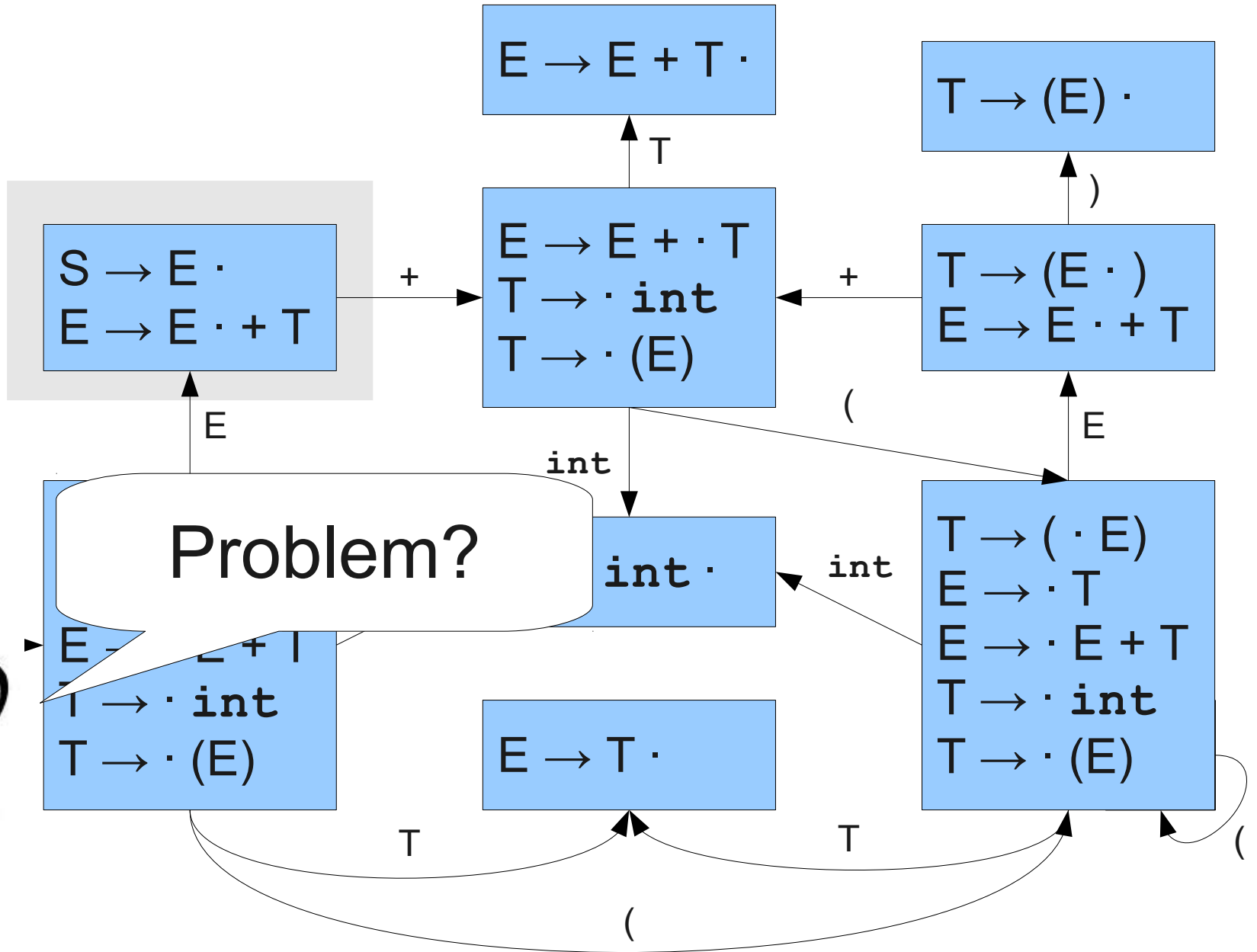
T → int

T → (E)



A Non-LR(0) Grammar

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

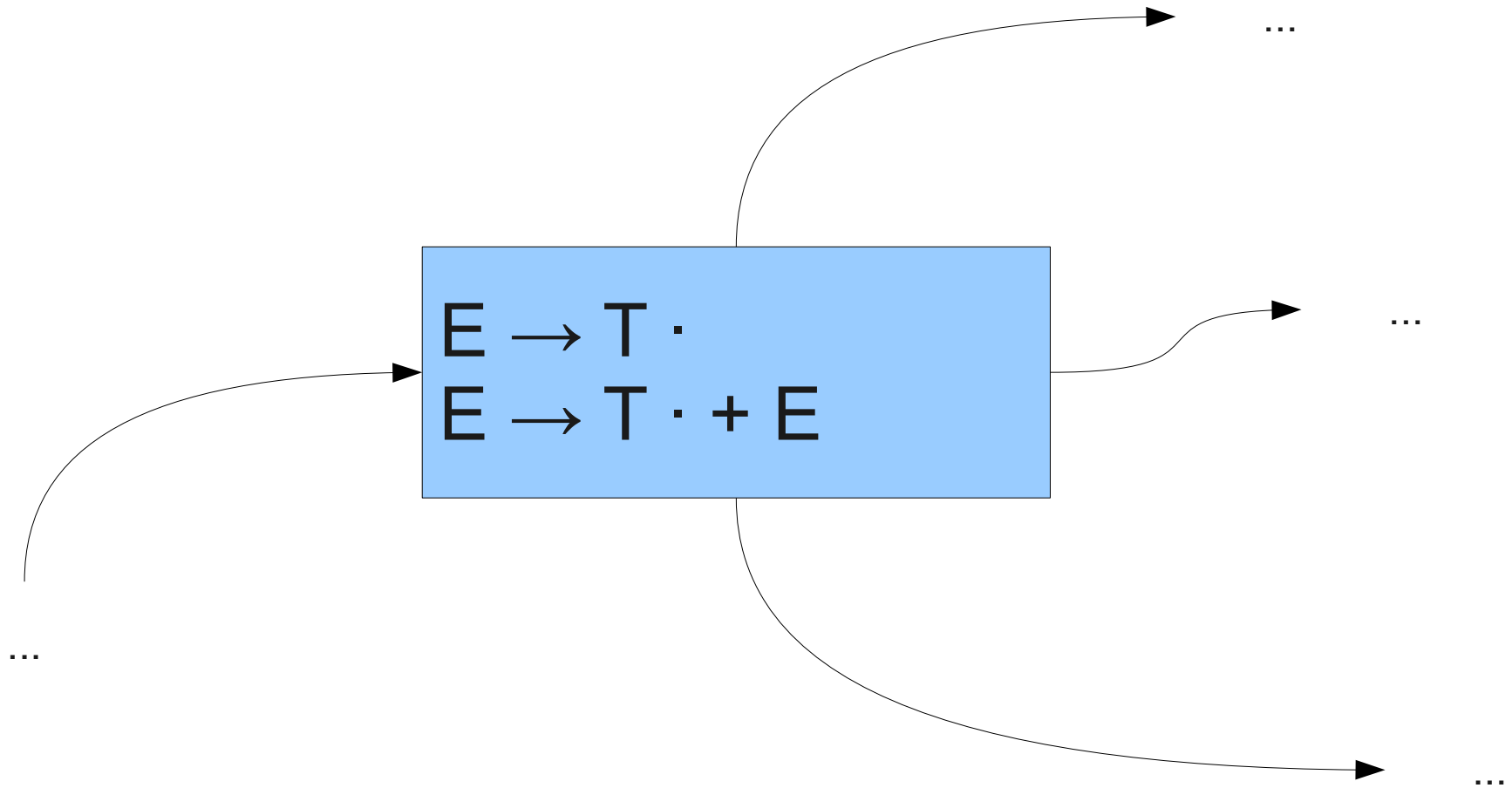


LR Conflicts

- 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).
- Can you have a **shift/shift conflict**?

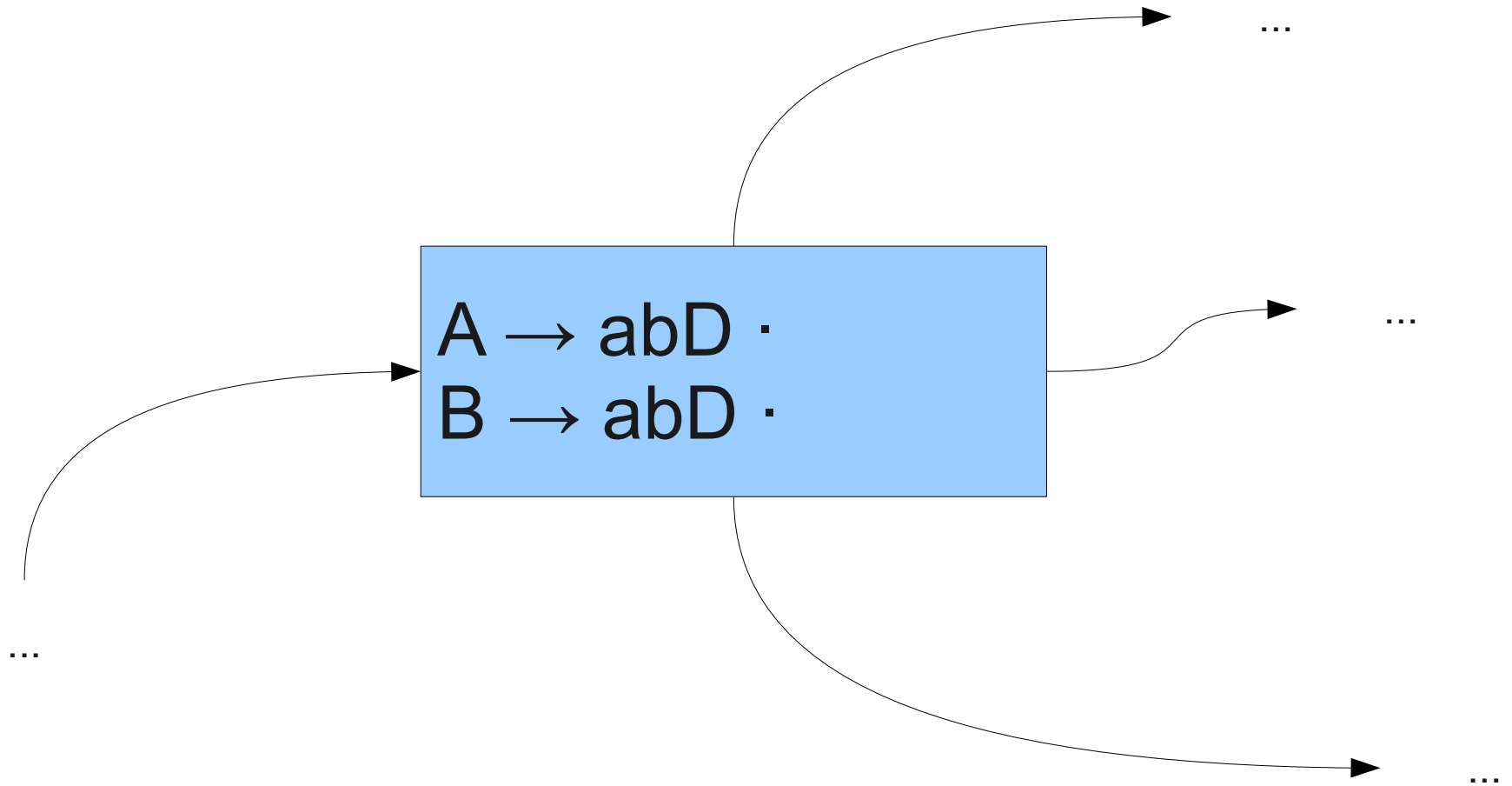
What error is this?

What error is this?



What about this?

What about this?



And what about this?

And what about this?



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?

A Powerful Parser: LR(1)

- Bottom-up predictive parsing with
 - **L**: Left-to-right scan
 - **R**: Rightmost derivation
 - **(1)**: One token lookahead
- **Substantially** more powerful than the other methods we've covered so far (more on that later).
- Tries to more intelligently find handles by using a lookahead token at each step.

LR(1) Parsing: The Intuition

$S \rightarrow E$

$E \rightarrow T$

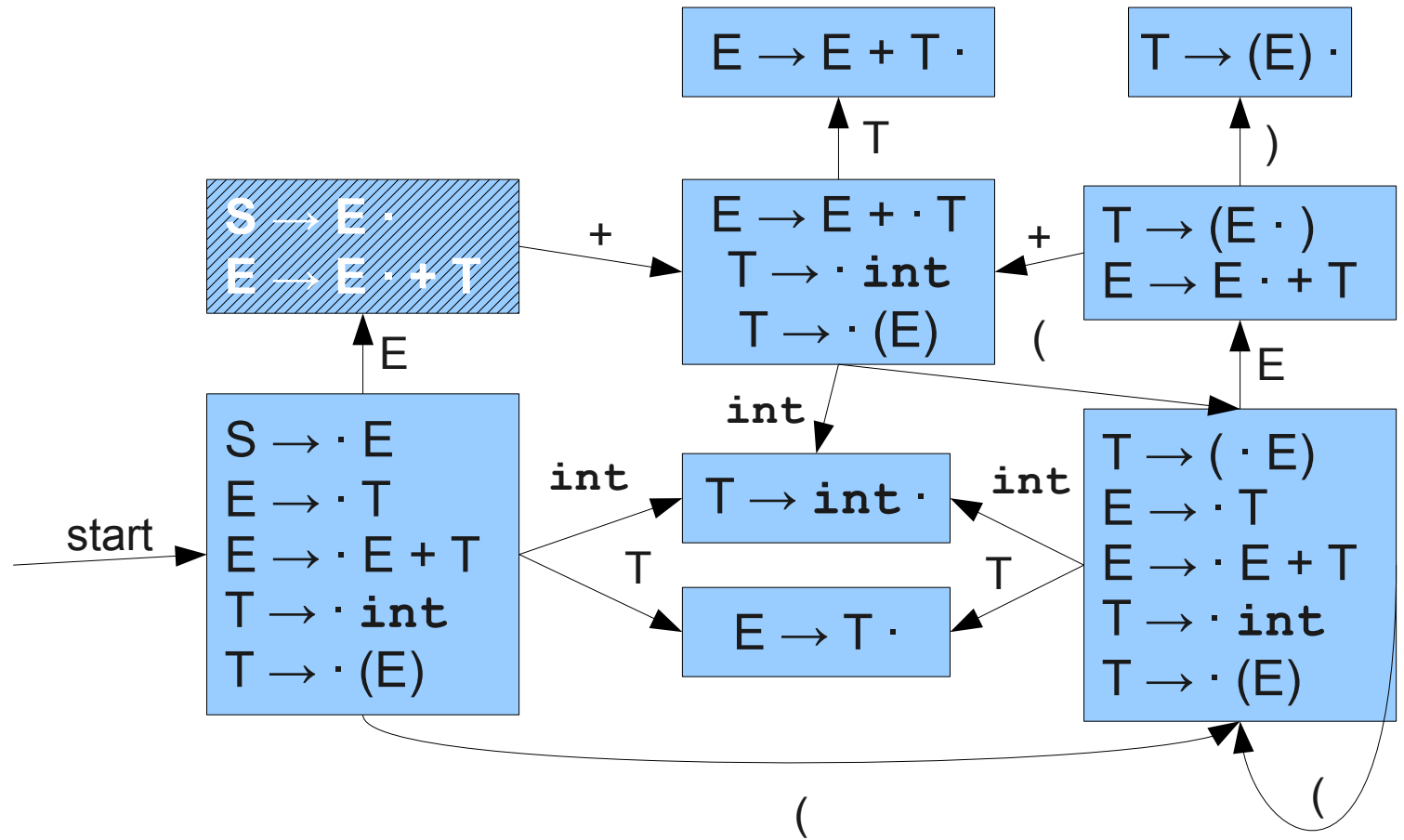
$E \rightarrow E + T$

$T \rightarrow \mathbf{int}$

$T \rightarrow (E)$

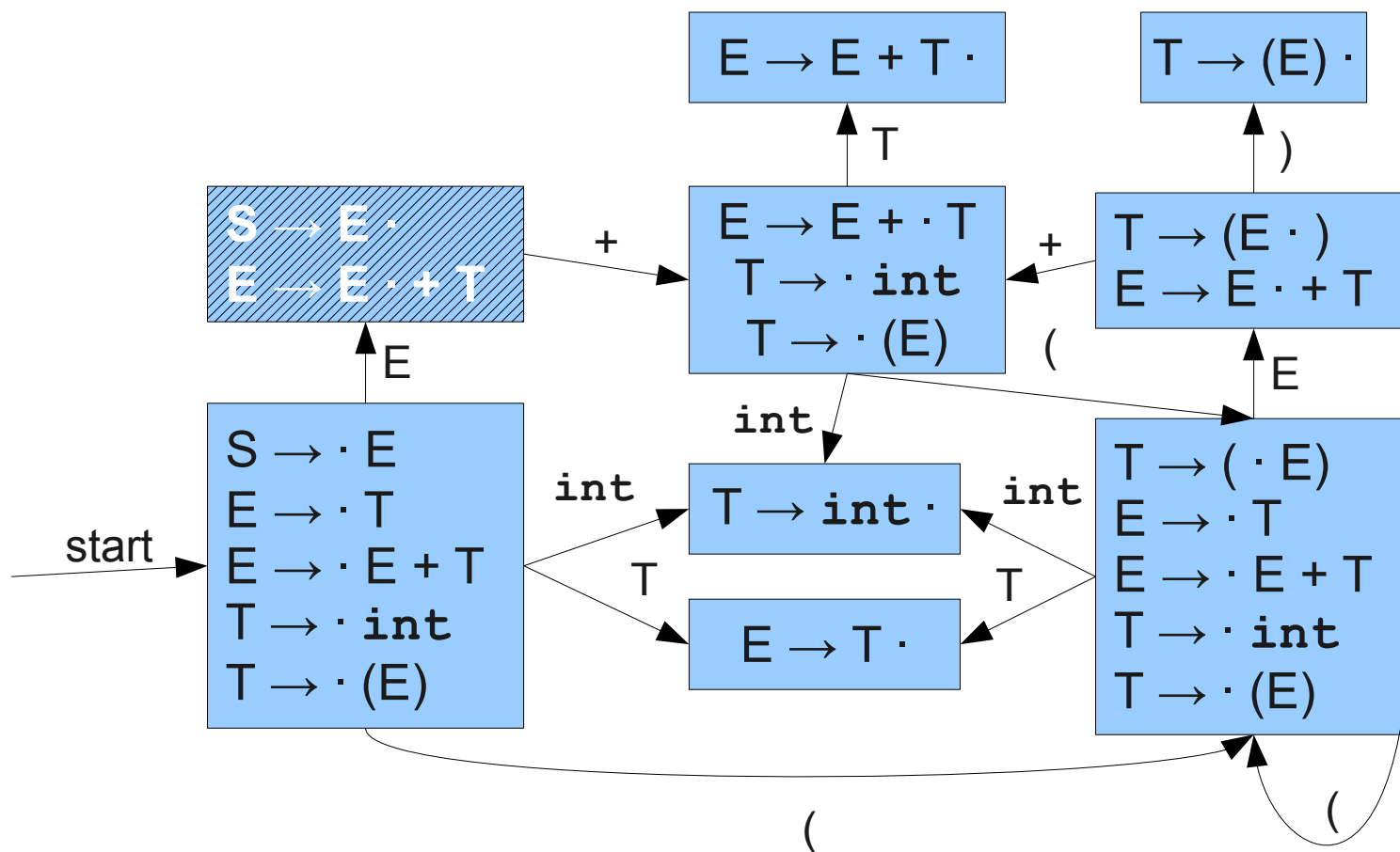
LR(1) Parsing: The Intuition

$S \rightarrow E$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$



LR(1) Parsing: The Intuition

$S \rightarrow E$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$

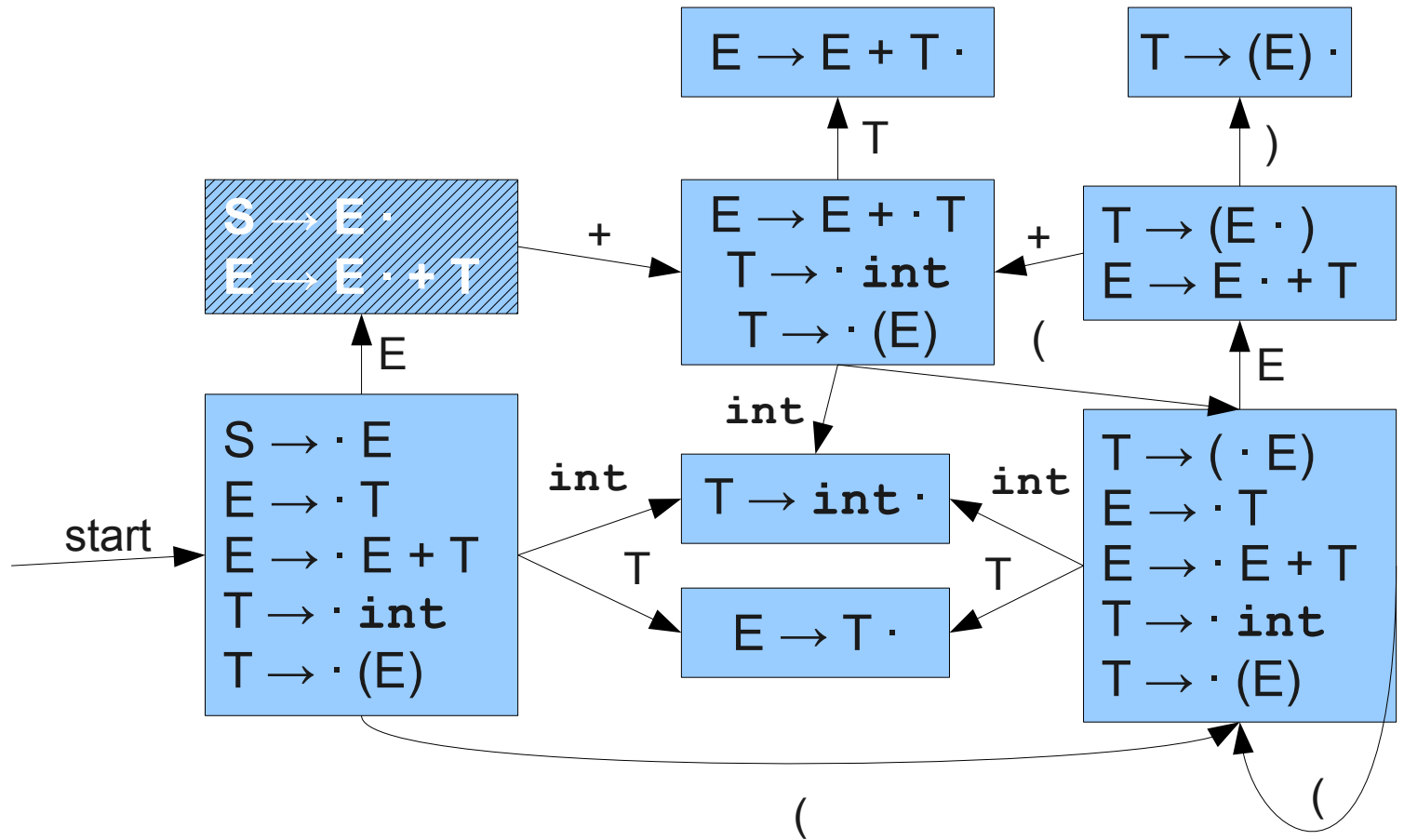


int	+	(int	+	int	+	int)	\$
-----	---	---	-----	---	-----	---	-----	---	----

LR(1) Parsing: The Intuition

$S \rightarrow E$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$

$S \rightarrow \cdot E$ \$



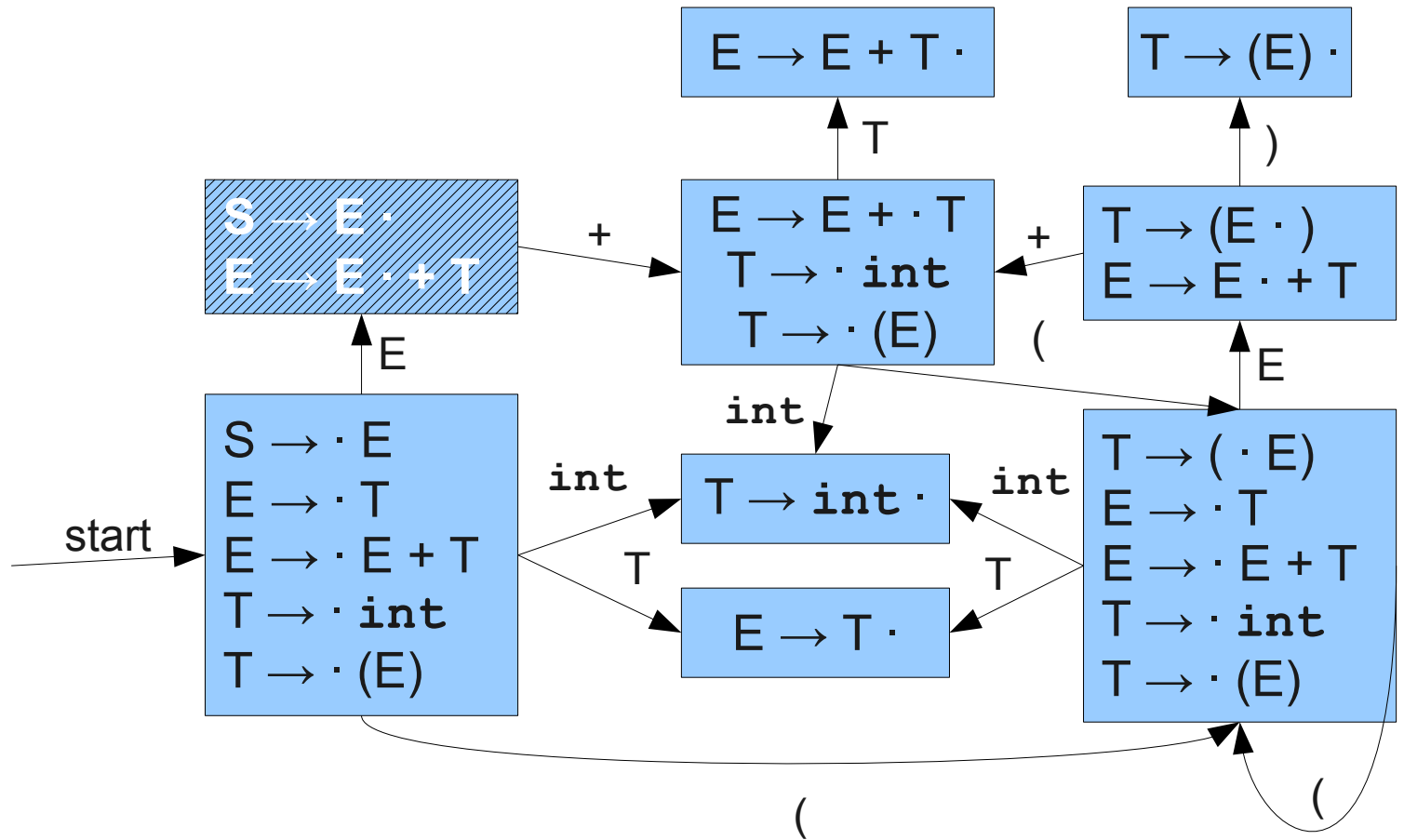
|

int	+	(int	+	int	+	int)	\$
-----	---	---	-----	---	-----	---	-----	---	----

LR(1) Parsing: The Intuition

$S \rightarrow E$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$

$S \rightarrow \cdot E$	\$
$E \rightarrow \cdot E + T$	\$

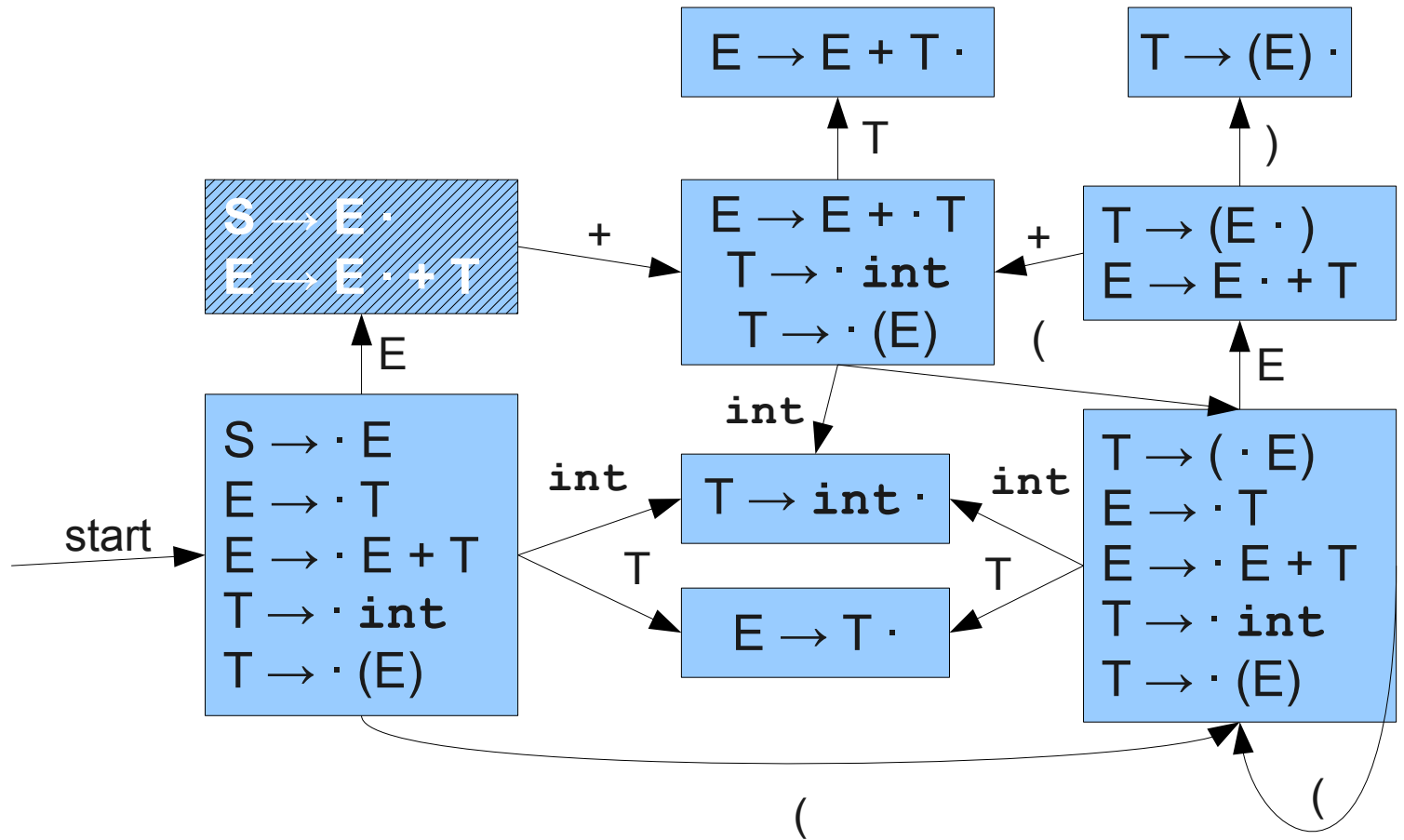


int + (int + int + int) \$

LR(1) Parsing: The Intuition

$S \rightarrow E$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$

$S \rightarrow \cdot E$	\$
$E \rightarrow \cdot E + T$	\$
$E \rightarrow \cdot T$	+

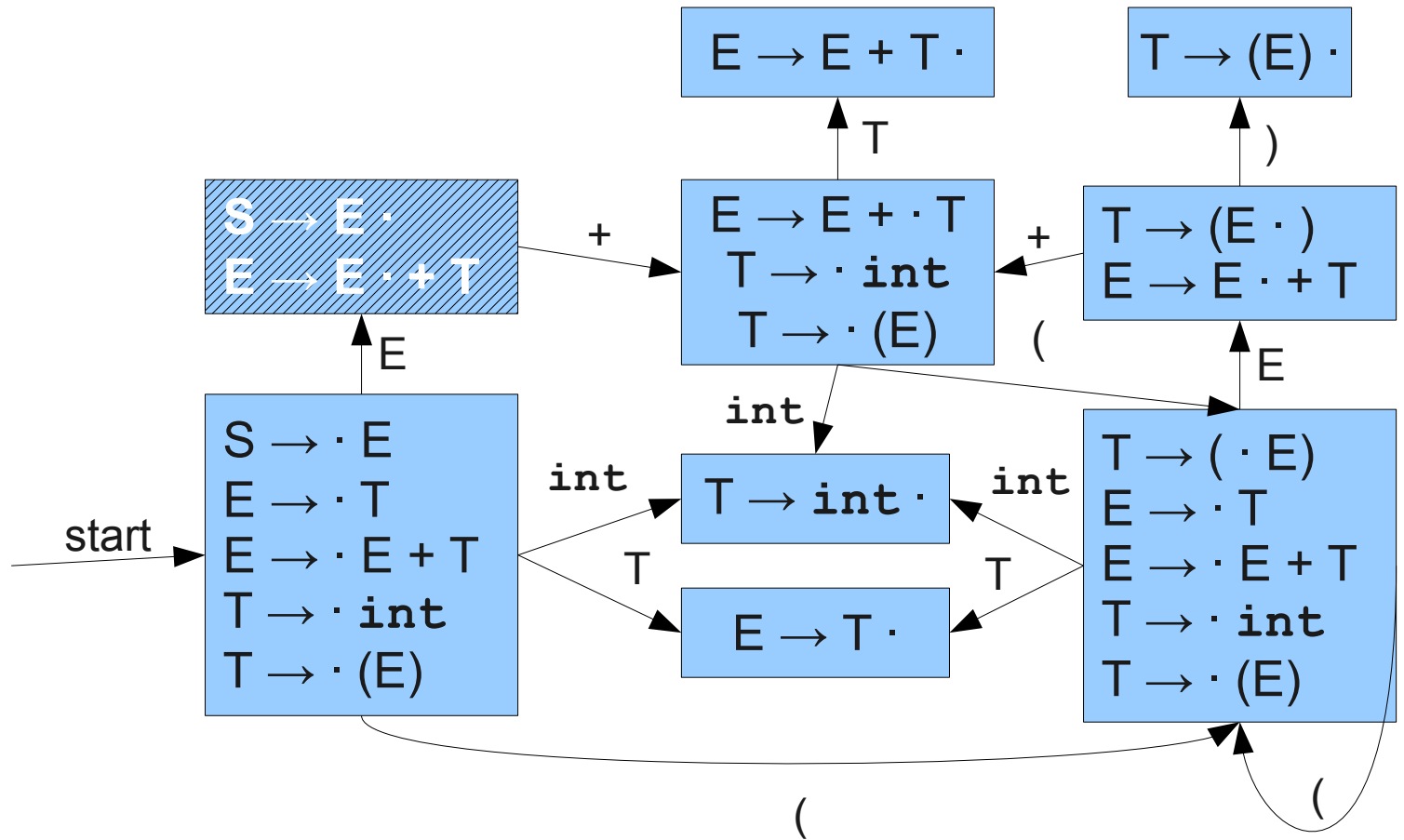


int	+	(int	+	int	+	int)	\$
-----	---	---	-----	---	-----	---	-----	---	----

LR(1) Parsing: The Intuition

$S \rightarrow E$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$

$S \rightarrow \cdot E$	\$
$E \rightarrow \cdot E + T$	\$
$E \rightarrow \cdot T$	+
$T \rightarrow \cdot \text{int}$	+

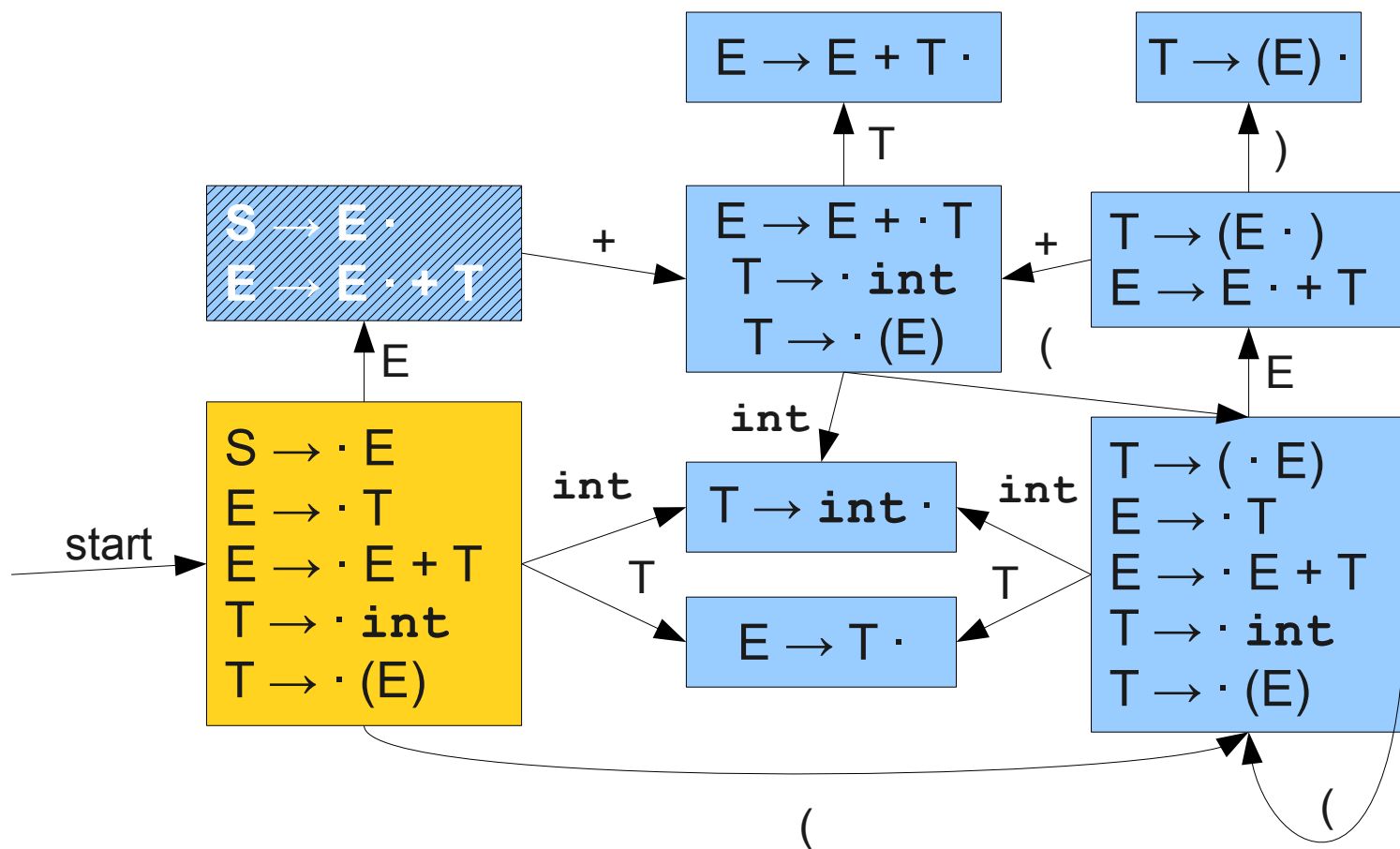


int + (int + int + int) \$

LR(1) Parsing: The Intuition

$S \rightarrow E$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$

$S \rightarrow \cdot E$	\$
$E \rightarrow \cdot E + T$	\$
$E \rightarrow \cdot T$	+
$T \rightarrow \cdot \text{int}$	+

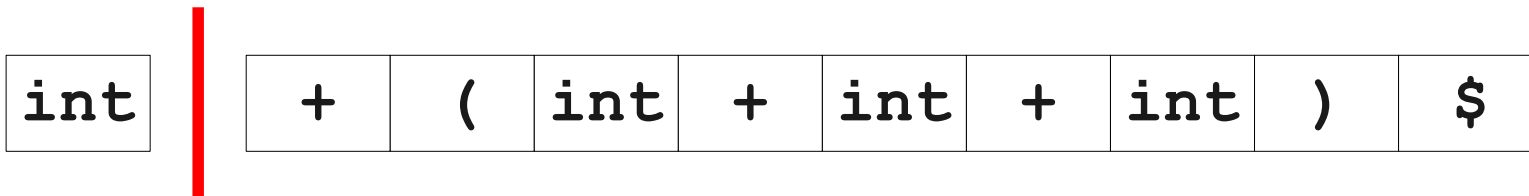
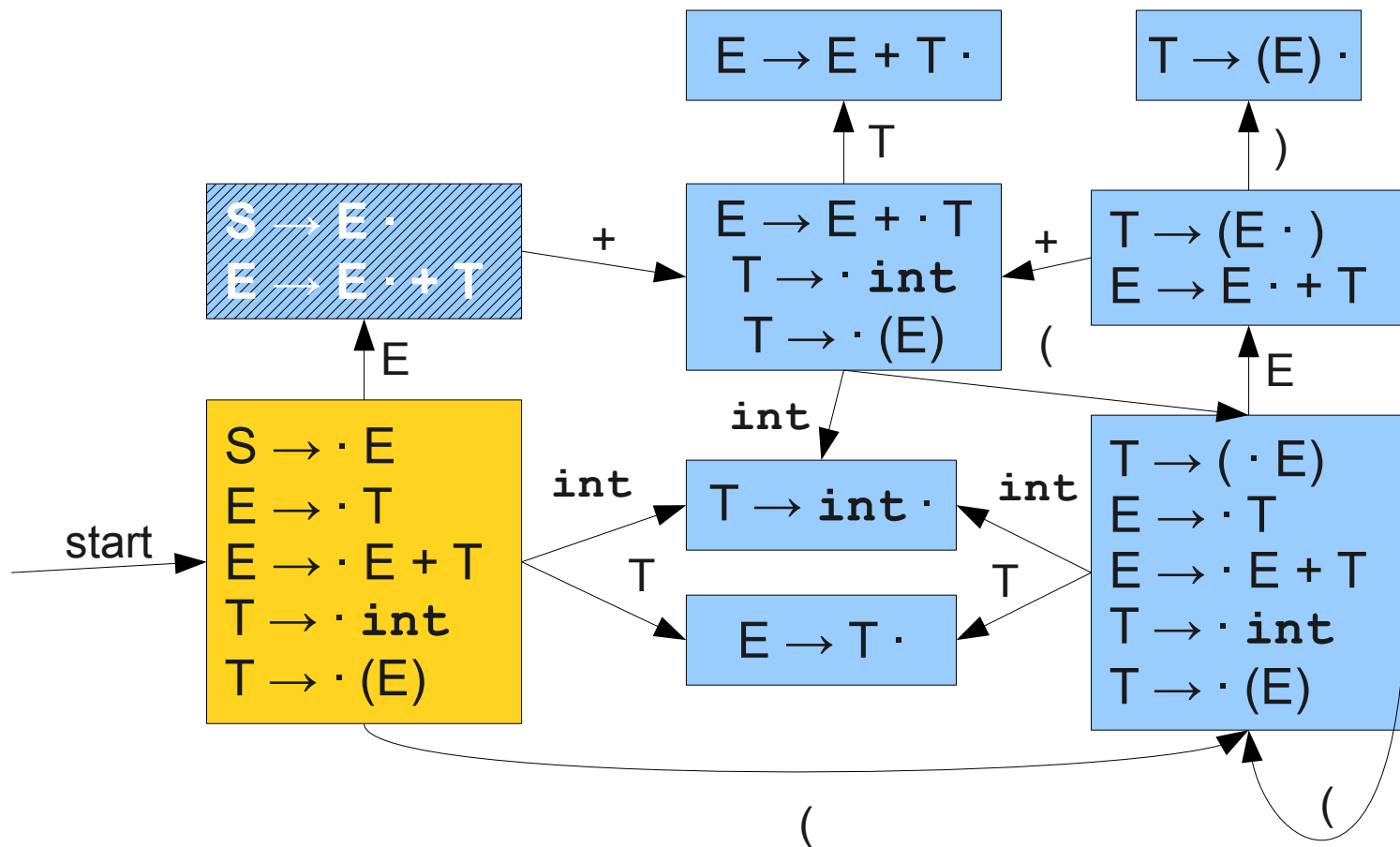


int + (int + int + int) \$

LR(1) Parsing: The Intuition

$S \rightarrow E$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$

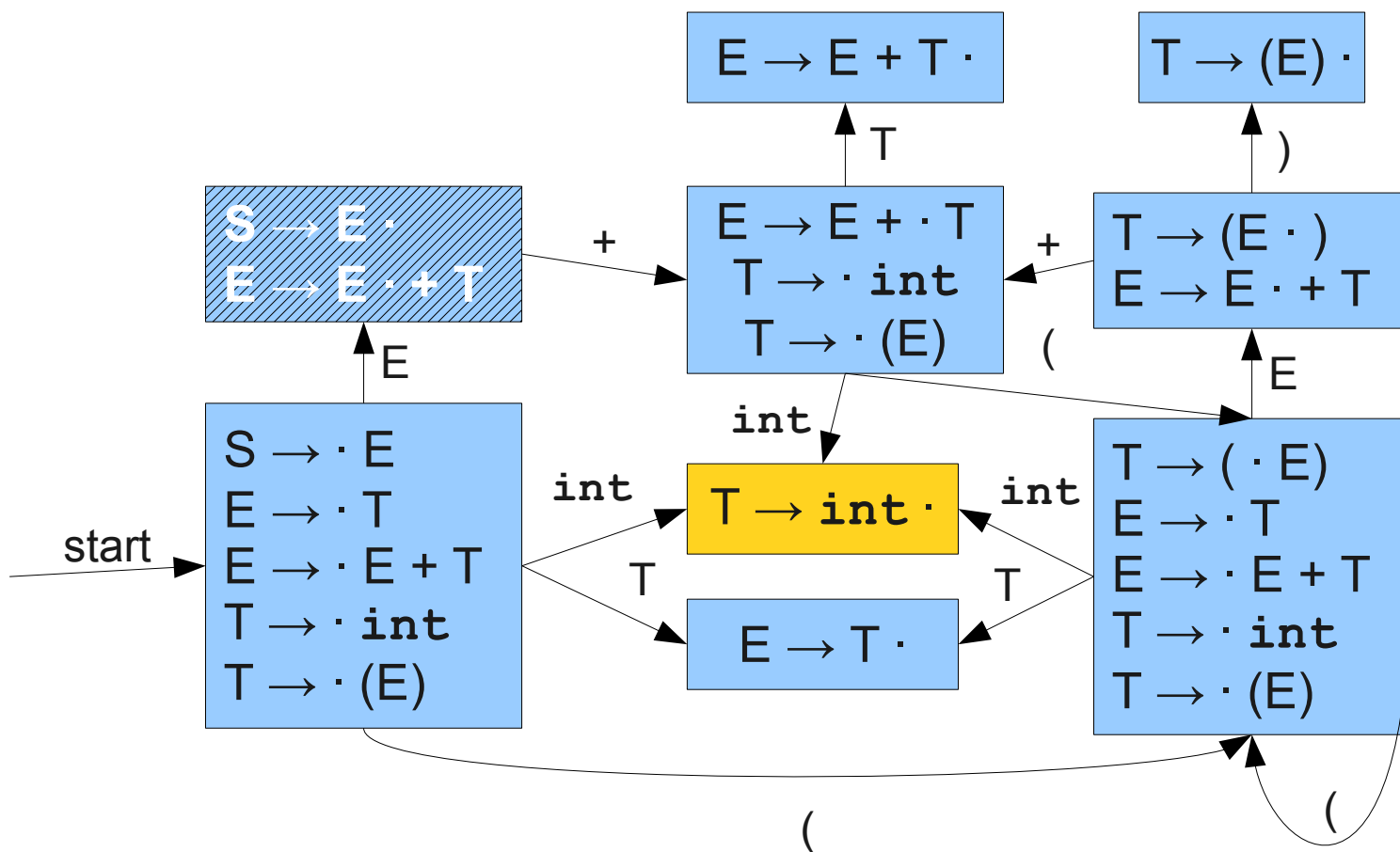
$S \rightarrow \cdot E$	\$
$E \rightarrow \cdot E + T$	\$
$E \rightarrow \cdot T$	+
$T \rightarrow \cdot \text{int}$	+



LR(1) Parsing: The Intuition

$S \rightarrow E$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$

$S \rightarrow \cdot E$	\$
$E \rightarrow \cdot E + T$	\$
$E \rightarrow \cdot T$	+
$T \rightarrow \cdot \text{int}$	+



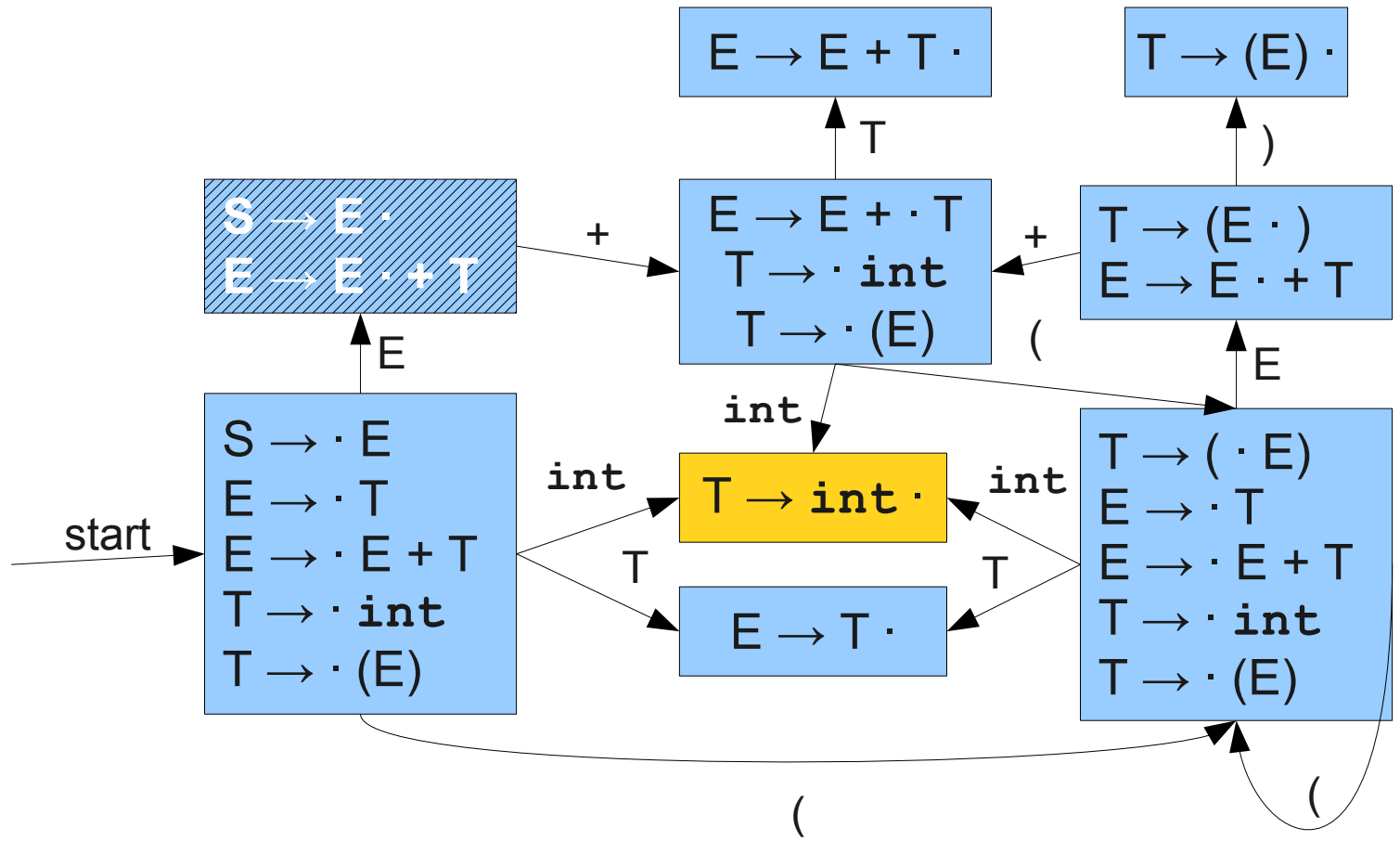
int

| + (int + int + int) \$

LR(1) Parsing: The Intuition

$S \rightarrow E$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$

$S \rightarrow \cdot E$	\$
$E \rightarrow \cdot E + T$	\$
$E \rightarrow \cdot T$	+
$T \rightarrow \text{int} \cdot$	+



int

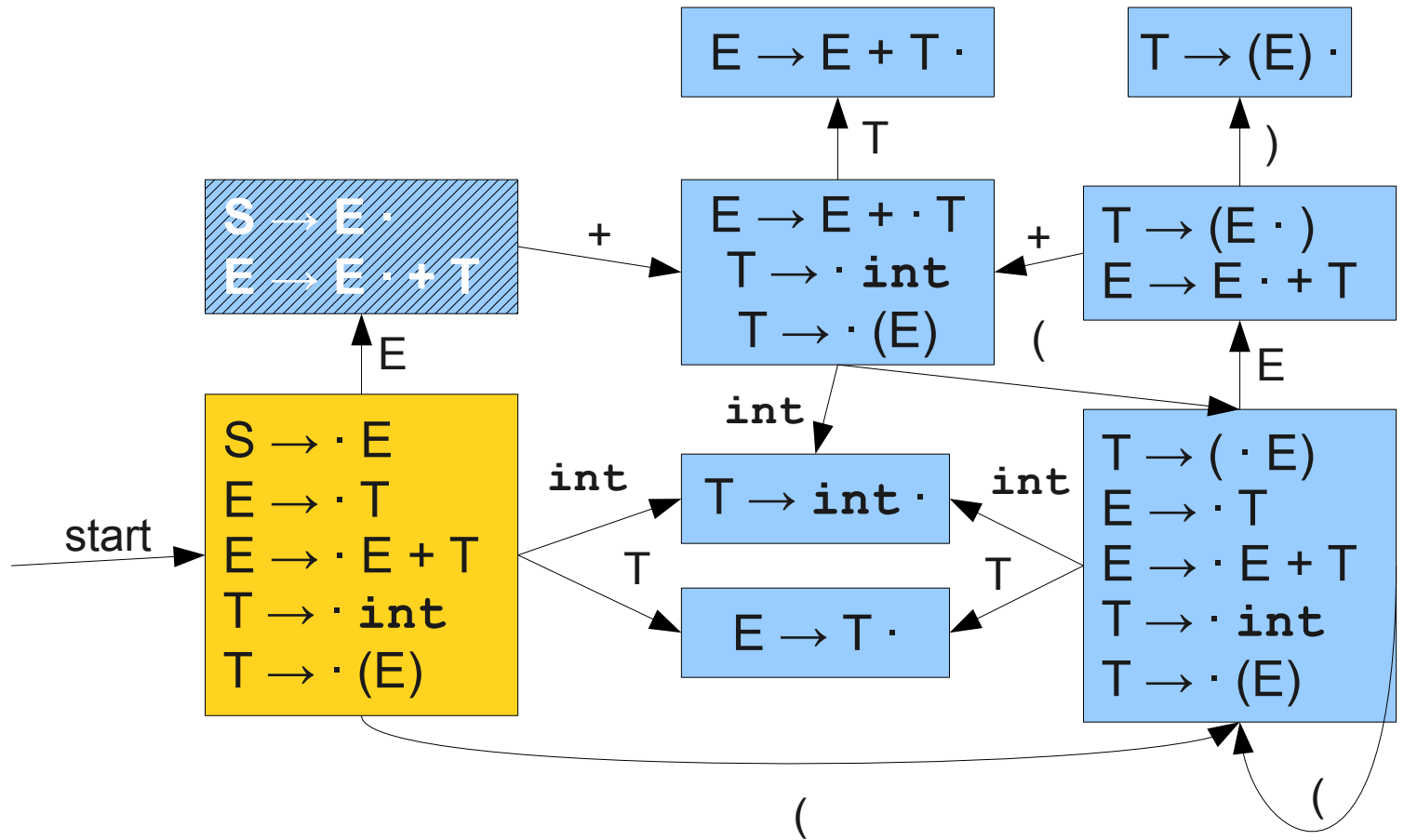


+ (int + int + int) \$

LR(1) Parsing: The Intuition

$S \rightarrow E$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$

$S \rightarrow \cdot E$	\$
$E \rightarrow \cdot E + T$	\$
$E \rightarrow \cdot T$	+



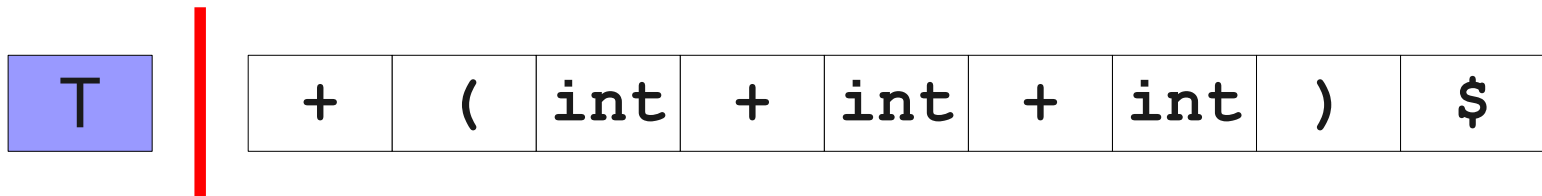
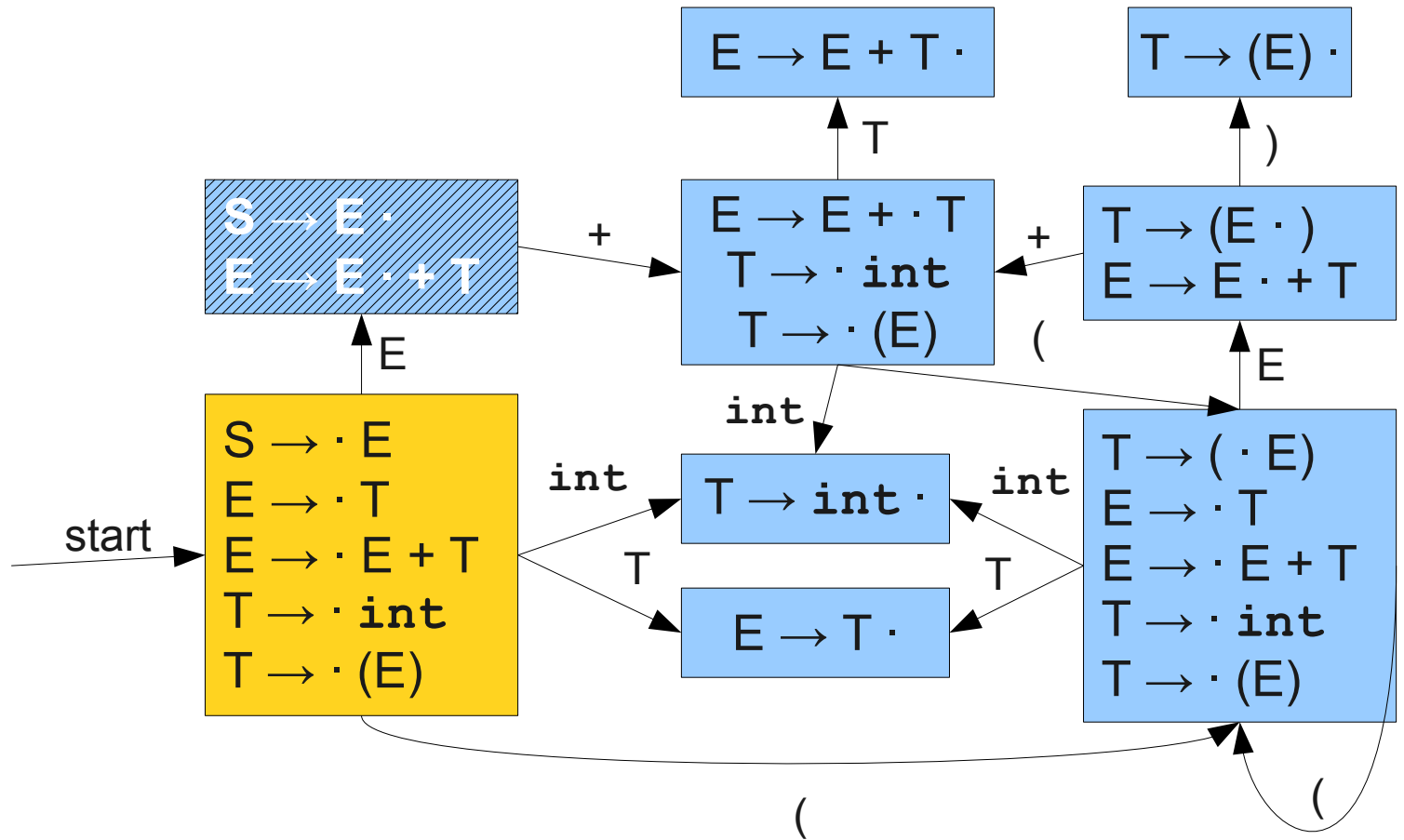
|

+	(int	+	int	+	int)	\$
---	---	-----	---	-----	---	-----	---	----

LR(1) Parsing: The Intuition

$S \rightarrow E$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$

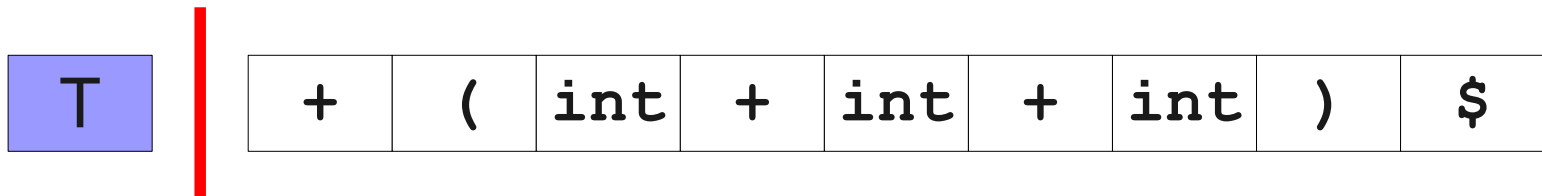
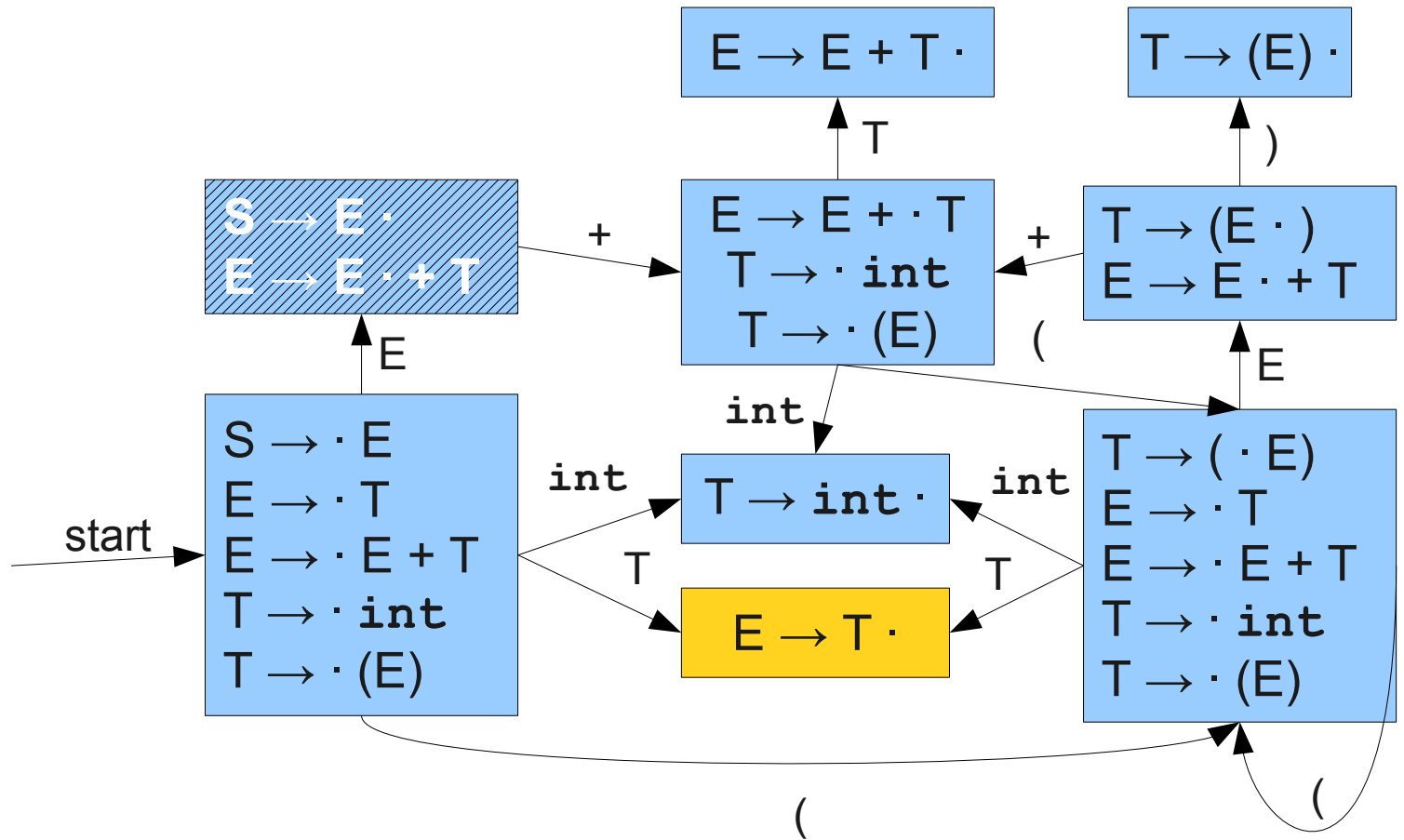
$S \rightarrow \cdot E$	\$
$E \rightarrow \cdot E + T$	\$
$E \rightarrow \cdot T$	+



LR(1) Parsing: The Intuition

$S \rightarrow E$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$

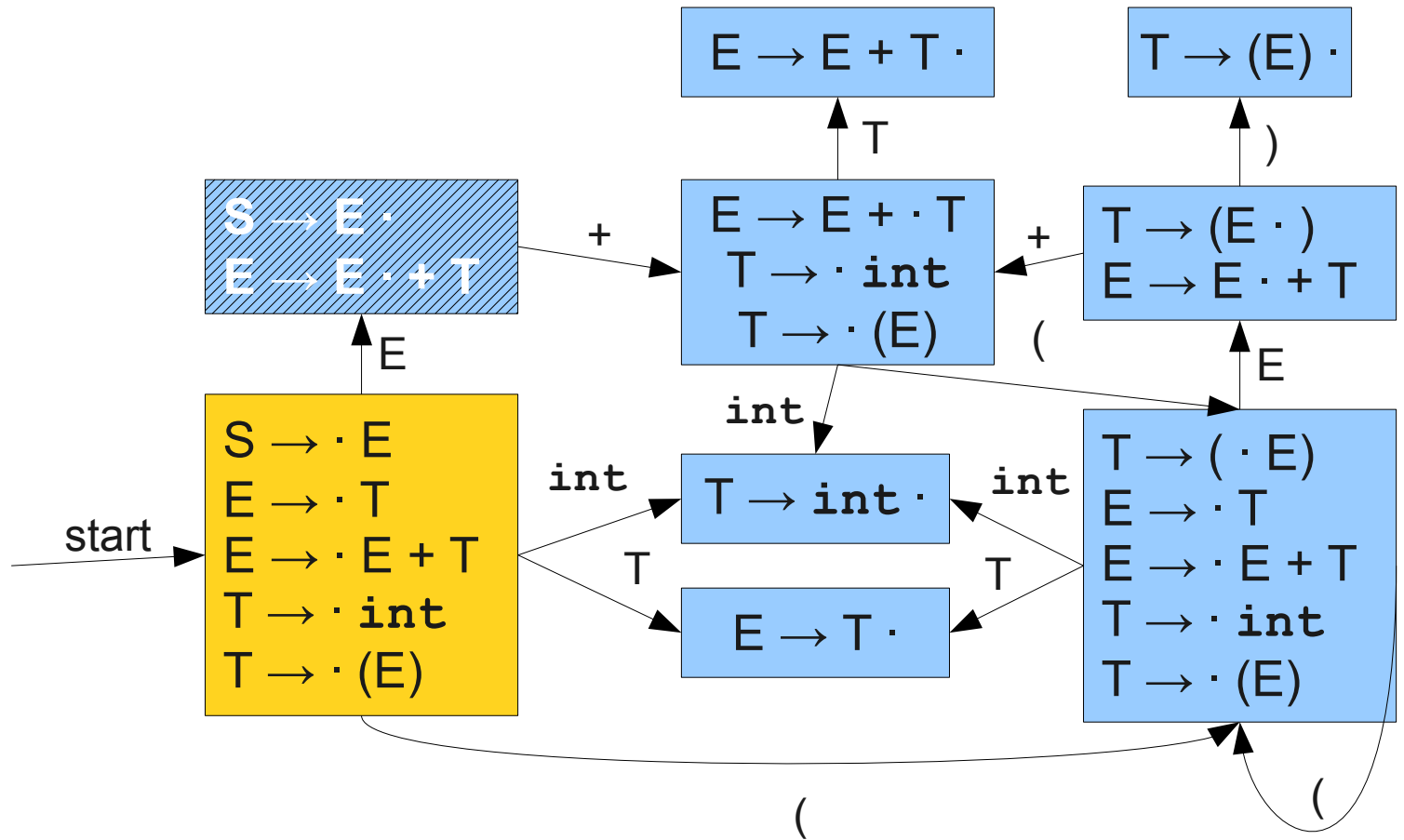
$S \rightarrow \cdot E$	\$
$E \rightarrow \cdot E + T$	\$
$E \rightarrow T \cdot$	+



LR(1) Parsing: The Intuition

$S \rightarrow E$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$

$S \rightarrow \cdot E$	\$
$E \rightarrow \cdot E + T$	\$



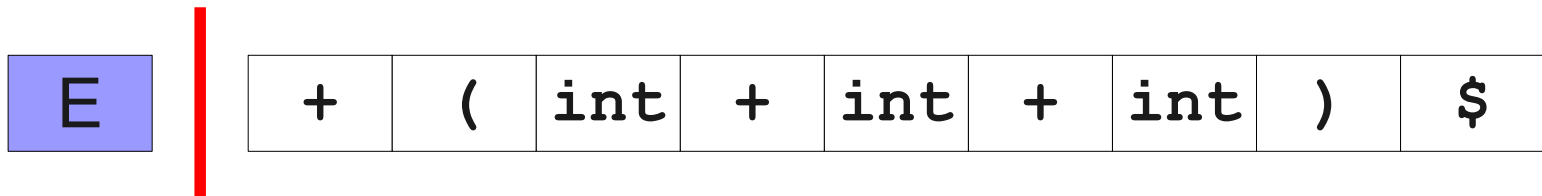
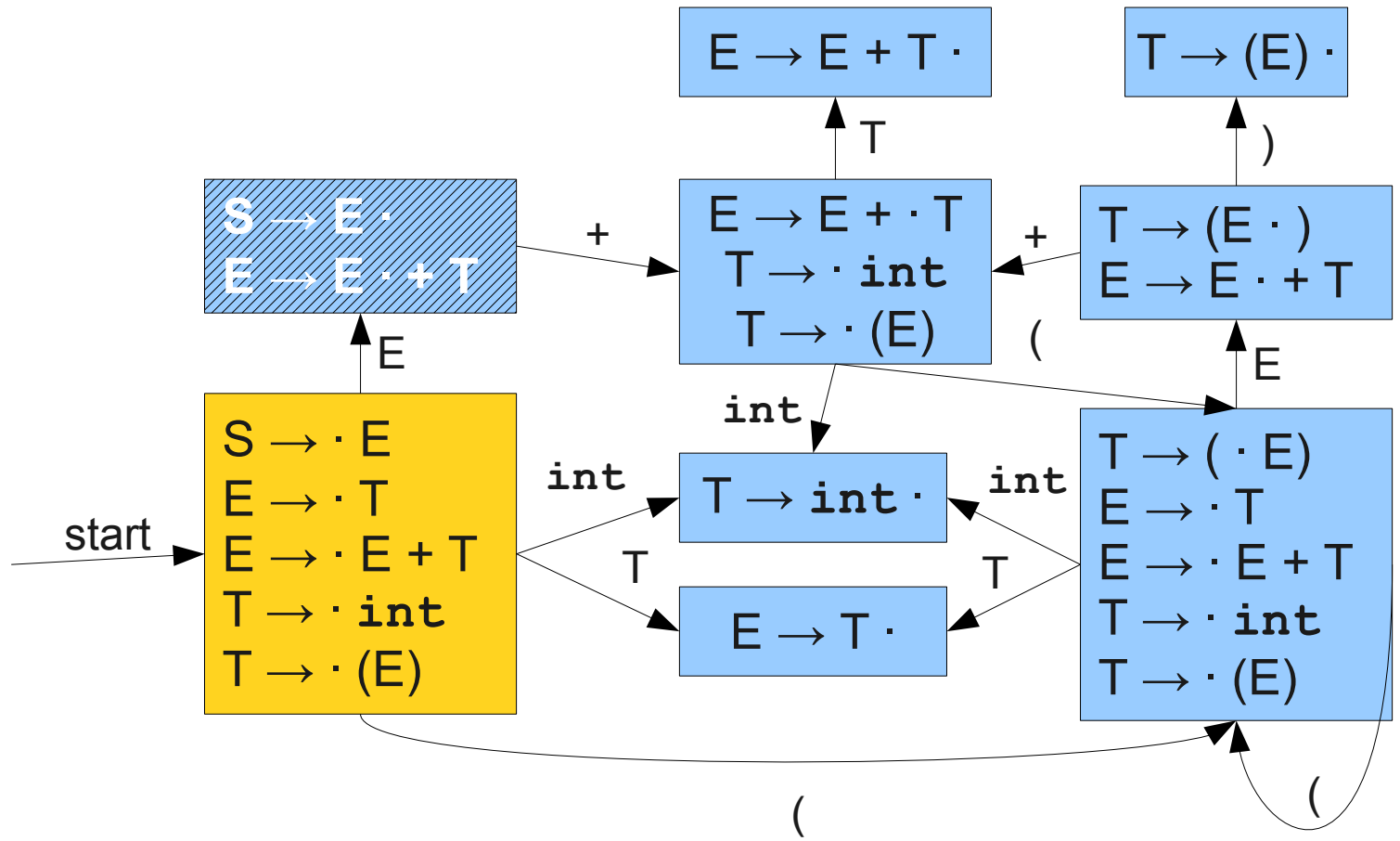
|

+	(int	+	int	+	int)	\$
---	---	-----	---	-----	---	-----	---	----

LR(1) Parsing: The Intuition

$S \rightarrow E$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$

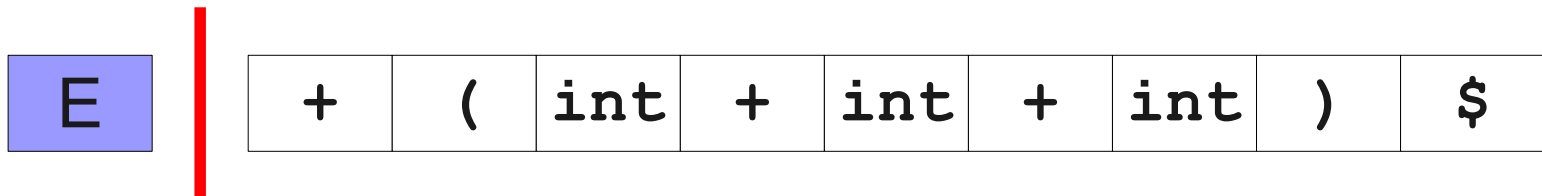
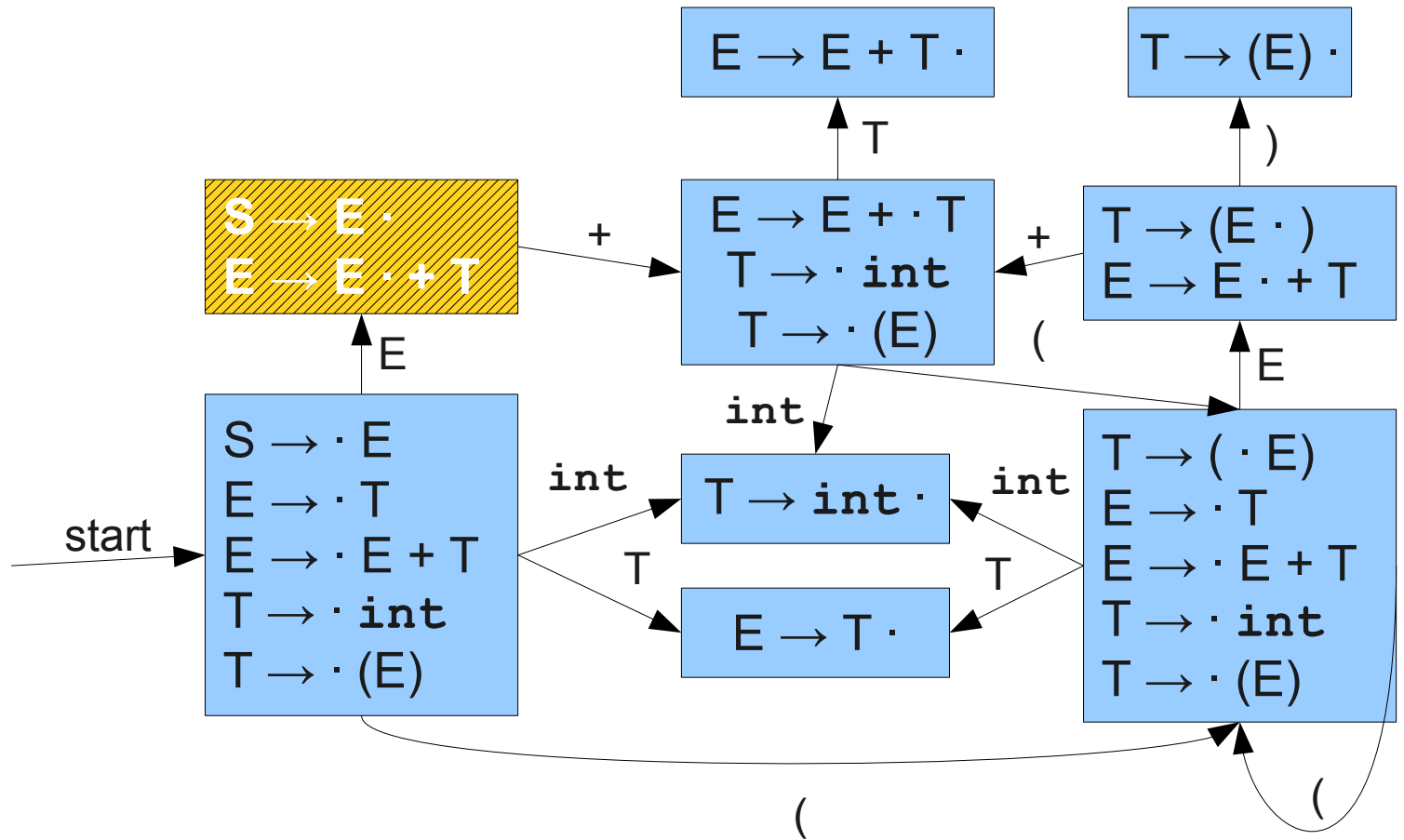
$S \rightarrow \cdot E$	\$
$E \rightarrow \cdot E + T$	\$



LR(1) Parsing: The Intuition

$S \rightarrow E$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$

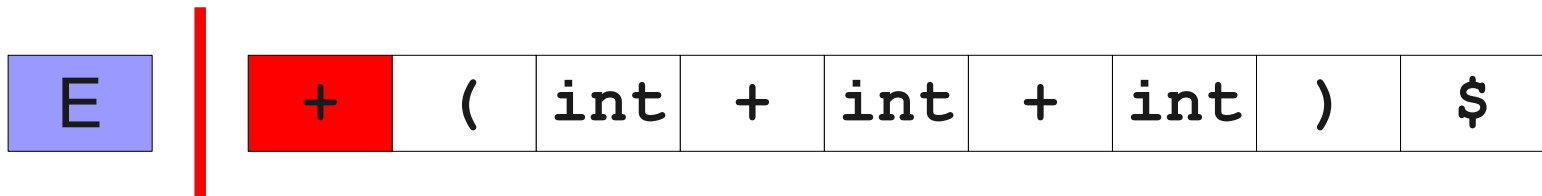
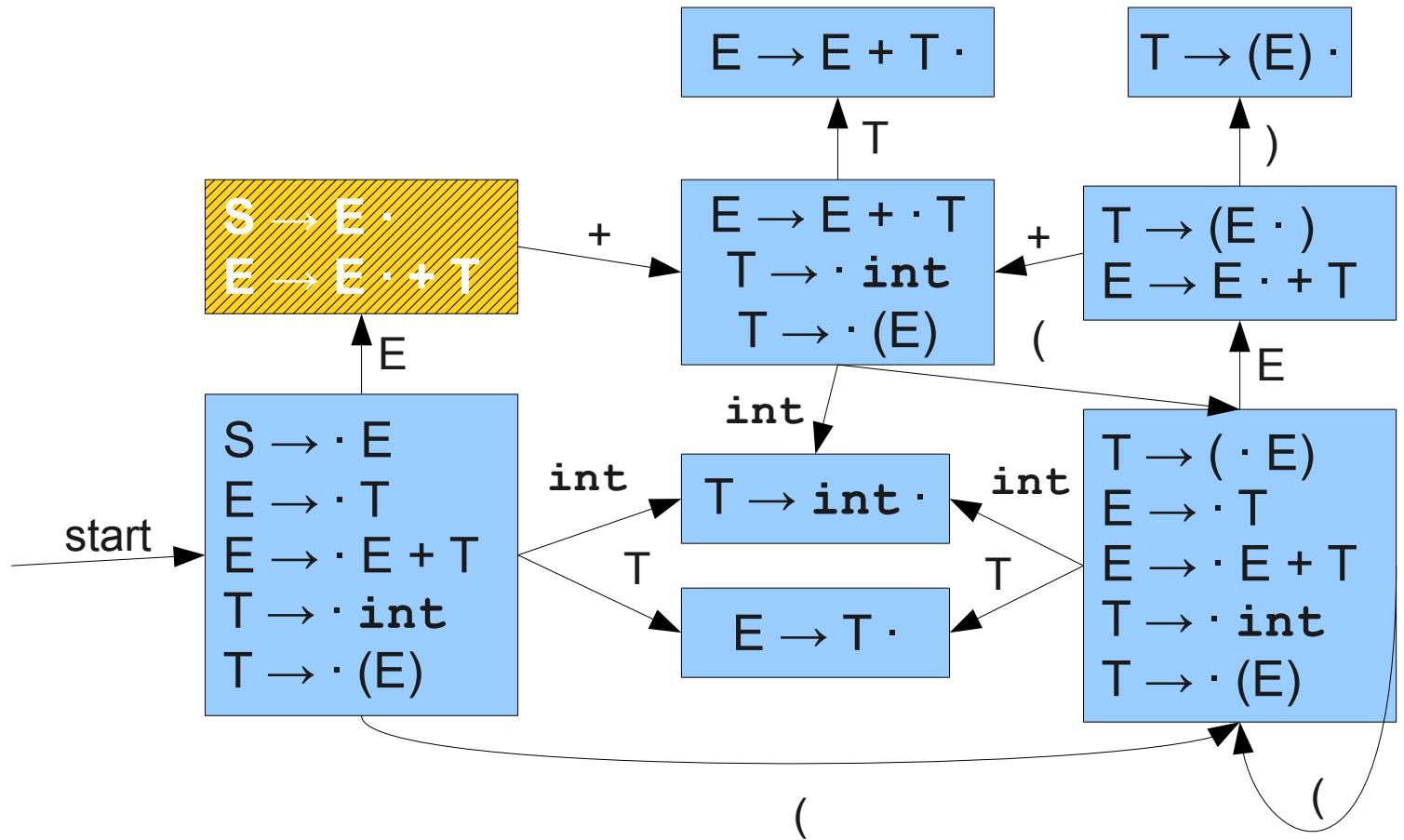
$S \rightarrow \cdot E$	\$
$E \rightarrow E \cdot + T$	\$



LR(1) Parsing: The Intuition

$S \rightarrow E$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$

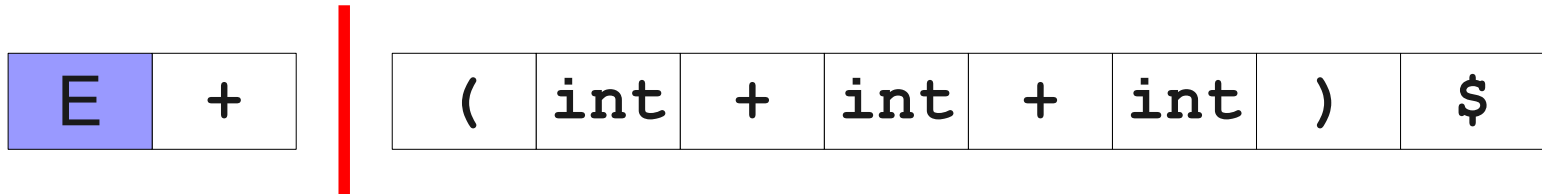
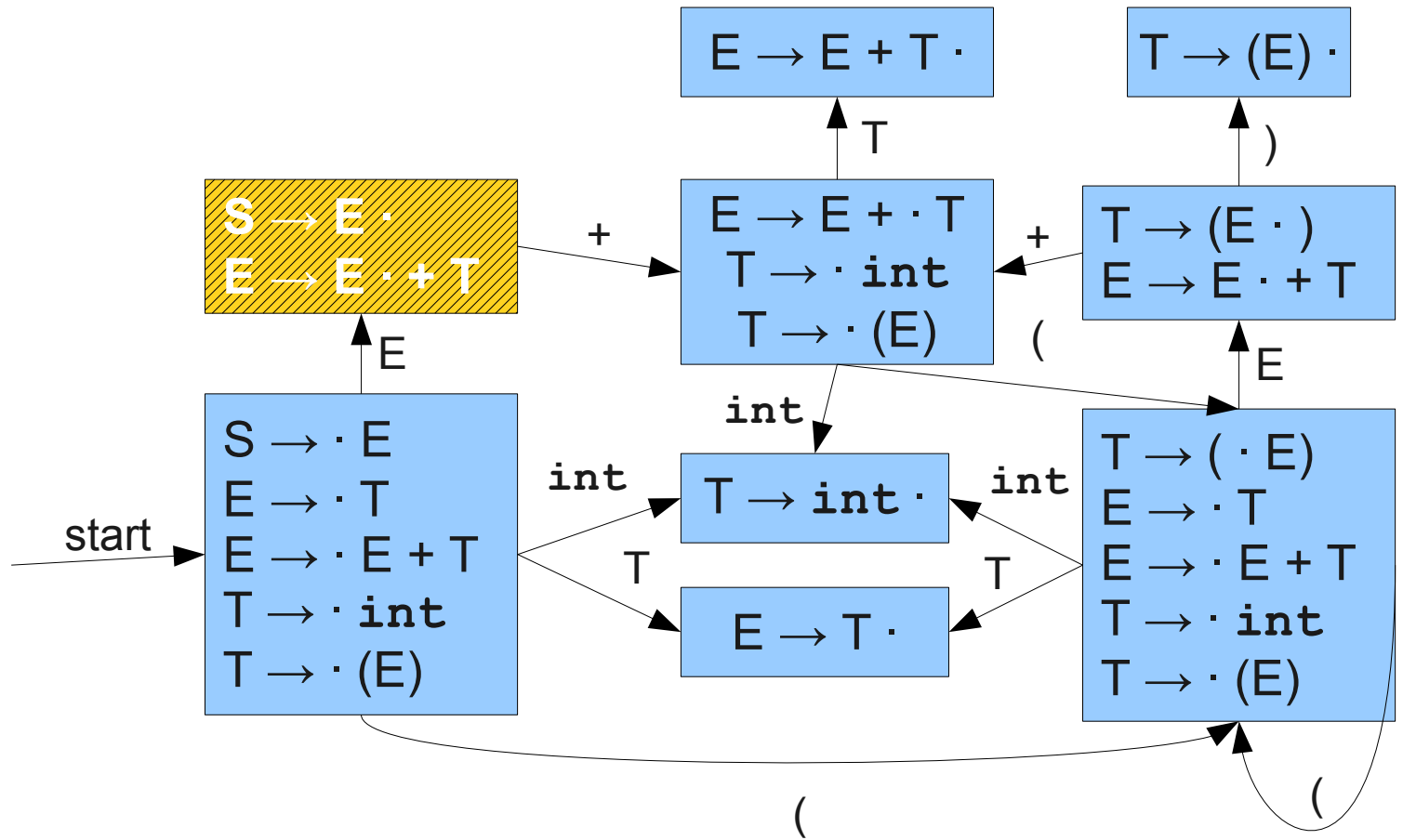
$S \rightarrow \cdot E$	\$
$E \rightarrow E \cdot + T$	\$



LR(1) Parsing: The Intuition

$S \rightarrow E$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$

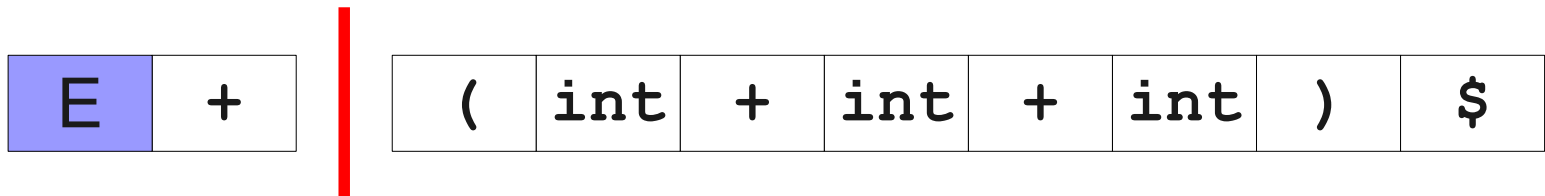
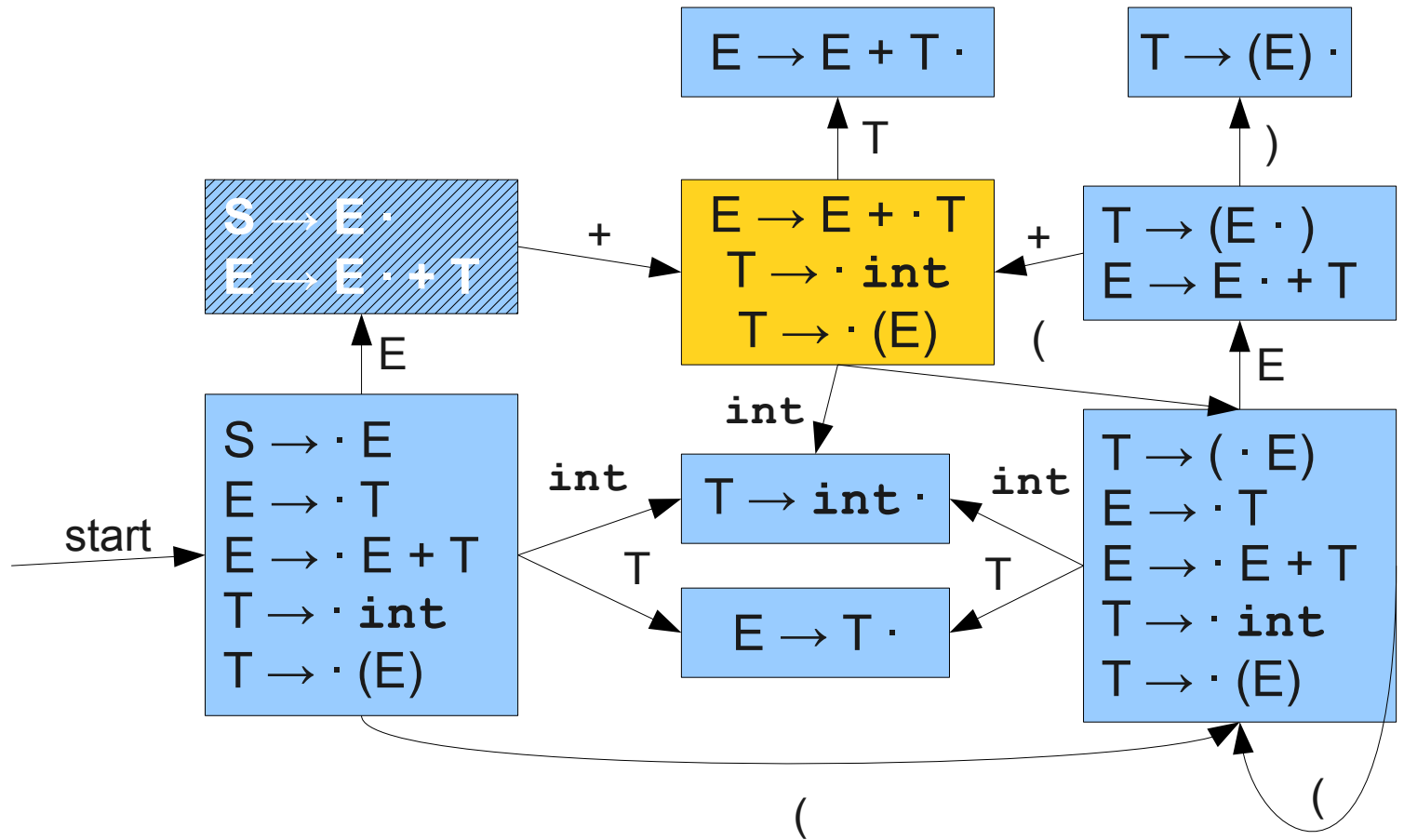
$S \rightarrow \cdot E$	\$
$E \rightarrow E \cdot + T$	\$



LR(1) Parsing: The Intuition

$S \rightarrow E$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$

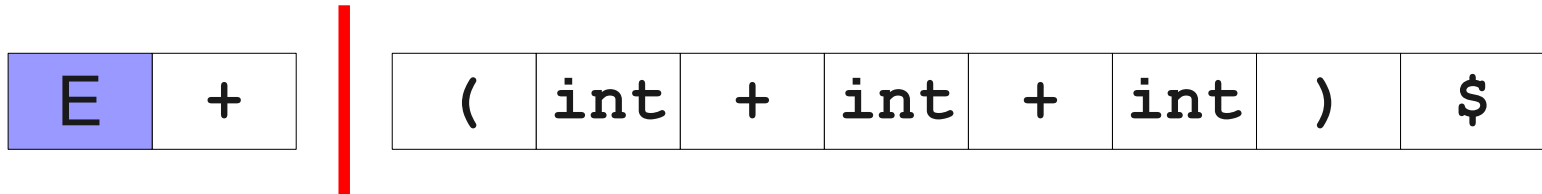
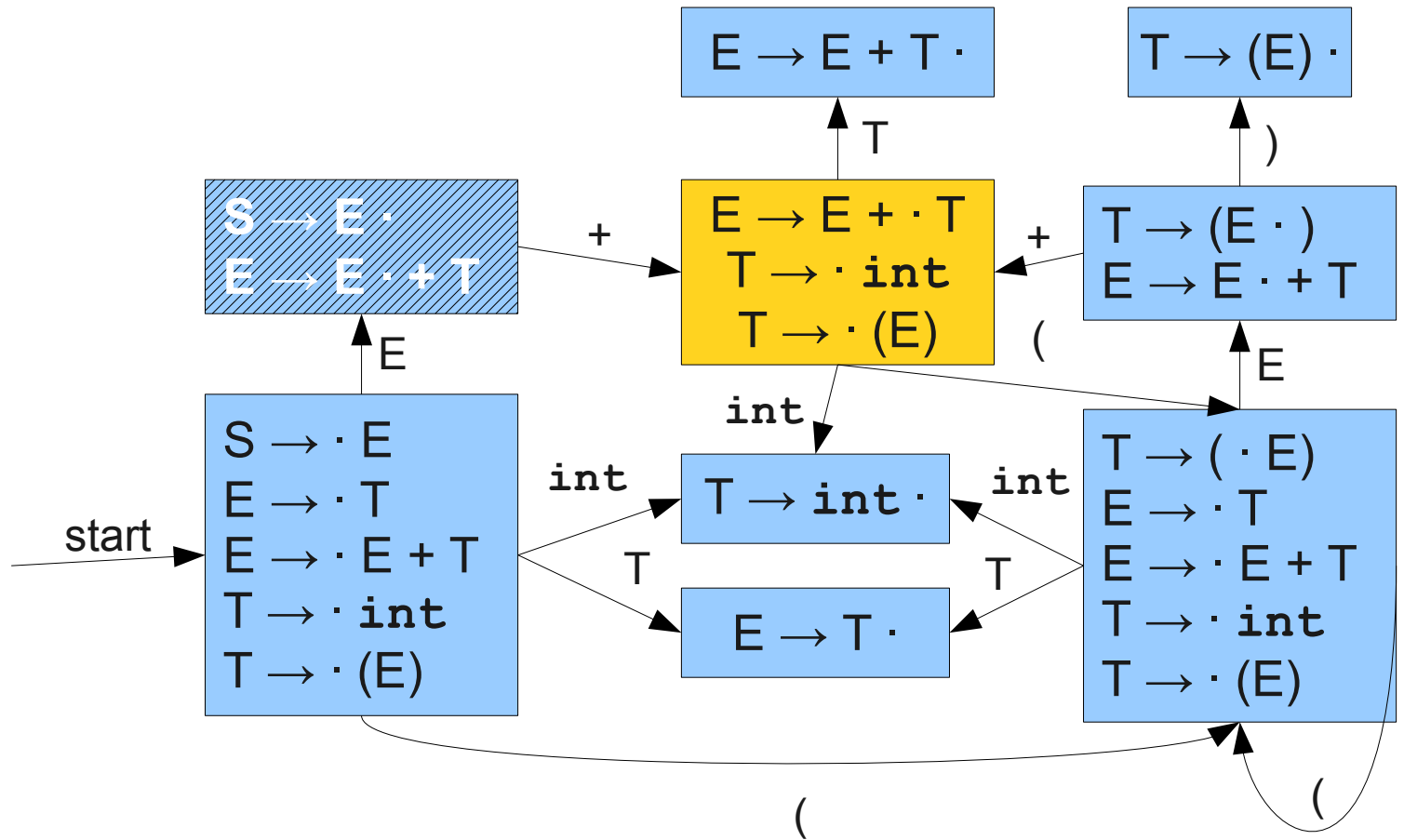
$S \rightarrow \cdot E$	\$
$E \rightarrow E + \cdot T$	\$



LR(1) Parsing: The Intuition

$S \rightarrow E$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$

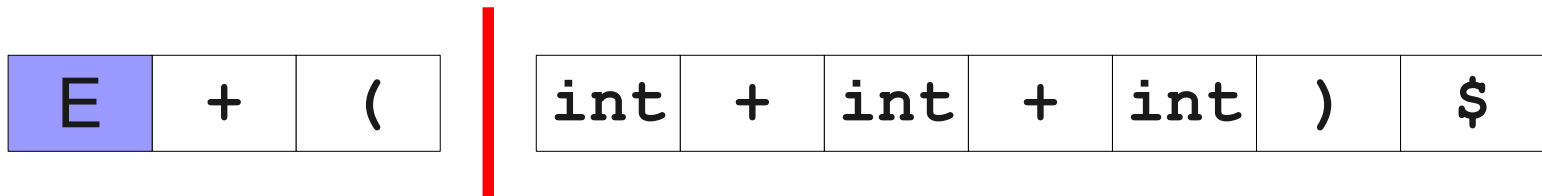
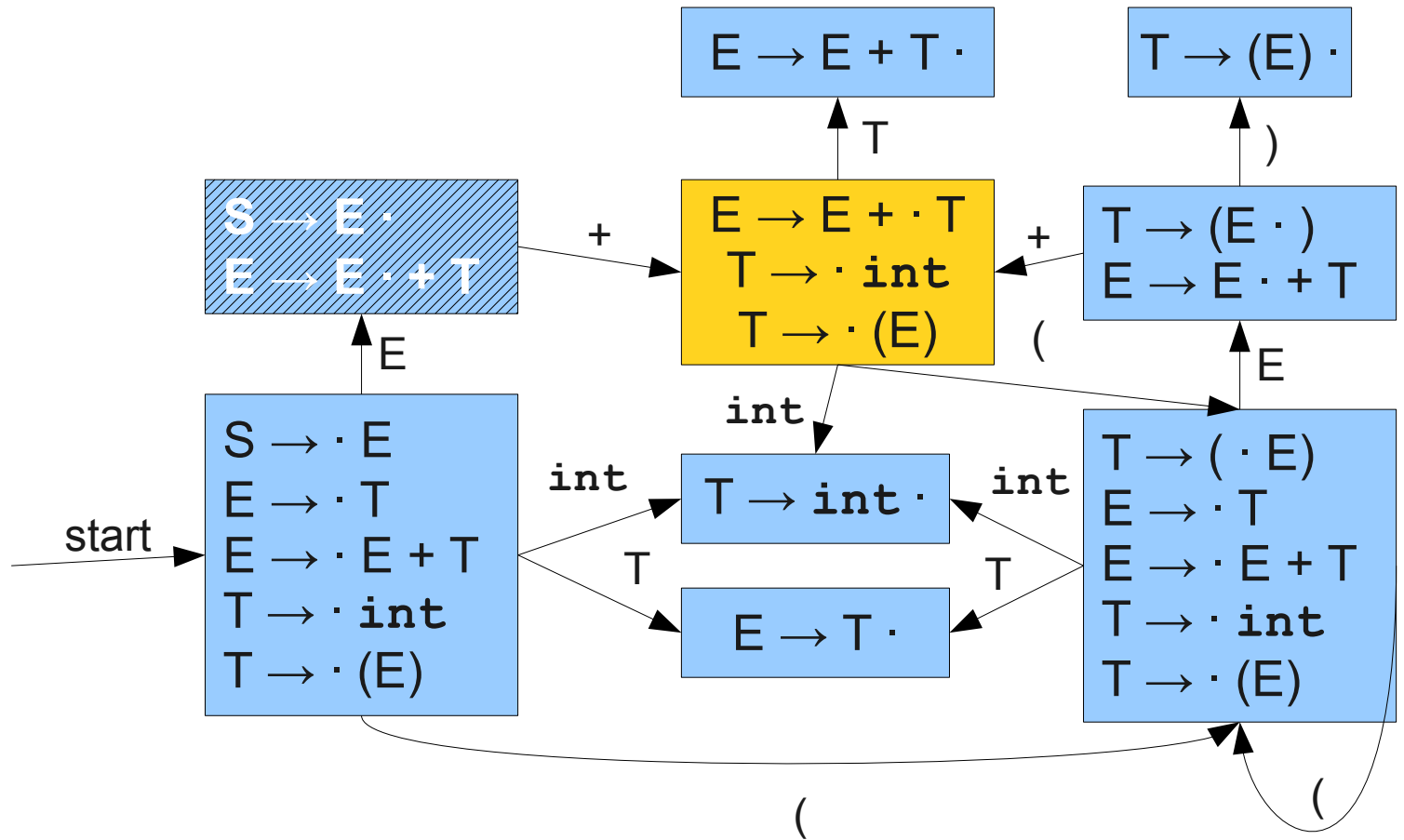
$S \rightarrow \cdot E$	\$
$E \rightarrow E + \cdot T$	\$
$T \rightarrow \cdot (E)$	\$



LR(1) Parsing: The Intuition

$S \rightarrow E$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$

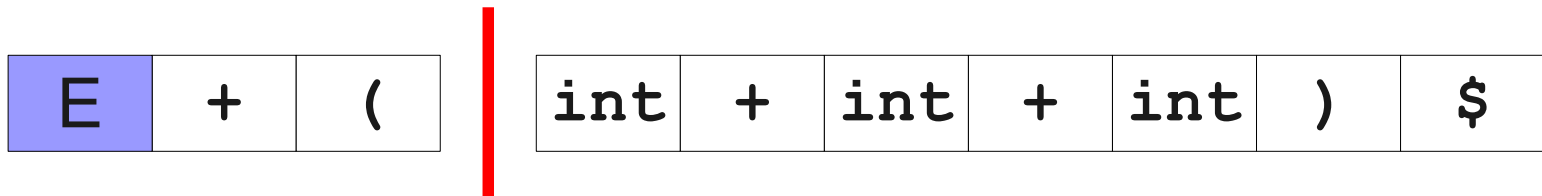
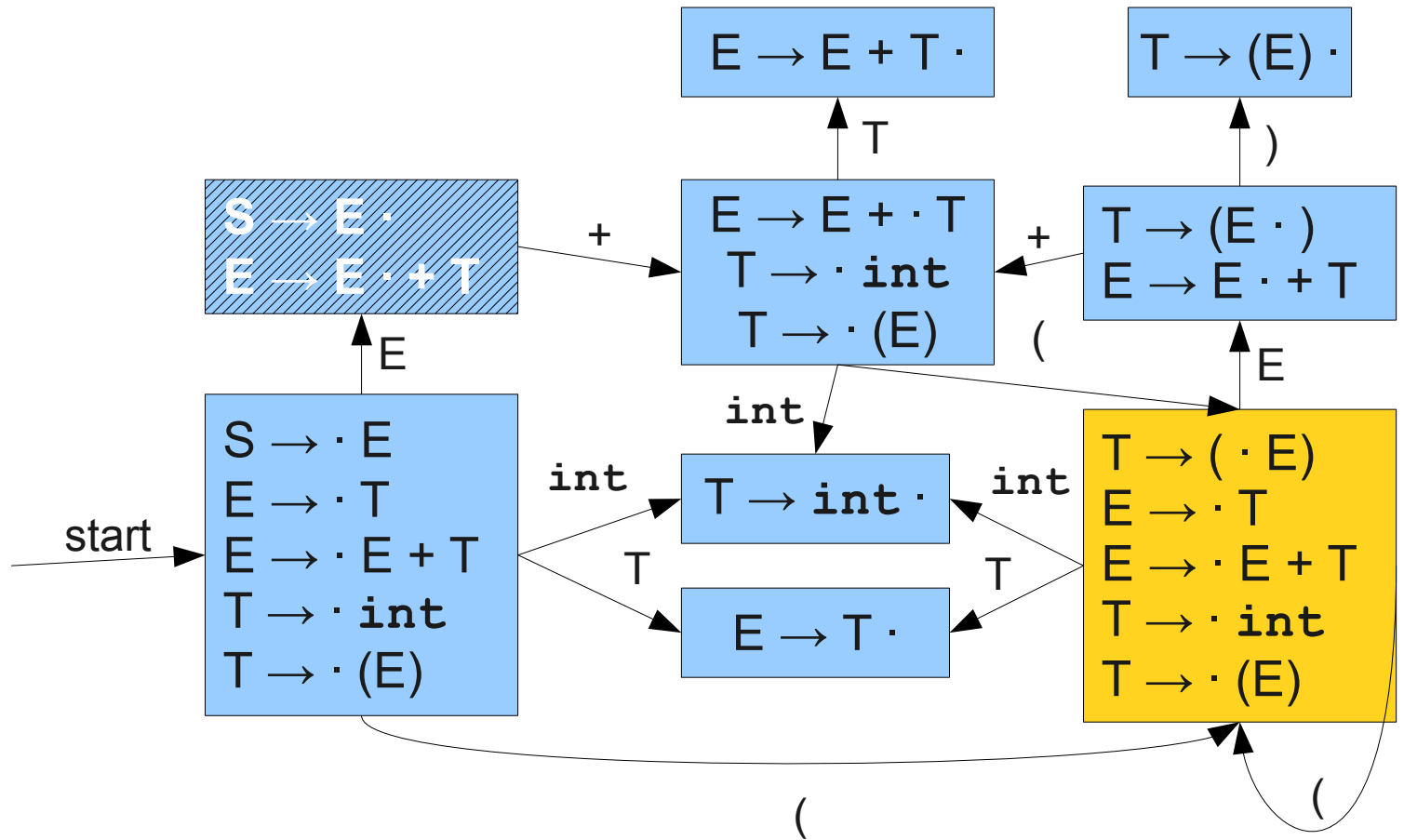
$S \rightarrow \cdot E$	\$
$E \rightarrow E + \cdot T$	\$
$T \rightarrow \cdot (E)$	\$



LR(1) Parsing: The Intuition

$S \rightarrow E$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$

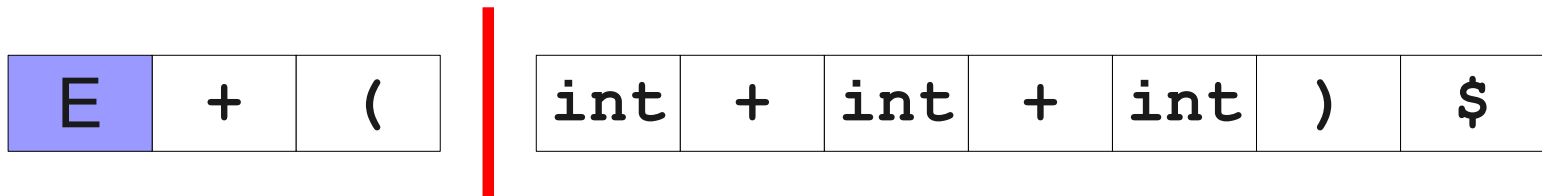
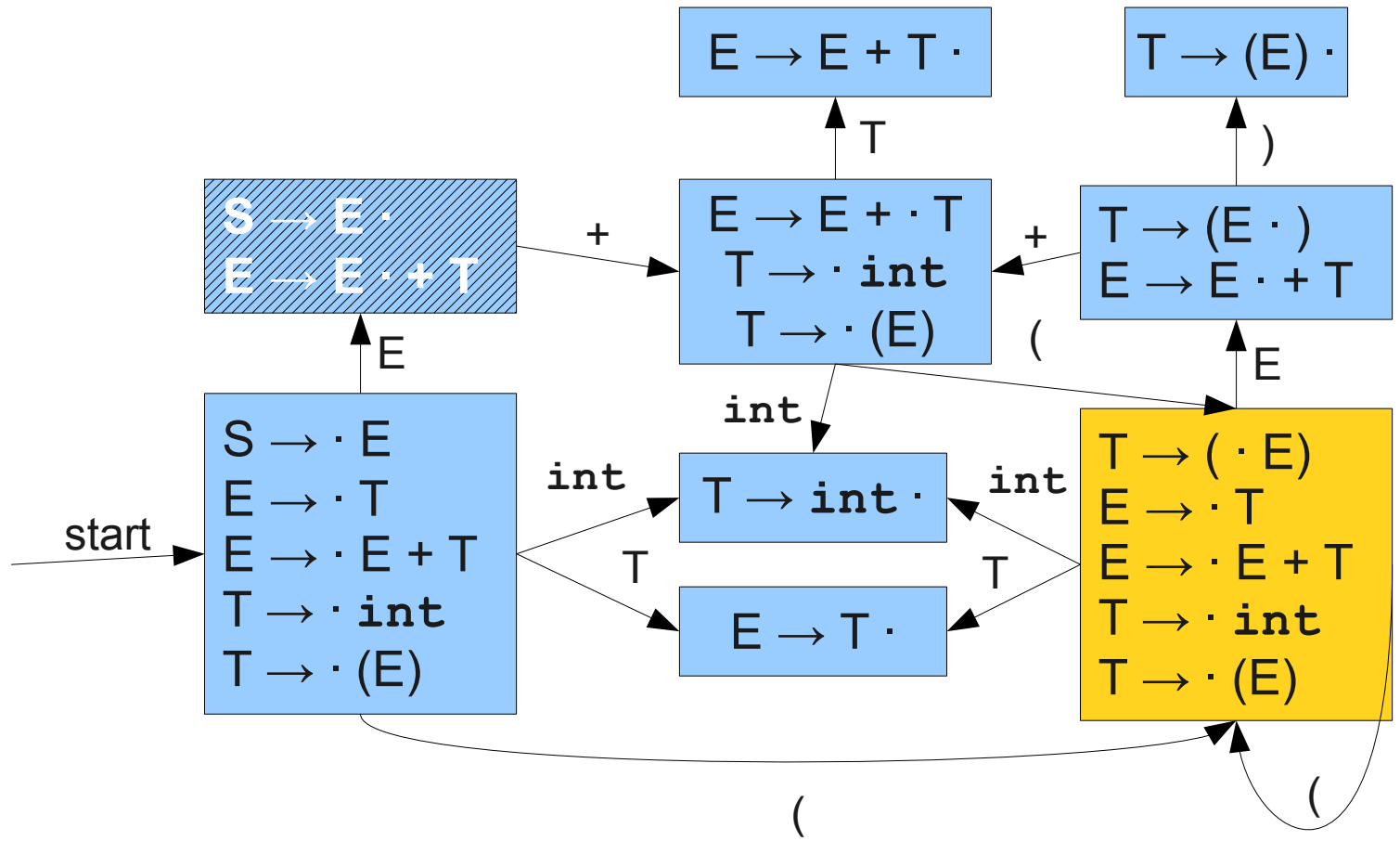
$S \rightarrow \cdot E$	\$
$E \rightarrow E + \cdot T$	\$
$T \rightarrow (\cdot E)$	\$



LR(1) Parsing: The Intuition

$S \rightarrow E$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$

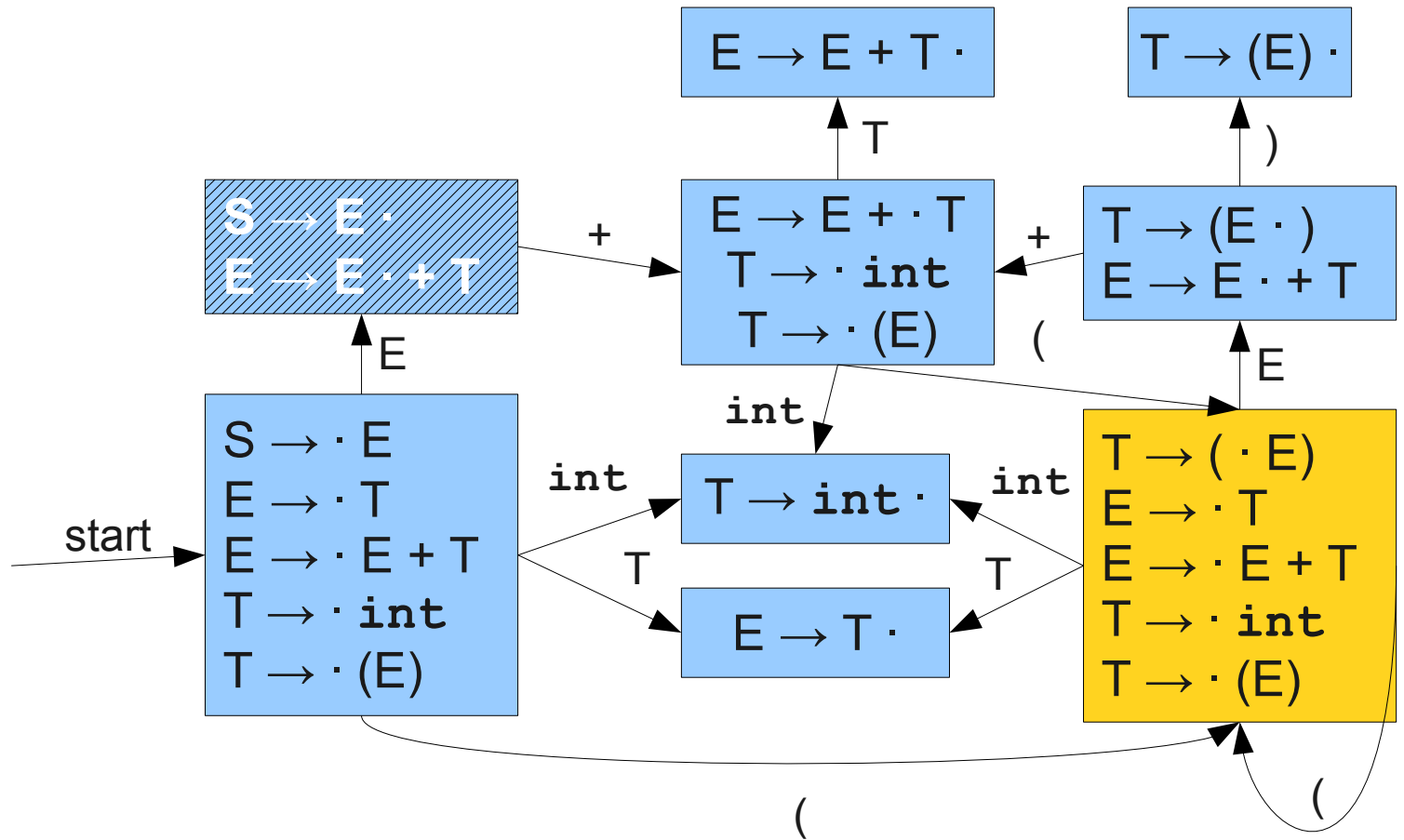
$S \rightarrow \cdot E$	\$
$E \rightarrow E + \cdot T$	\$
$T \rightarrow (\cdot E)$	\$
$E \rightarrow \cdot E + T$)



LR(1) Parsing: The Intuition

$S \rightarrow E$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$

$S \rightarrow \cdot E$	\$
$E \rightarrow E + \cdot T$	\$
$T \rightarrow (\cdot E)$	\$
$E \rightarrow \cdot E + T$)
$E \rightarrow \cdot T$	+



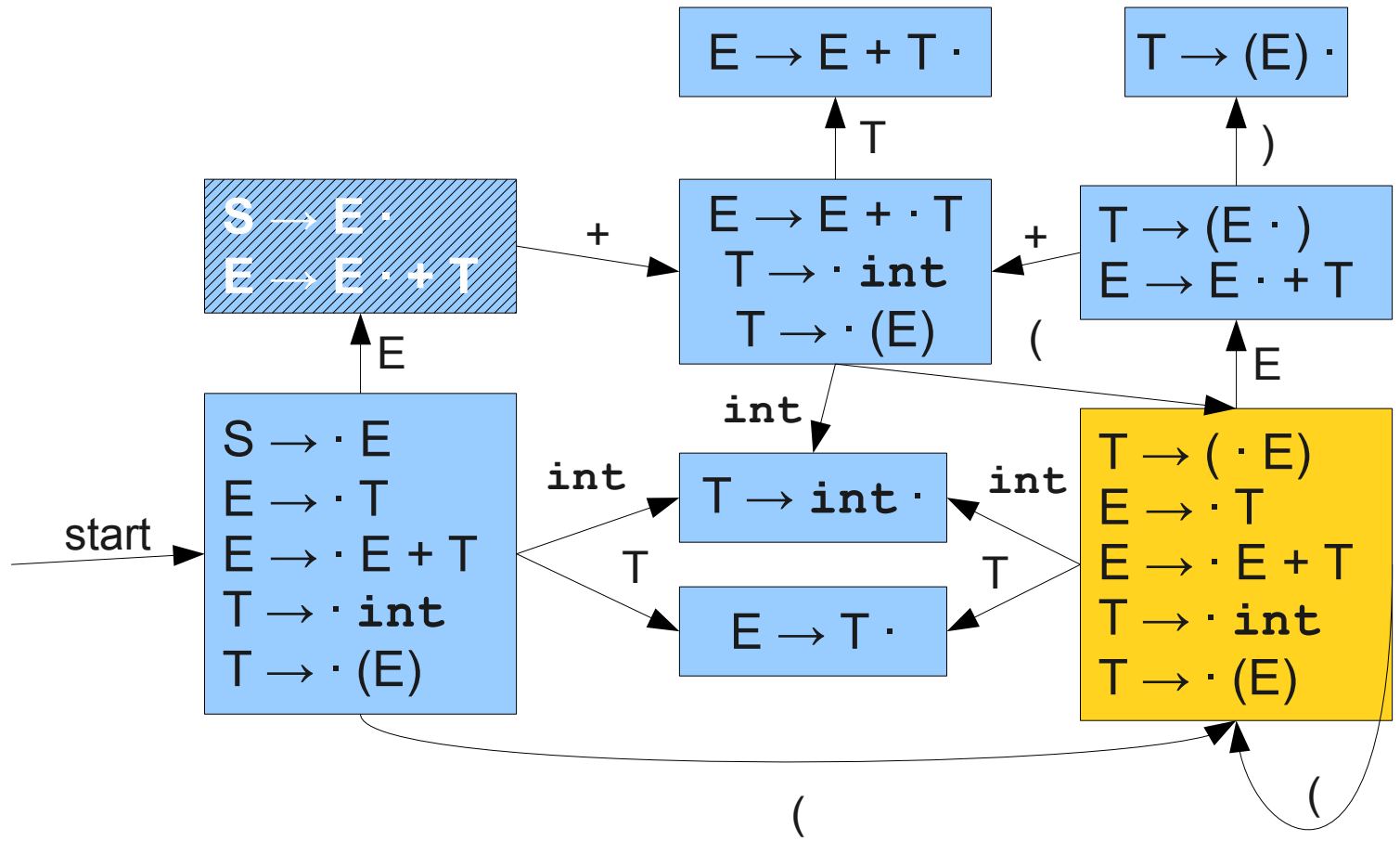
E + (

| int + int + int) \$

LR(1) Parsing: The Intuition

$S \rightarrow E$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$

$S \rightarrow \cdot E$	\$
$E \rightarrow E + \cdot T$	\$
$T \rightarrow (\cdot E)$	\$
$E \rightarrow \cdot E + T$)
$E \rightarrow \cdot T$	+
$T \rightarrow \cdot \text{int}$	+



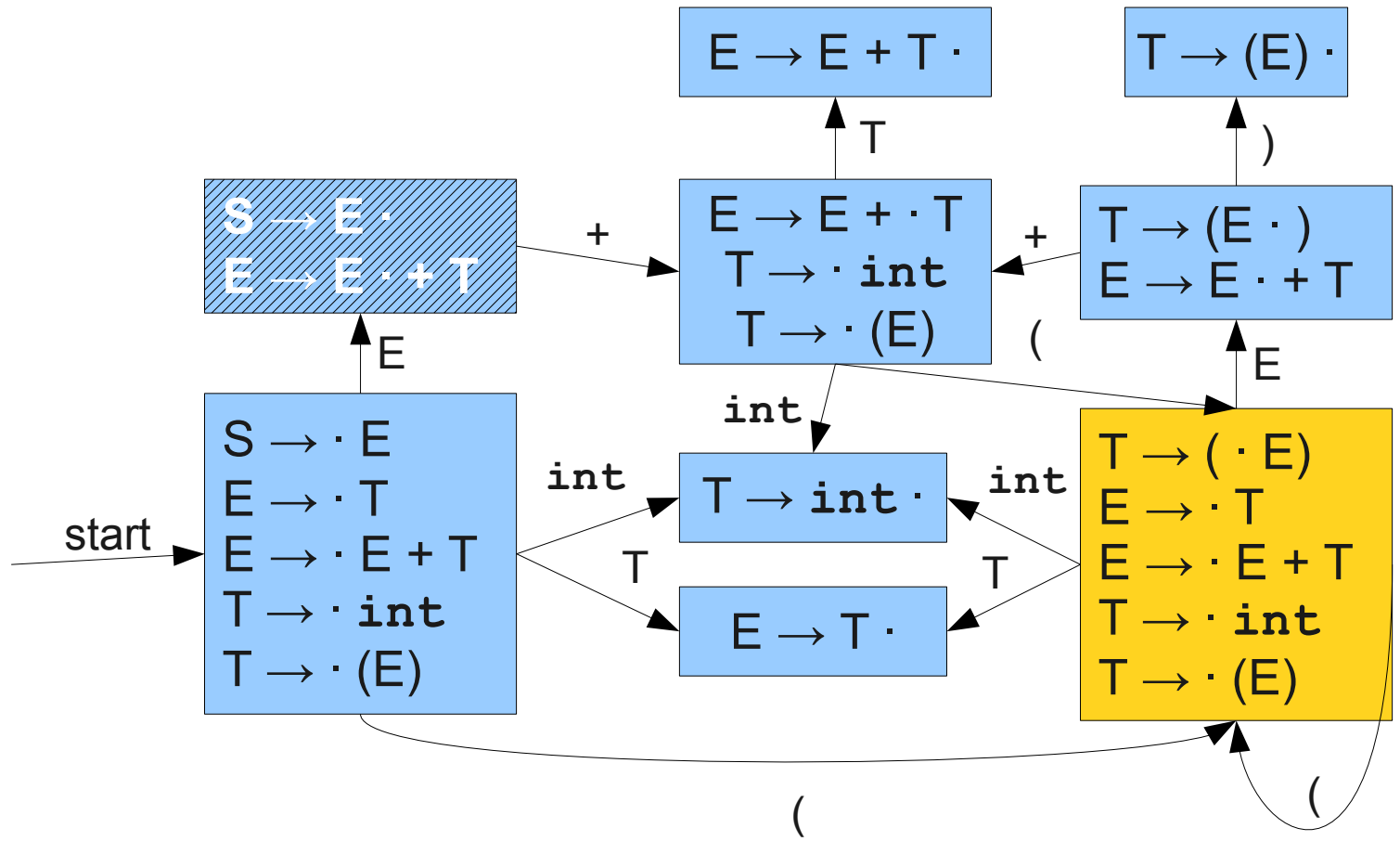
E + (

int + int + int) \$

LR(1) Parsing: The Intuition

$S \rightarrow E$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$

$S \rightarrow \cdot E$	\$
$E \rightarrow E + \cdot T$	\$
$T \rightarrow (\cdot E)$	\$
$E \rightarrow \cdot E + T$)
$E \rightarrow \cdot T$	+
$T \rightarrow \cdot \text{int}$	+



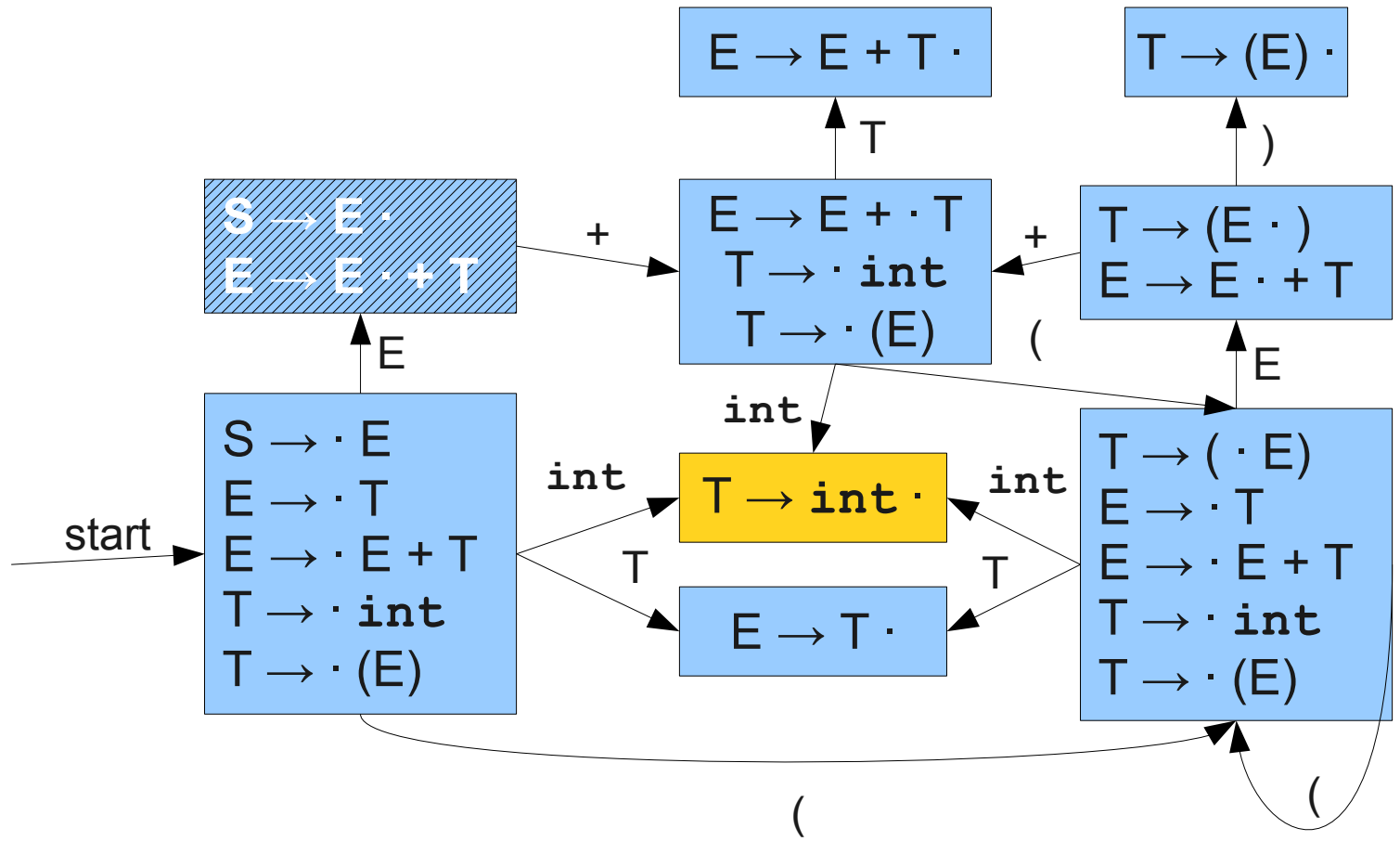
E + (int

+ int + int) \$

LR(1) Parsing: The Intuition

$S \rightarrow E$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$

$S \rightarrow \cdot E$	\$
$E \rightarrow E + \cdot T$	\$
$T \rightarrow (\cdot E)$	\$
$E \rightarrow \cdot E + T$)
$E \rightarrow \cdot T$	+
$T \rightarrow \text{int} \cdot$	+



E + (int

+ int + int) \$

The Intuition behind LR(1)

- Guess which series of productions we are reversing.
- Use this information to maintain information about what lookahead to expect.
- When deciding whether to shift or reduce, use lookahead to disambiguate.

Tracking Lookaheads

- How do we know what lookahead to expect at each state?
- Observation:
 - There are only finitely many productions we can be in at any point.
 - There are only finitely many positions we can be in each production.
 - **There are only finitely many lookahead sets** at each point.
- Construct an automaton to track lookaheads!

Constructing LR(1) Automata

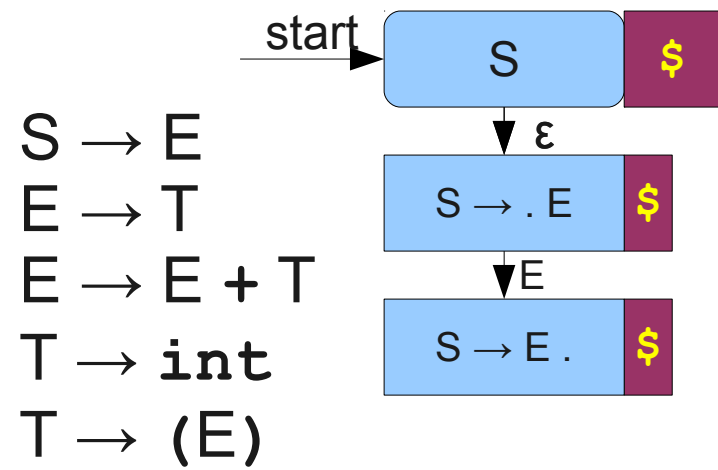
$S \rightarrow E$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \mathbf{int}$
 $T \rightarrow (E)$

Constructing LR(1) Automata

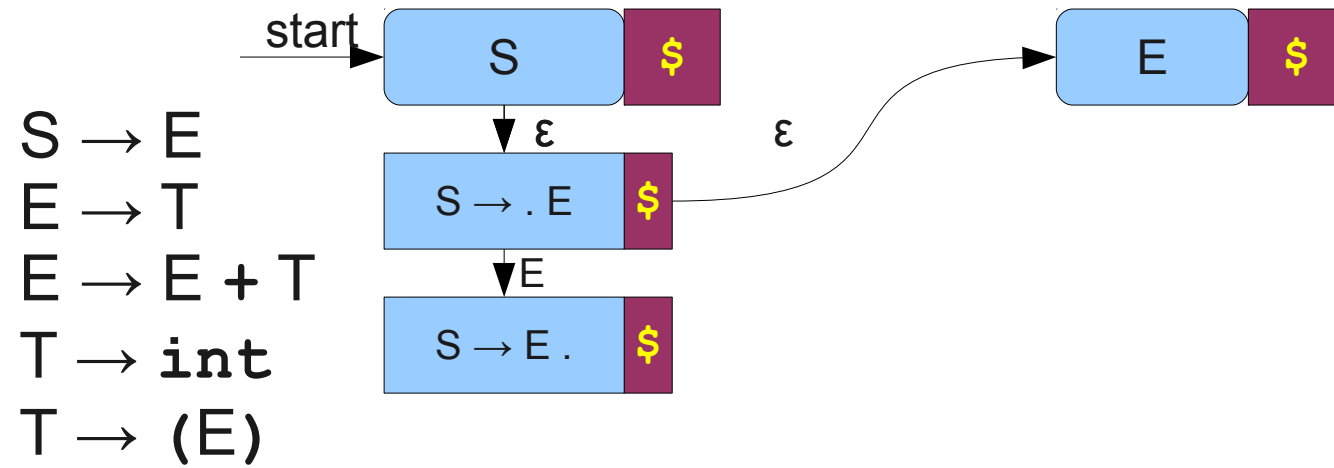


$S \rightarrow E$
 $E \rightarrow T$
 $E \rightarrow E + T$
 $T \rightarrow \mathbf{int}$
 $T \rightarrow (E)$

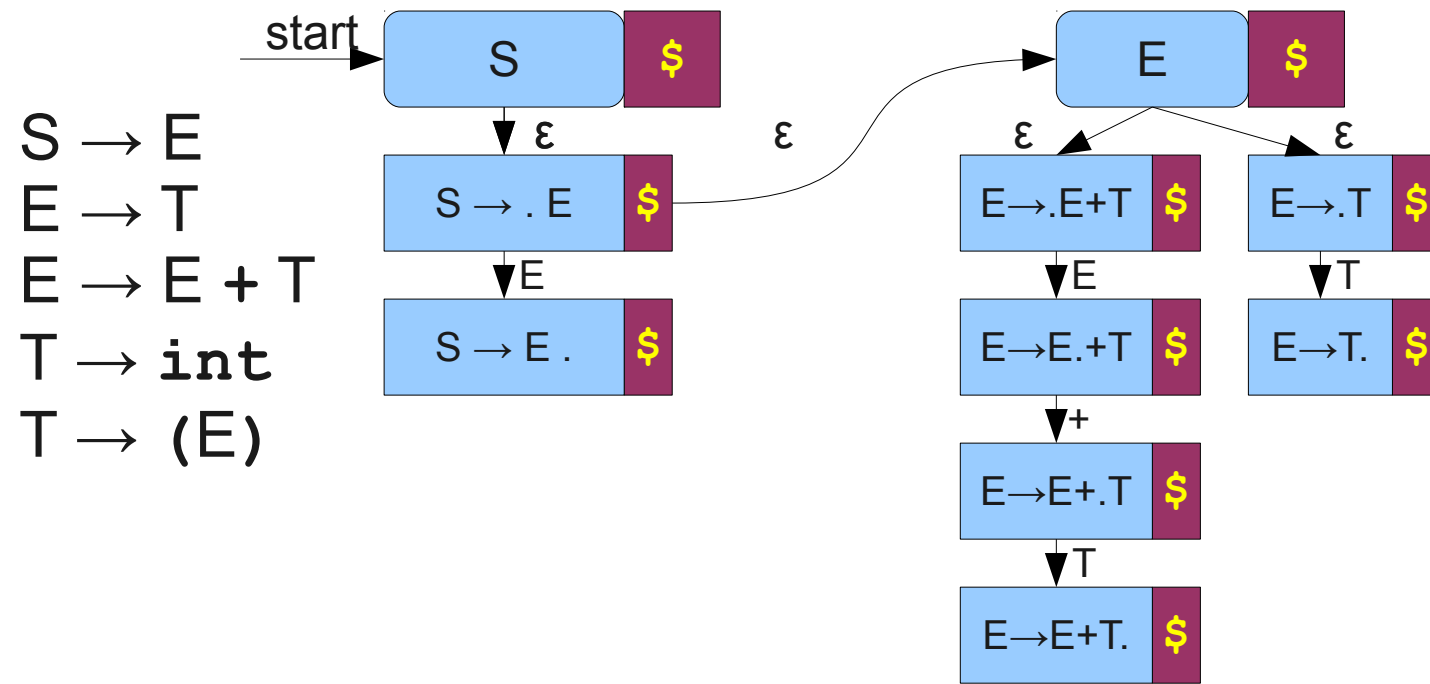
Constructing LR(1) Automata



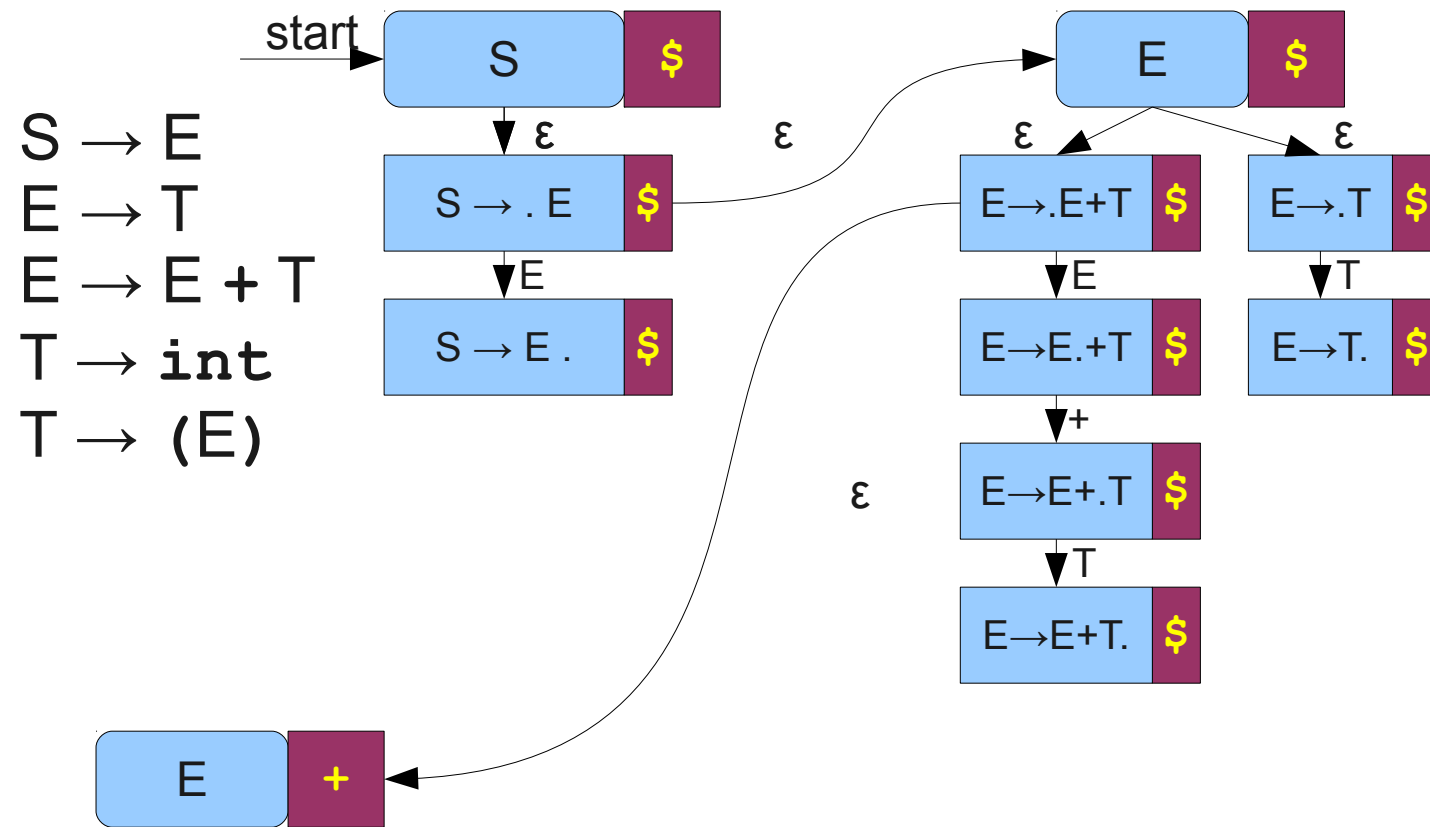
Constructing LR(1) Automata



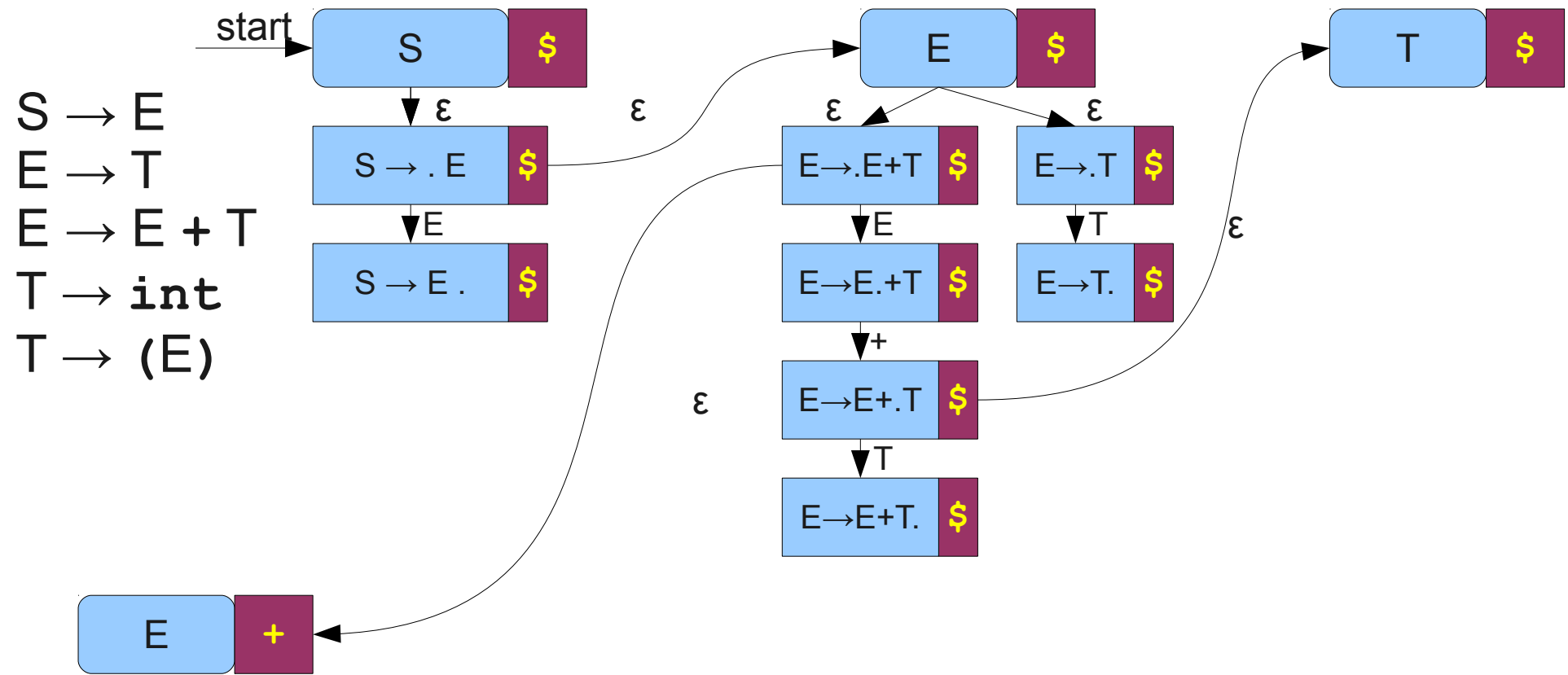
Constructing LR(1) Automata



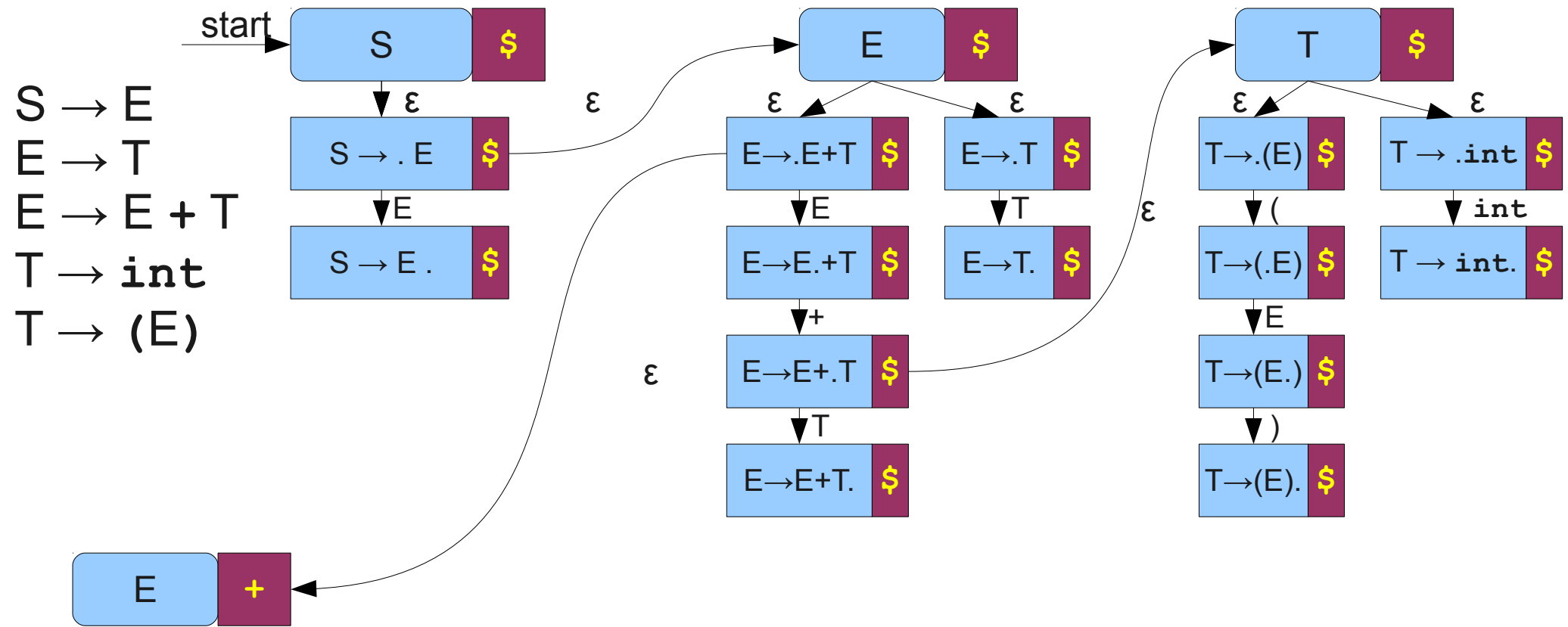
Constructing LR(1) Automata



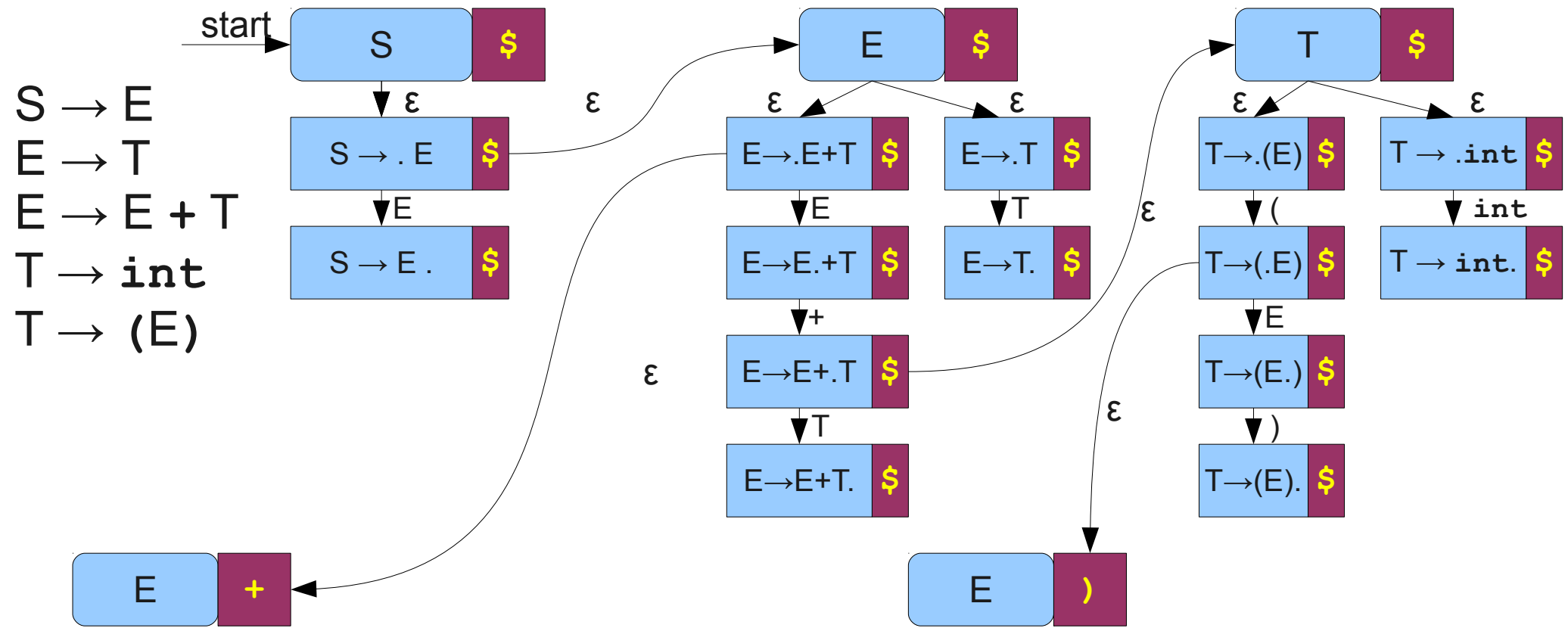
Constructing LR(1) Automata



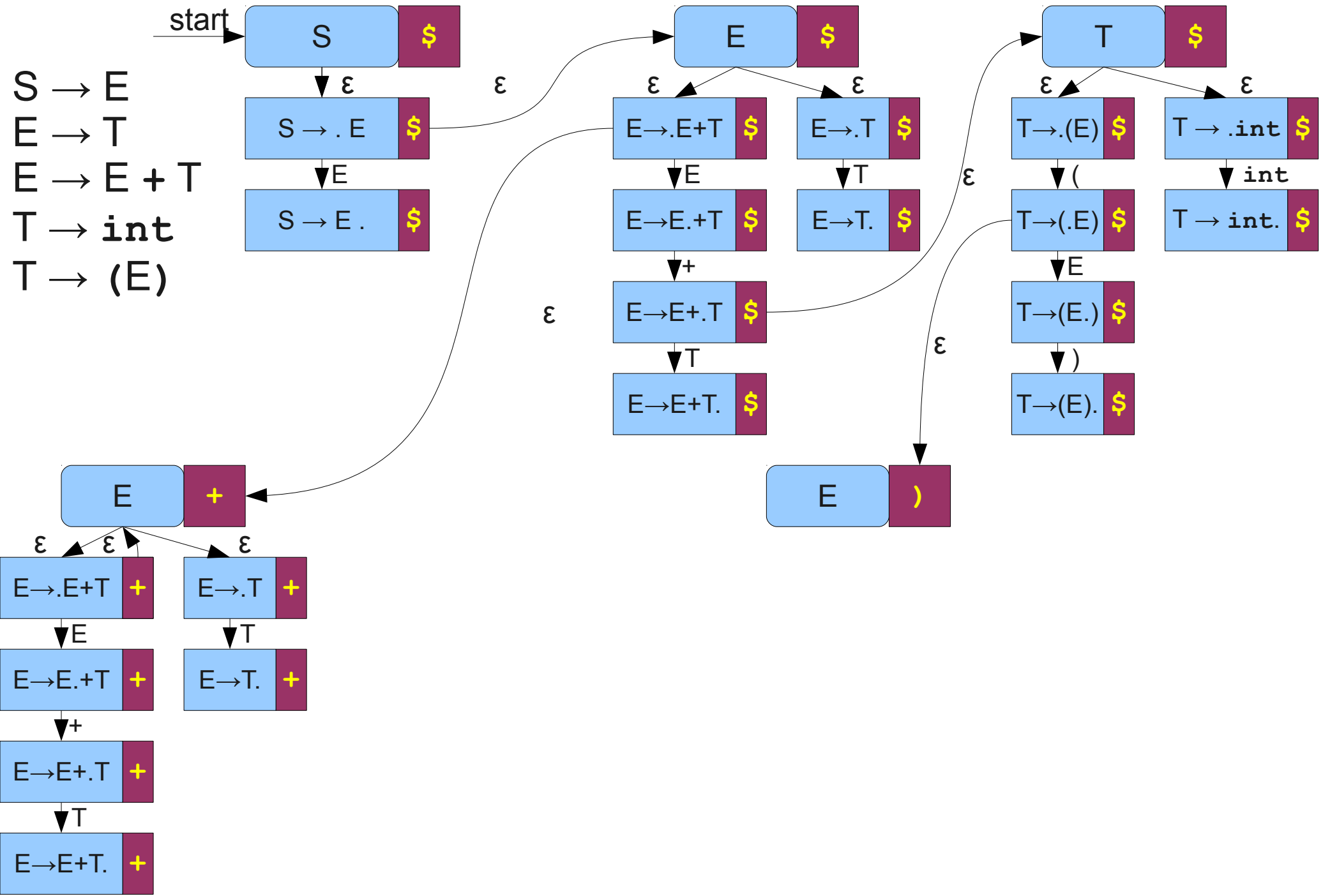
Constructing LR(1) Automata



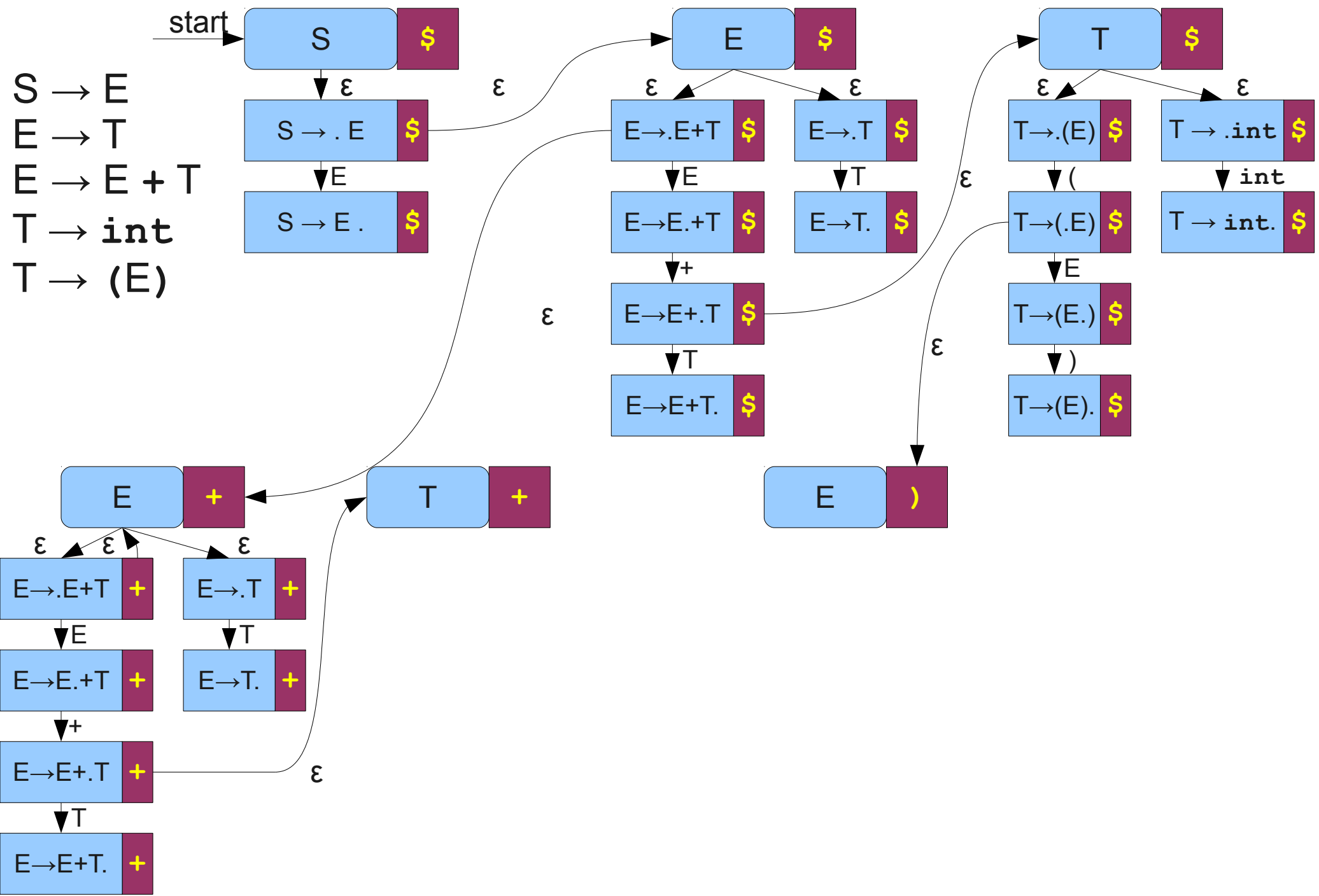
Constructing LR(1) Automata



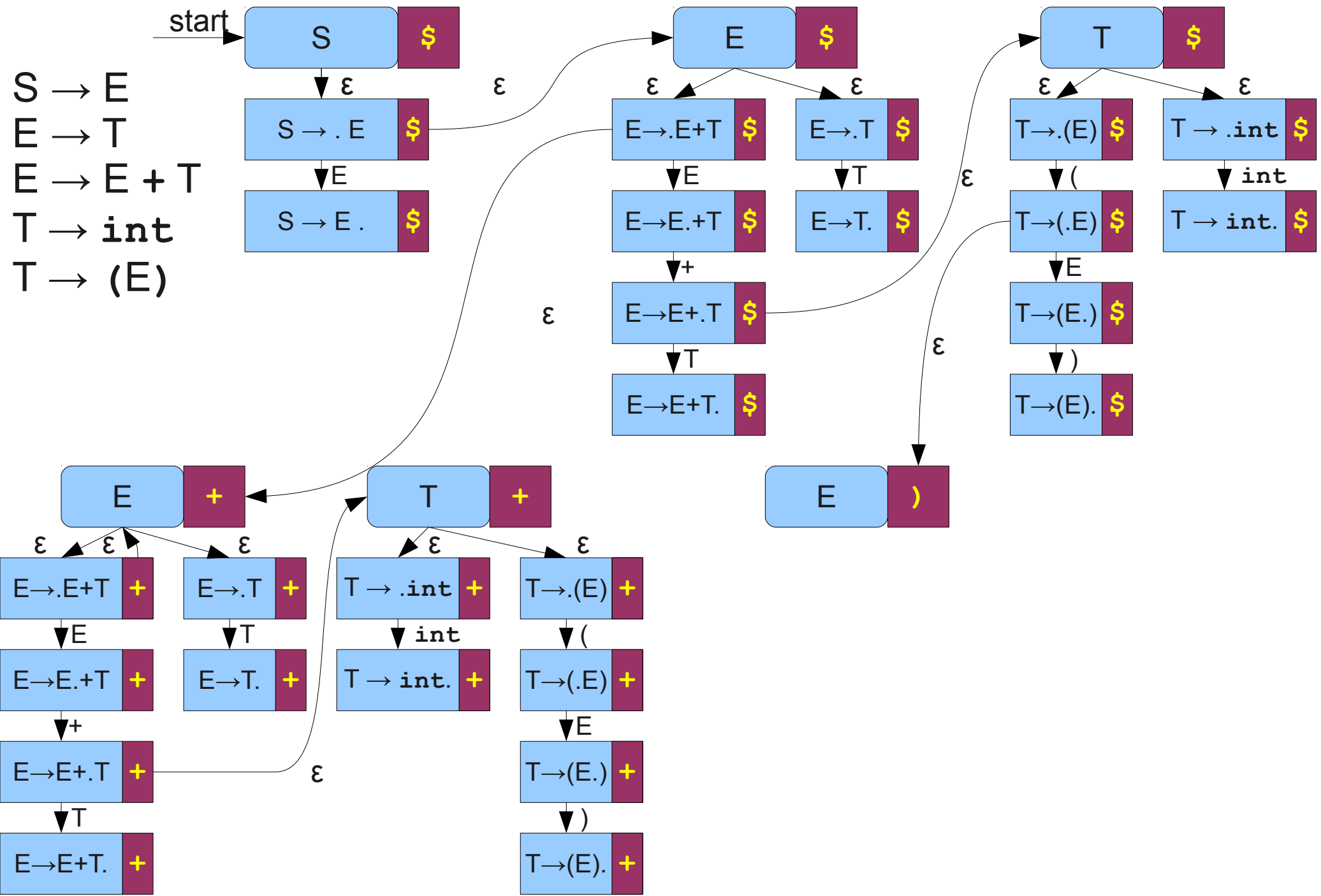
Constructing LR(1) Automata



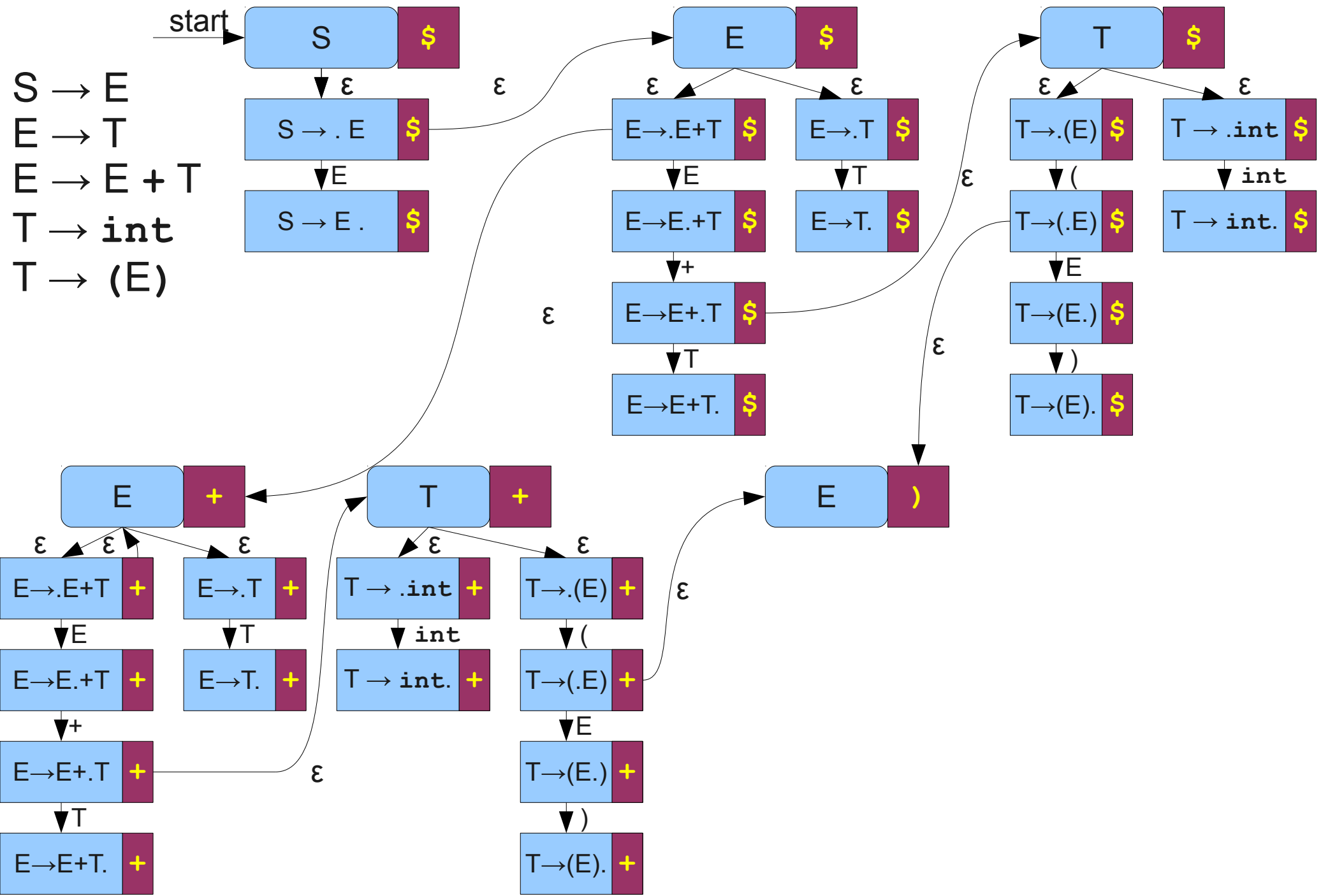
Constructing LR(1) Automata



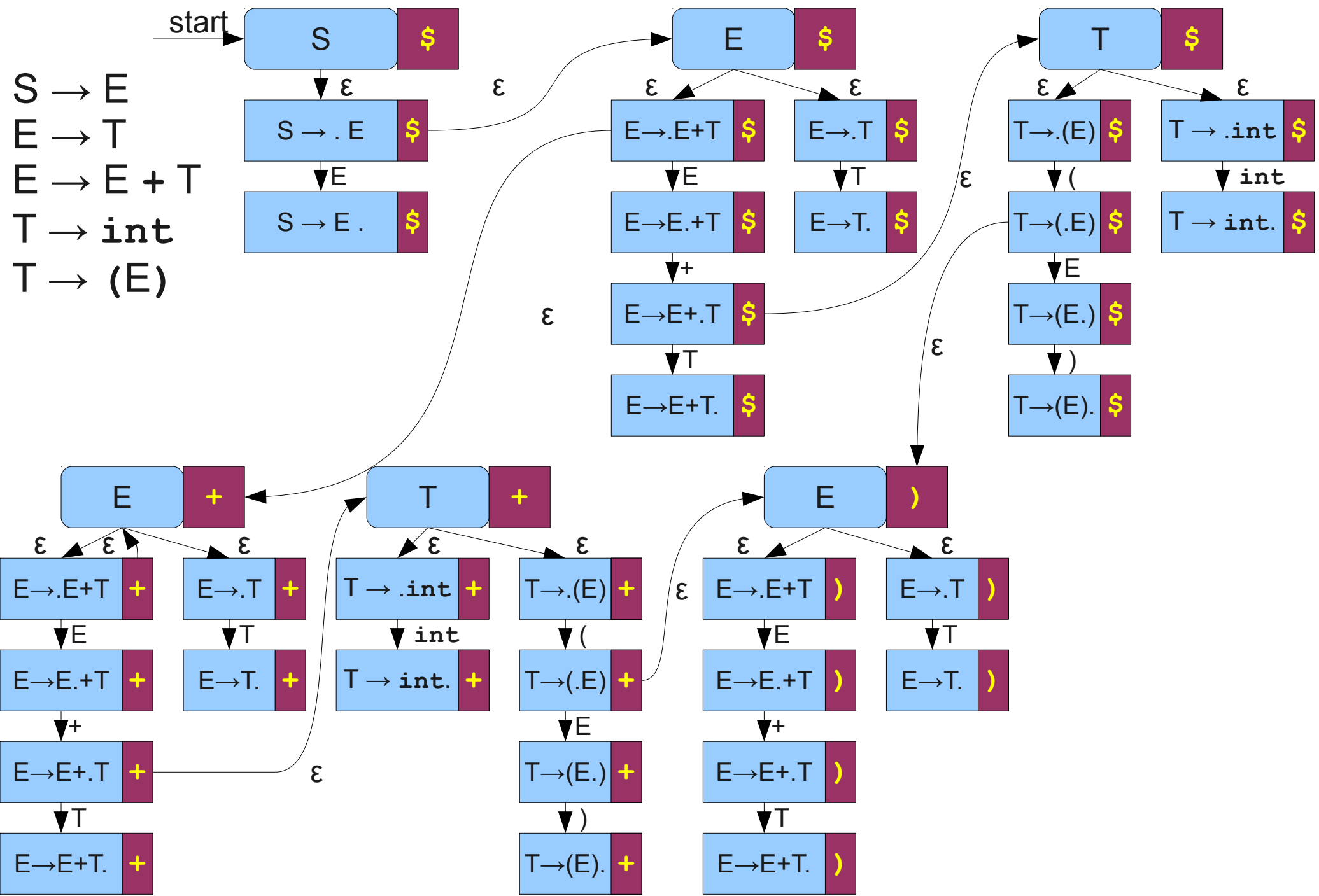
Constructing LR(1) Automata



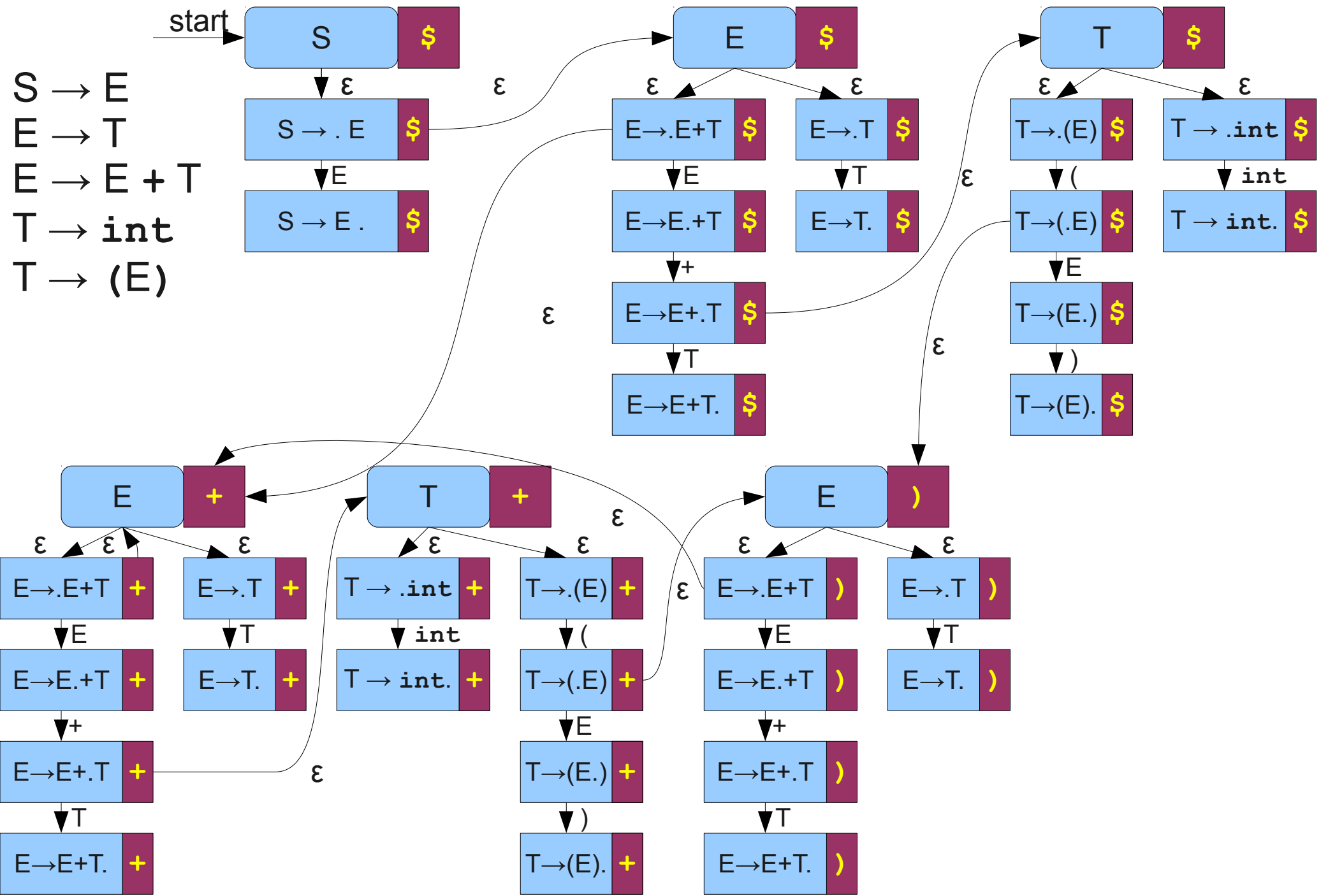
Constructing LR(1) Automata



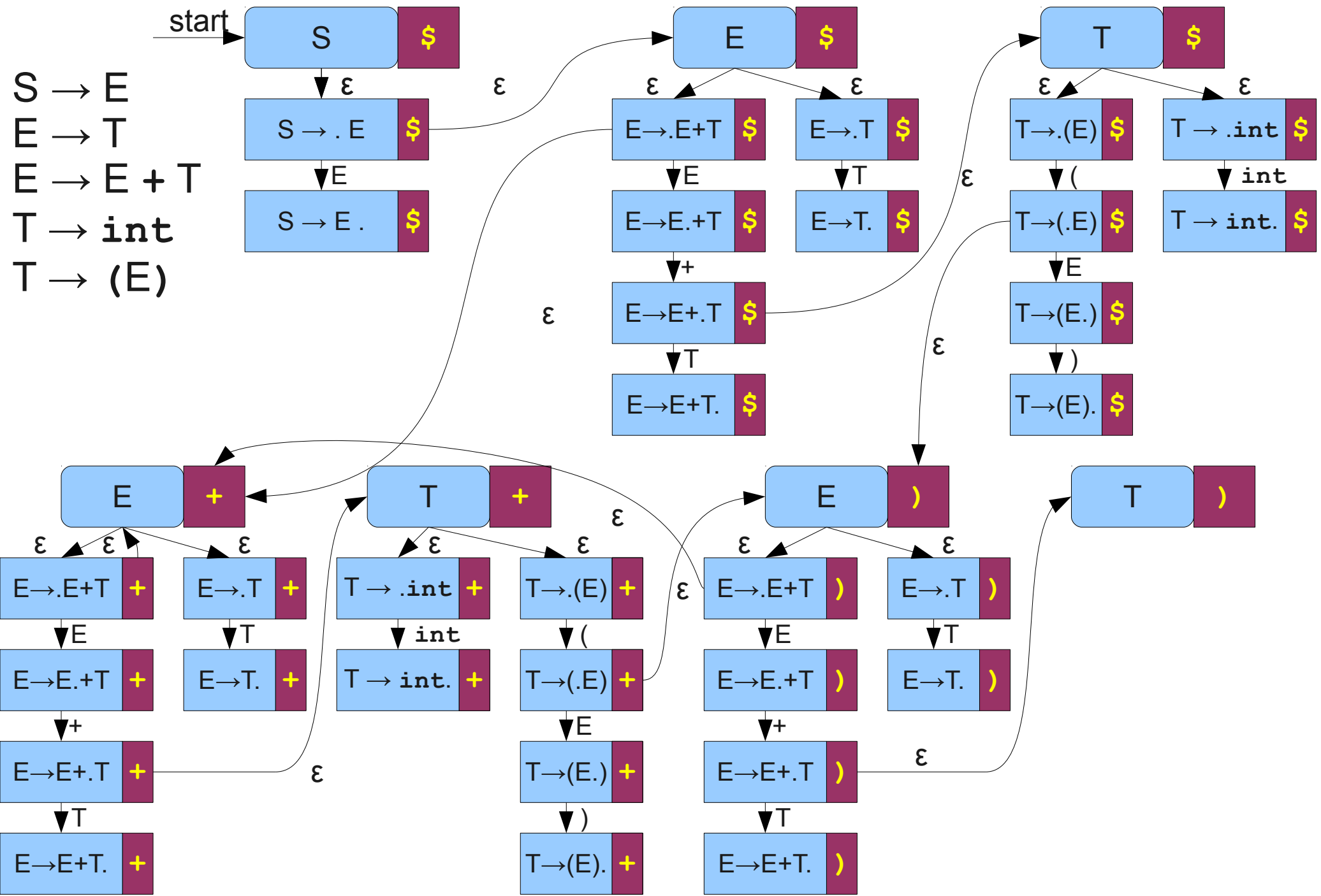
Constructing LR(1) Automata



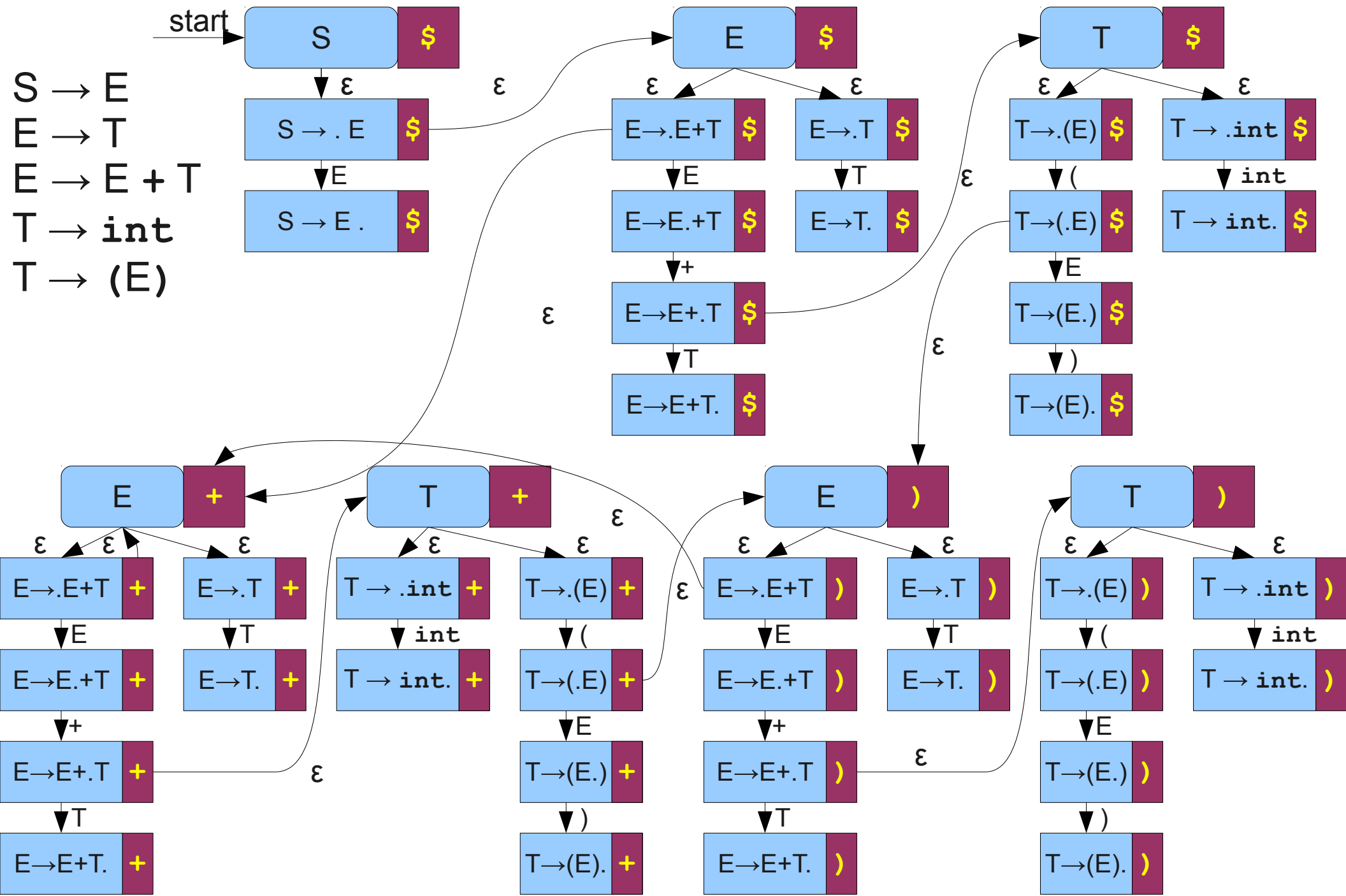
Constructing LR(1) Automata



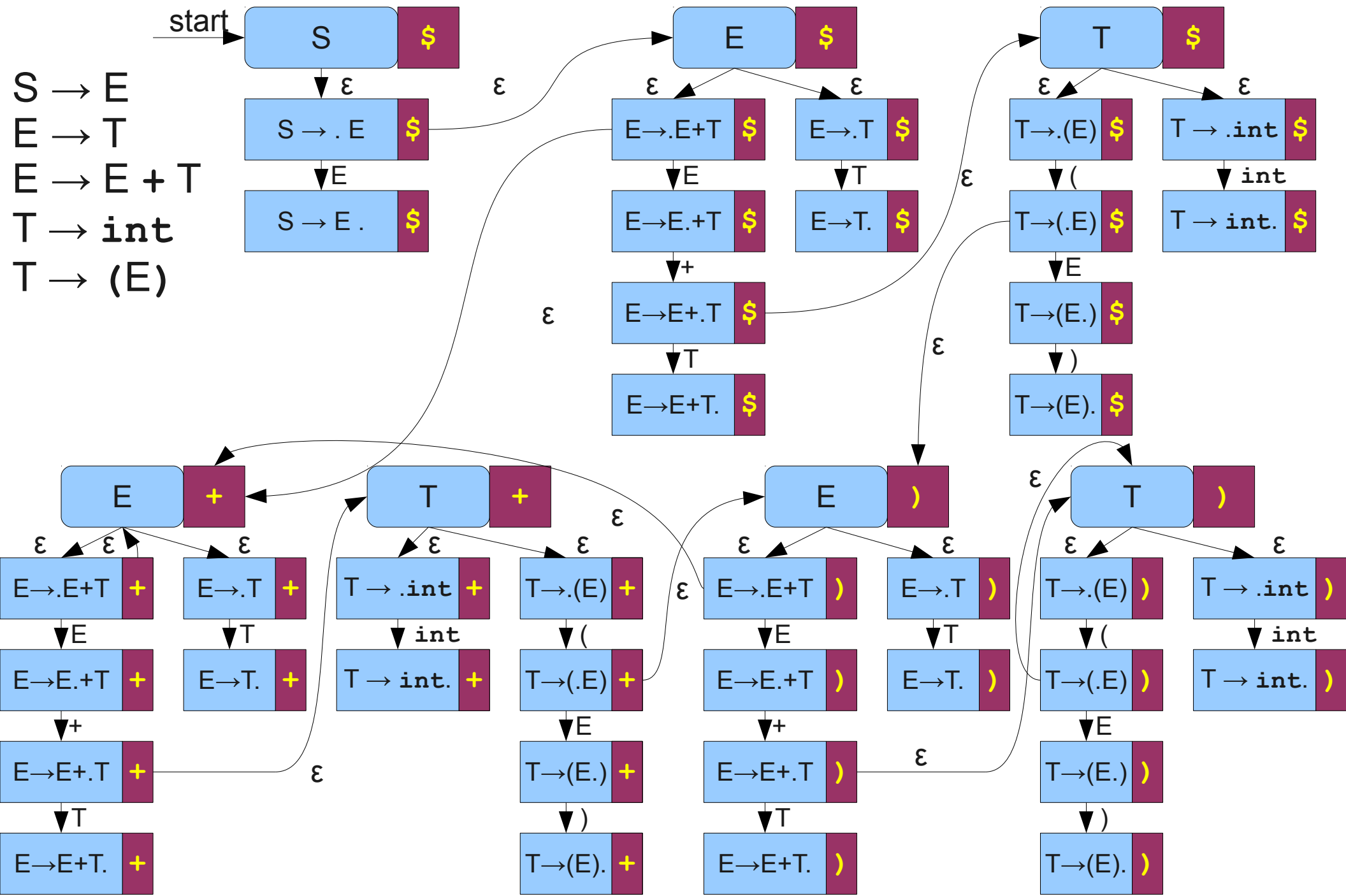
Constructing LR(1) Automata



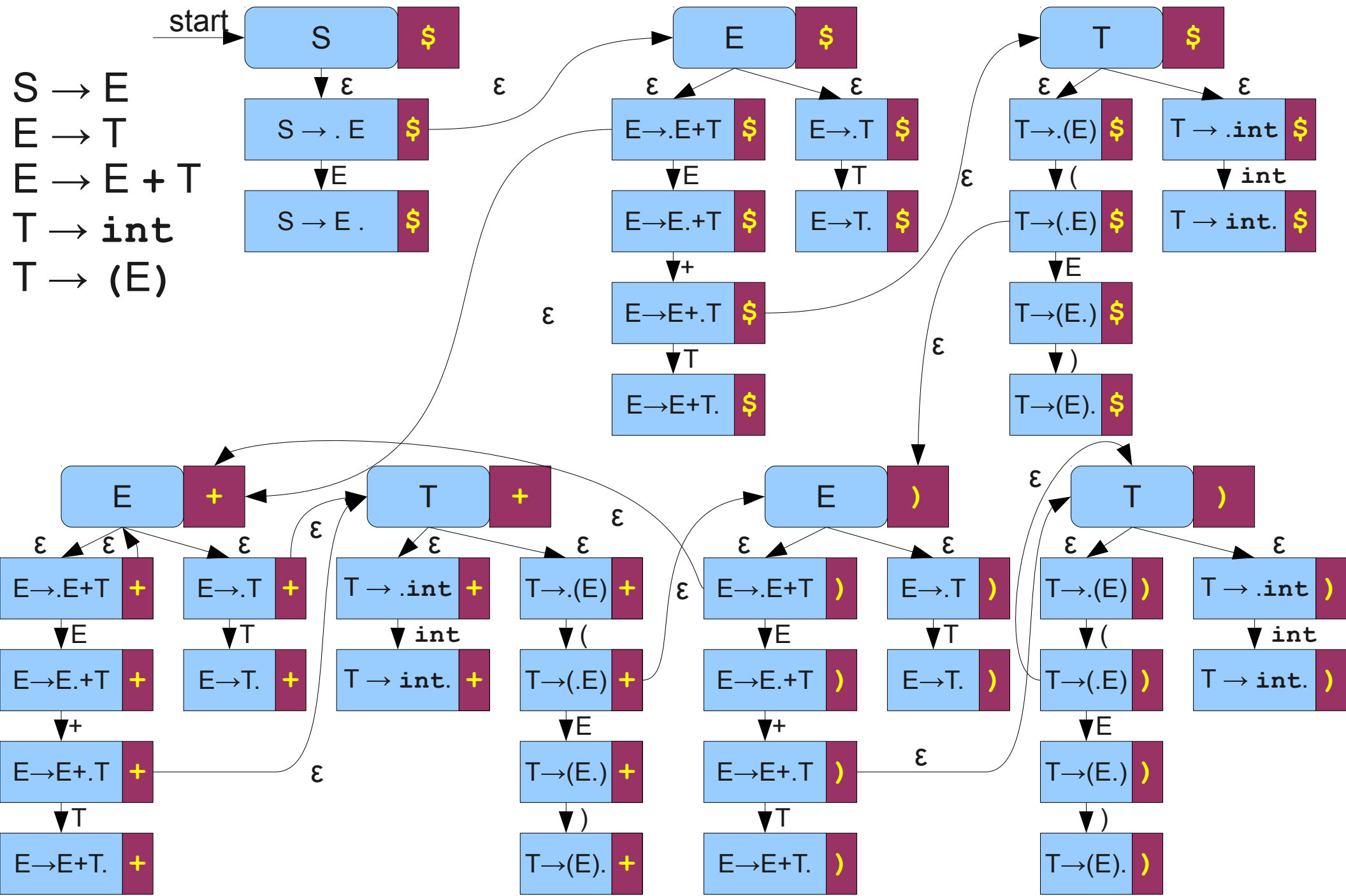
Constructing LR(1) Automata



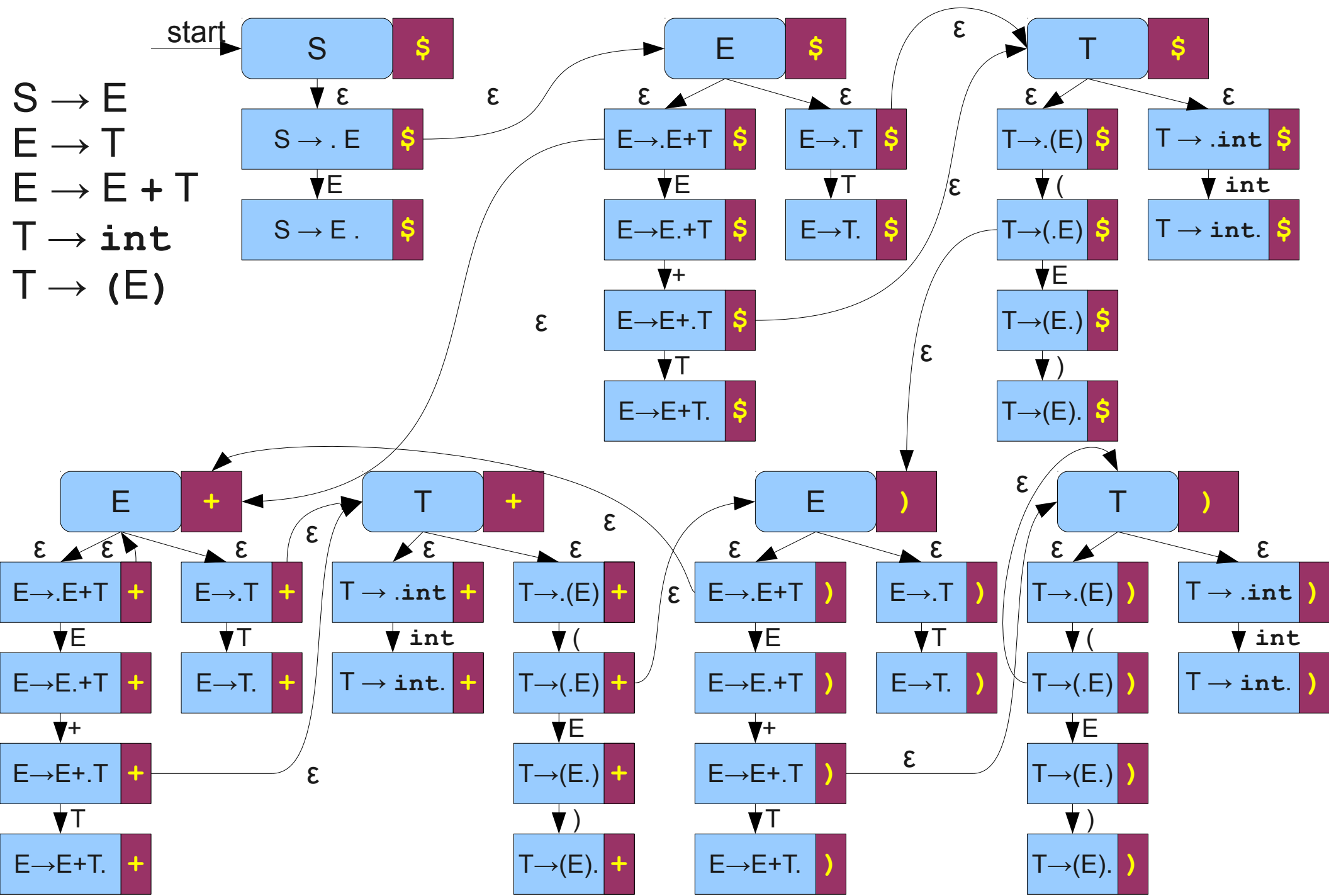
Constructing LR(1) Automata



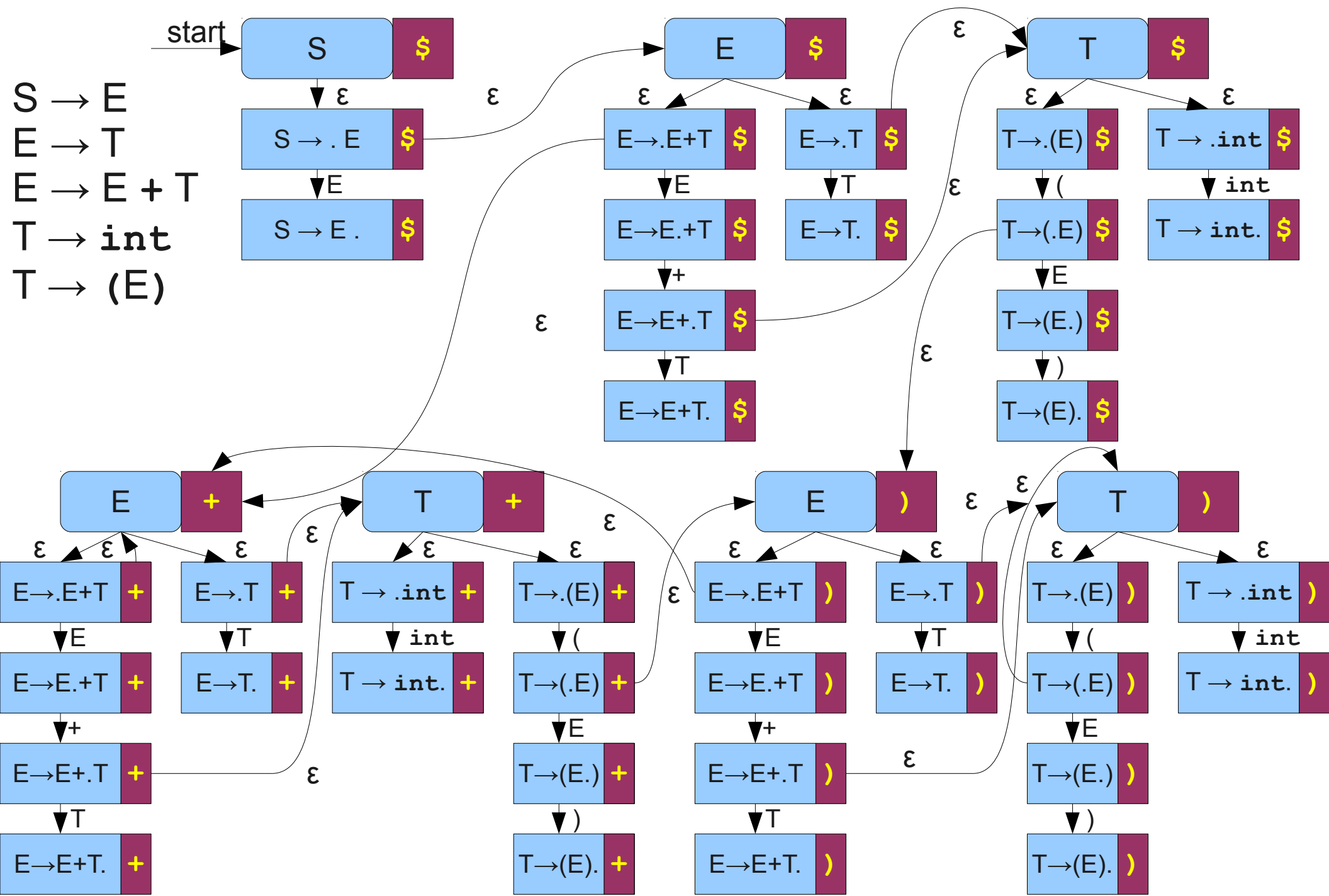
Constructing LR(1) Automata



Constructing LR(1) Automata



Constructing LR(1) Automata

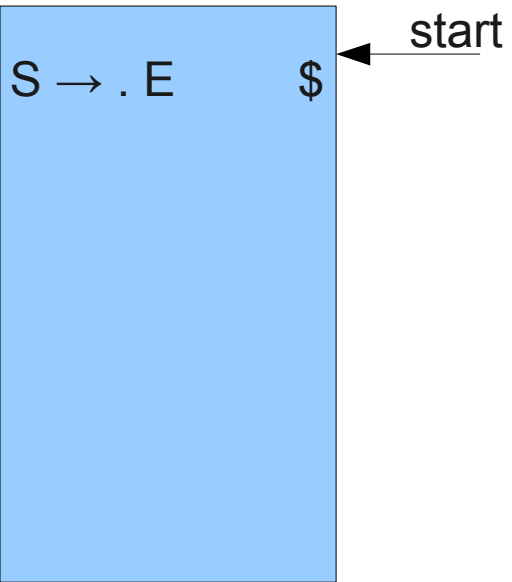


Constructing LR(1) Automata

- Begin with a state $S [\$]$.
- For each state $A [t]$, for each production $A \rightarrow a_1 a_2 \dots a_n$:
 - Construct states $A \rightarrow a_1 a_2 \dots a_i \cdot a_{i+1} \dots a_n [t]$ for $0 \leq i \leq n$.
 - Add an ε -transition from $A [t]$ to each of these states.
 - Add transitions on x between $A \rightarrow u \cdot xv [t]$ and $A \rightarrow ux \cdot v [t]$
- For each state $A \rightarrow u \cdot Bv [t]$ for nonterminal B , add an ε -transition from $A \rightarrow u \cdot Bv [t]$ to $B [r]$ for each terminal r in $\text{FIRST}(tv)$

Building Deterministic LR(1) Automata

Building Deterministic LR(1) Automata



Building Deterministic LR(1) Automata

$S \rightarrow . E$	\$	← start
$E \rightarrow . T$	\$	
$E \rightarrow . E + T$	\$	

Building Deterministic LR(1) Automata

$S \rightarrow . E$	\$	← start
$E \rightarrow . T$	\$	
$E \rightarrow . E + T$	\$	
$E \rightarrow . T$	+	
$E \rightarrow . E + T$	+	

Building Deterministic LR(1) Automata

$S \rightarrow . E$	\$
$E \rightarrow . T$	\$
$E \rightarrow . E + T$	\$
$E \rightarrow . T$	+
$E \rightarrow . E + T$	+
$T \rightarrow . \text{int}$	\$
$T \rightarrow . (E)$	\$

← start

Building Deterministic LR(1) Automata

$S \rightarrow . E$	\$
$E \rightarrow . T$	\$
$E \rightarrow . E + T$	\$
$E \rightarrow . T$	+
$E \rightarrow . E + T$	+
$T \rightarrow . \text{int}$	\$
$T \rightarrow . (E)$	\$
$T \rightarrow . \text{int}$	+
$T \rightarrow . (E)$	+

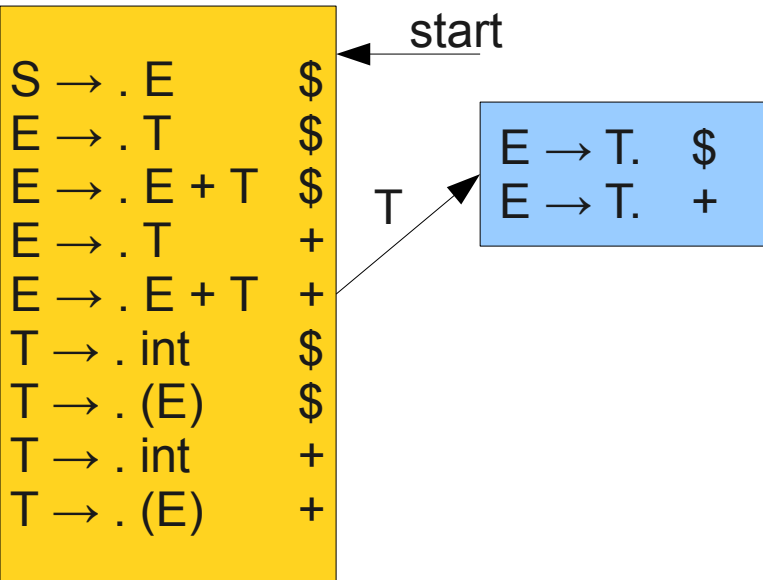
← start

Building Deterministic LR(1) Automata

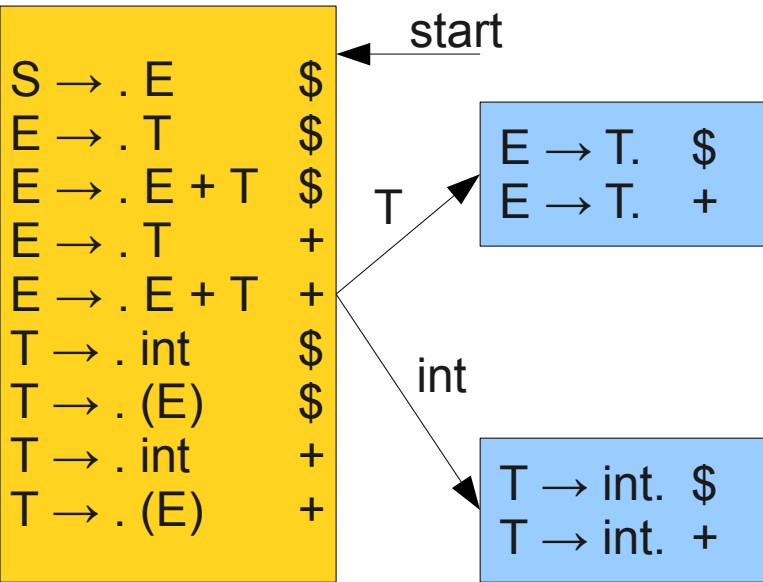
$S \rightarrow . E$	\$
$E \rightarrow . T$	\$
$E \rightarrow . E + T$	\$
$E \rightarrow . T$	+
$E \rightarrow . E + T$	+
$T \rightarrow . \text{int}$	\$
$T \rightarrow . (E)$	\$
$T \rightarrow . \text{int}$	+
$T \rightarrow . (E)$	+

← start

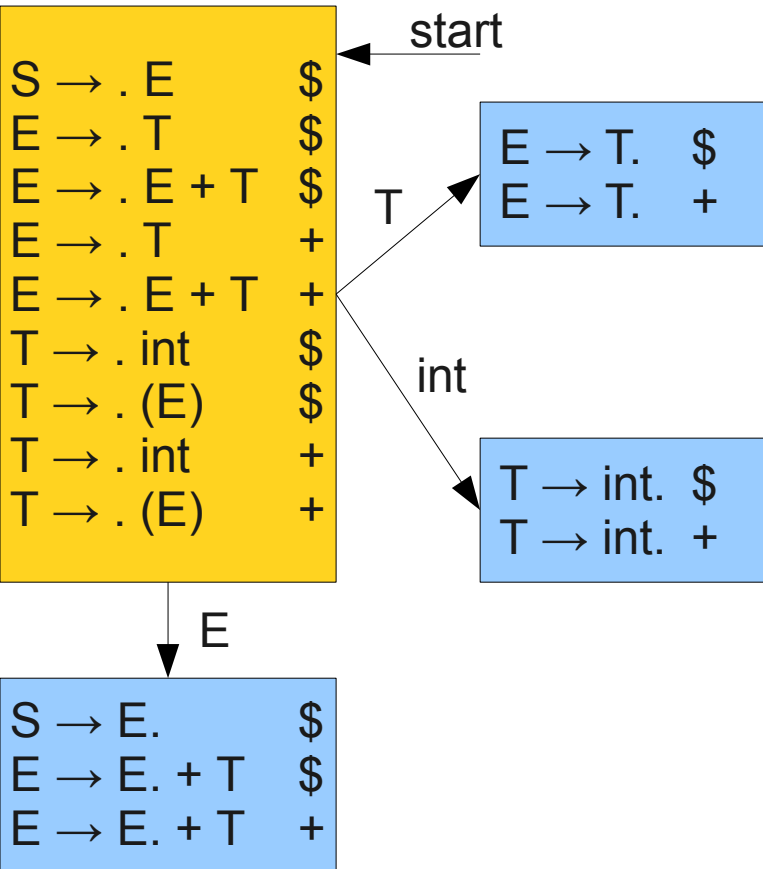
Building Deterministic LR(1) Automata



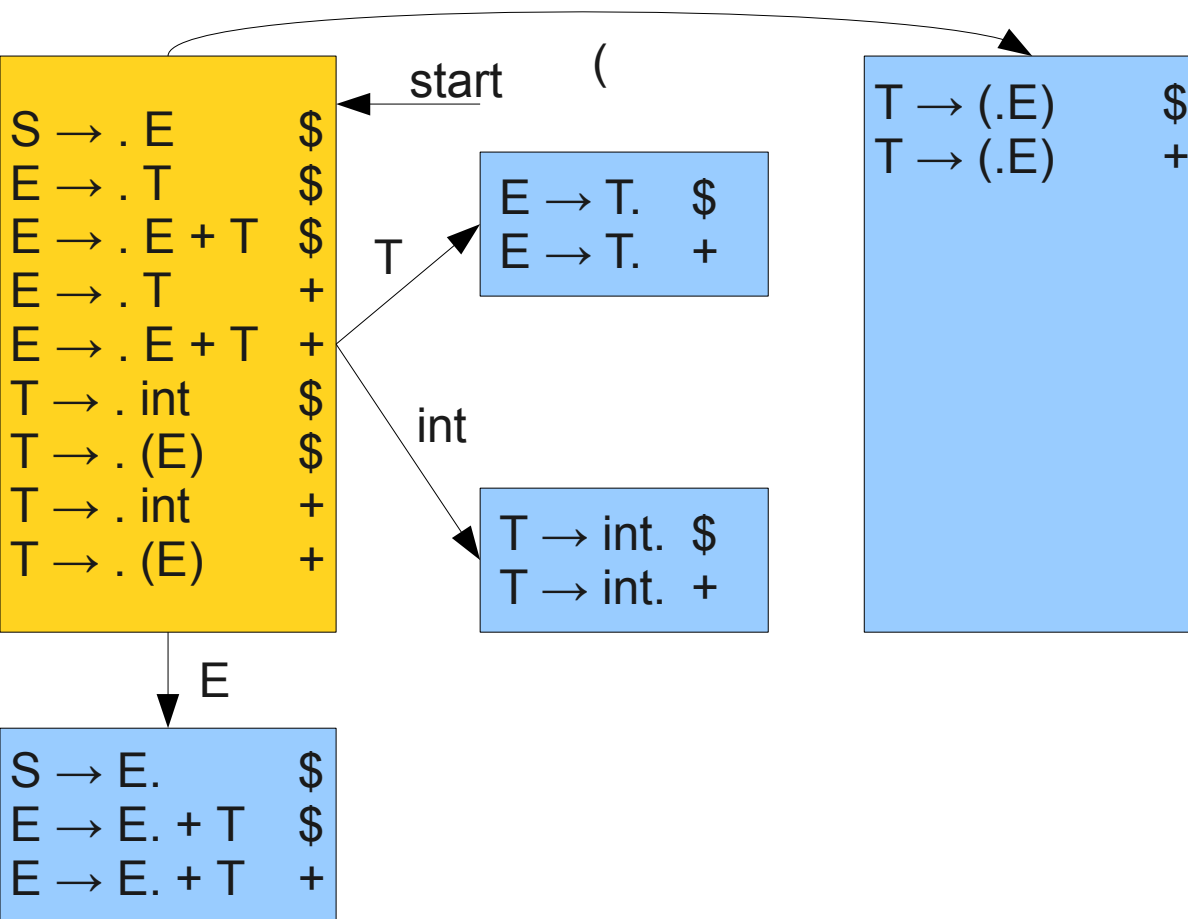
Building Deterministic LR(1) Automata



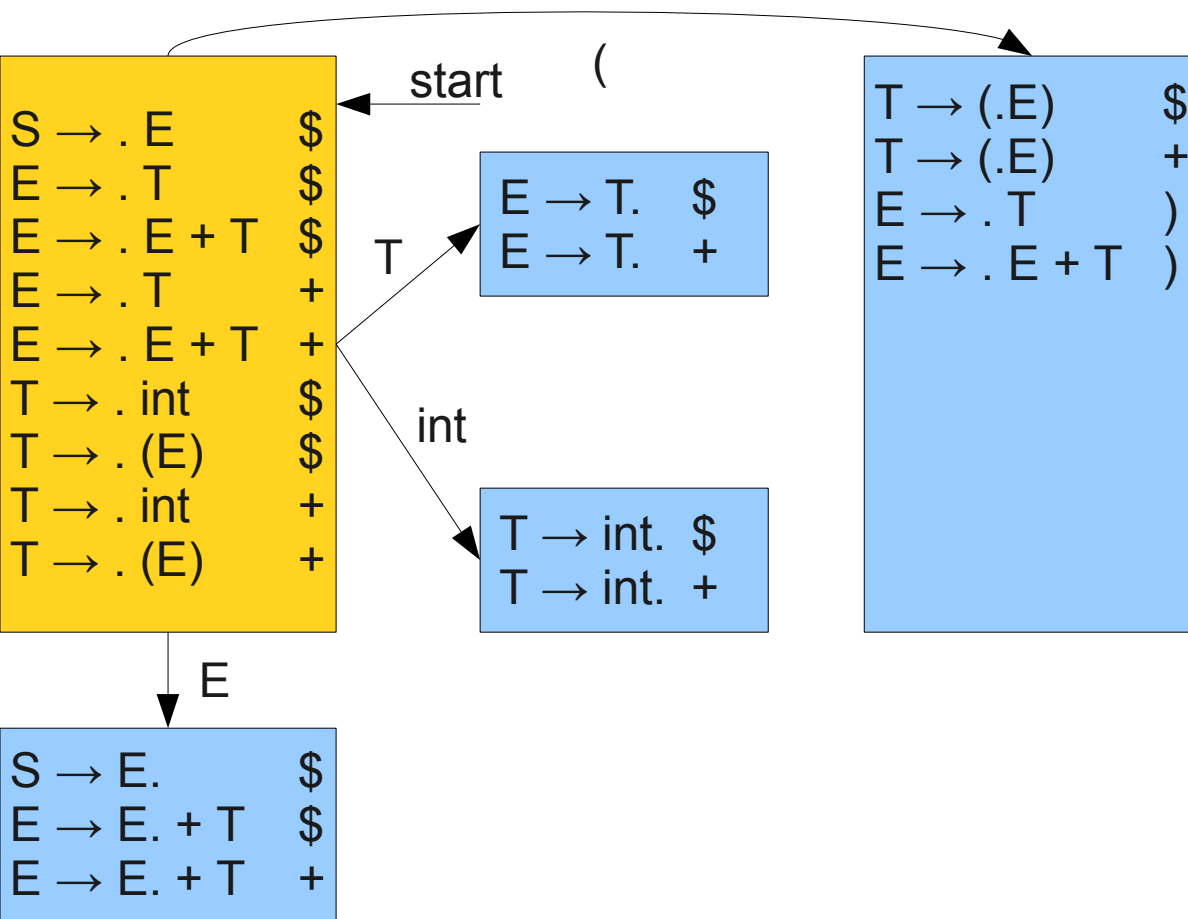
Building Deterministic LR(1) Automata



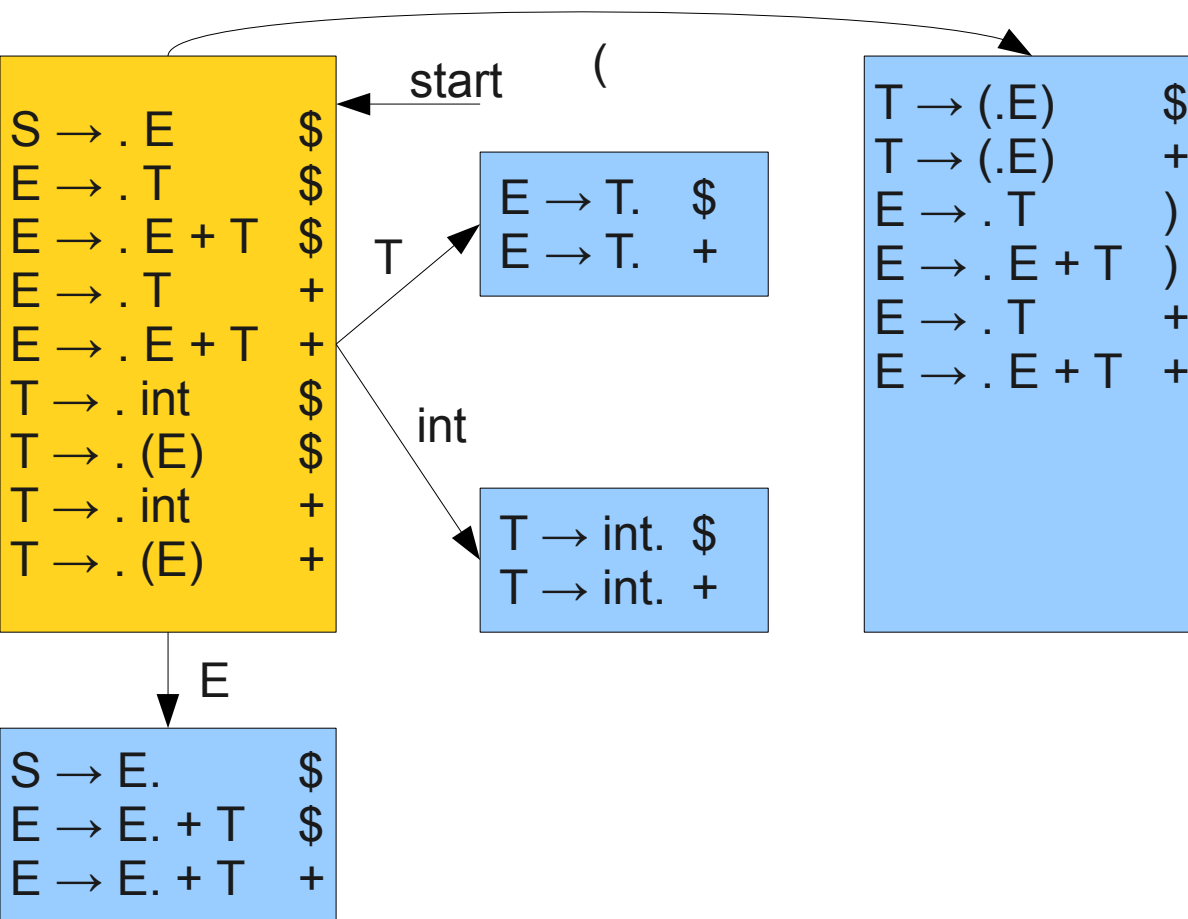
Building Deterministic LR(1) Automata



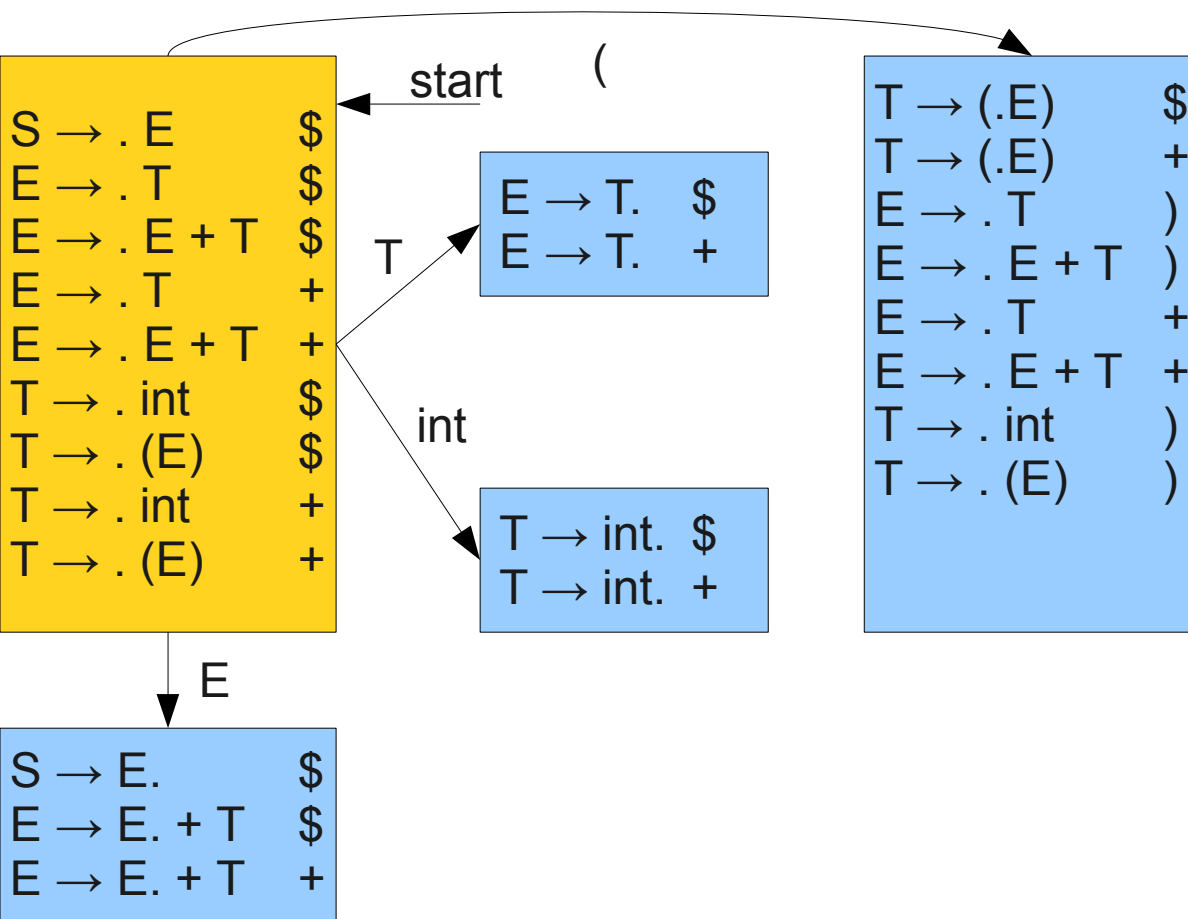
Building Deterministic LR(1) Automata



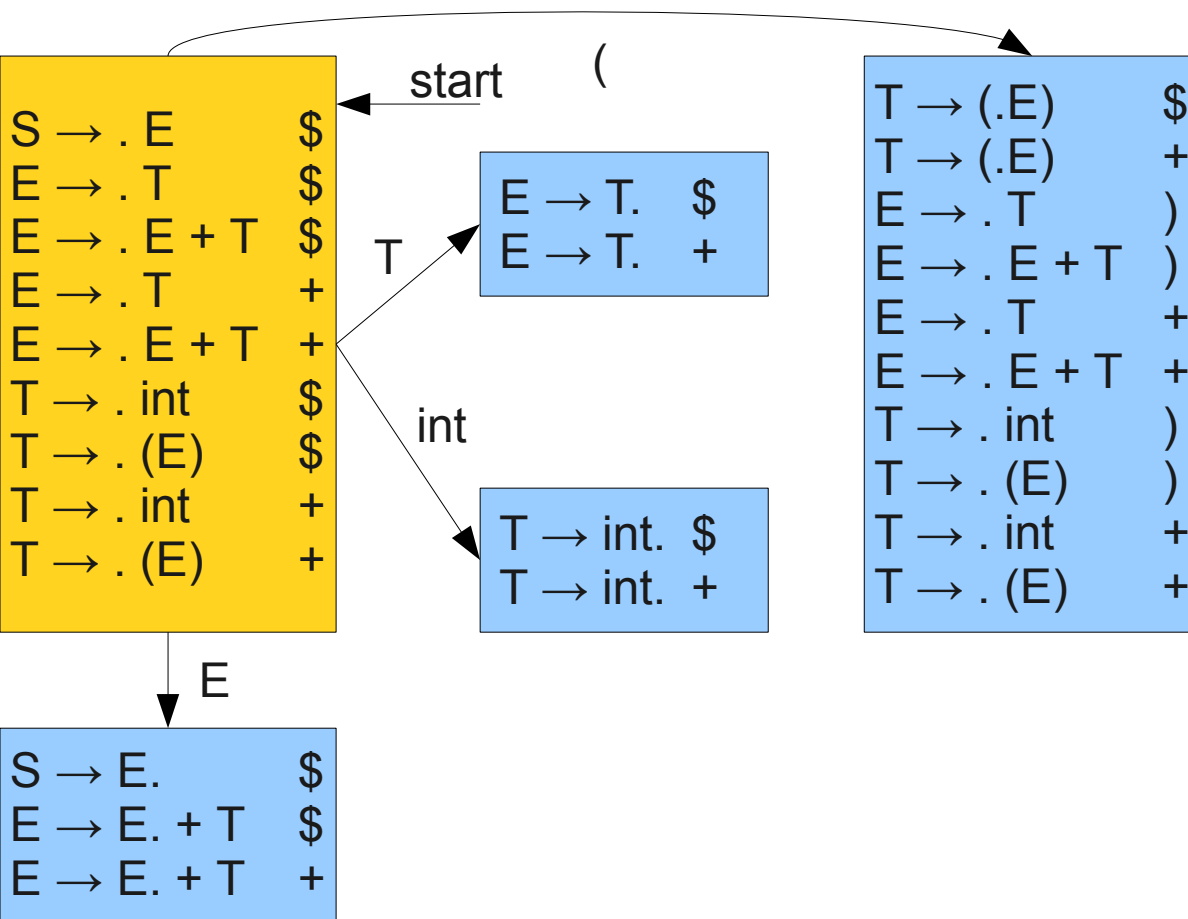
Building Deterministic LR(1) Automata



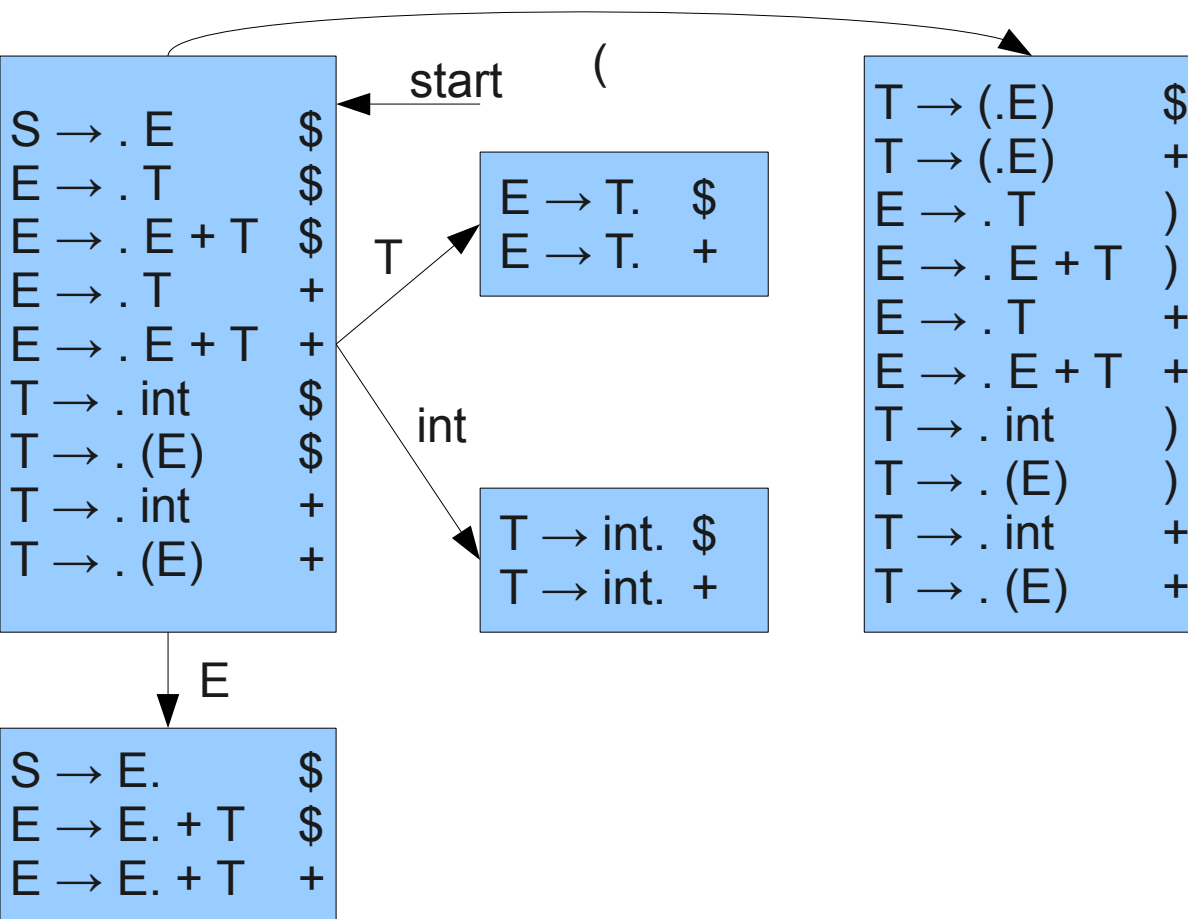
Building Deterministic LR(1) Automata



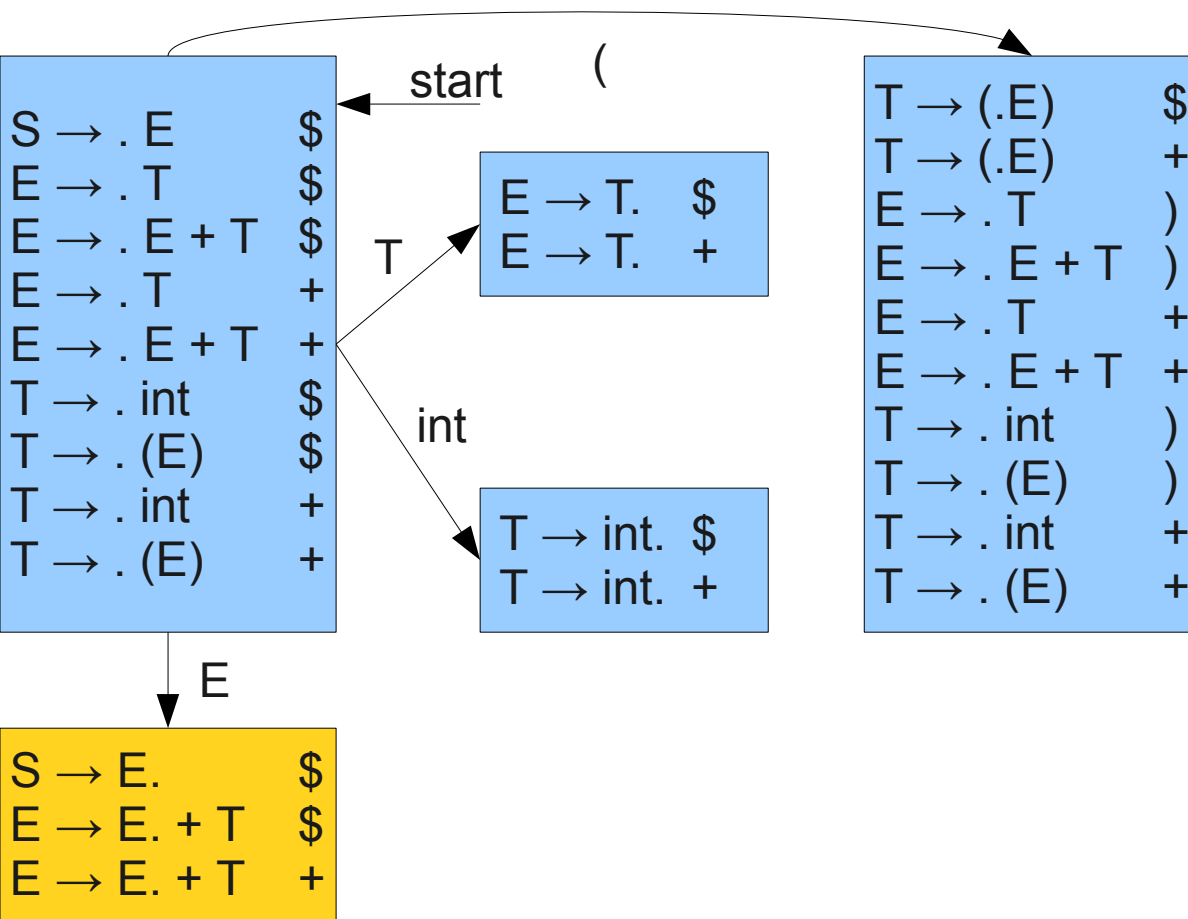
Building Deterministic LR(1) Automata



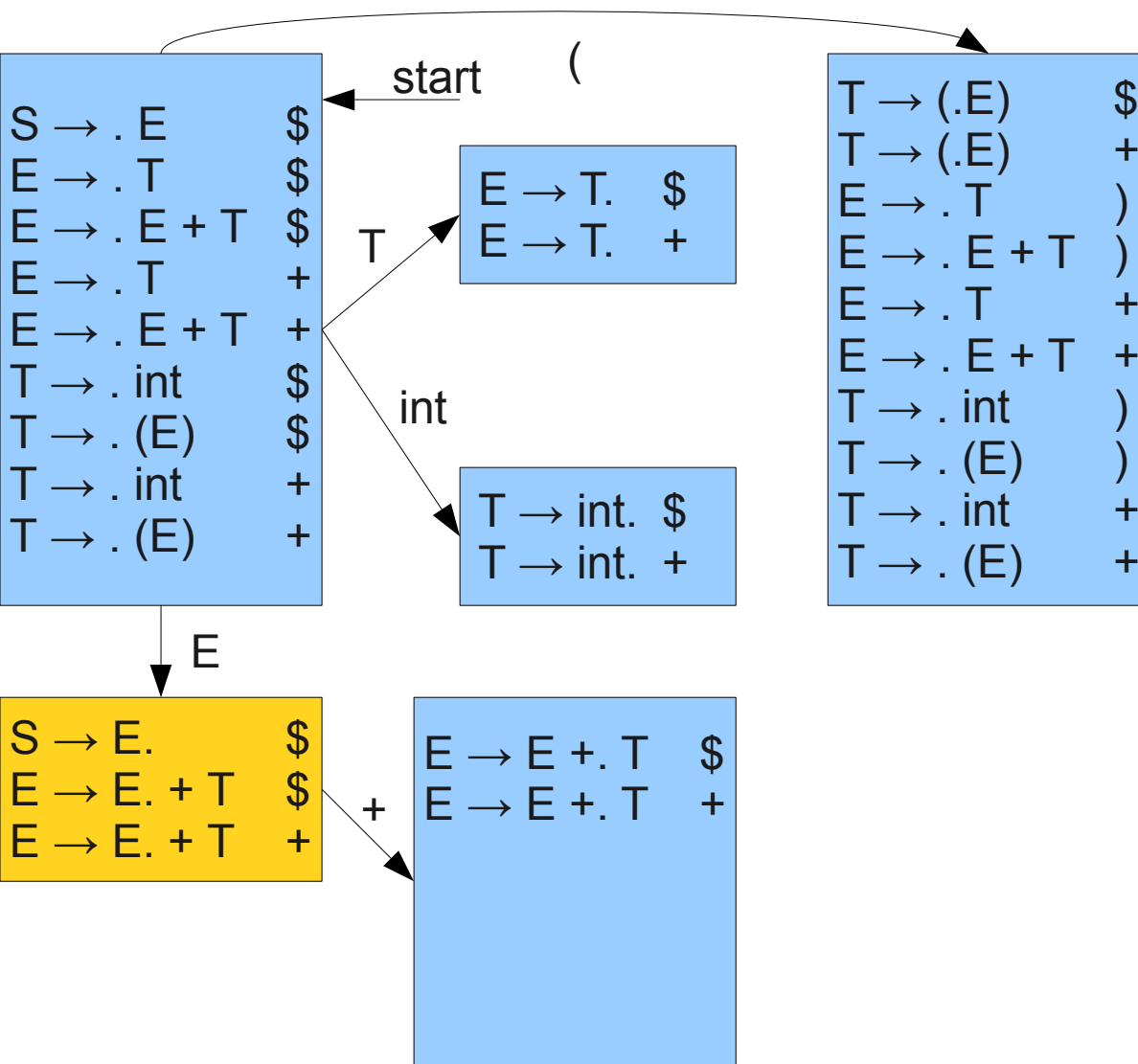
Building Deterministic LR(1) Automata



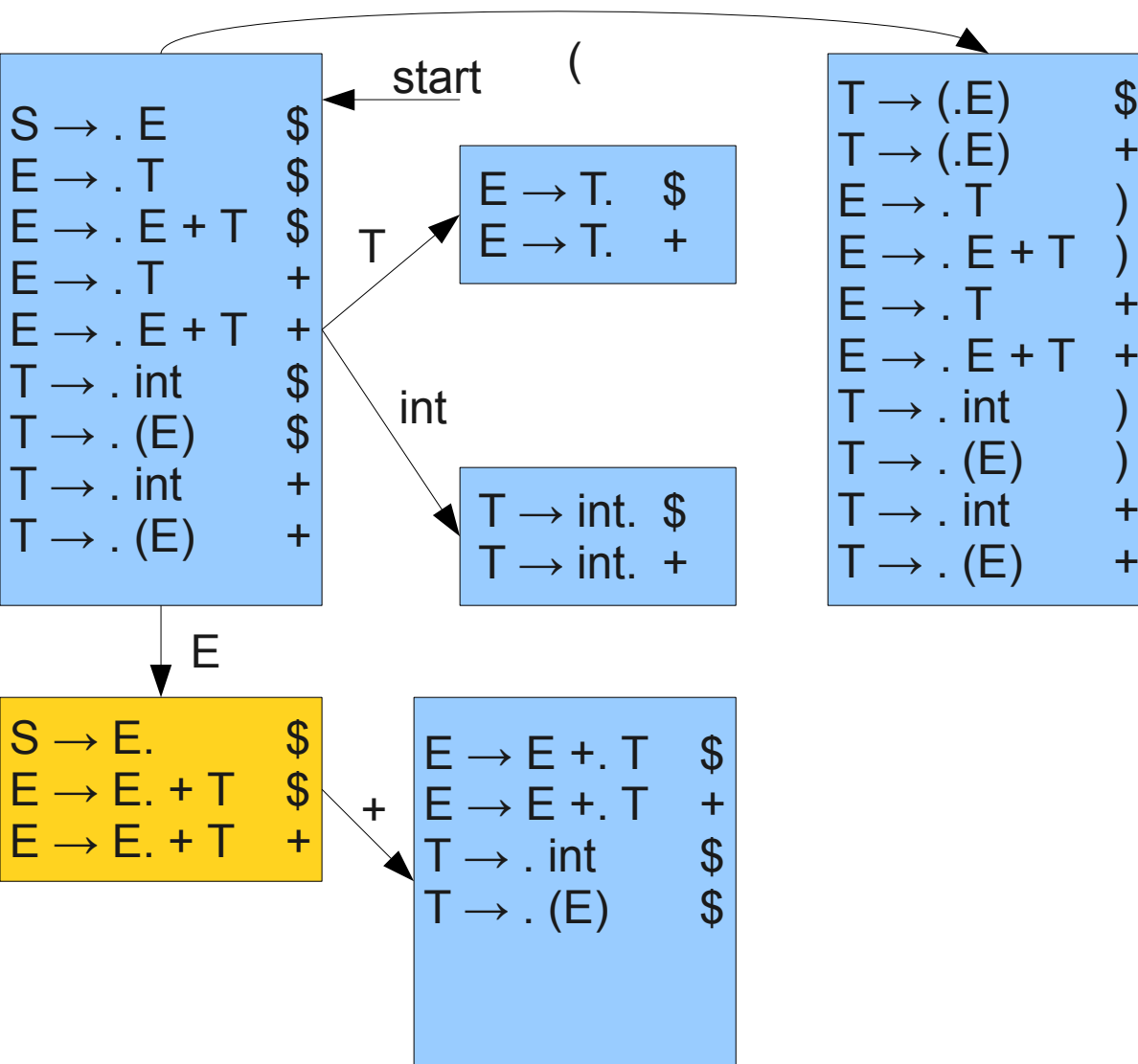
Building Deterministic LR(1) Automata



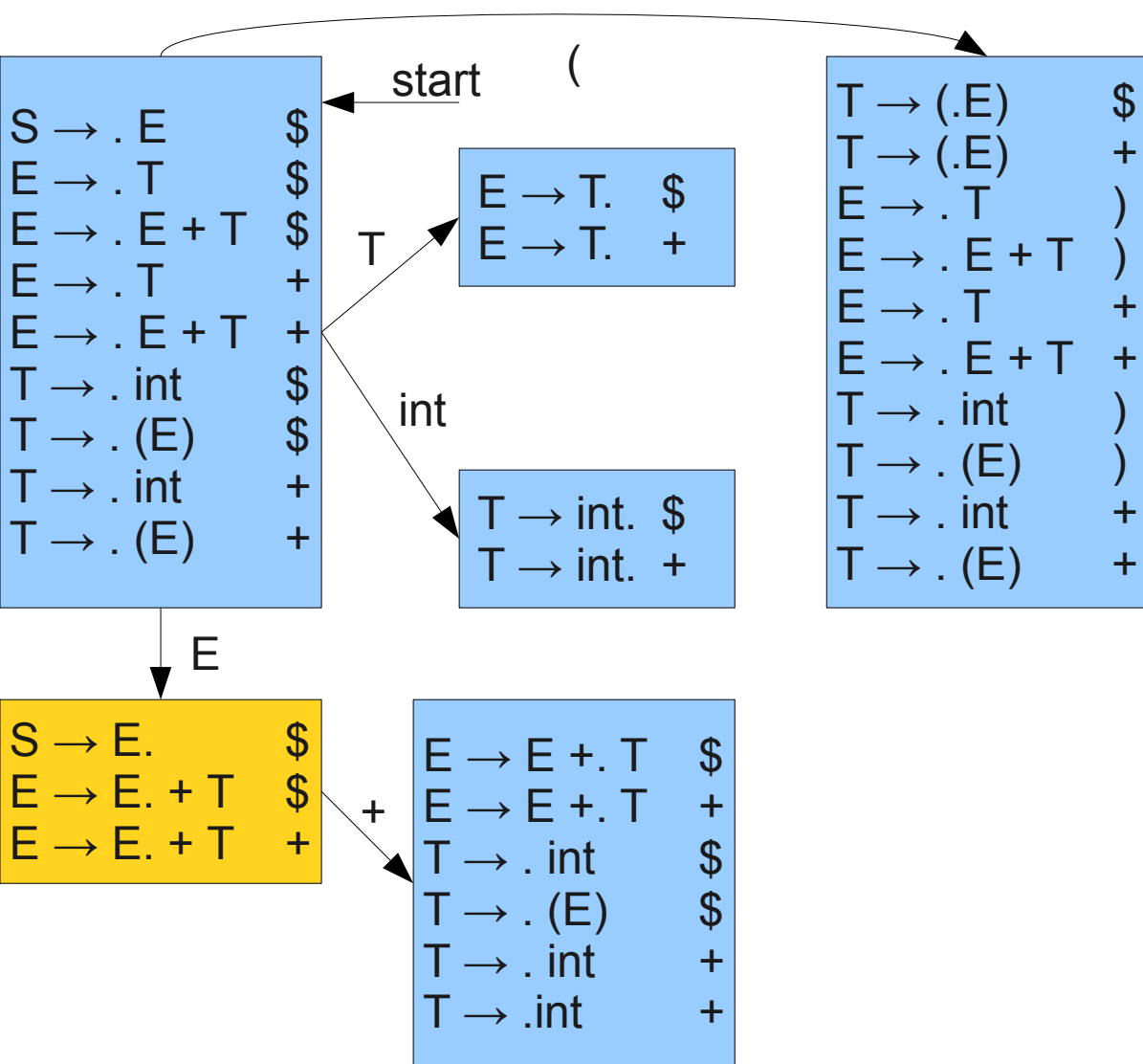
Building Deterministic LR(1) Automata



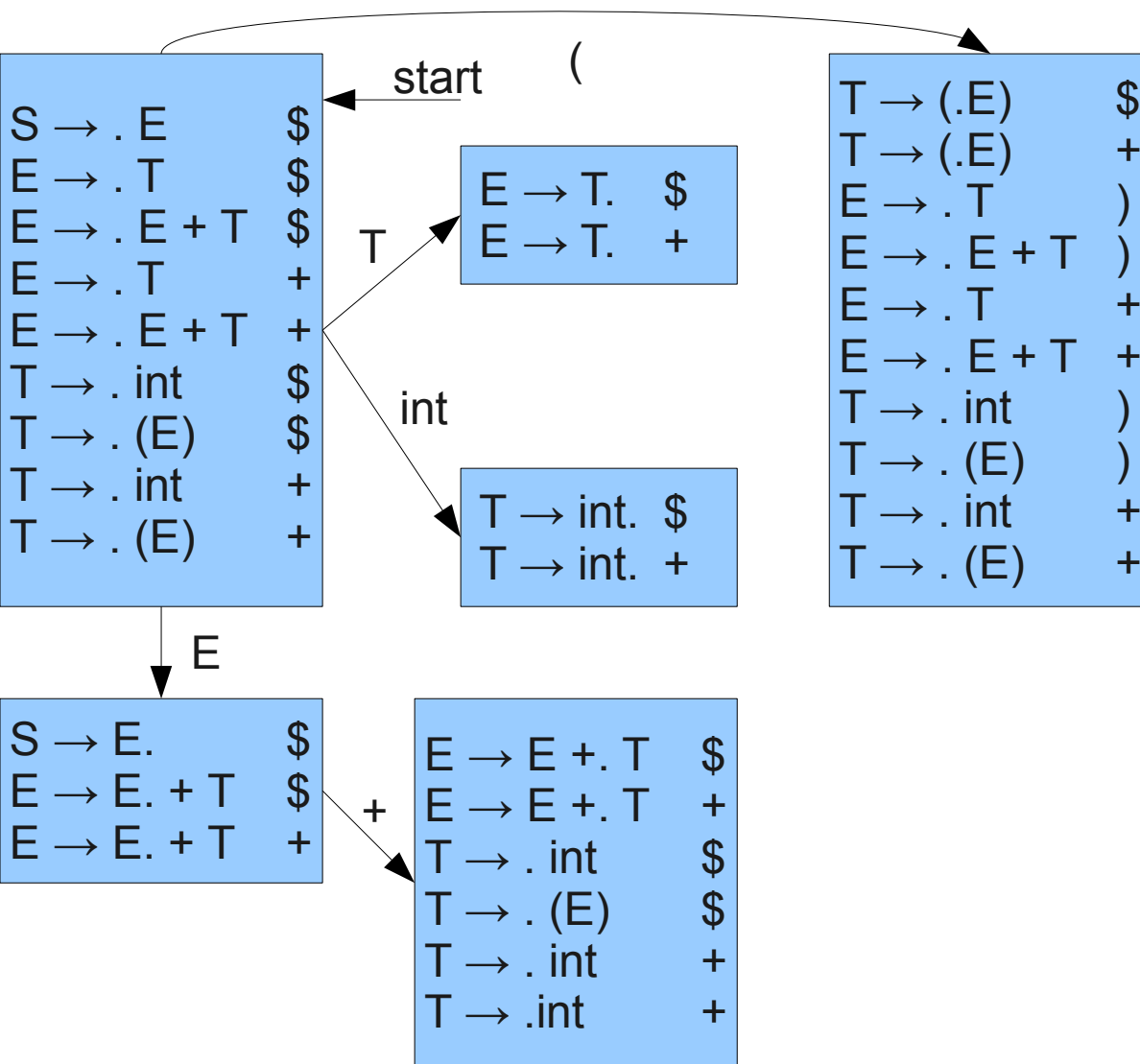
Building Deterministic LR(1) Automata



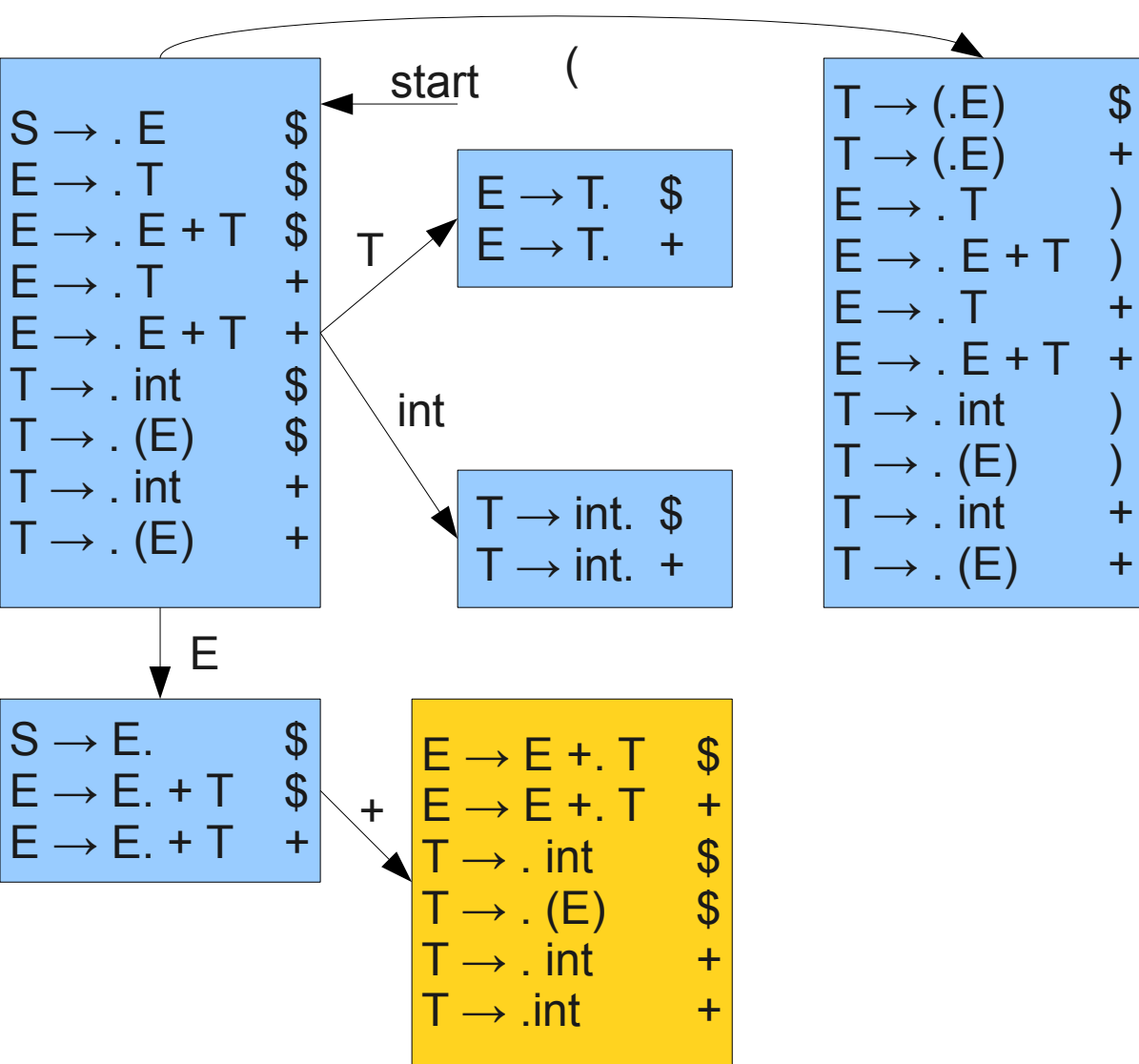
Building Deterministic LR(1) Automata



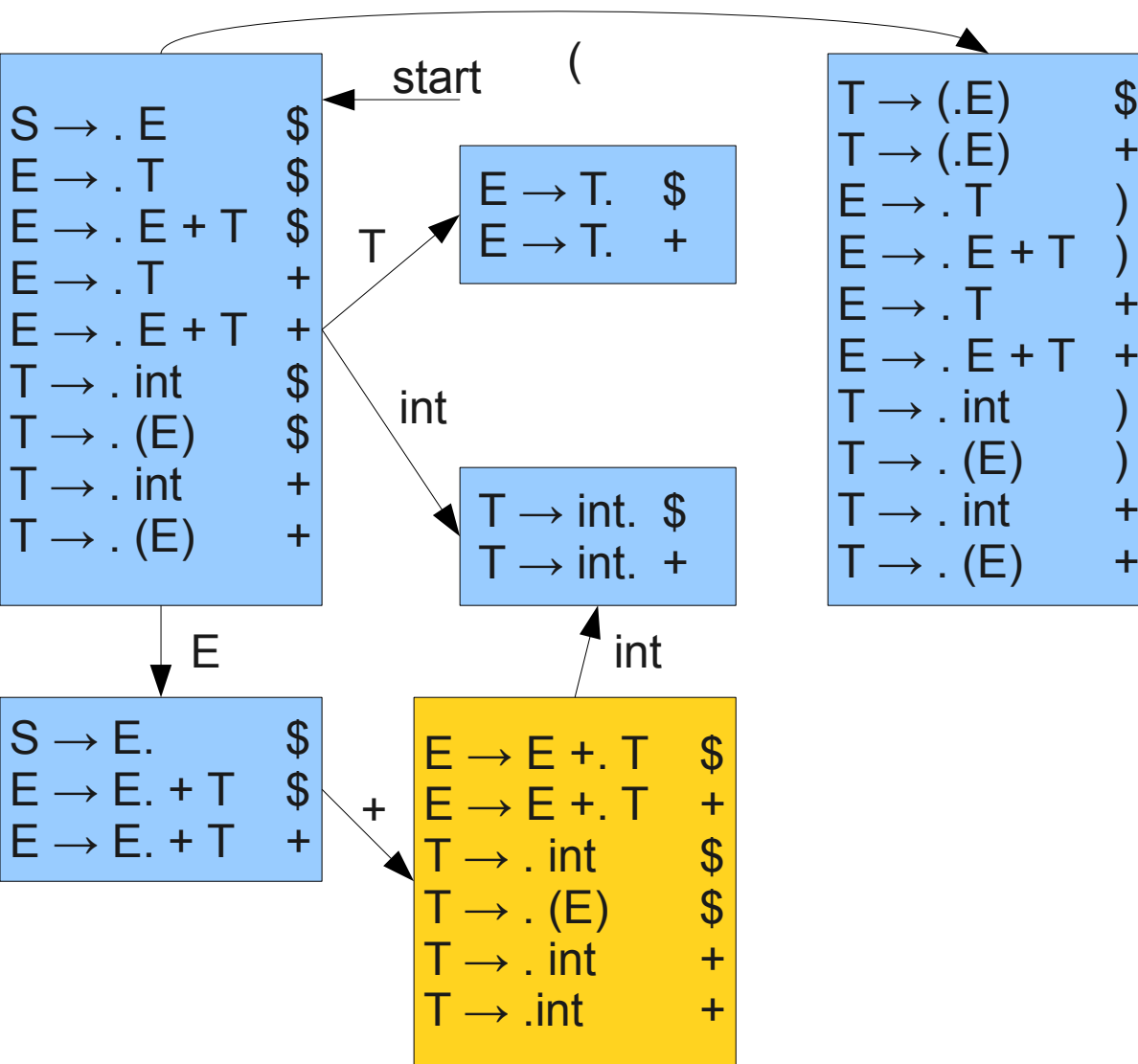
Building Deterministic LR(1) Automata



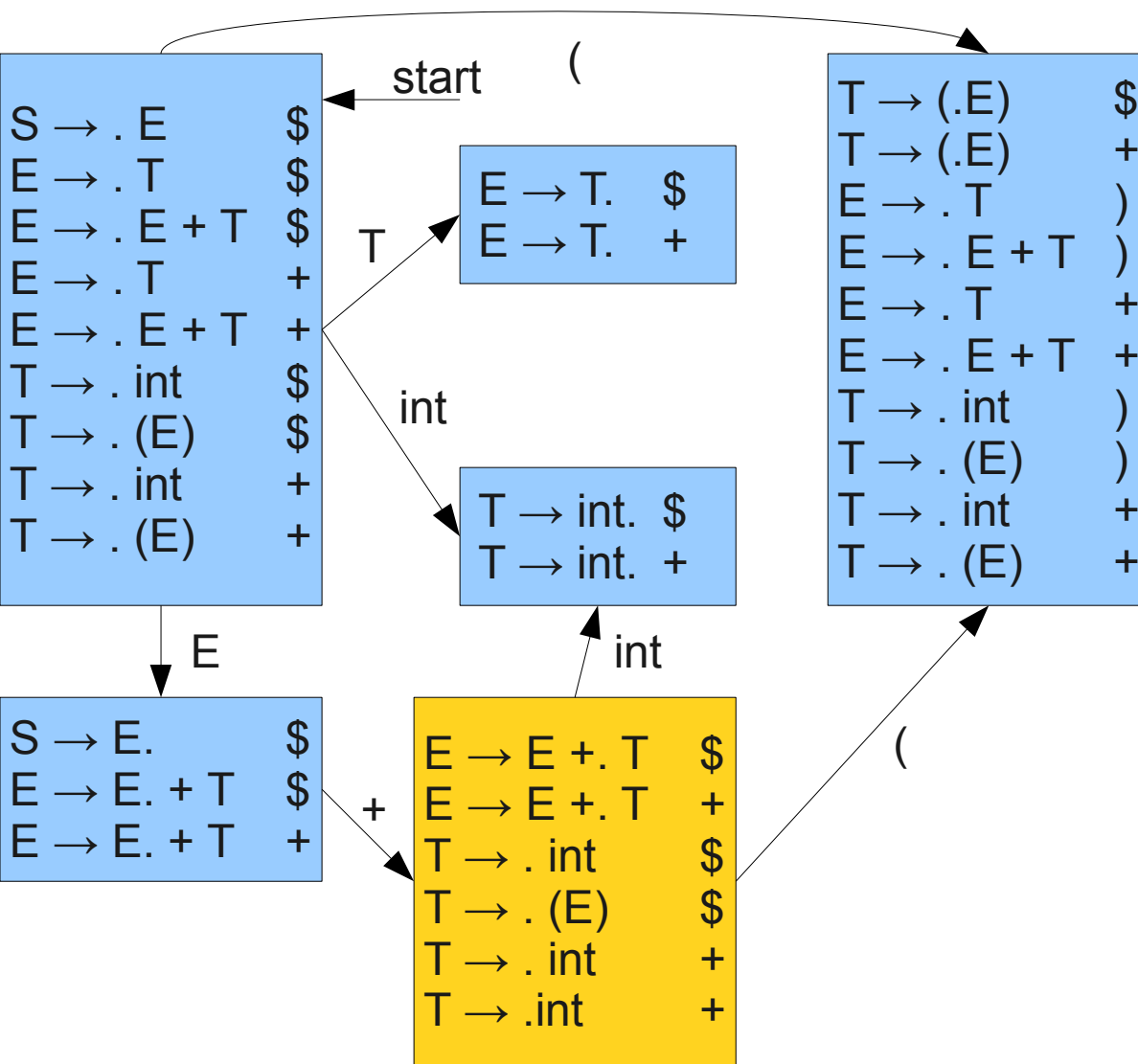
Building Deterministic LR(1) Automata



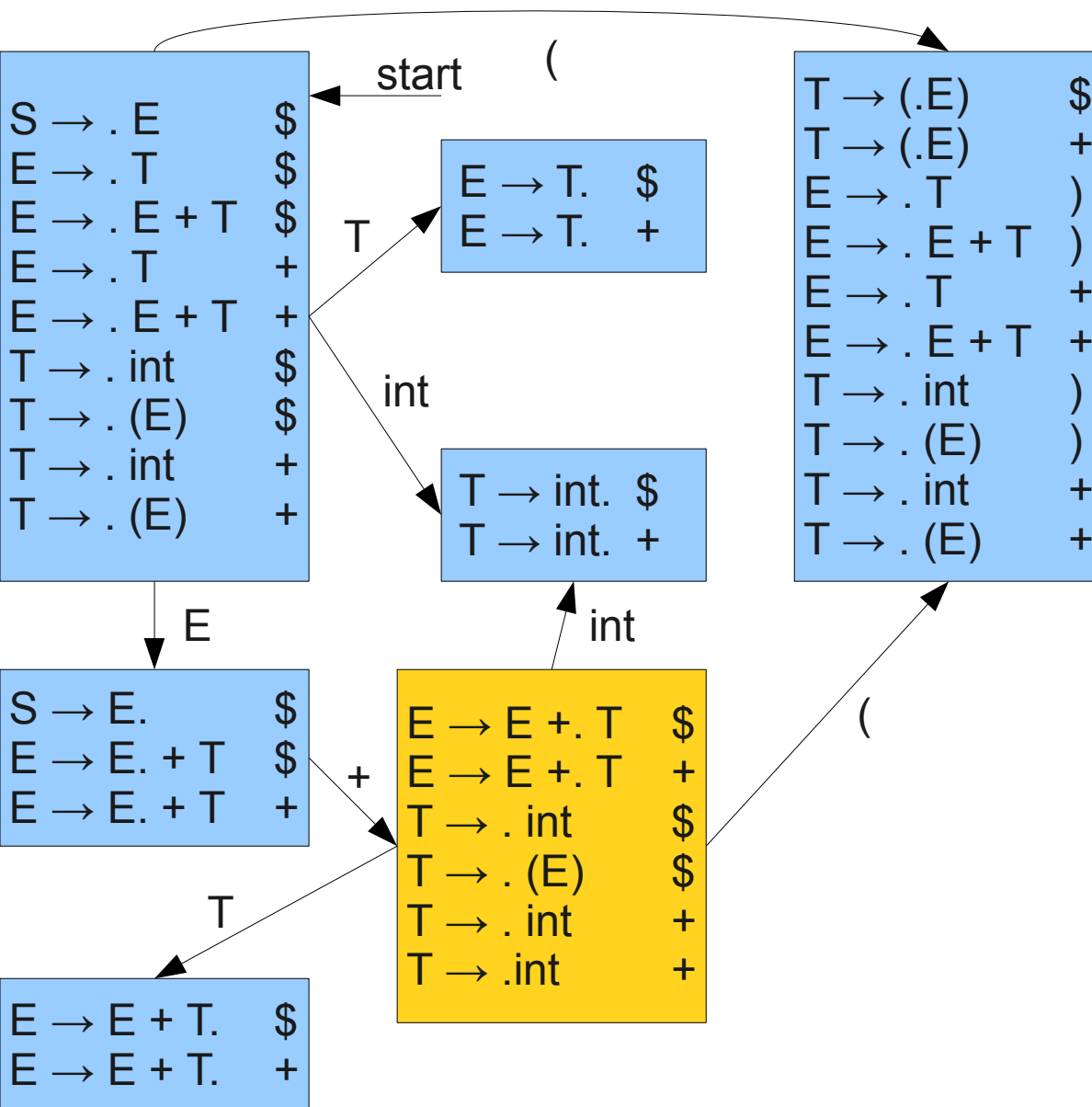
Building Deterministic LR(1) Automata



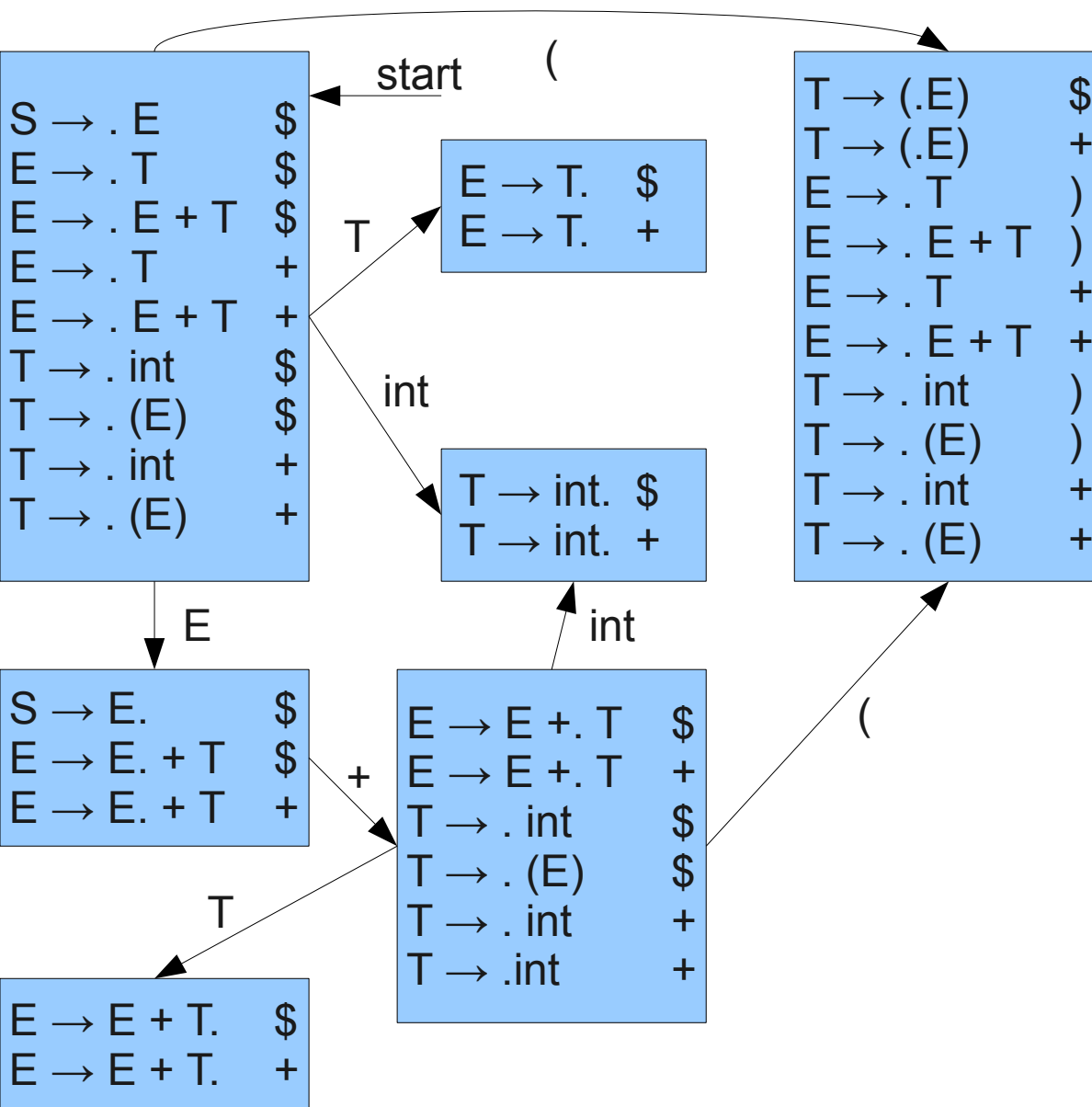
Building Deterministic LR(1) Automata



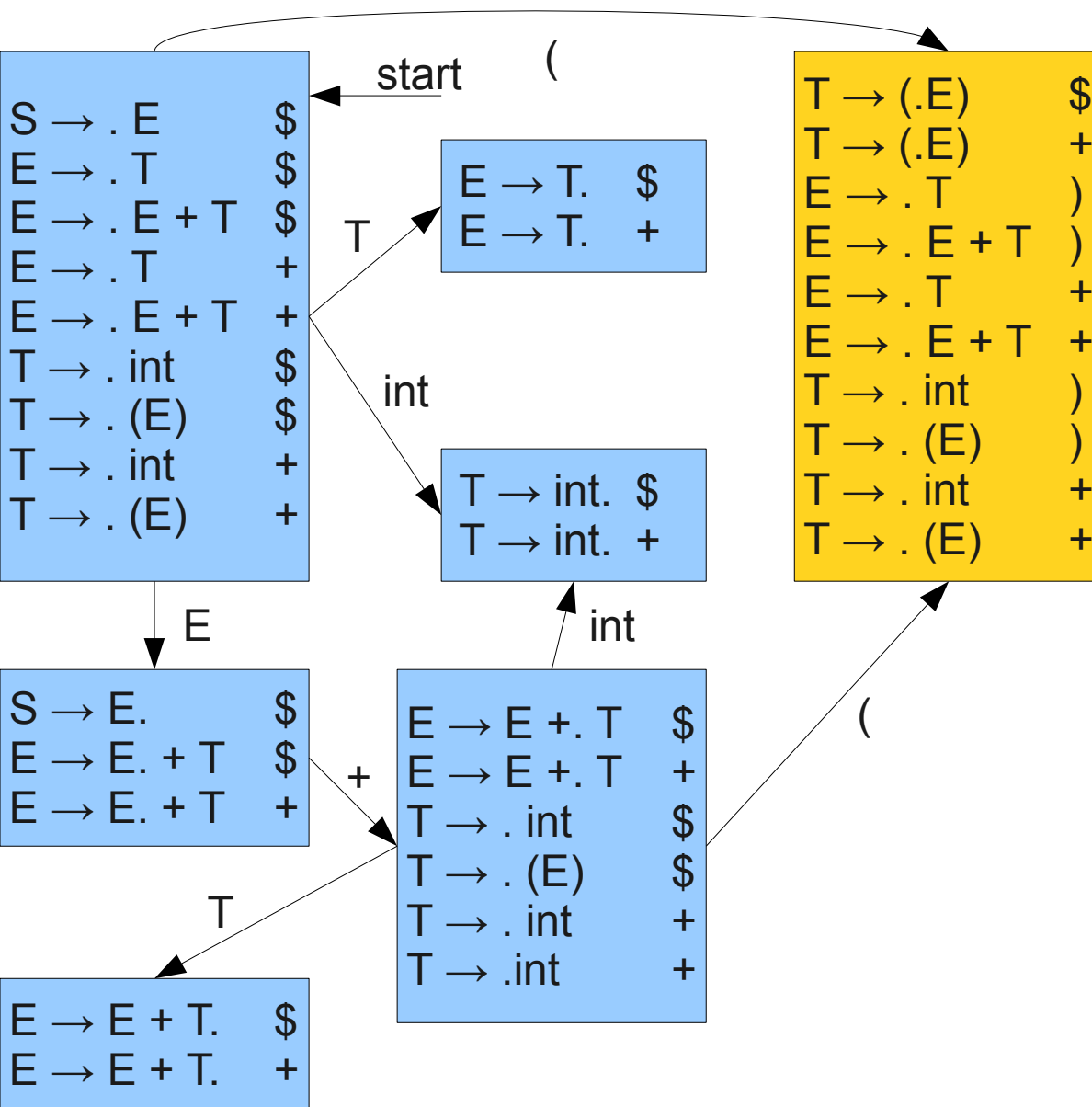
Building Deterministic LR(1) Automata



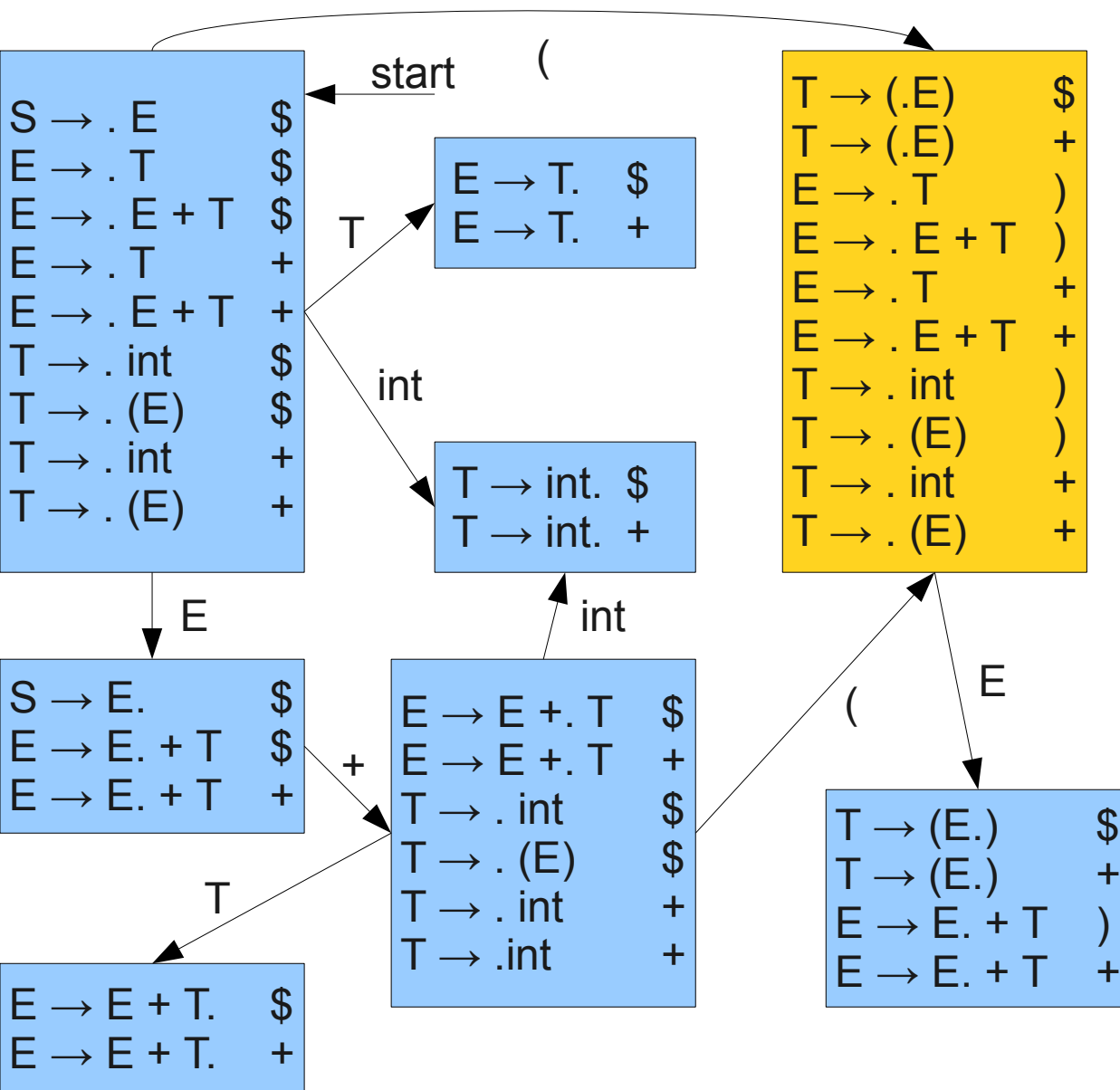
Building Deterministic LR(1) Automata



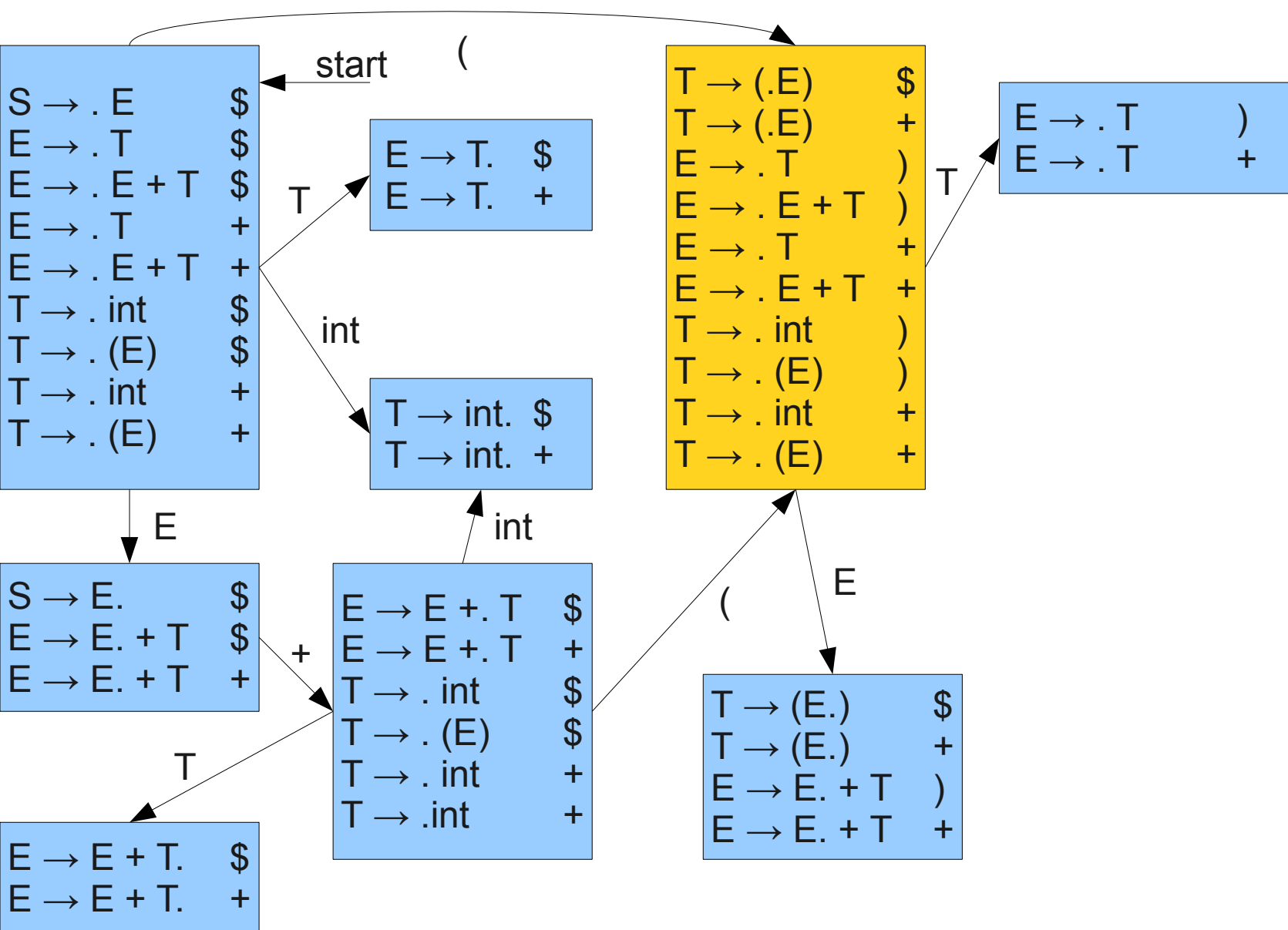
Building Deterministic LR(1) Automata



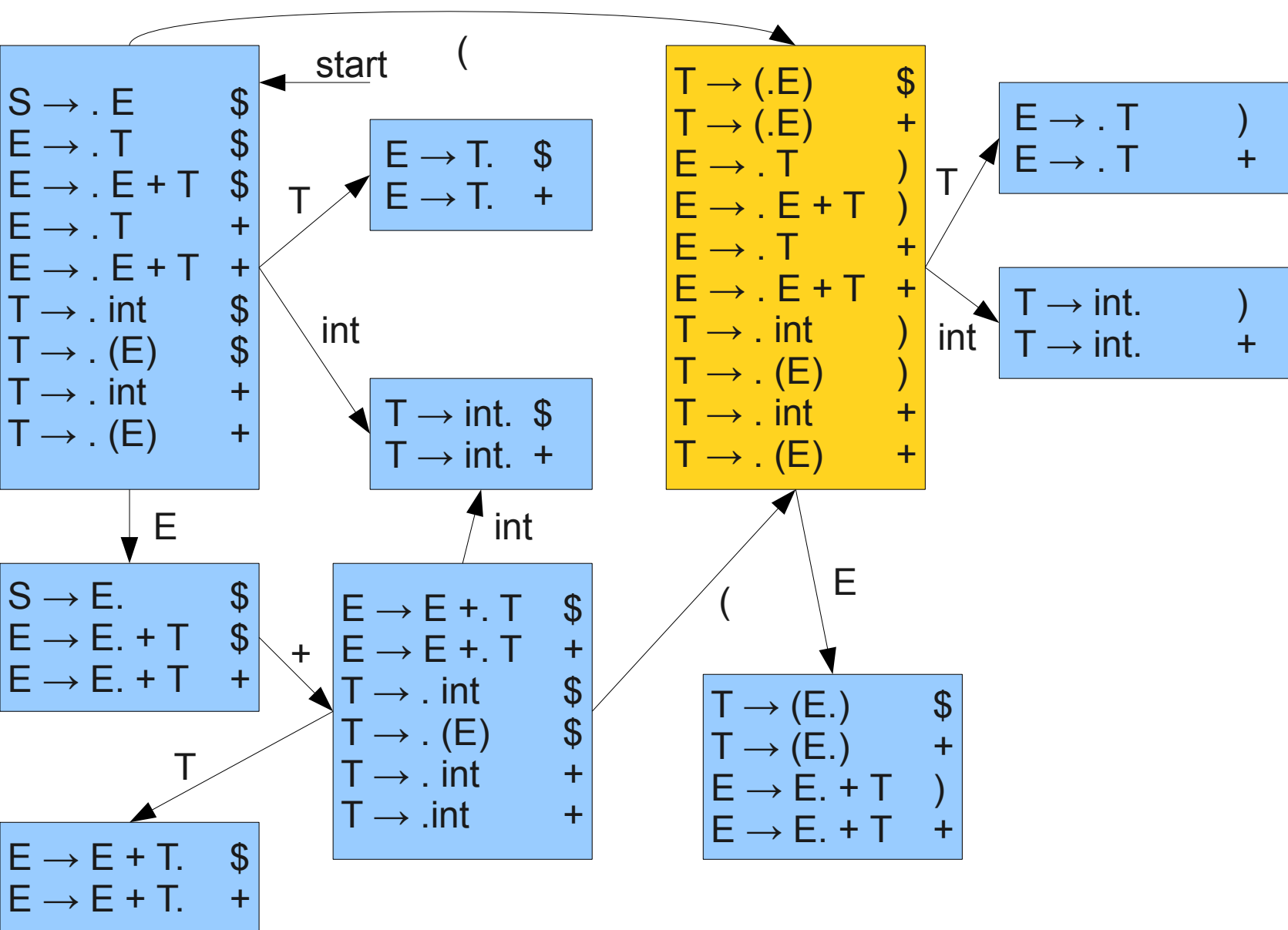
Building Deterministic LR(1) Automata



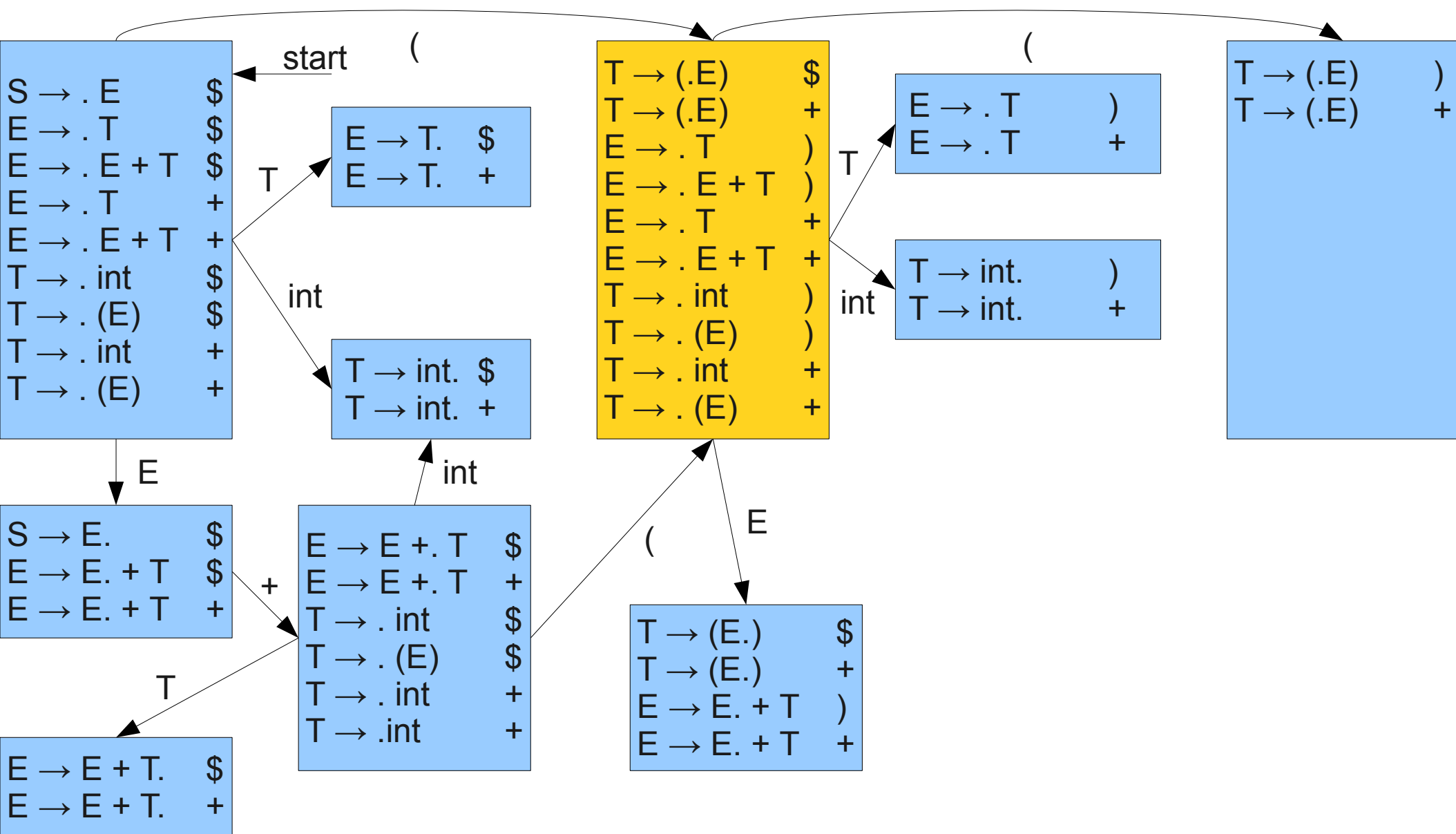
Building Deterministic LR(1) Automata



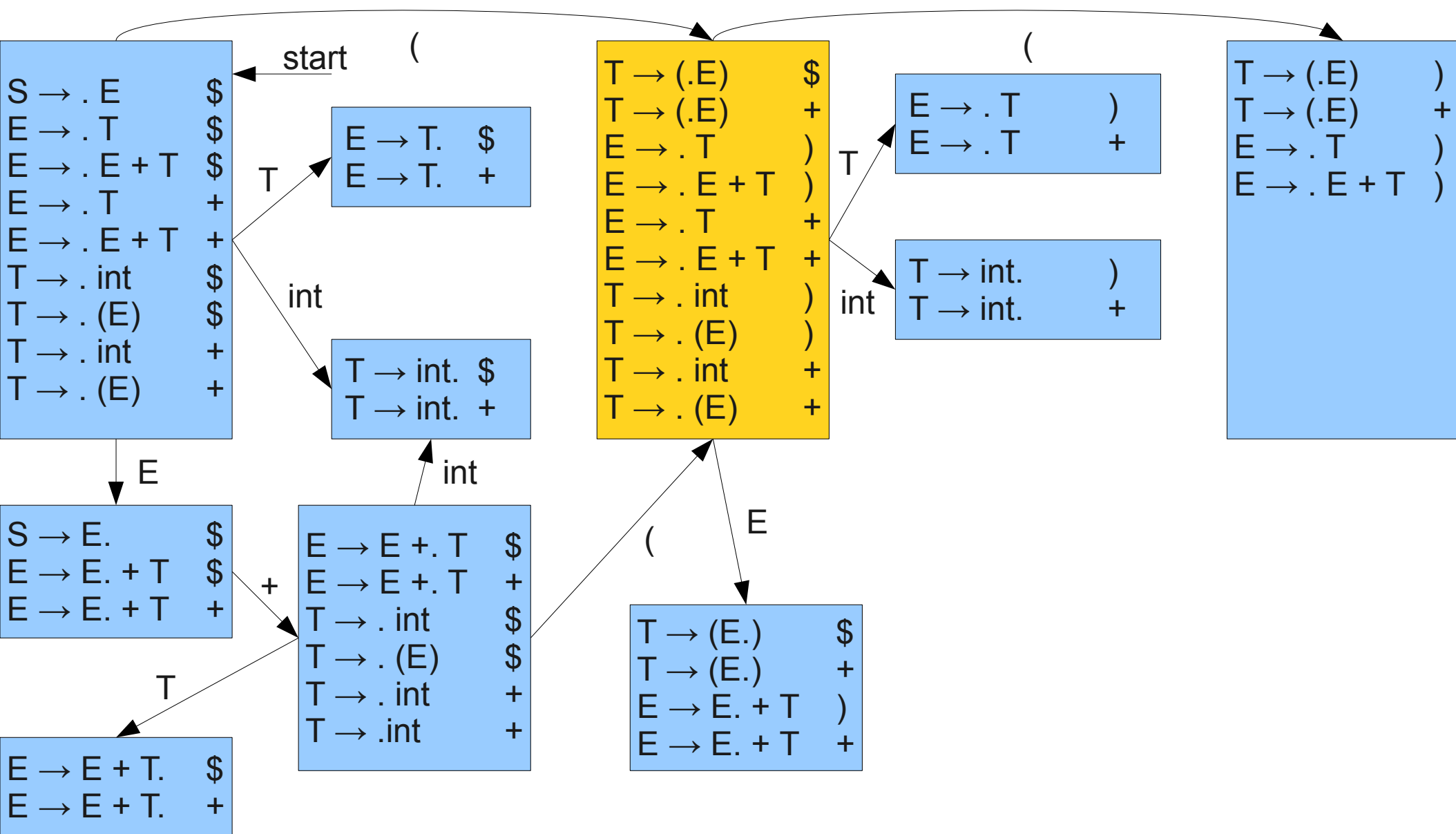
Building Deterministic LR(1) Automata



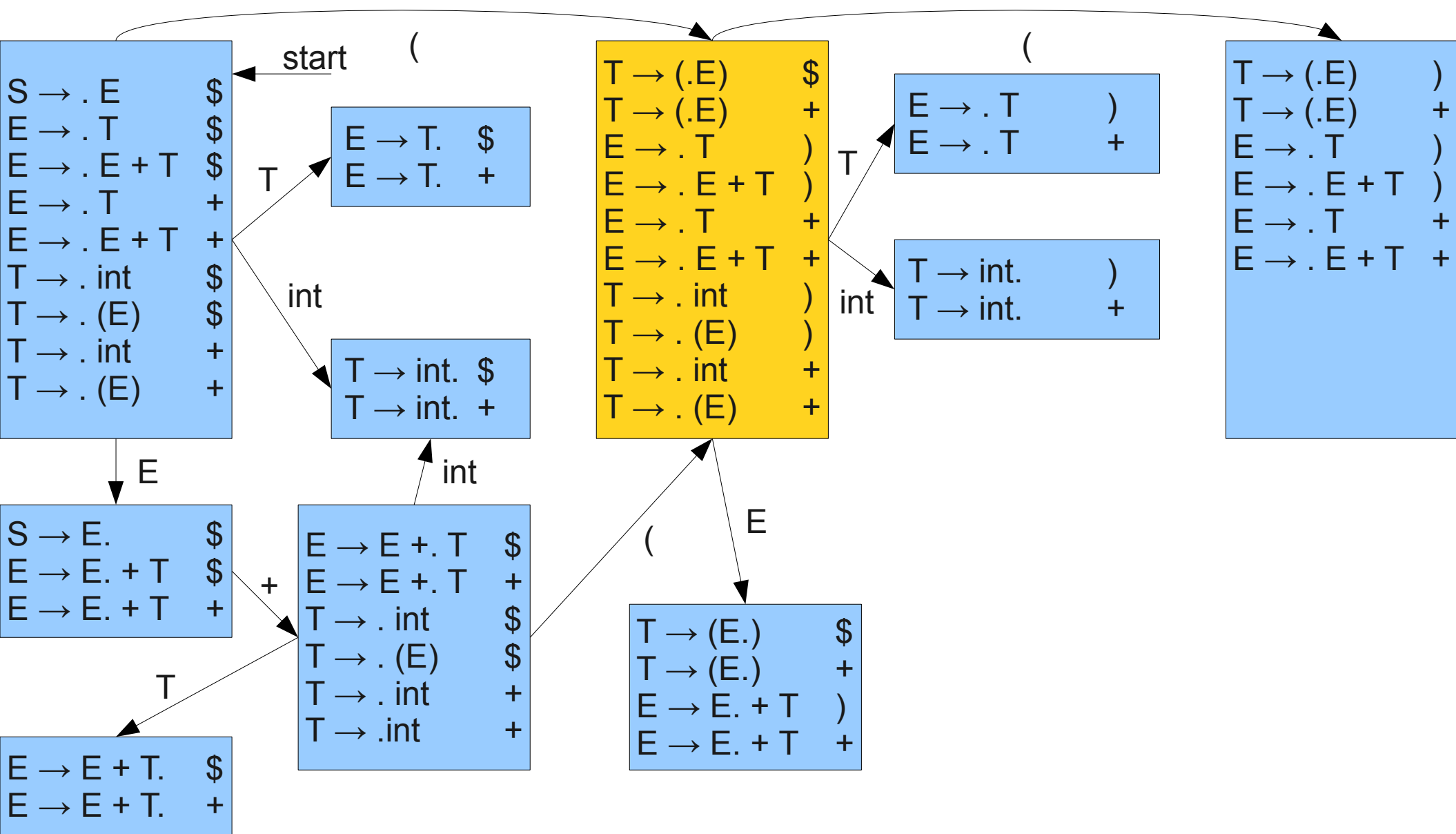
Building Deterministic LR(1) Automata



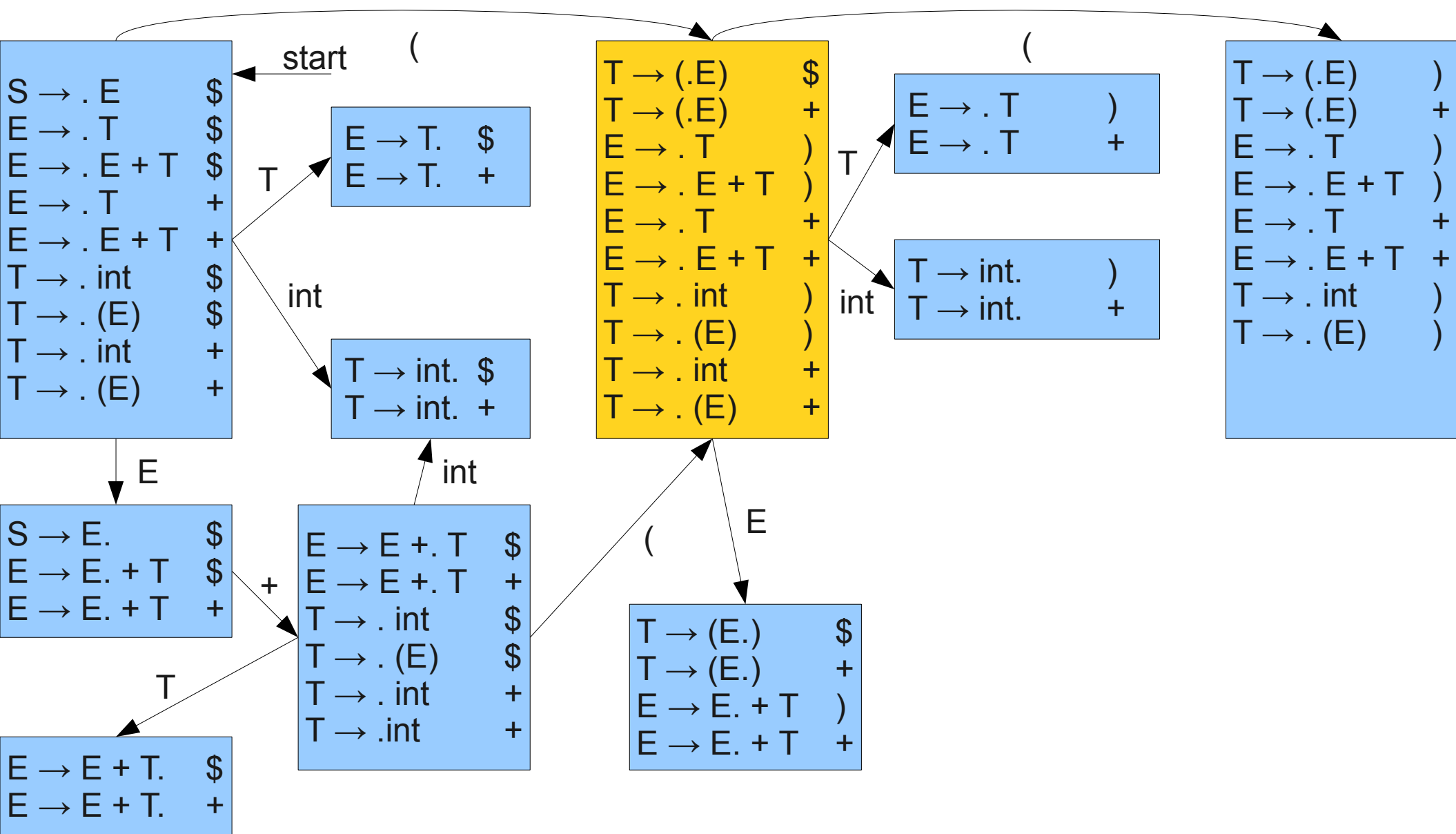
Building Deterministic LR(1) Automata



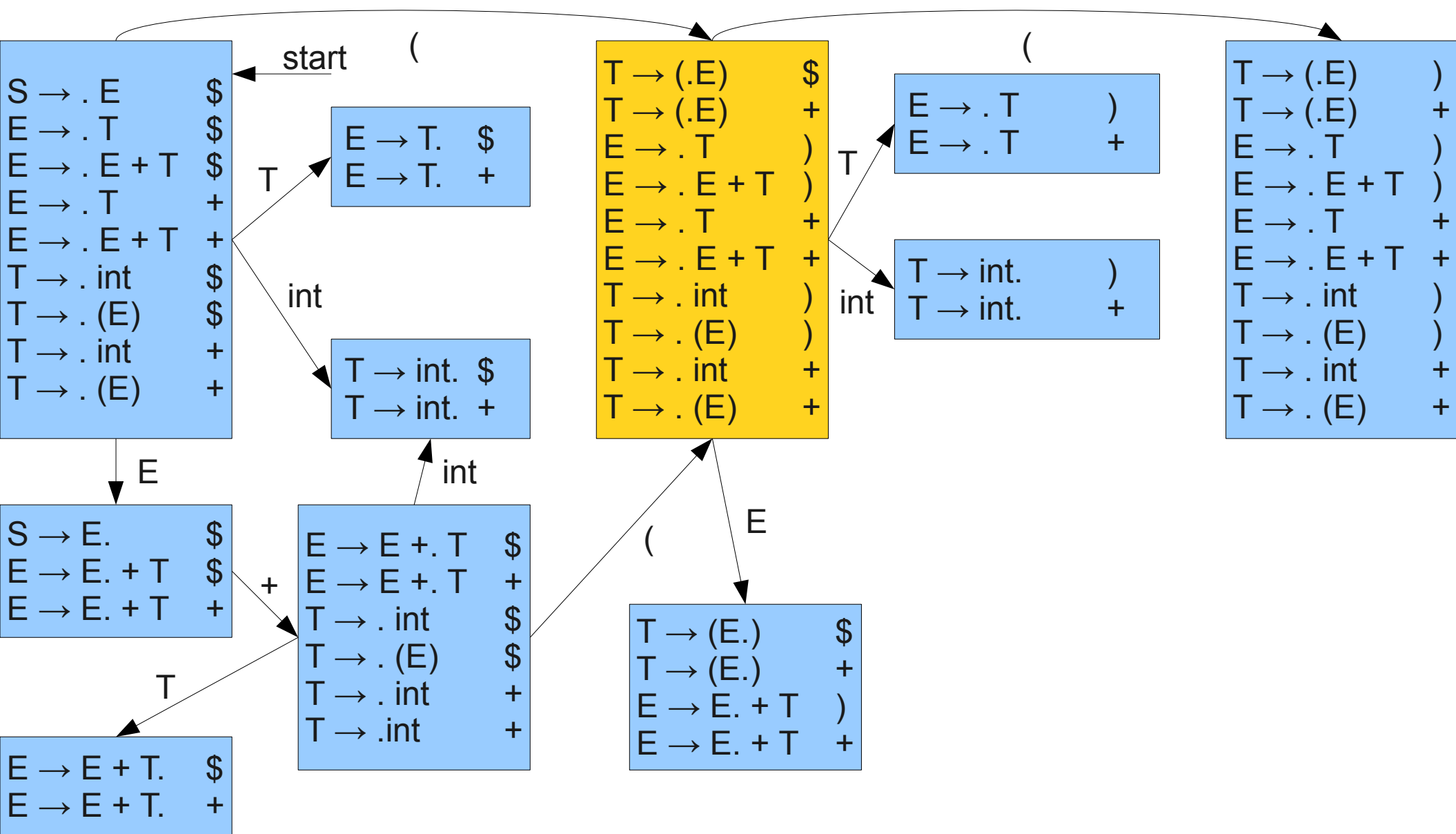
Building Deterministic LR(1) Automata



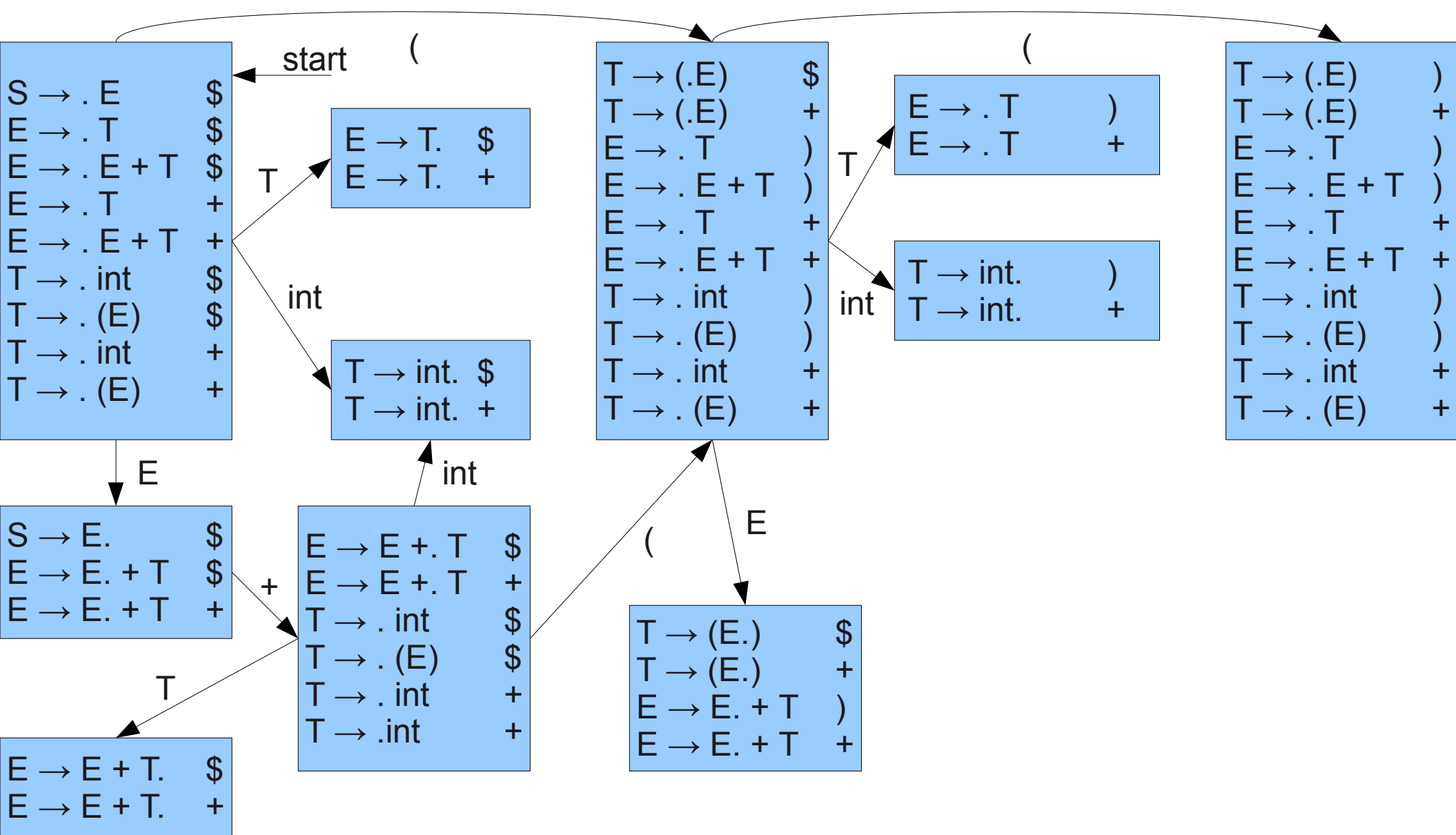
Building Deterministic LR(1) Automata



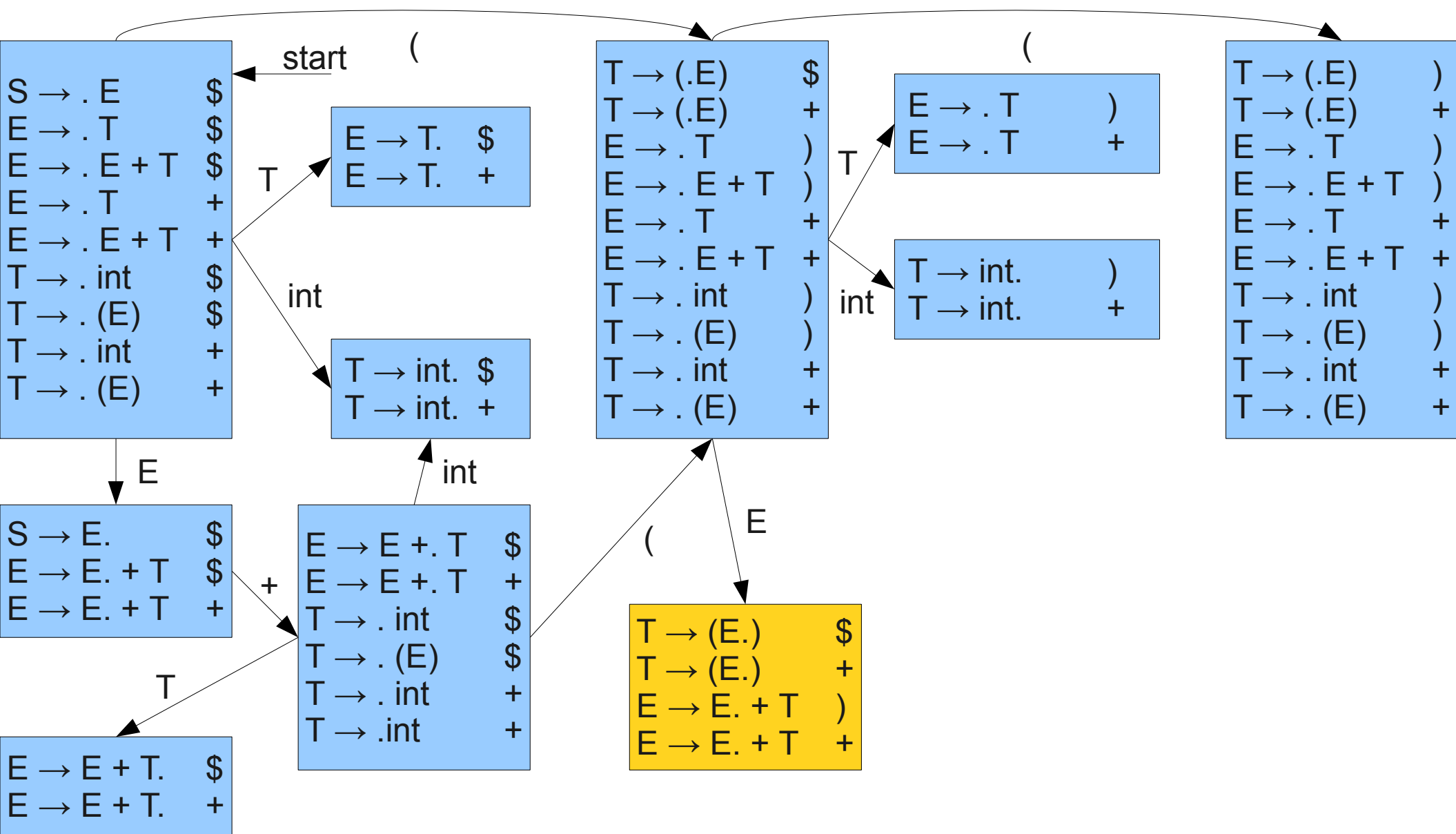
Building Deterministic LR(1) Automata



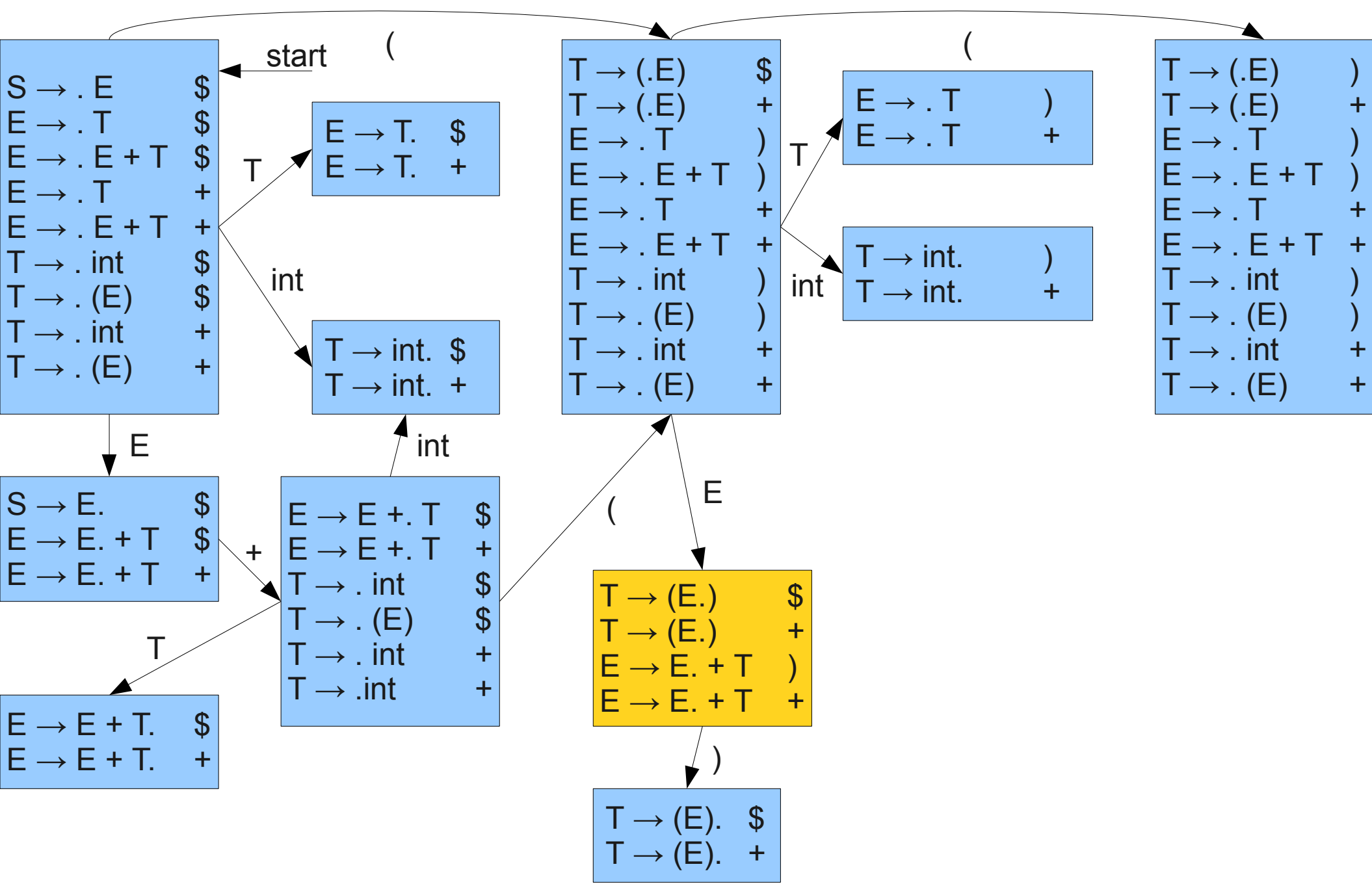
Building Deterministic LR(1) Automata



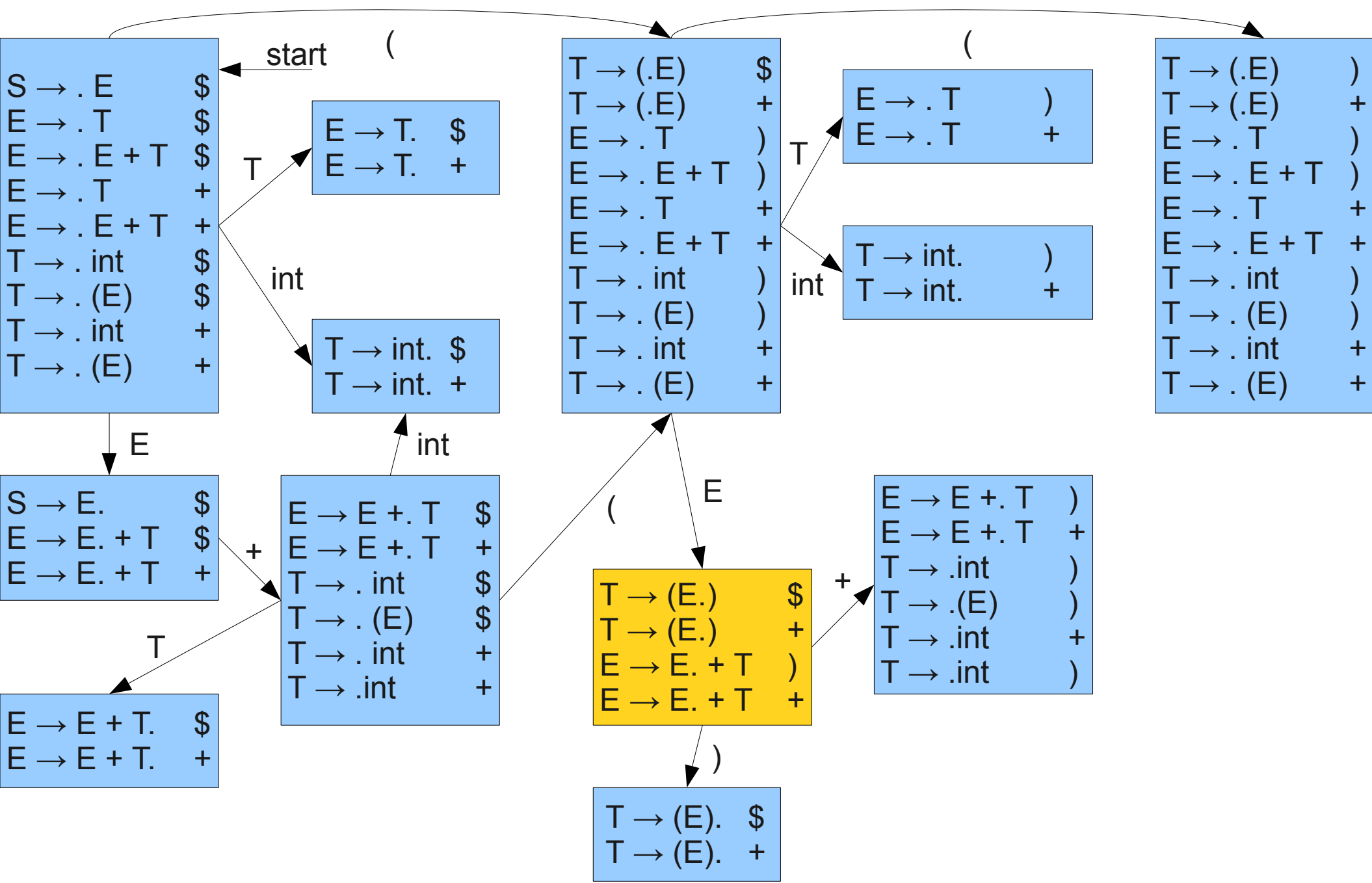
Building Deterministic LR(1) Automata



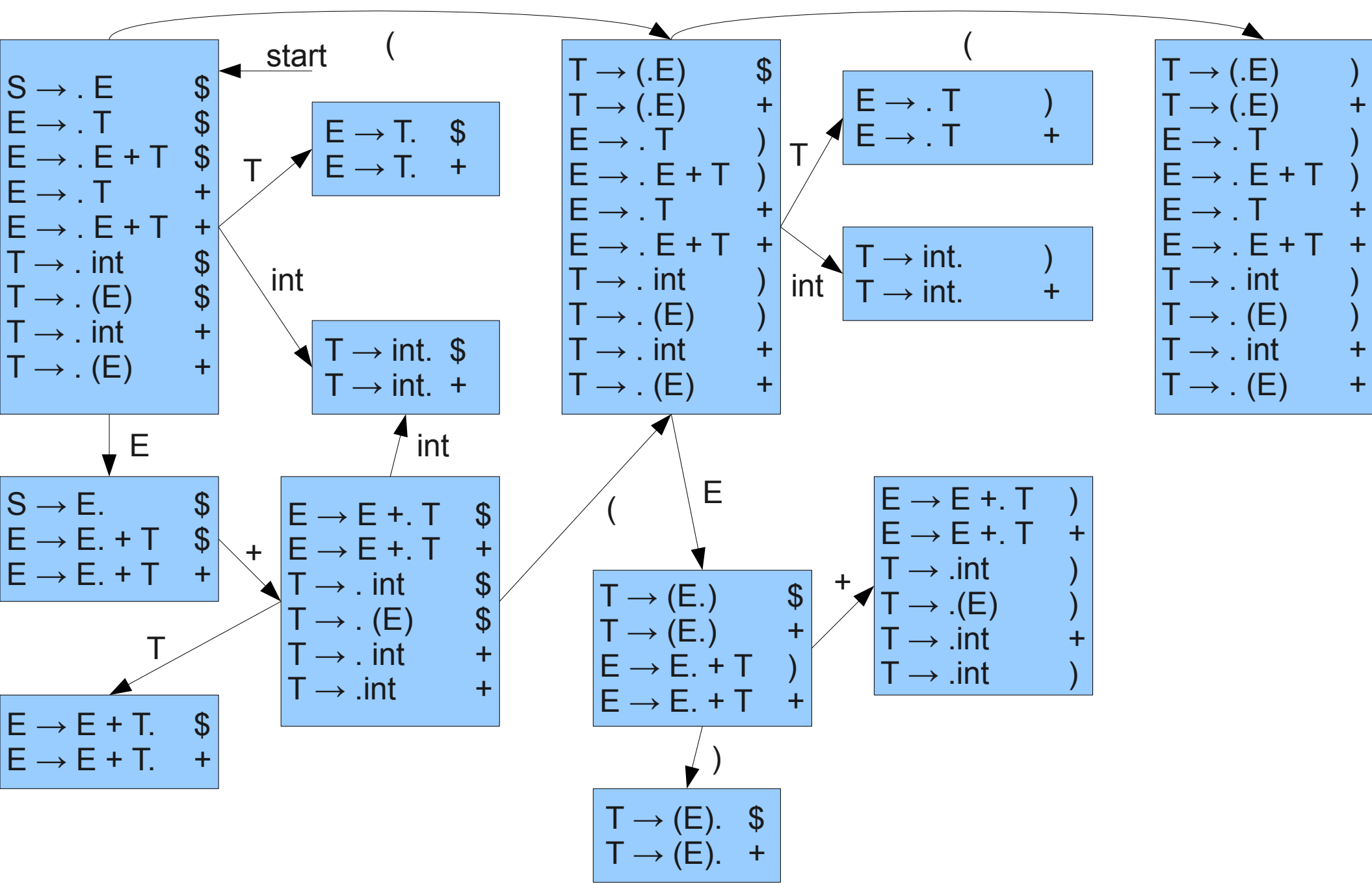
Building Deterministic LR(1) Automata



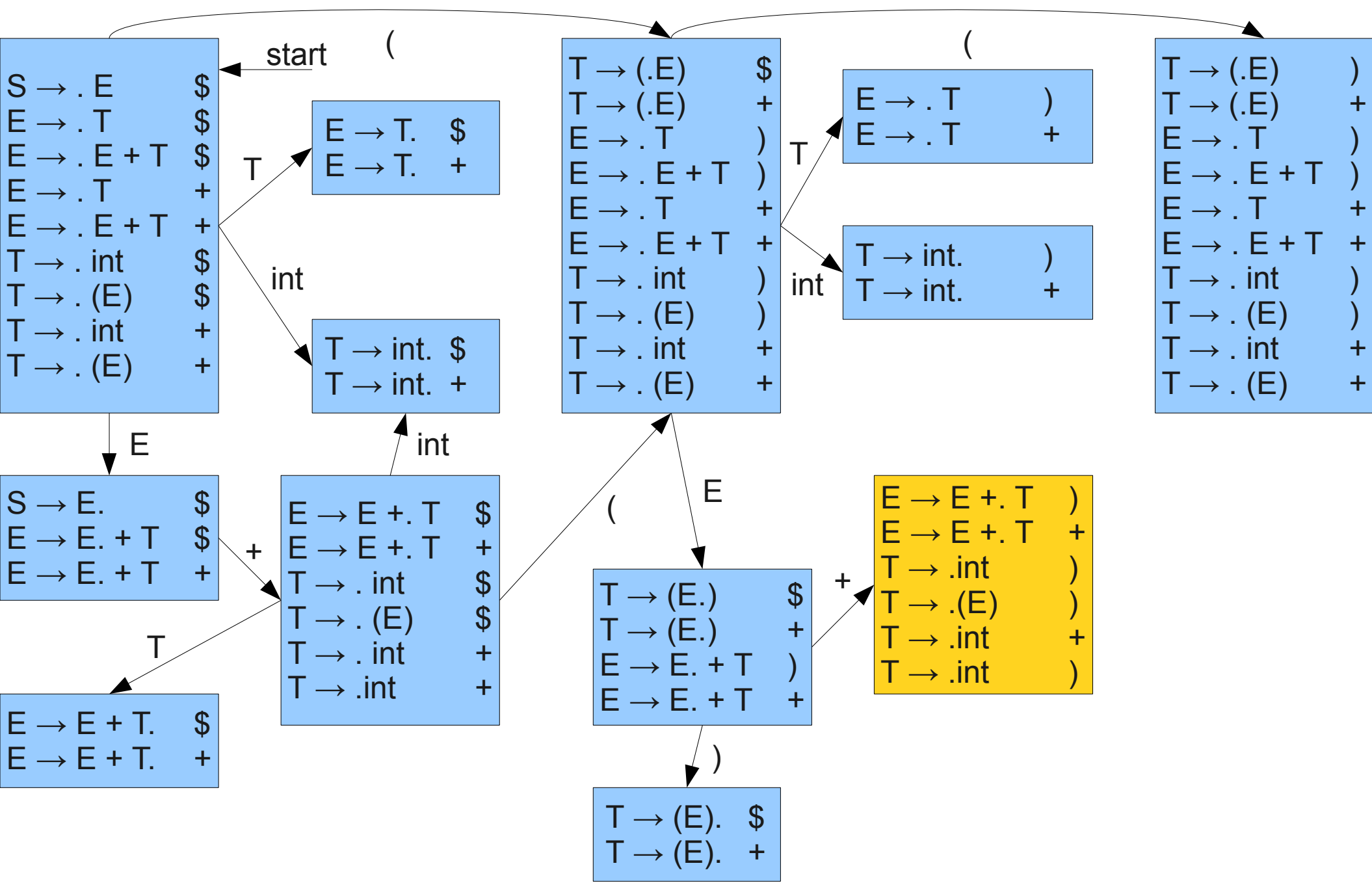
Building Deterministic LR(1) Automata



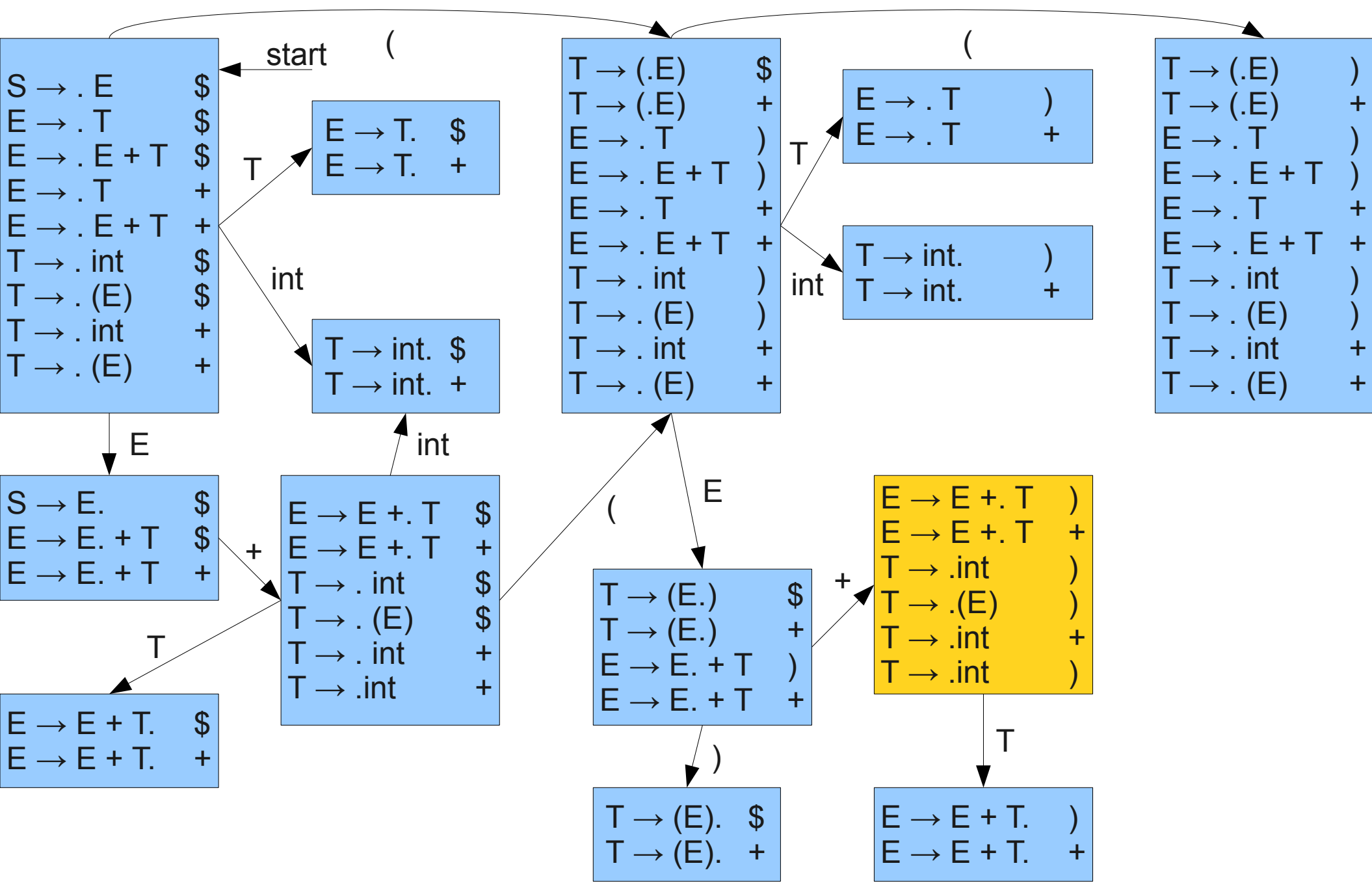
Building Deterministic LR(1) Automata



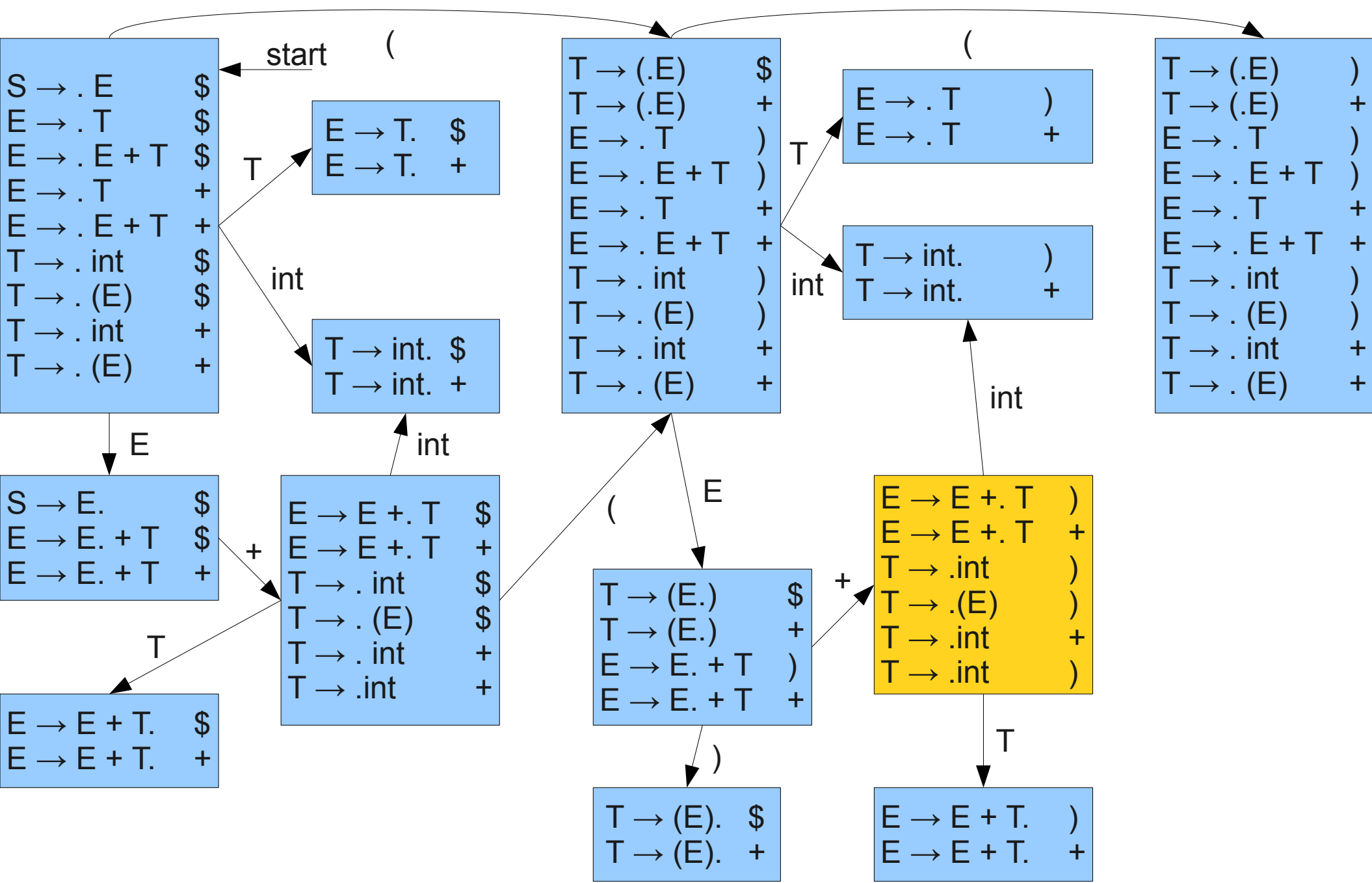
Building Deterministic LR(1) Automata



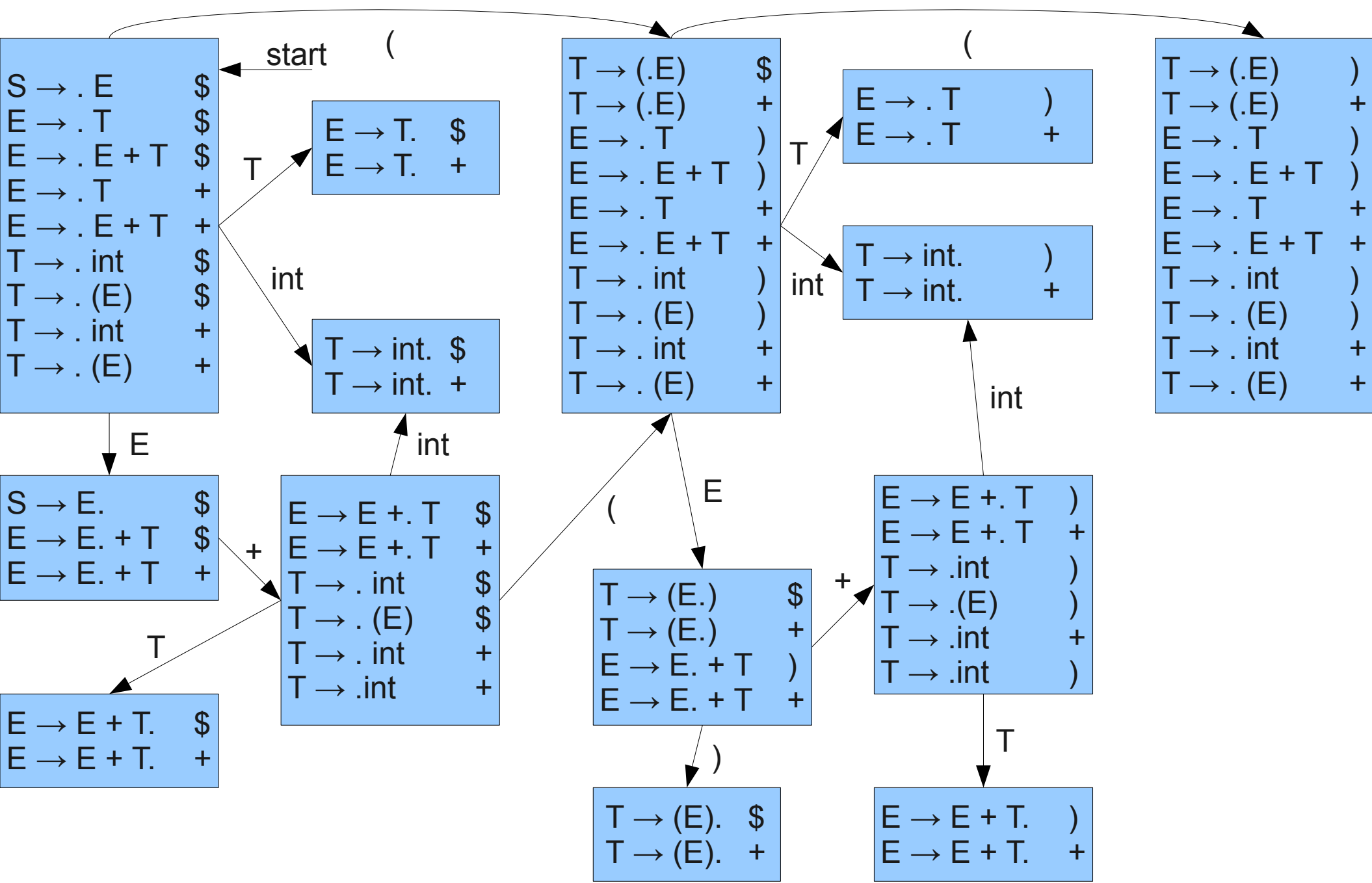
Building Deterministic LR(1) Automata



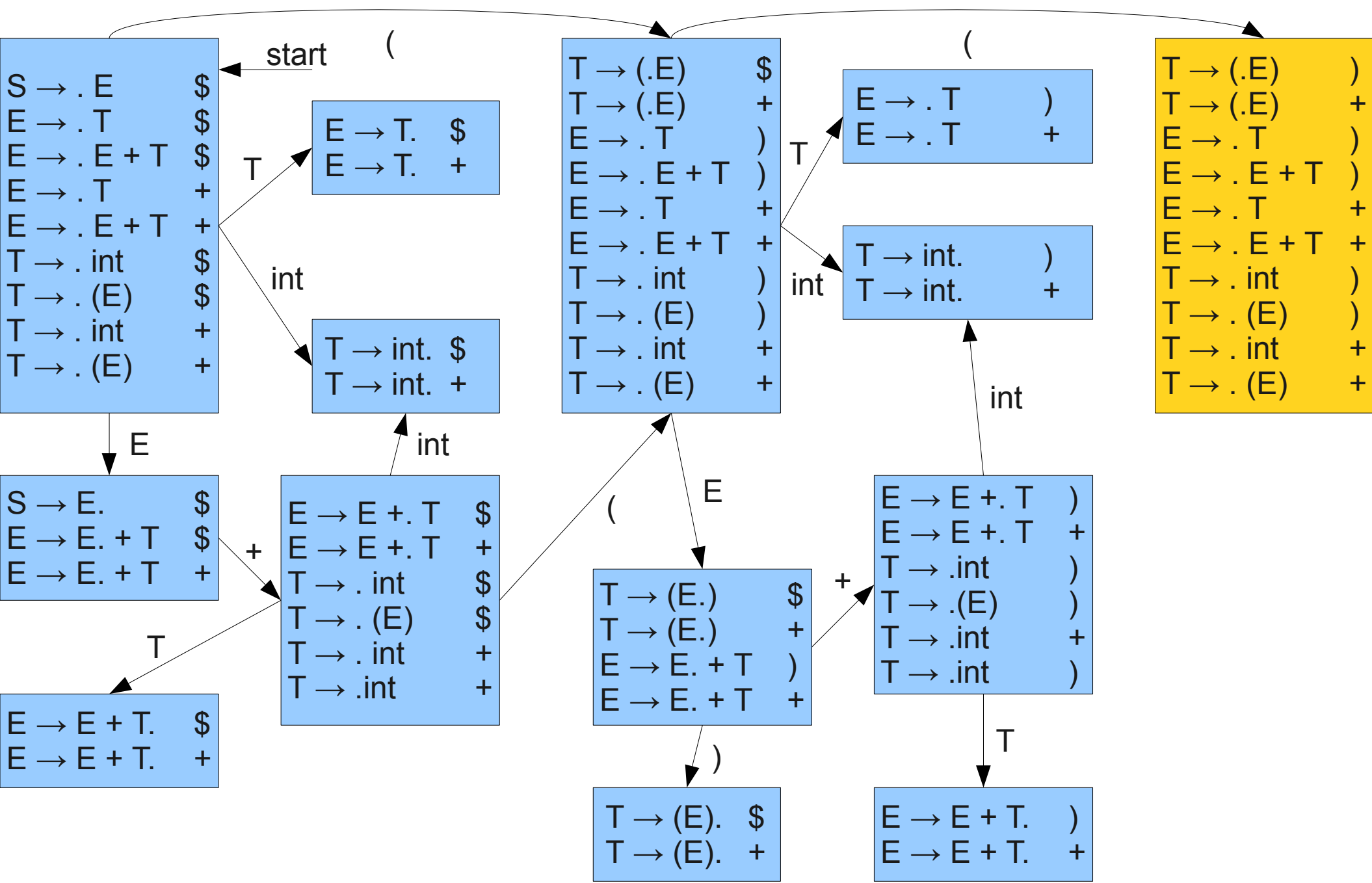
Building Deterministic LR(1) Automata



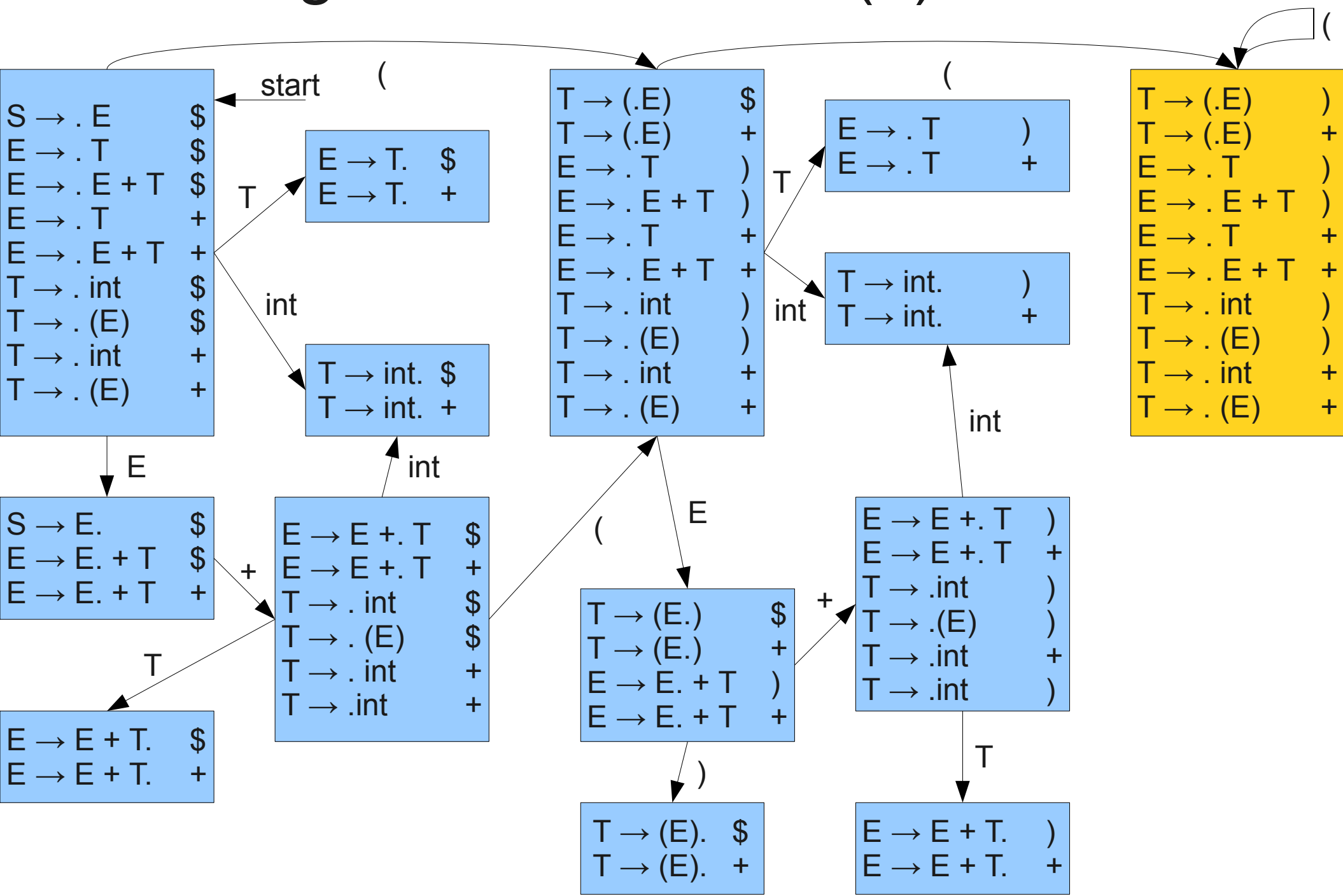
Building Deterministic LR(1) Automata



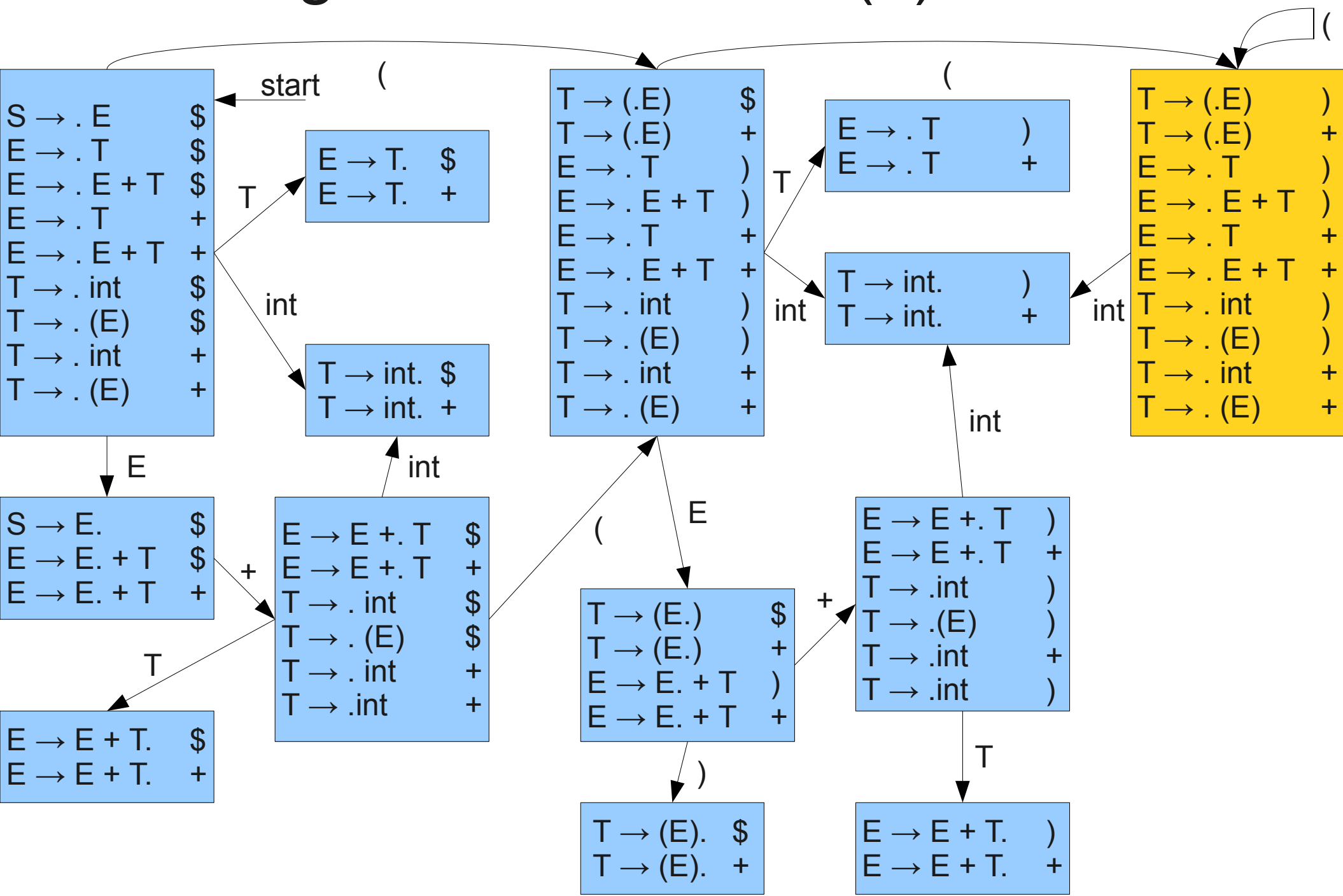
Building Deterministic LR(1) Automata



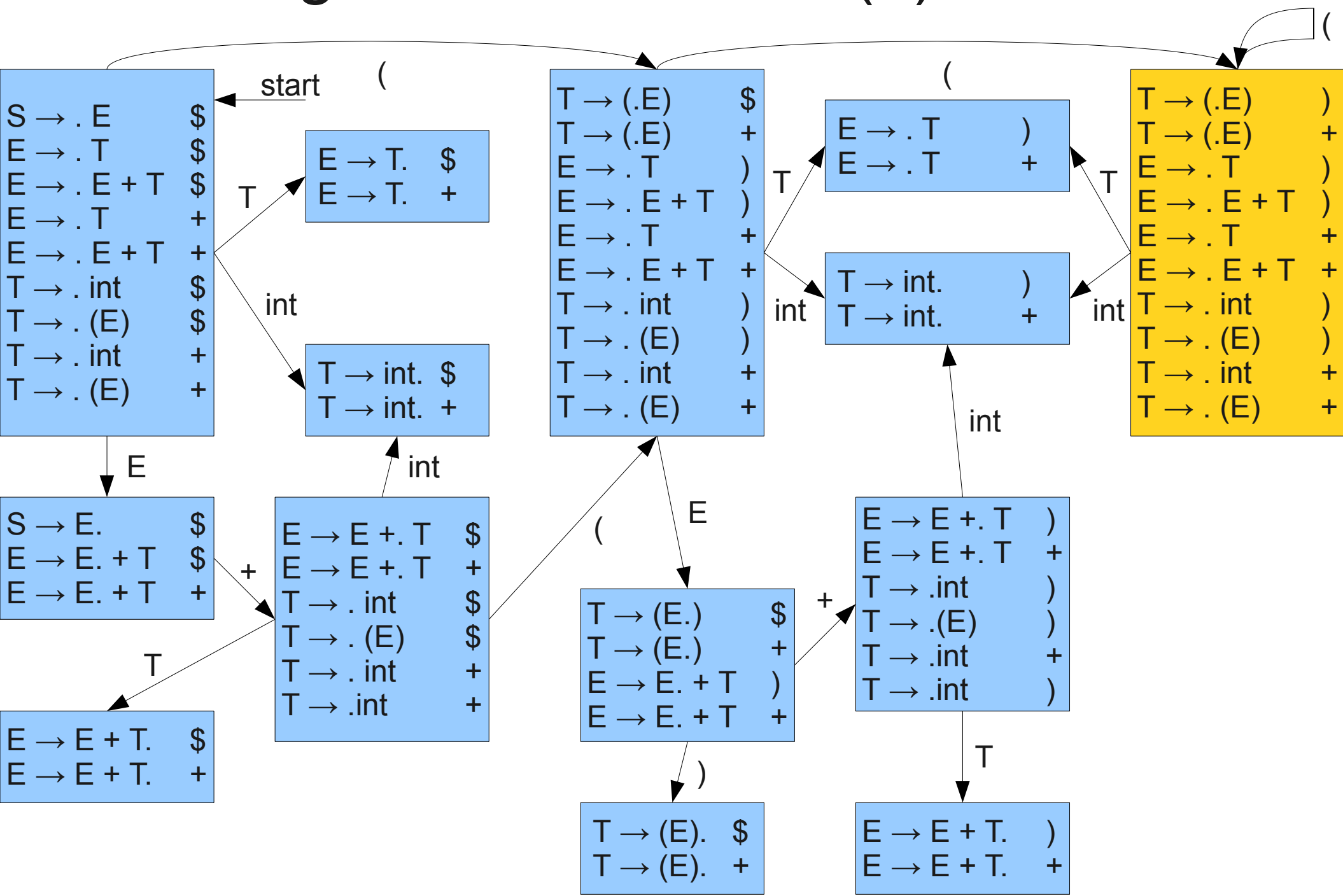
Building Deterministic LR(1) Automata



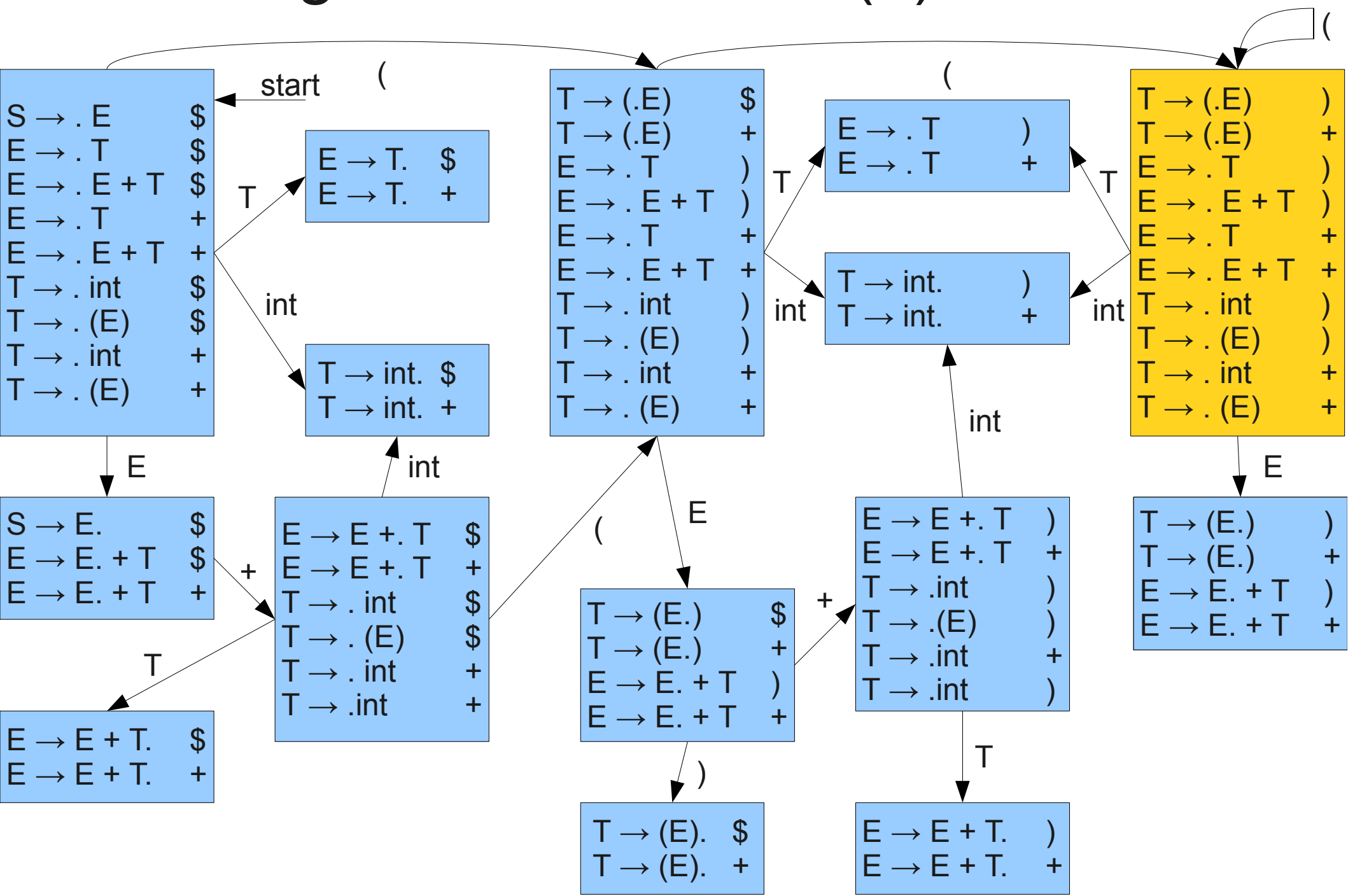
Building Deterministic LR(1) Automata



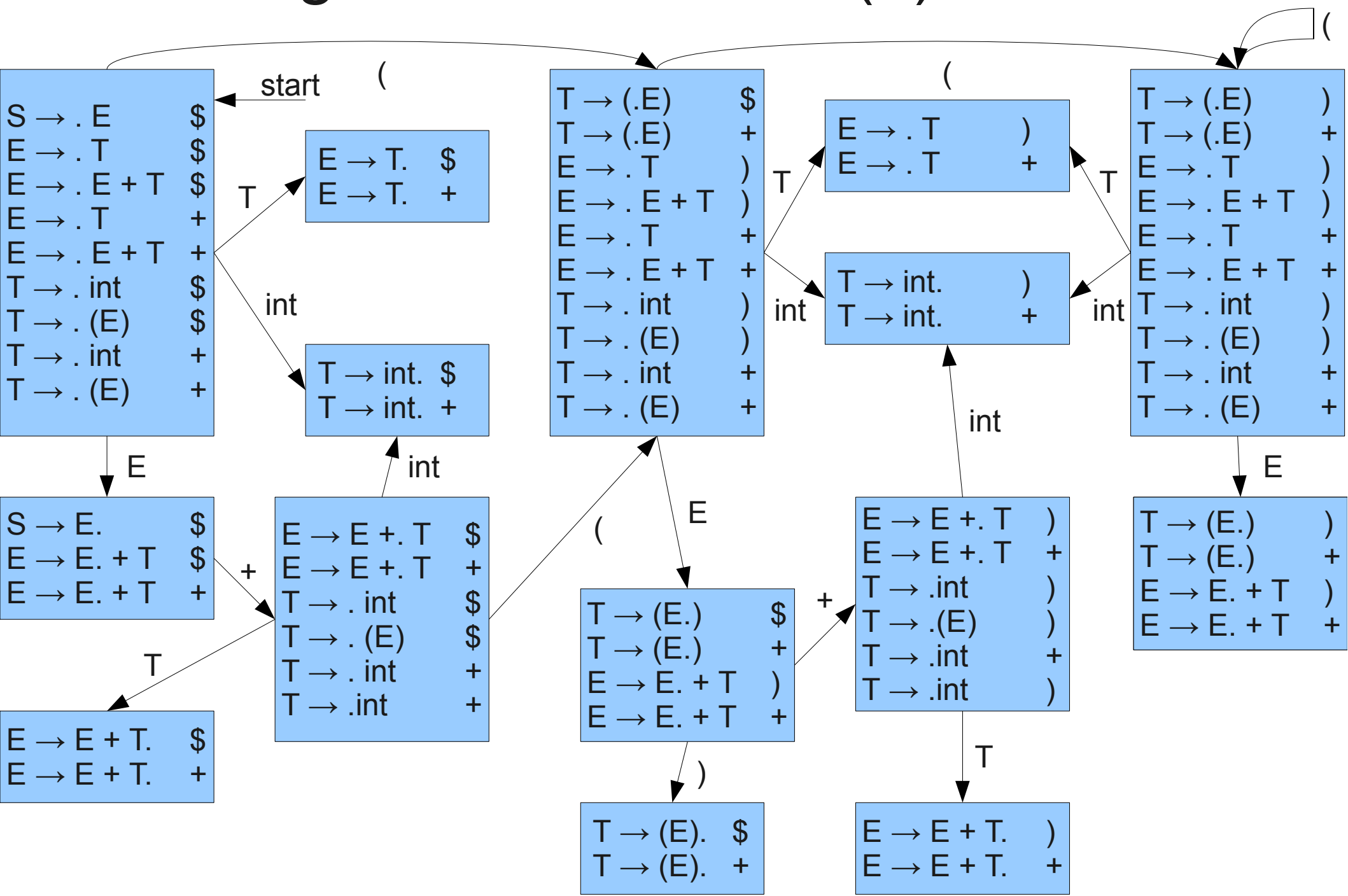
Building Deterministic LR(1) Automata



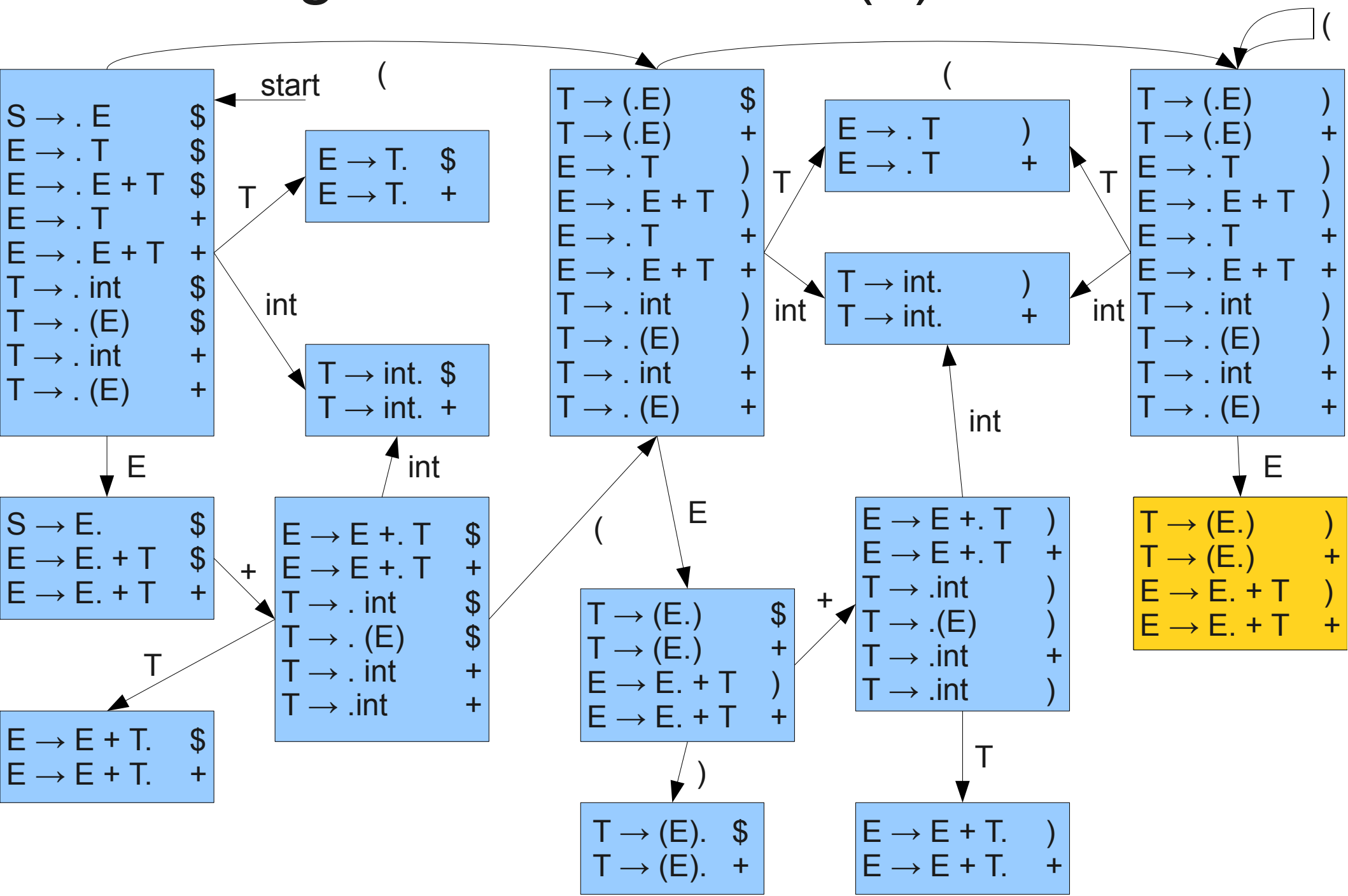
Building Deterministic LR(1) Automata



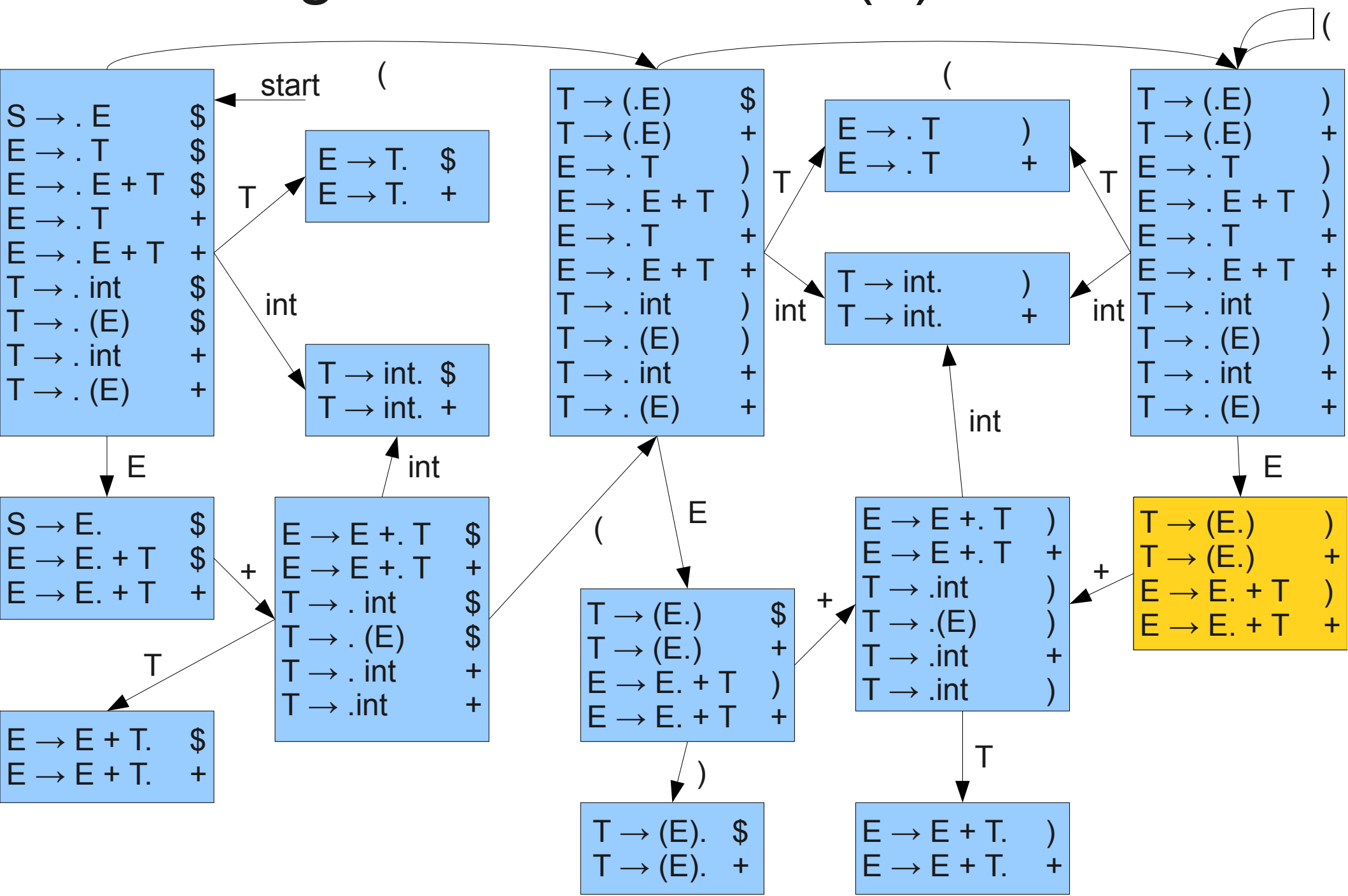
Building Deterministic LR(1) Automata



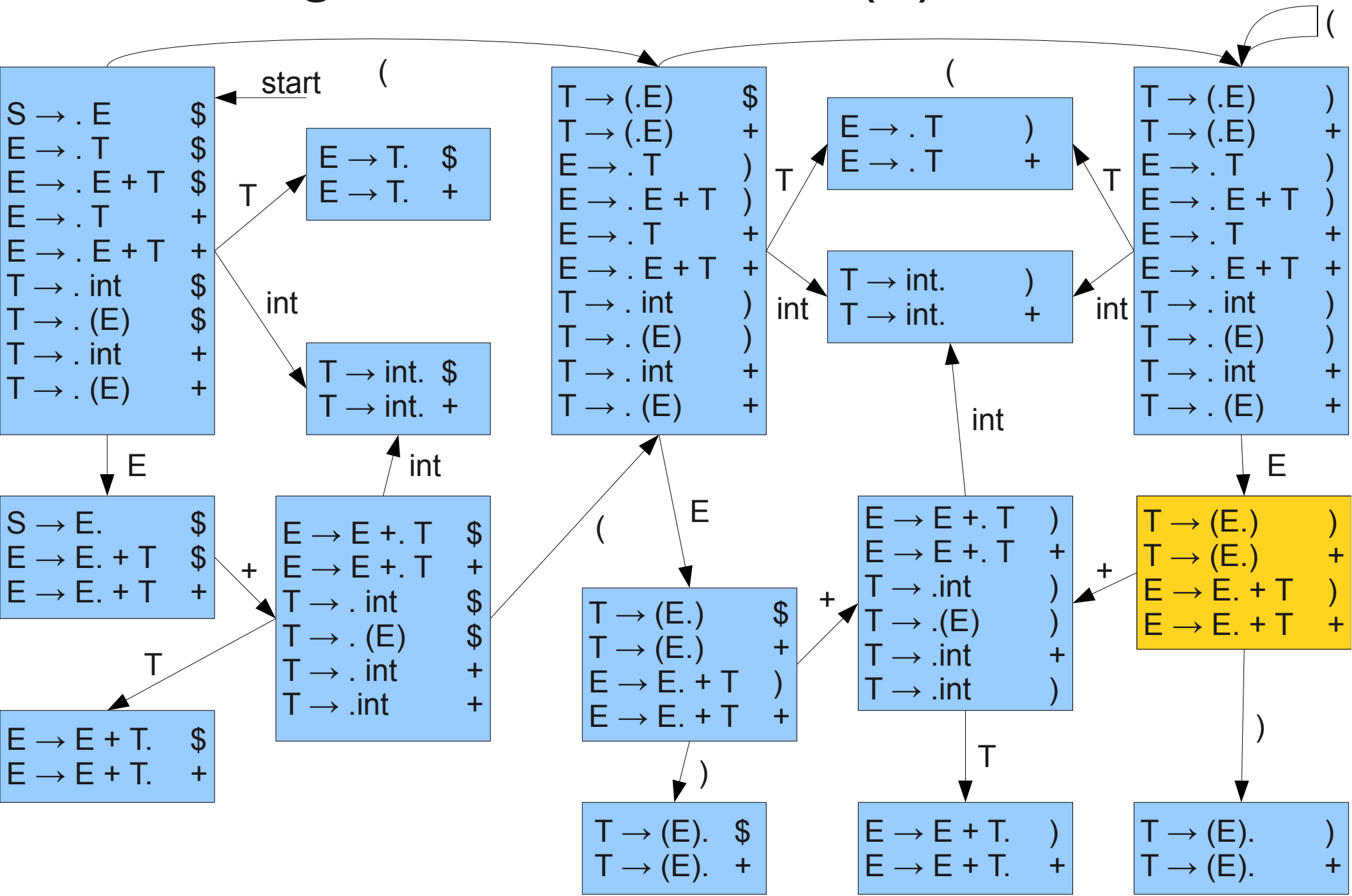
Building Deterministic LR(1) Automata



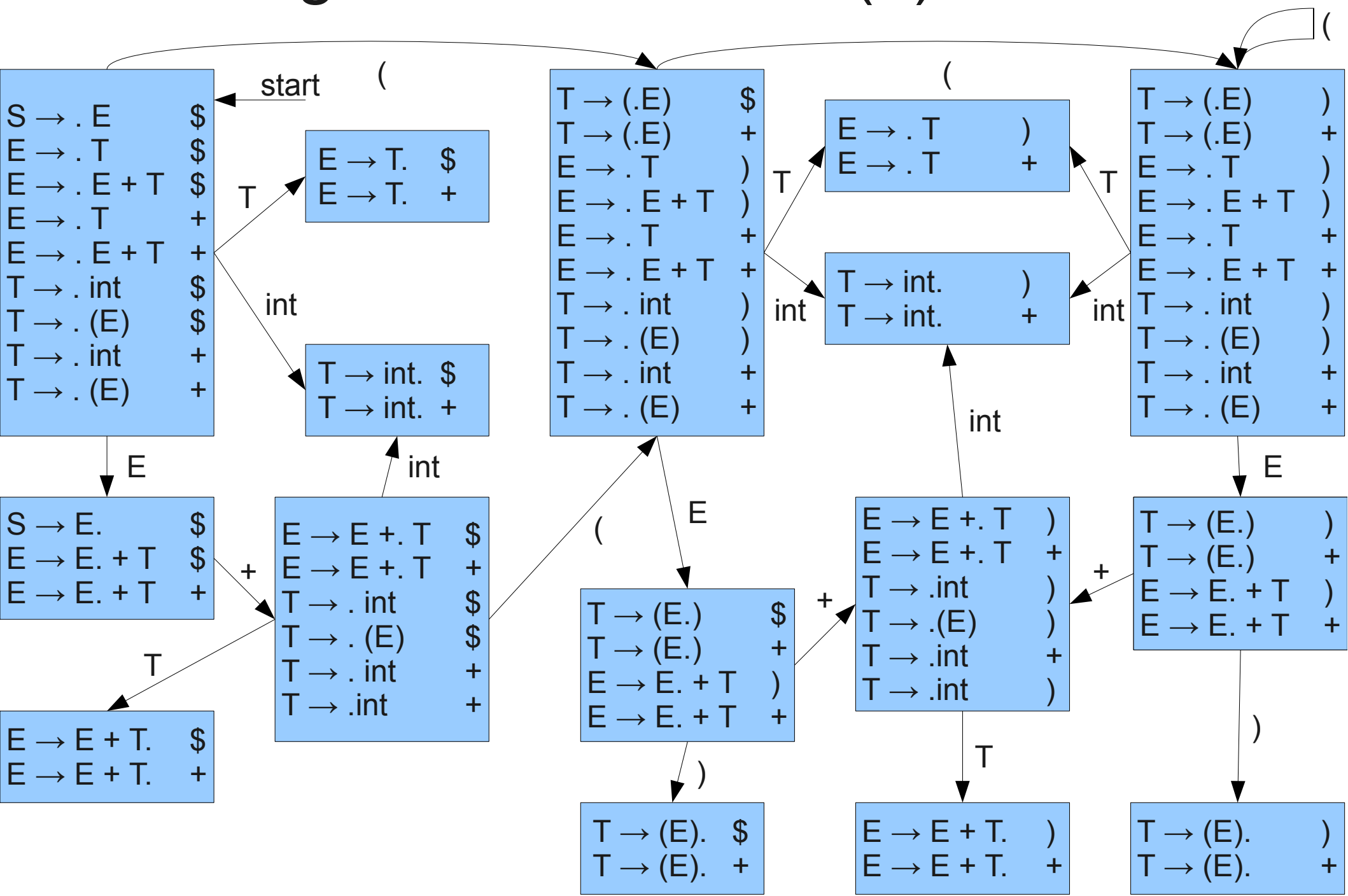
Building Deterministic LR(1) Automata



Building Deterministic LR(1) Automata



Building Deterministic LR(1) Automata



Constructing LR(1) Automata II

- Begin in a state containing $S \rightarrow \cdot E [\$]$, where S is the start symbol.
- Compute the **closure** of the state:
 - If $A \rightarrow v \cdot Bw [t]$ is in the state, add $B \rightarrow \cdot x [r]$ to the state for each production $B \rightarrow x$ and for each terminal r in $\text{FIRST}(wt)$
- Repeat until no new states are added:
 - If a state contains a production $A \rightarrow v \cdot \mathbf{x}w [r]$ for symbol \mathbf{x} , add a transition on t from that state to the state containing the closure of $A \rightarrow v\mathbf{x} \cdot w [r]$

The Structure of LR(1) Automata

- Every LR(1) automaton simulates two processes simultaneously:
 - An **LR(0) automaton** for finding handles.
 - A **lookahead tracker** for determining what the lookahead is.
- Removing the lookaheads from an LR(1) automaton results in a (much larger) LR(0) automaton for the same grammar.

Representing LR(1) Automata

- As with LR(0), use **action** and **goto** tables.
- **goto** table defined as before; encodes transition table as map from (state, token) to states.
- **action** table maps pairs (state, lookahead) to actions.
- Commonly combined into **action/goto** table.

$S \rightarrow E$ (1)
 $E \rightarrow T$ (2)
 $E \rightarrow E + T$ (3)
 $T \rightarrow \text{int}$ (4)
 $T \rightarrow (E)$ (5)

	int	()	+	\$	T	E
1	s5					s4	s2
2				s6	ACCEPT		
3				r3	r3		
4				r2	r2		
5				r5	r5		
6	s5	s7				s3	
7	s10	s14				s10	s8
8			s9	s12			
9				r5	r5		
10			r2	r2			
11			r4	r4			
12	s11					s13	
13			r3	r3			
14	s11		s14			s10	s15
15			s16	s12			
16			r5	r5			

$S \rightarrow E$ (1)
 $E \rightarrow T$ (2)
 $E \rightarrow E + T$ (3)
 $T \rightarrow \text{int}$ (4)
 $T \rightarrow (E)$ (5)

	int	()	+	\$	T	E
1	s5					s4	s2
2				s6	ACCEPT		
3				r3	r3		
4				r2	r2		
5				r5	r5		
6	s5	s7				s3	
7	s10	s14				s10	s8
8			s9	s12			
9				r5	r5		
10			r2	r2			
11			r4	r4			
12	s11					s13	
13			r3	r3			
14	s11		s14			s10	s15
15			s16	s12			
16			r5	r5			

The LR(1) Parsing Algorithm

- Begin with an empty stack and the input set to $S\$. Set **state** to the initial state.$
- Repeat the following:
 - Let the next symbol of input be **t**.
 - If **action[state, t]** is **shift**, then shift the input and set **state = goto[state, t]**.
 - If **action[state, t]** is **reduce** $A \rightarrow v$:
 - Pop $|v|$ symbols off the stack; replace them with **A**.
 - Let the state atop the stack be **top-state**.
 - Set **state = goto[top-state, A]**
 - If **action[state, t]** is **accept**, then the parse is done.
 - If **action[state, t]** is **error**, report an error.

Constructing LR(1) Parse Tables

- For each state s :
 - If there is a production $A \rightarrow v \cdot [t]$, set **action[s, t] = reduce $A \rightarrow v$.**
 - If there is the special production $S \rightarrow E \cdot [t]$, where S is the start state, set **action[s, t] = accept.**
 - If there is a transition out of s on symbol t , set **action[s, t] = shift.**
- Set all other actions to **error.**
- If any table entry contains two or more actions, the grammar is not LR(1).

The Power of LR(1)

- LR(1) parsers are **extremely powerful**.
- Any LL(1) grammar is LR(1).
- Any LR(0) grammar is LR(1).
- Any **deterministic language** has an LR(1) grammar.
 - Any language that can be recognized by some automaton-driven shift/reduce parser has an LR(1) grammar.
- Any LR(k) **language** is LR(1), though individual LR(k) **grammars** might not be.

LR(1) Automata are **Huge**

- In a grammar with n terminals, could in theory be $O(2^n)$ times as large as the LR(0) automaton.
 - Replicate each state with all $O(2^n)$ possible lookaheads.
- LR(1) tables for practical programming languages can have hundreds of thousands or even **millions** of states.
- Consequently, LR(1) parsers are rarely used in practice.

Is there a way to get the power of LR(1) without the huge table size?

Why is LR(1) so powerful?

- Intuitively, for two reasons:
- **Lookahead makes handle-finding easier.**
 - The LR(0) automaton says whether there could be a handle later on based on no right context.
 - The LR(1) automaton can predict whether it needs to reduce based on more information.
- **More states encode more information.**
 - LR(1) lookaheads are very good because there's a greater number of states to be in.
- **Goal:** Incorporate lookahead without increasing the number of states.

Revisiting Shift/Reduce Conflicts

- A shift/reduce conflict is a state that looks like this:

$$A \rightarrow v \cdot$$

$$B \rightarrow w \cdot x$$

- In LR(0), this is simply not allowed.
- In LR(1), this can be avoided by using lookahead to determine whether to shift or reduce.

SLR(1)

- **Simple(1) LR**
- Minor modification to LR(0) automaton that uses lookahead to avoid shift/reduce conflicts.
- Idea: Only reduce $A \rightarrow v$ if the next token t is in $\text{FOLLOW}(A)$.
- Automaton identical to LR(0) automaton; only change is when we choose to reduce.

SLR(1) Parsing

$S \rightarrow E$

$E \rightarrow T$

$E \rightarrow T + E$

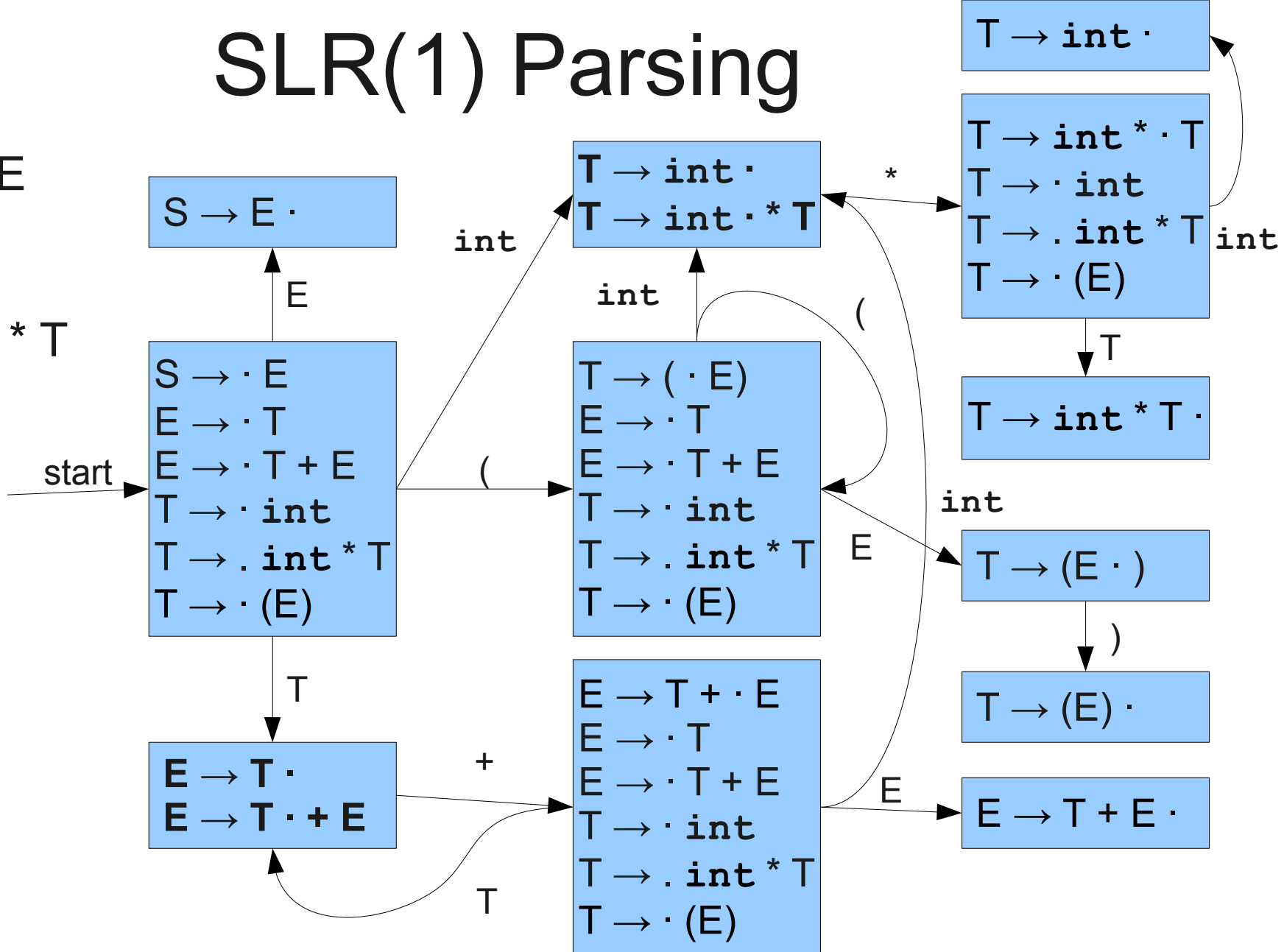
$T \rightarrow \mathbf{int}$

$T \rightarrow (E)$

$T \rightarrow \mathbf{int} * T$

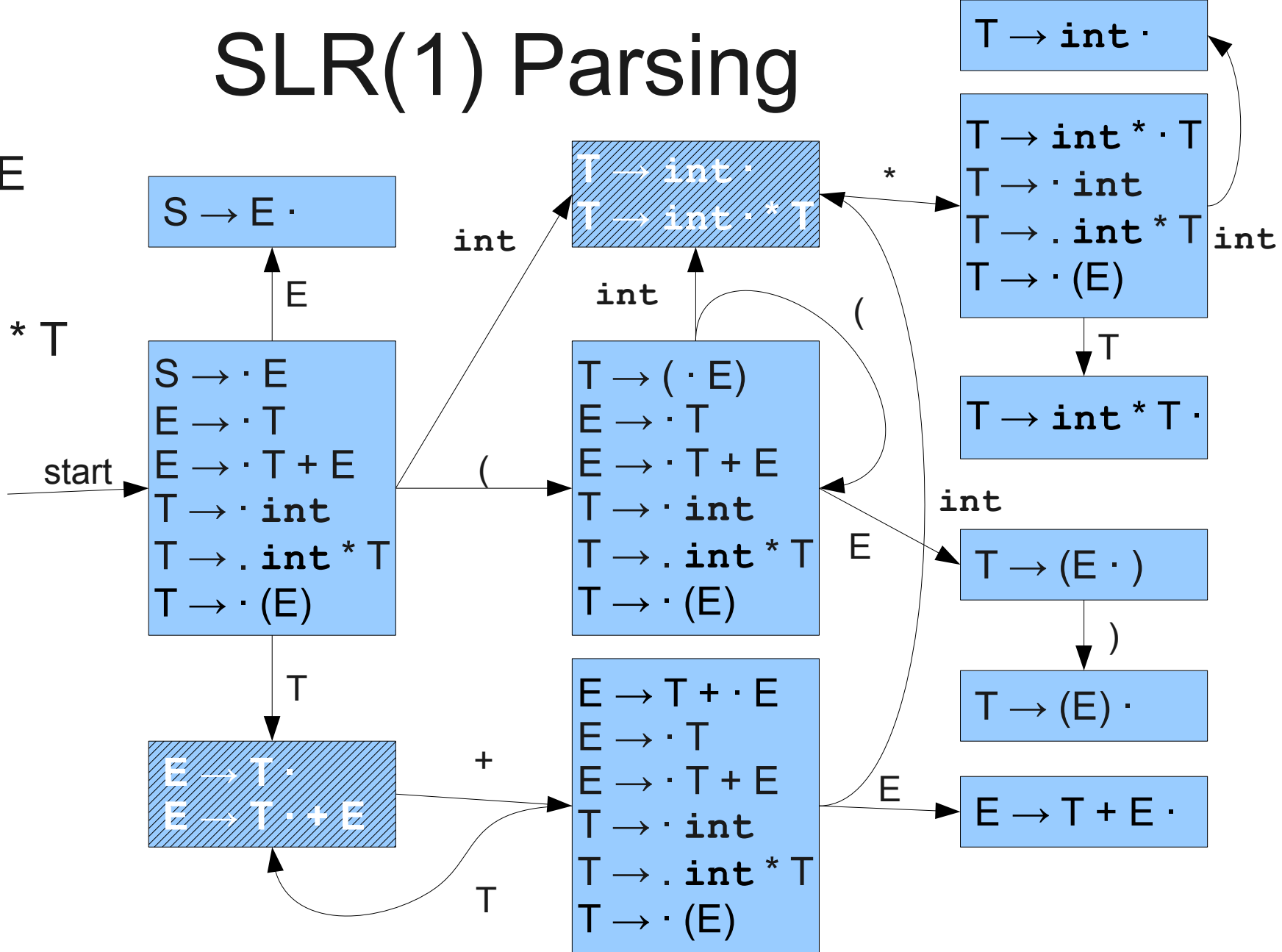
SLR(1) Parsing

$S \rightarrow E$
 $E \rightarrow T$
 $E \rightarrow T + E$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$
 $T \rightarrow \text{int} * T$



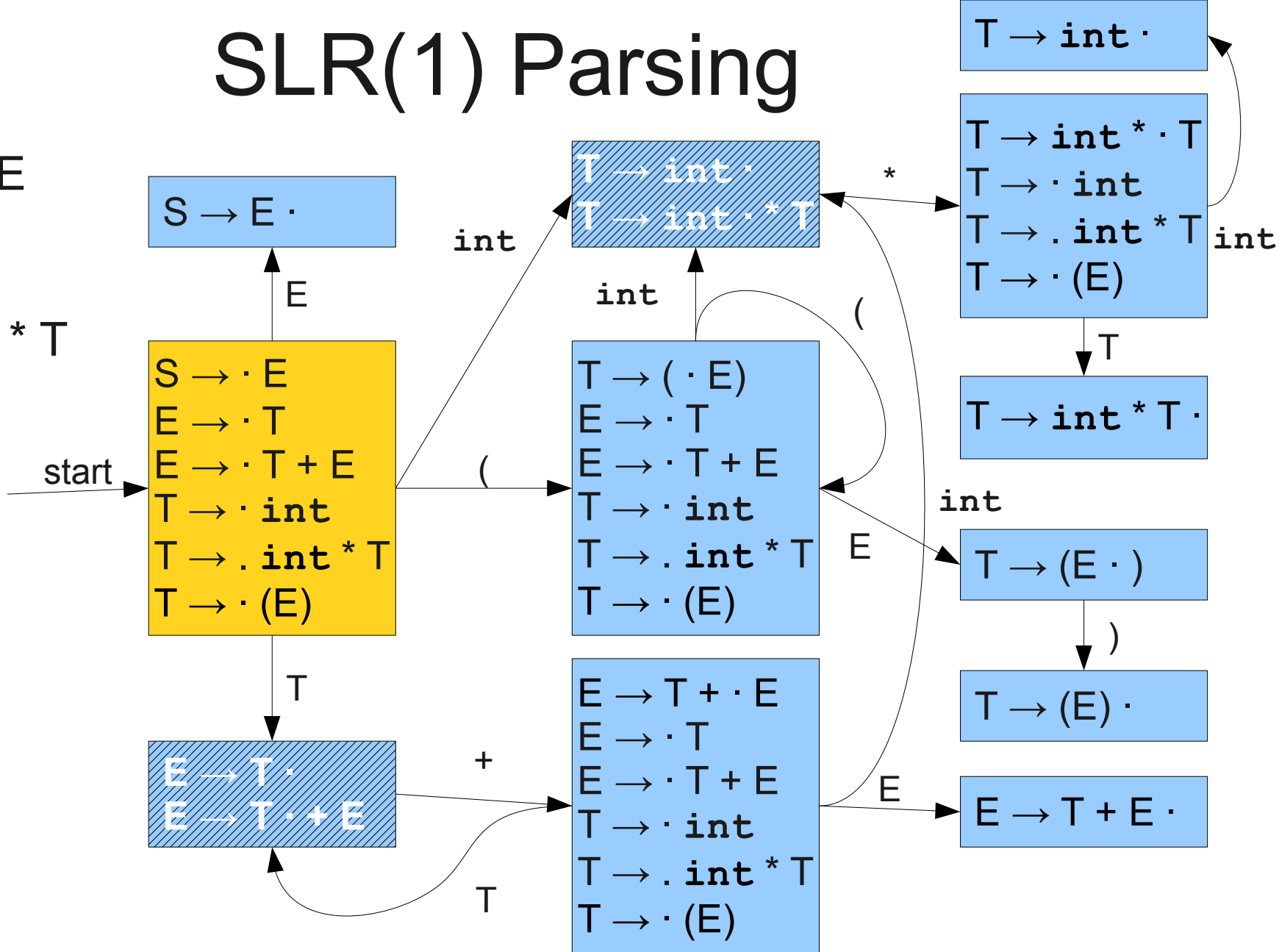
SLR(1) Parsing

$S \rightarrow E$
 $E \rightarrow T$
 $E \rightarrow T + E$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$
 $T \rightarrow \text{int} * T$



SLR(1) Parsing

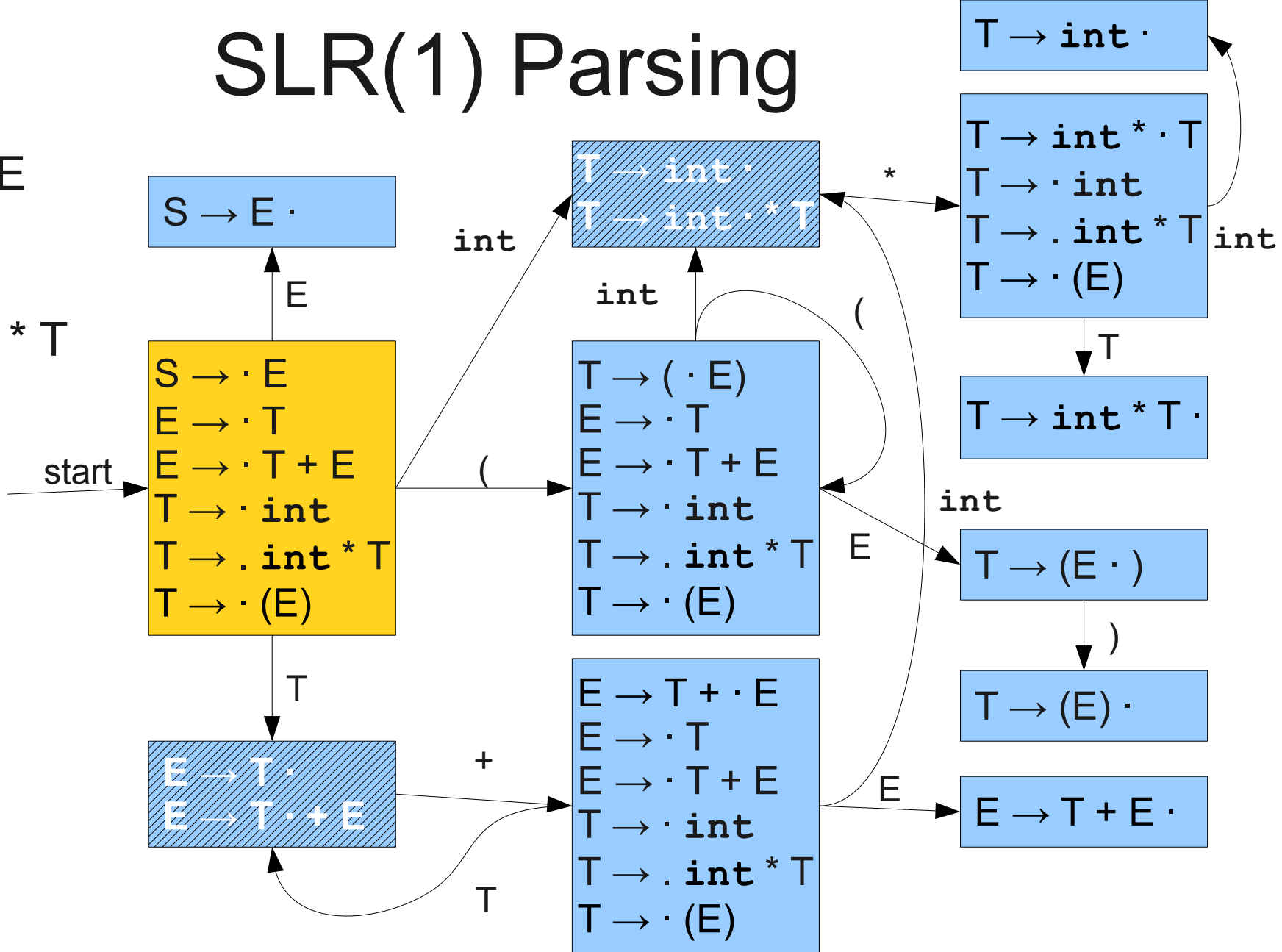
$S \rightarrow E$
 $E \rightarrow T$
 $E \rightarrow T + E$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$
 $T \rightarrow \text{int} * T$



int + int * int \$

SLR(1) Parsing

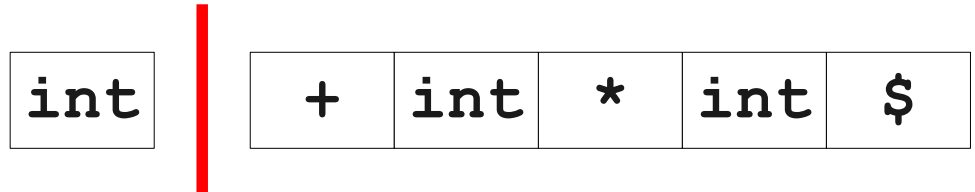
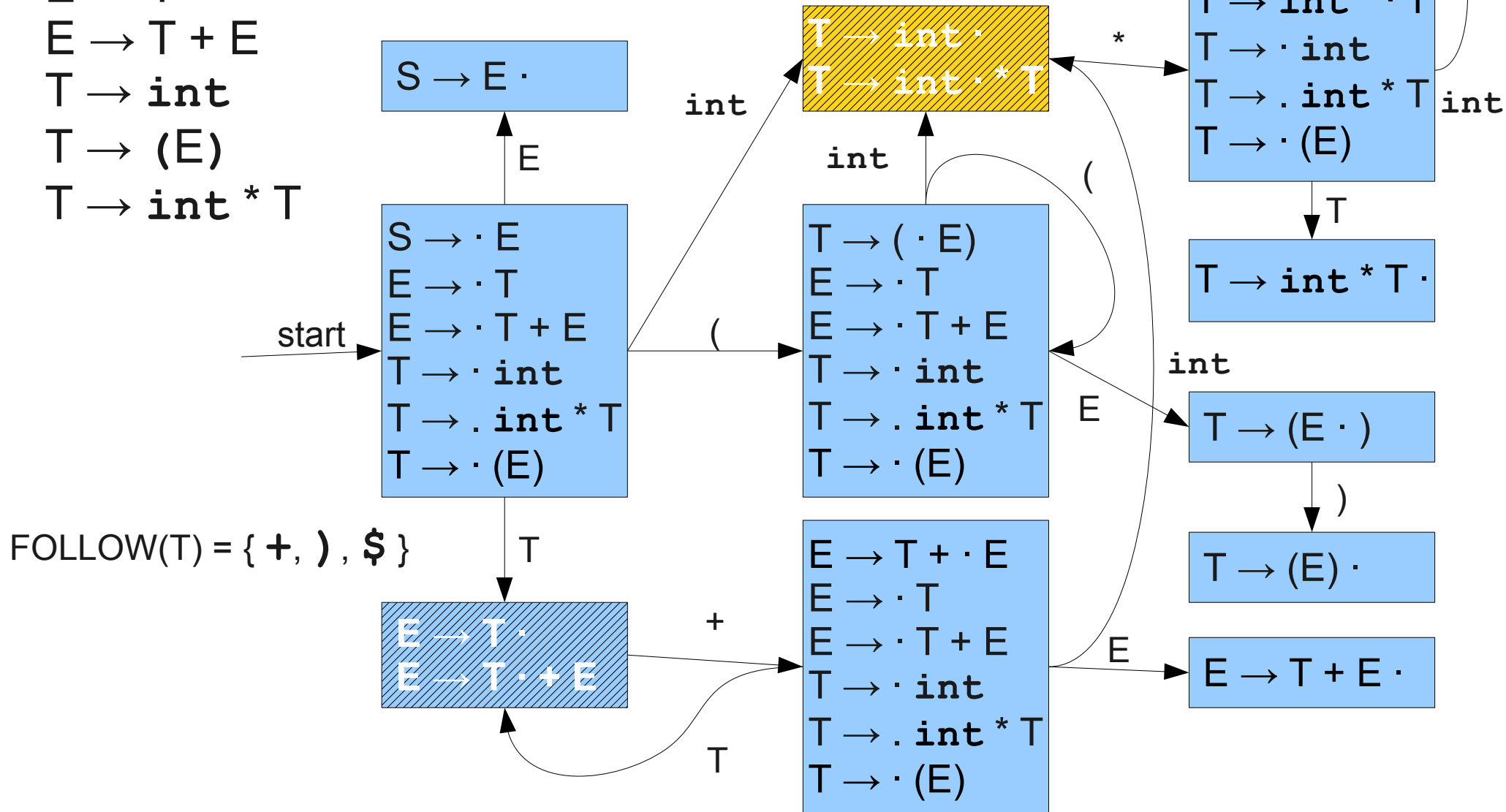
$S \rightarrow E$
 $E \rightarrow T$
 $E \rightarrow T + E$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$
 $T \rightarrow \text{int} * T$



int | + int * int \$

SLR(1) Parsing

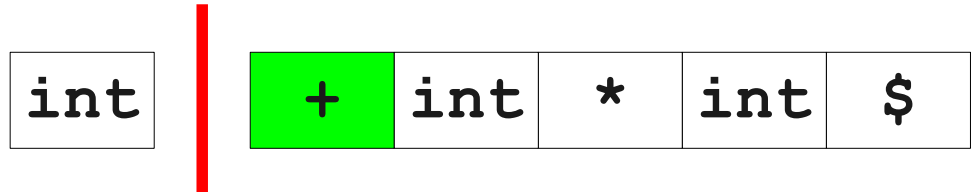
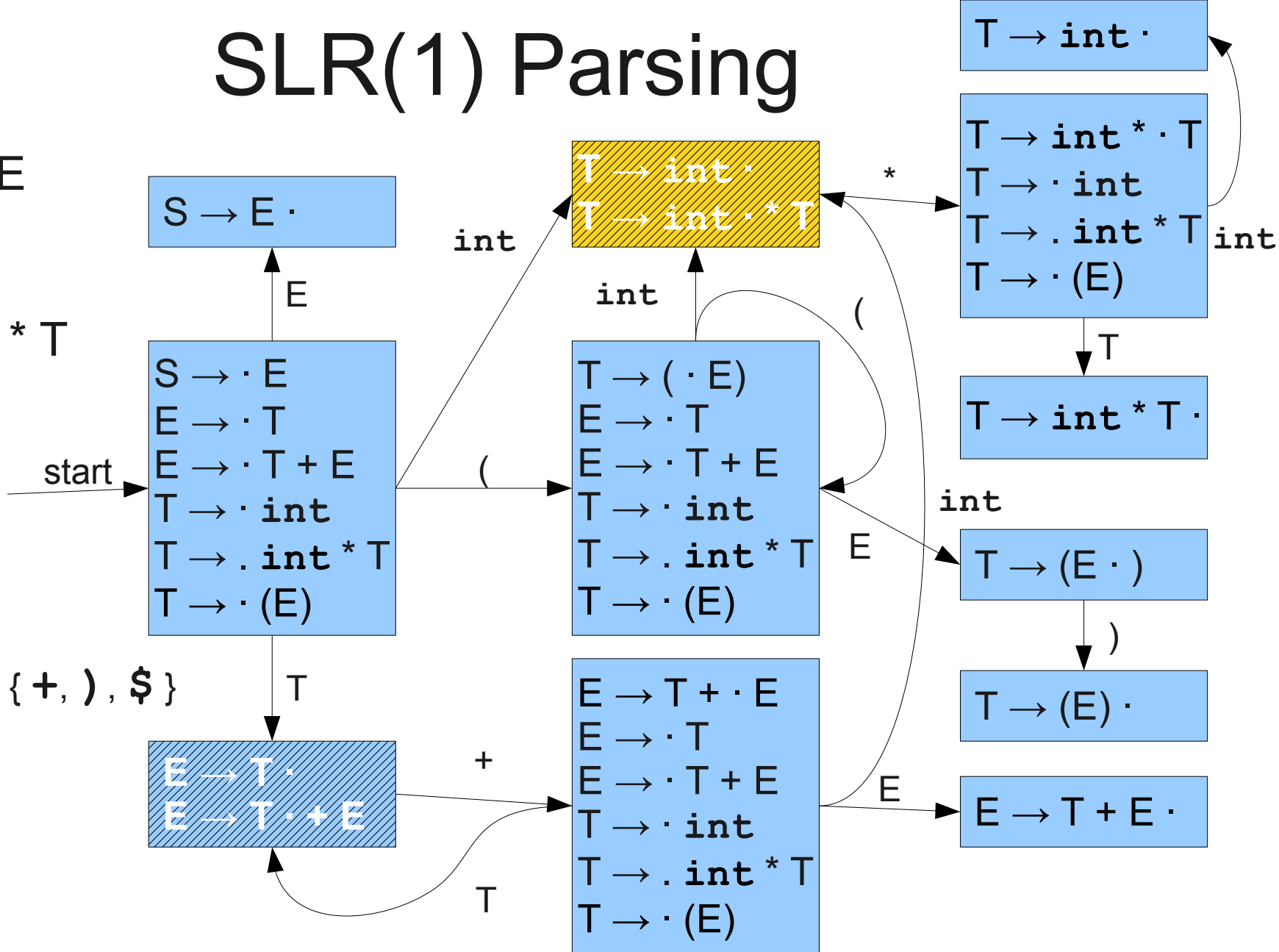
$S \rightarrow E$
 $E \rightarrow T$
 $E \rightarrow T + E$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$
 $T \rightarrow \text{int} * T$



SLR(1) Parsing

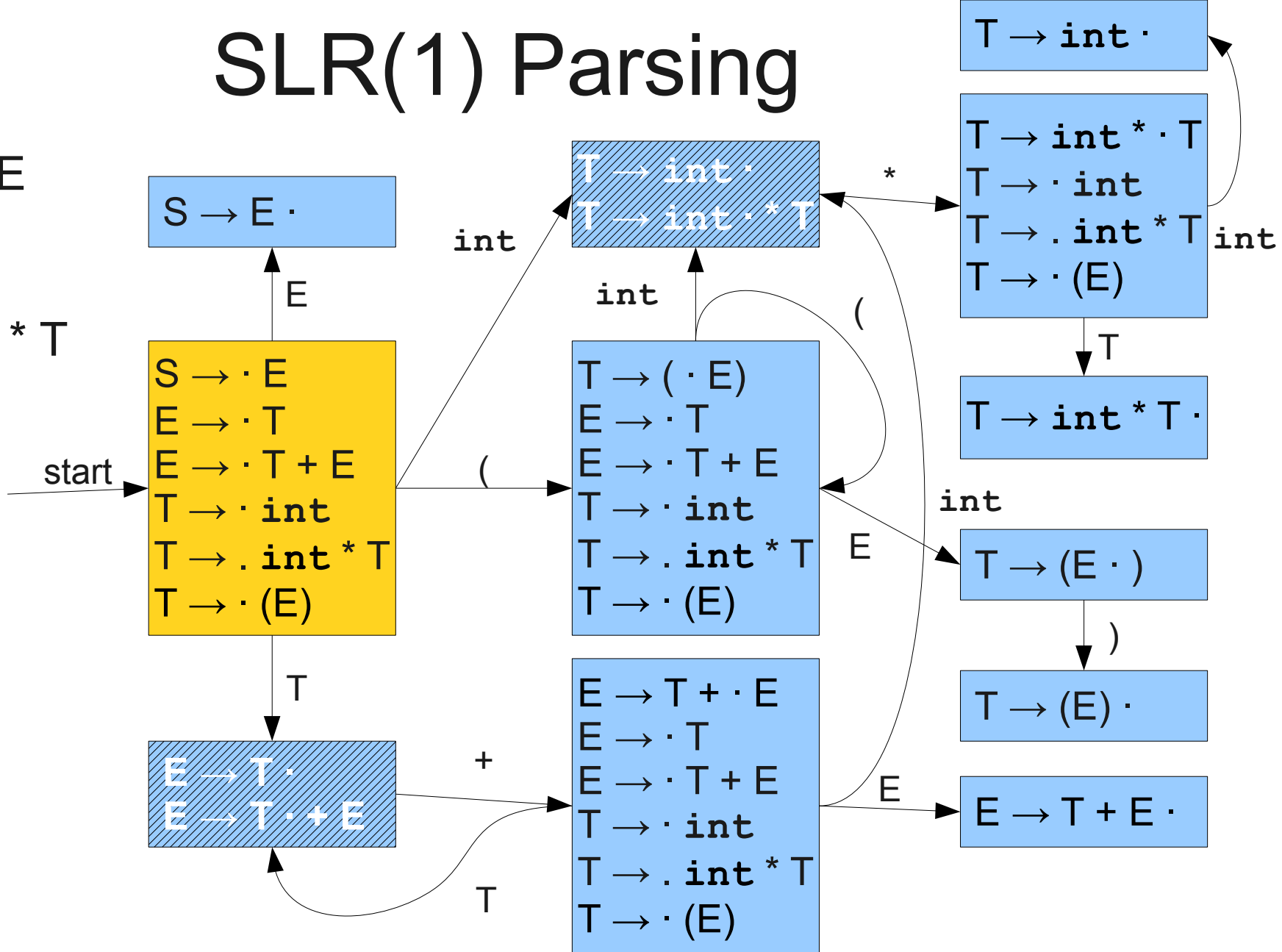
$S \rightarrow E$
 $E \rightarrow T$
 $E \rightarrow T + E$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$
 $T \rightarrow \text{int} * T$

$\text{FOLLOW}(T) = \{ +,), \$ \}$



SLR(1) Parsing

$S \rightarrow E$
 $E \rightarrow T$
 $E \rightarrow T + E$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$
 $T \rightarrow \text{int} * T$

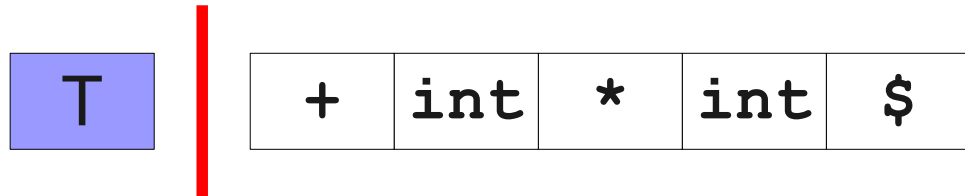
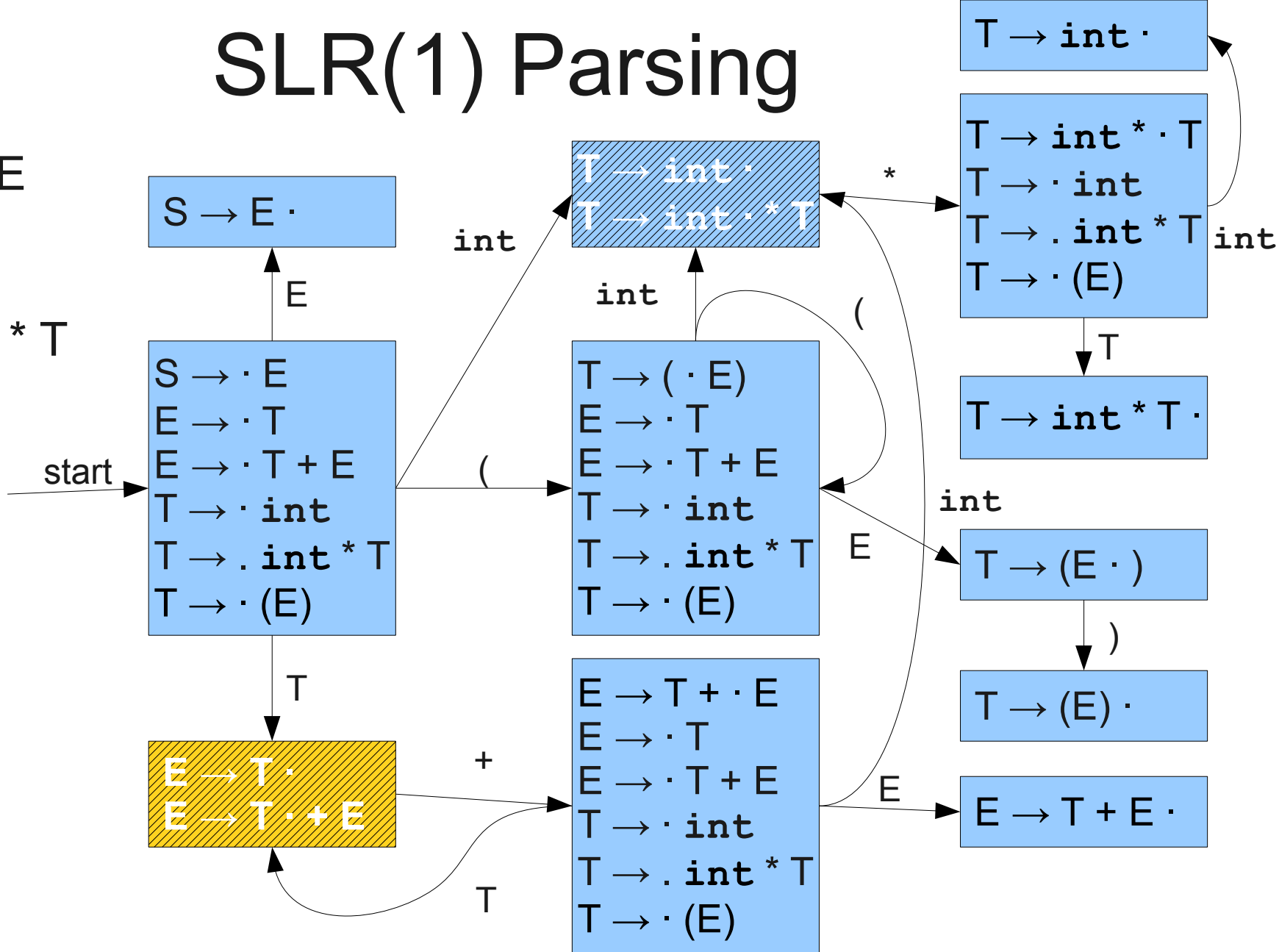


|

+	int	*	int	\$
---	-----	---	-----	----

SLR(1) Parsing

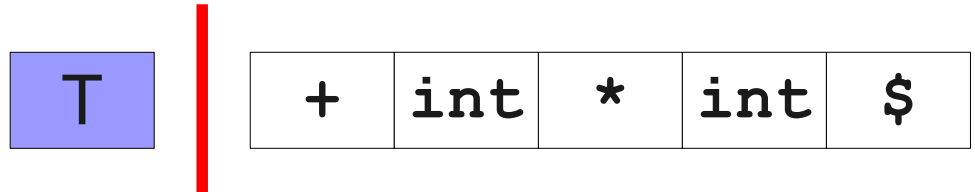
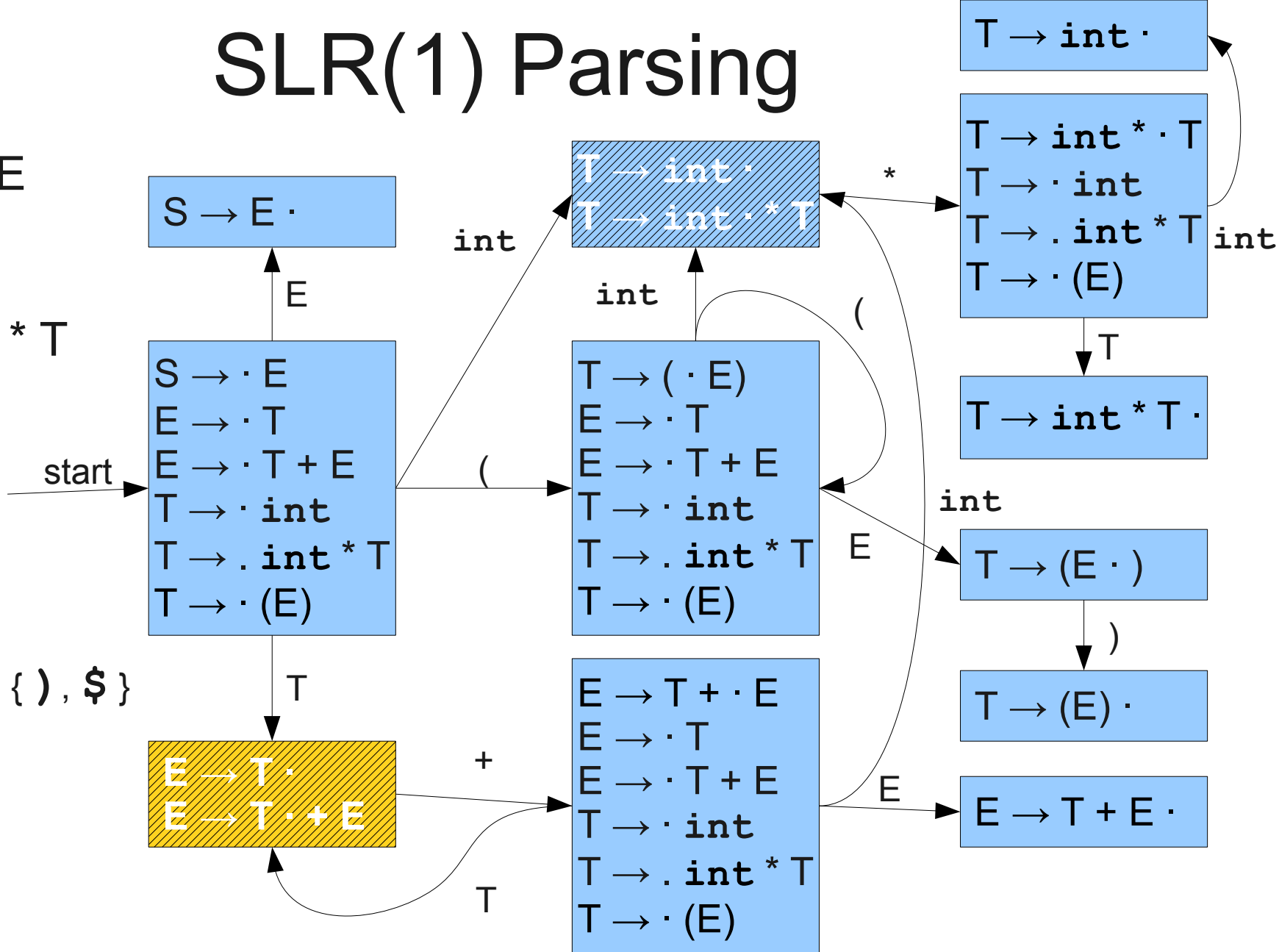
$S \rightarrow E$
 $E \rightarrow T$
 $E \rightarrow T + E$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$
 $T \rightarrow \text{int} * T$



SLR(1) Parsing

$S \rightarrow E$
 $E \rightarrow T$
 $E \rightarrow T + E$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$
 $T \rightarrow \text{int} * T$

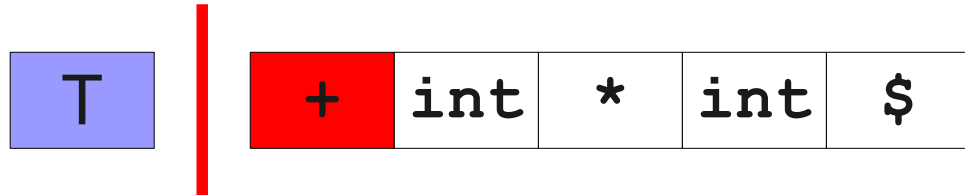
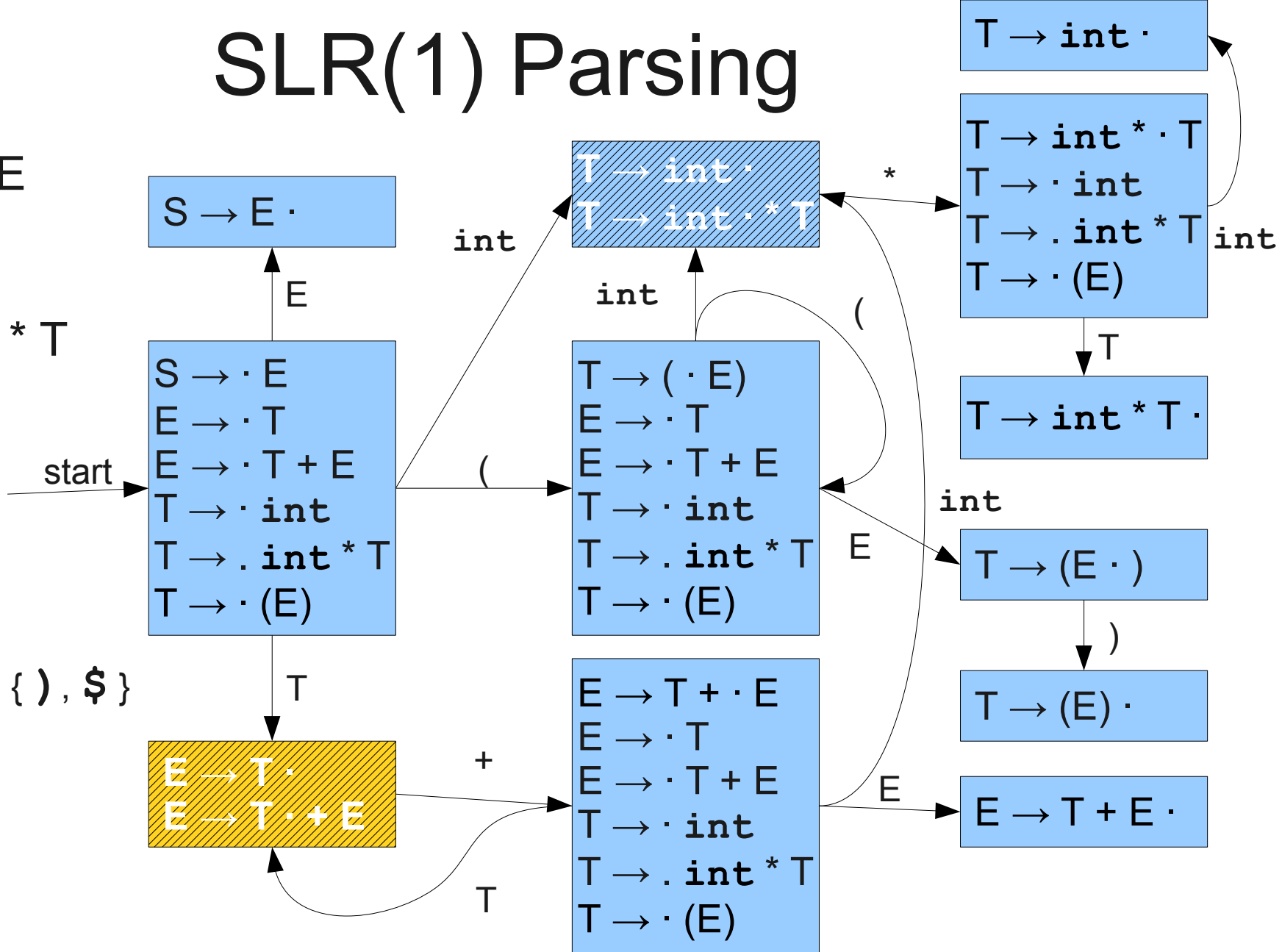
$\text{FOLLOW}(E) = \{), \$ \}$



SLR(1) Parsing

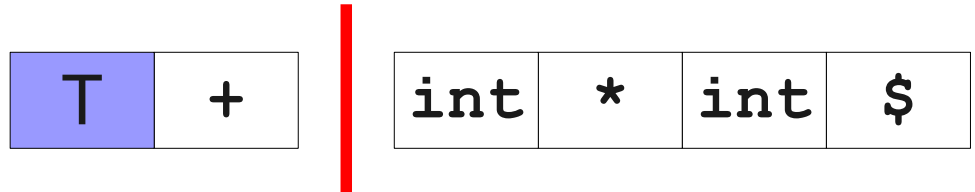
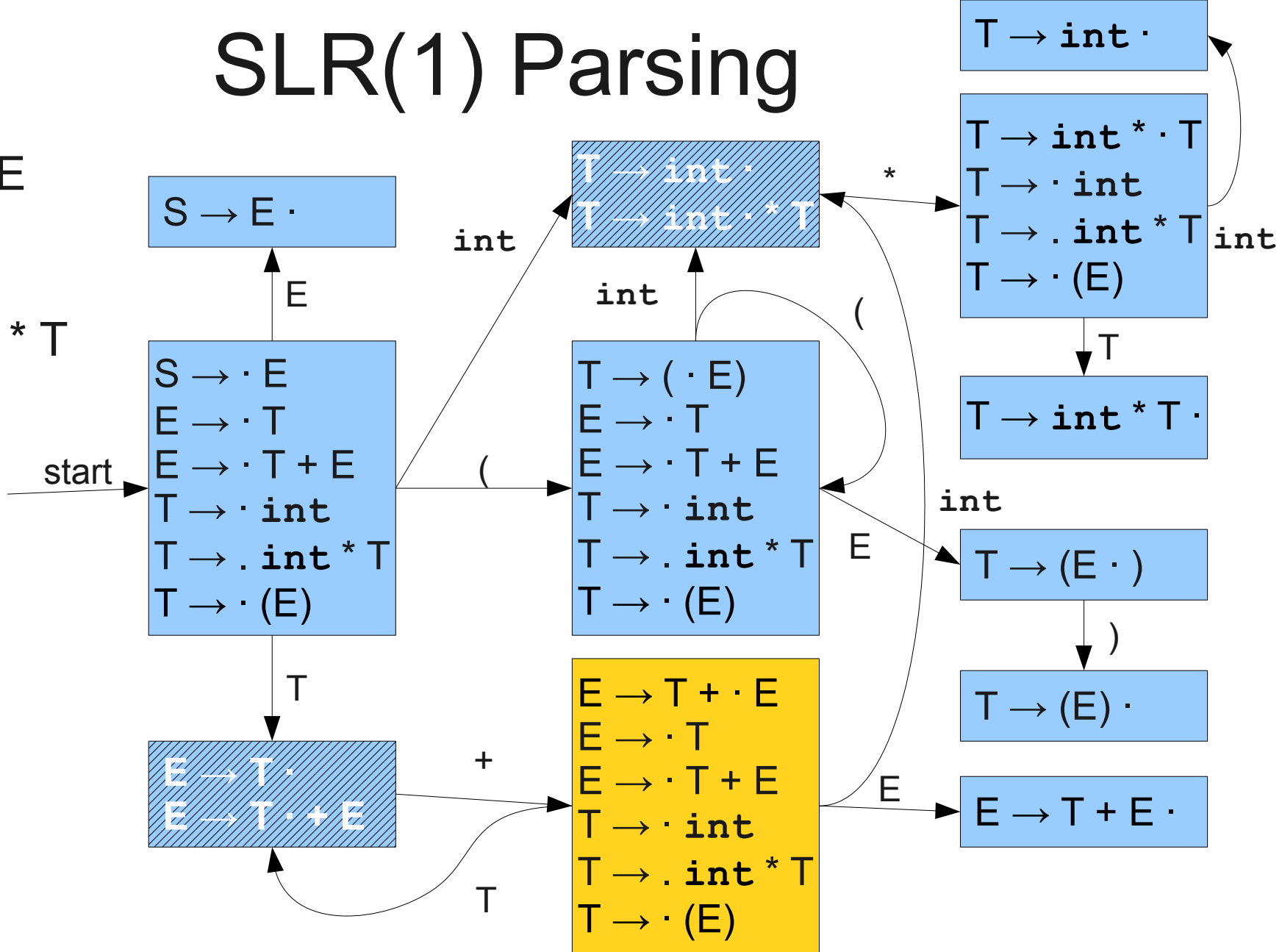
$S \rightarrow E$
 $E \rightarrow T$
 $E \rightarrow T + E$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$
 $T \rightarrow \text{int} * T$

$\text{FOLLOW}(E) = \{), \$ \}$



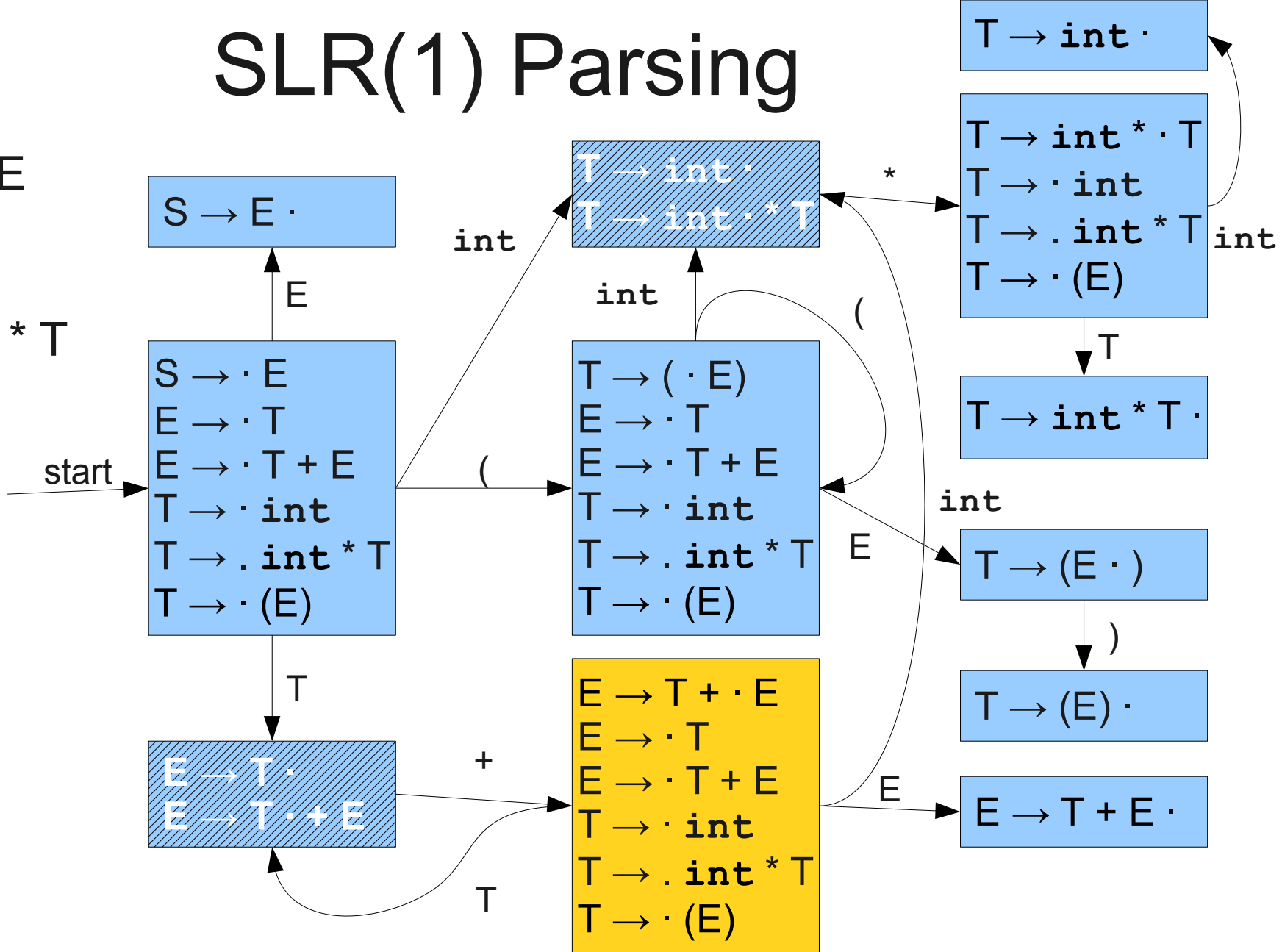
SLR(1) Parsing

$S \rightarrow E$
 $E \rightarrow T$
 $E \rightarrow T + E$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$
 $T \rightarrow \text{int} * T$



SLR(1) Parsing

$S \rightarrow E$
 $E \rightarrow T$
 $E \rightarrow T + E$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$
 $T \rightarrow \text{int} * T$

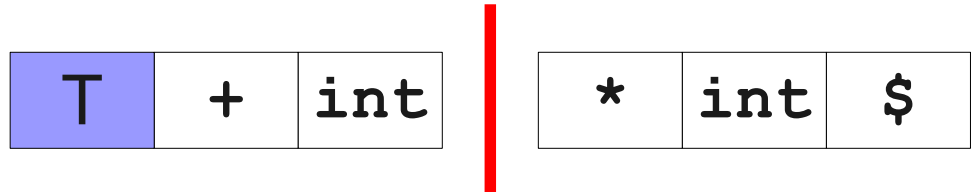
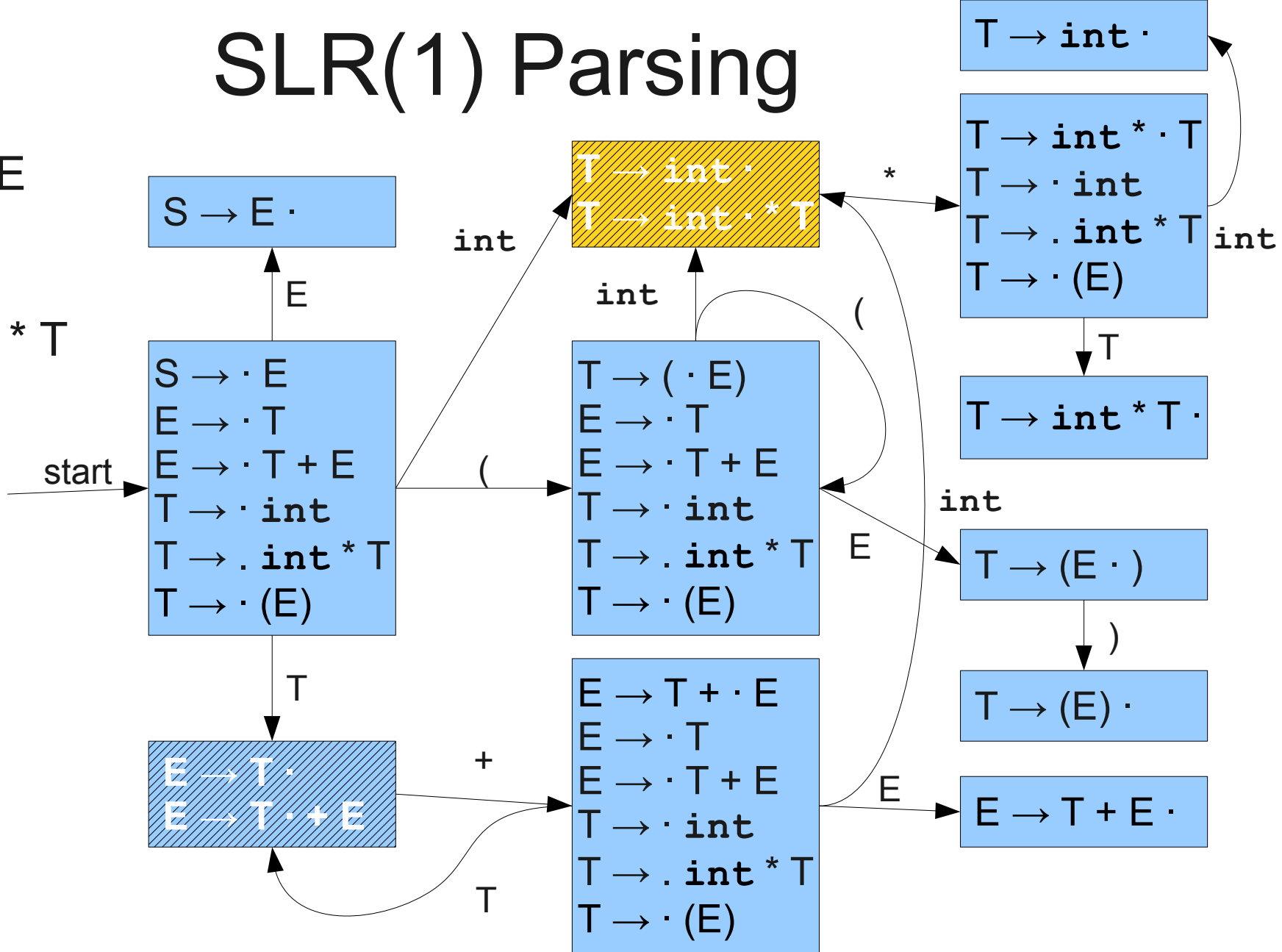


T	+	int
---	---	-----

*	int	\$
---	-----	----

SLR(1) Parsing

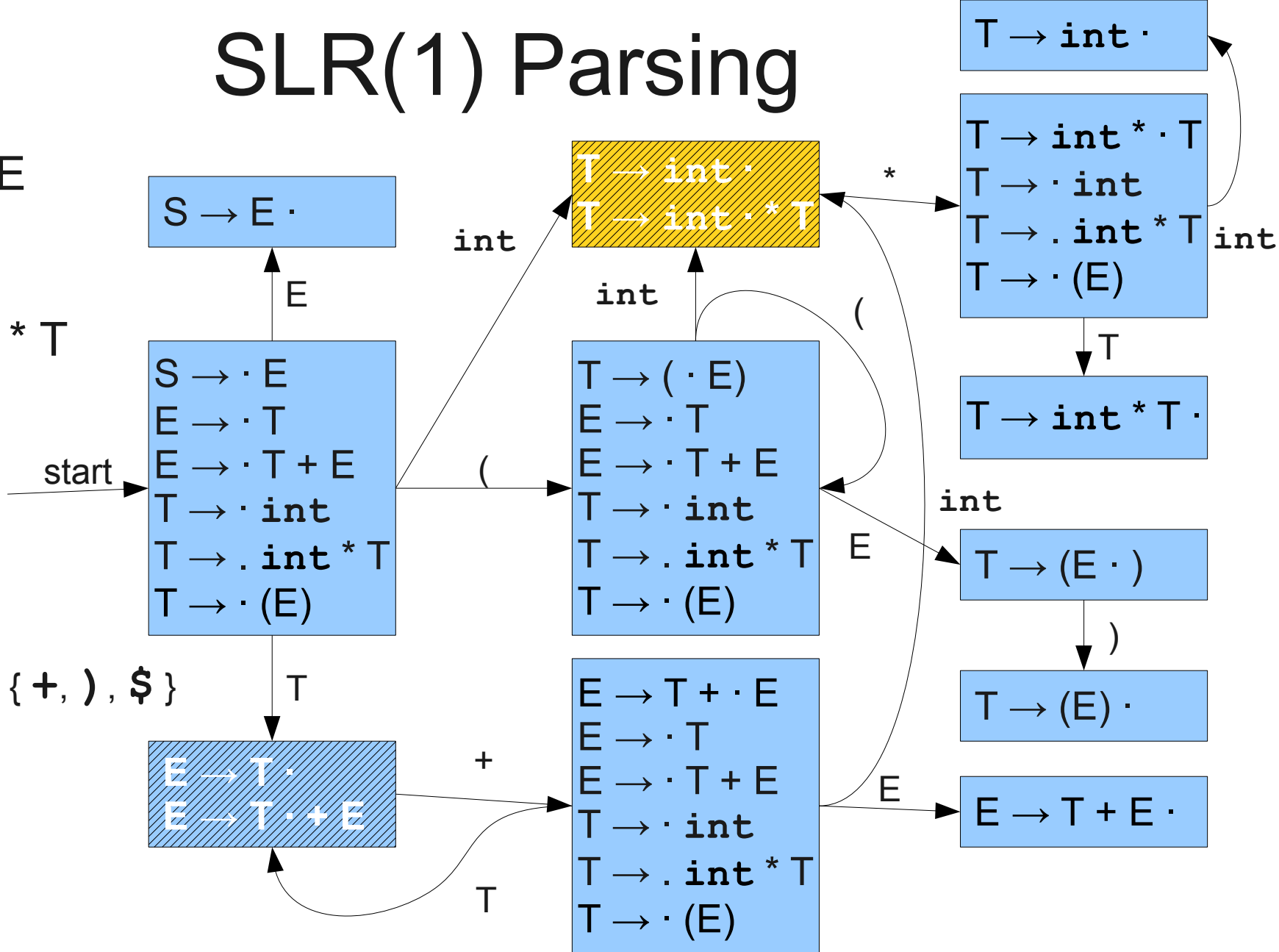
$S \rightarrow E$
 $E \rightarrow T$
 $E \rightarrow T + E$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$
 $T \rightarrow \text{int} * T$



SLR(1) Parsing

$S \rightarrow E$
 $E \rightarrow T$
 $E \rightarrow T + E$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$
 $T \rightarrow \text{int} * T$

$\text{FOLLOW}(T) = \{ +,), \$ \}$

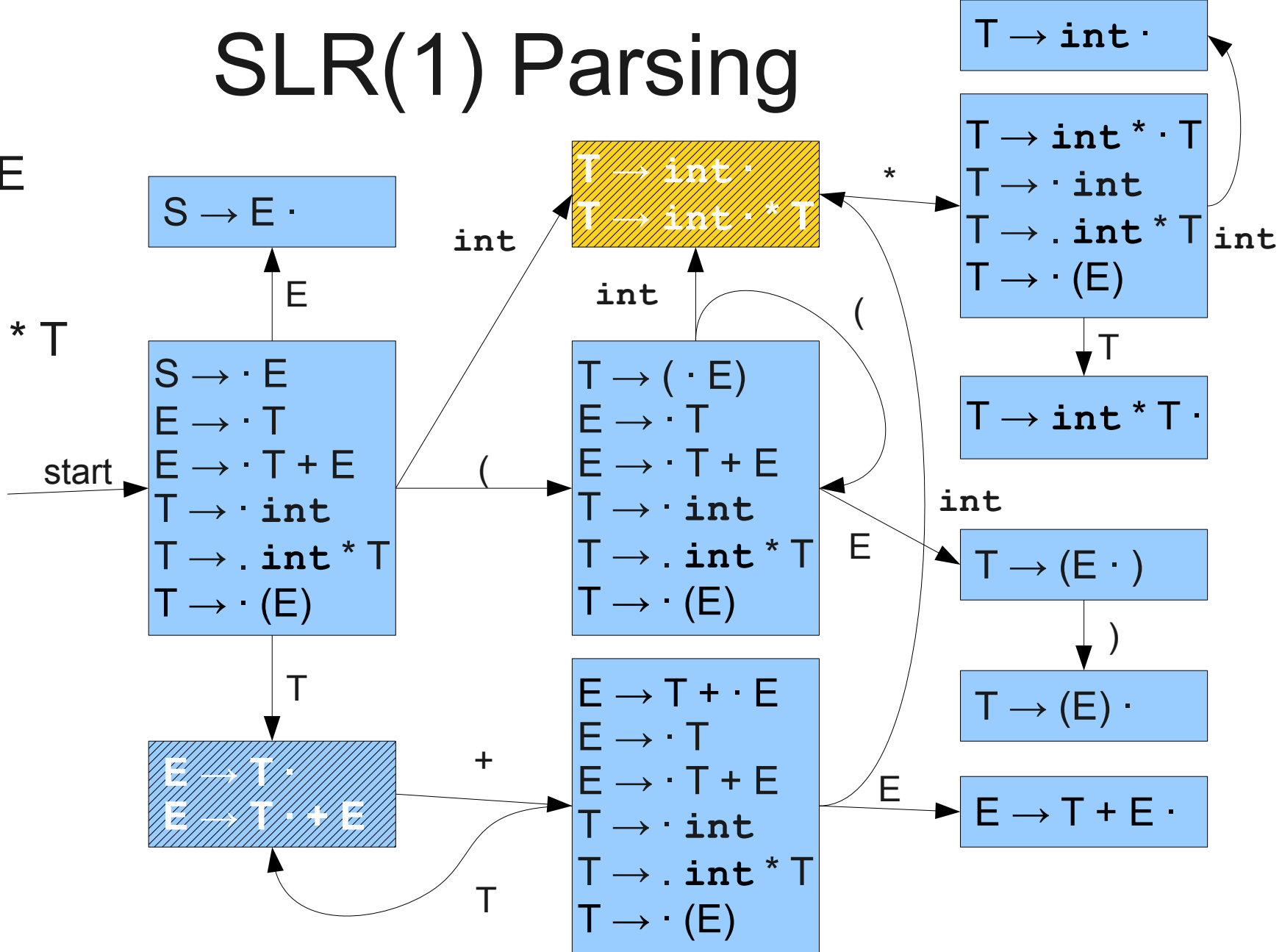


T + int

* int \$

SLR(1) Parsing

$S \rightarrow E$
 $E \rightarrow T$
 $E \rightarrow T + E$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$
 $T \rightarrow \text{int} * T$

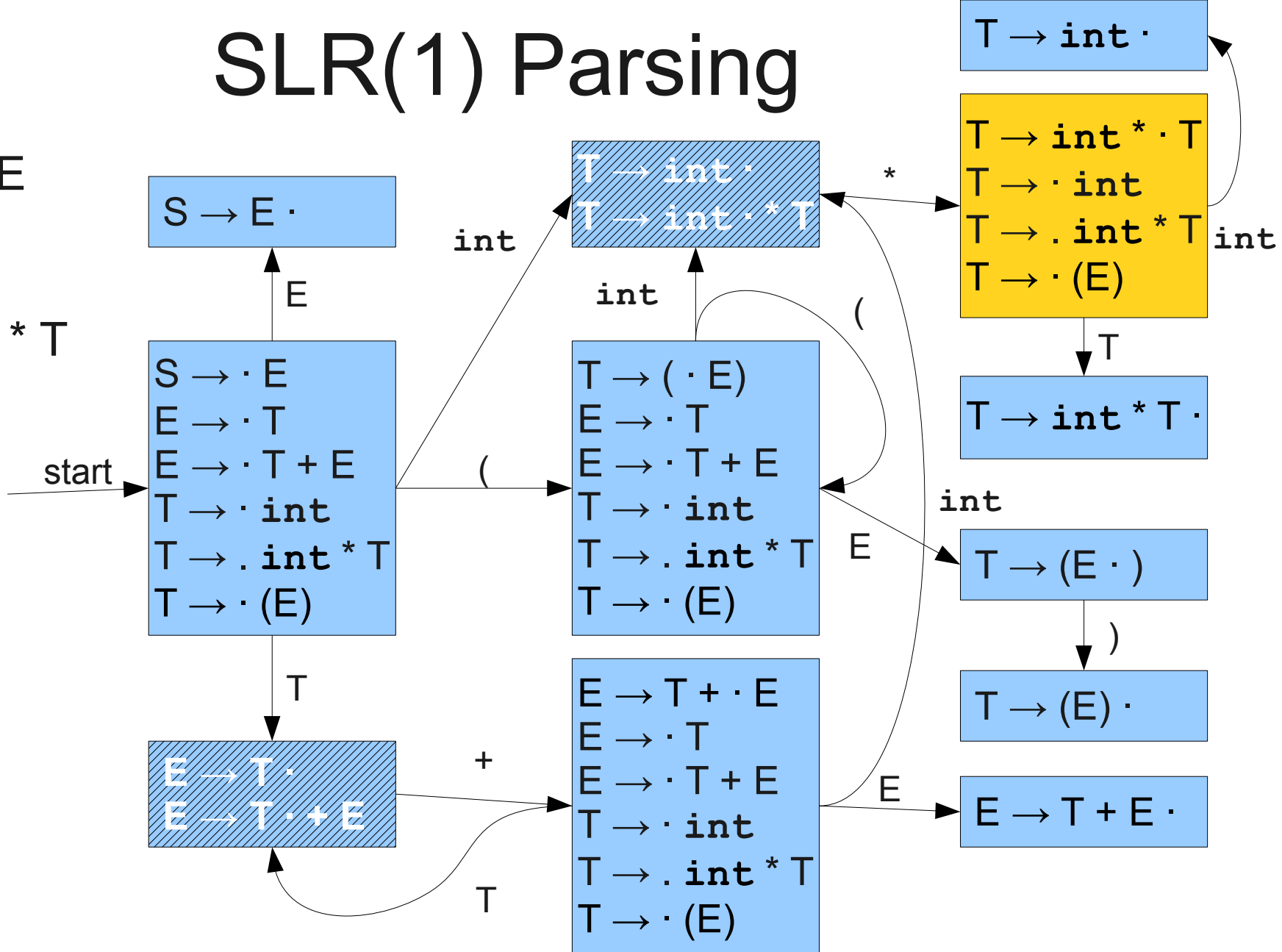


T	+	int	*
---	---	-----	---

int	\$
-----	----

SLR(1) Parsing

$S \rightarrow E$
 $E \rightarrow T$
 $E \rightarrow T + E$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$
 $T \rightarrow \text{int} * T$

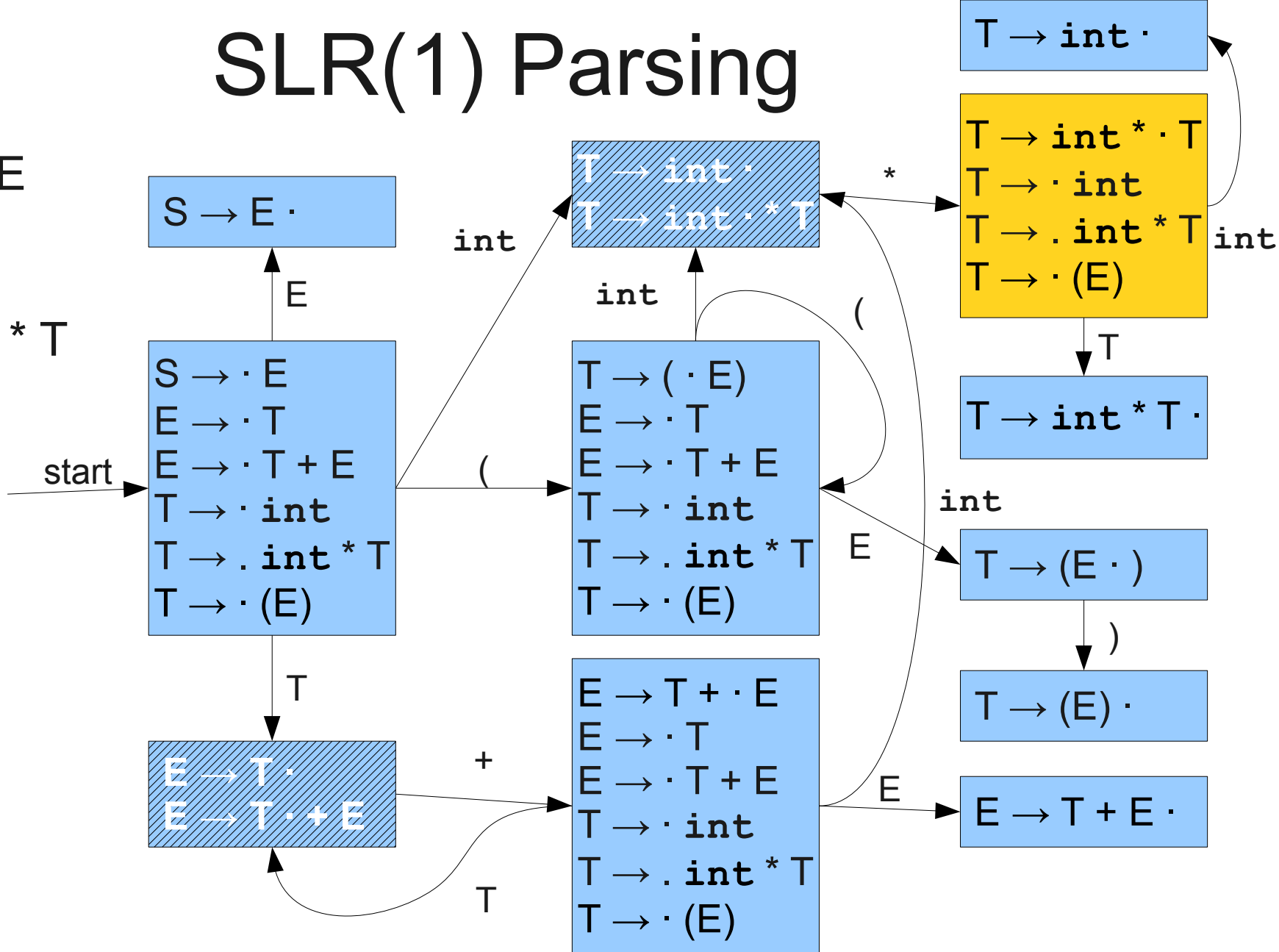


T	+	int	*
---	---	-----	---

int	\$
-----	----

SLR(1) Parsing

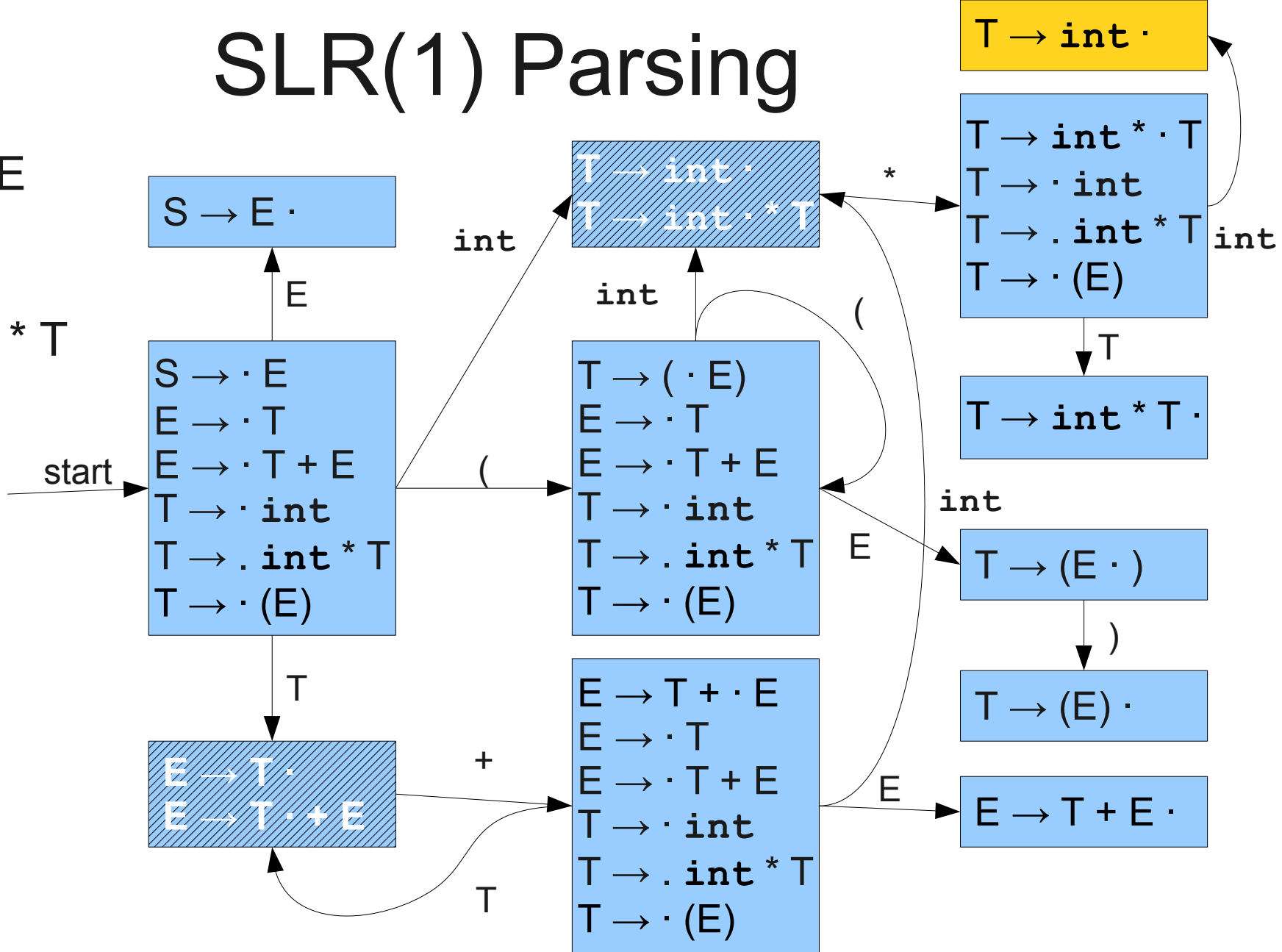
$S \rightarrow E$
 $E \rightarrow T$
 $E \rightarrow T + E$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$
 $T \rightarrow \text{int} * T$



T
+
int
*
int
|
\$

SLR(1) Parsing

$S \rightarrow E$
 $E \rightarrow T$
 $E \rightarrow T + E$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$
 $T \rightarrow \text{int} * T$

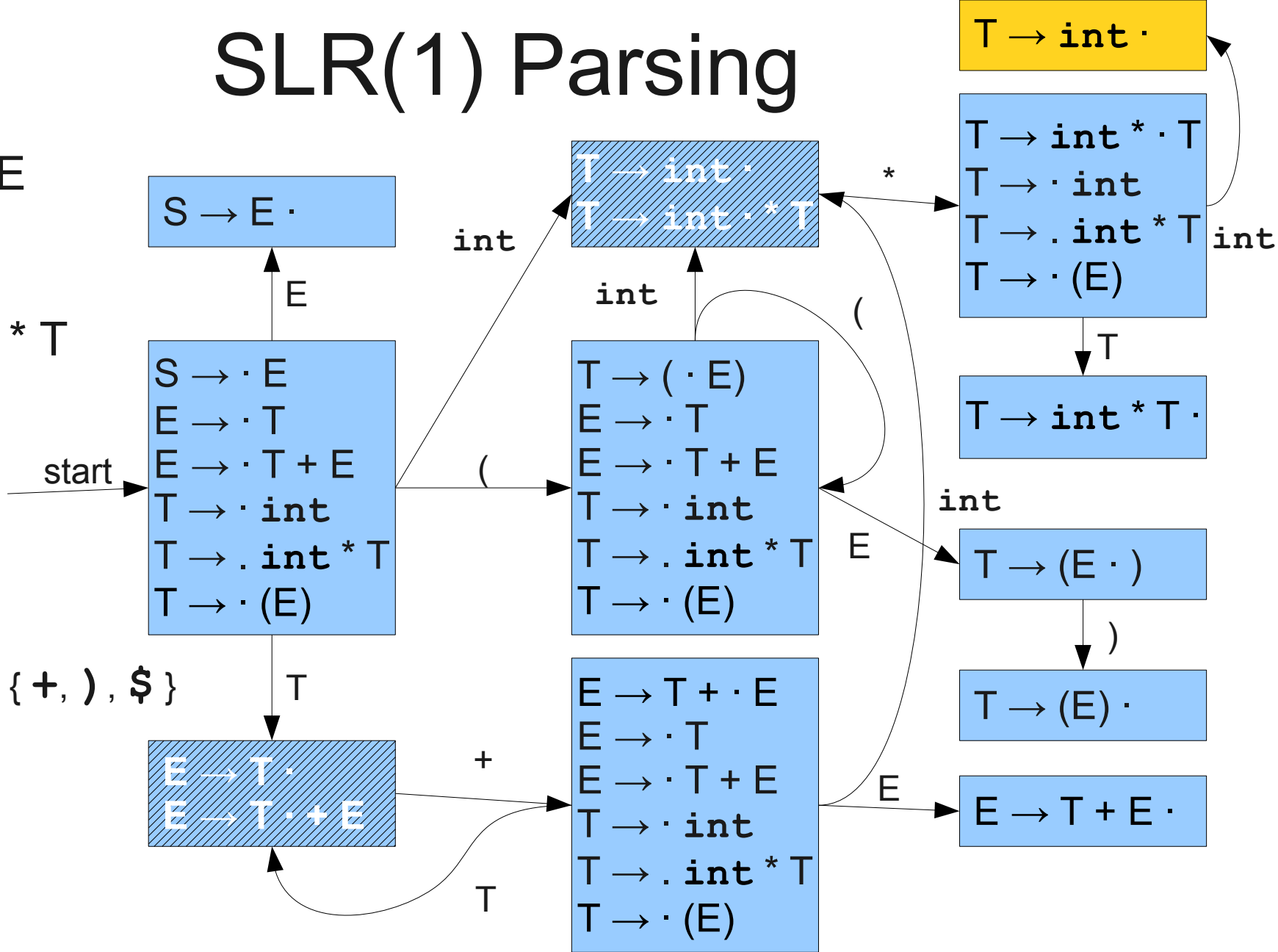


T
+
int
*
int
|
\$

SLR(1) Parsing

$S \rightarrow E$
 $E \rightarrow T$
 $E \rightarrow T + E$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$
 $T \rightarrow \text{int} * T$

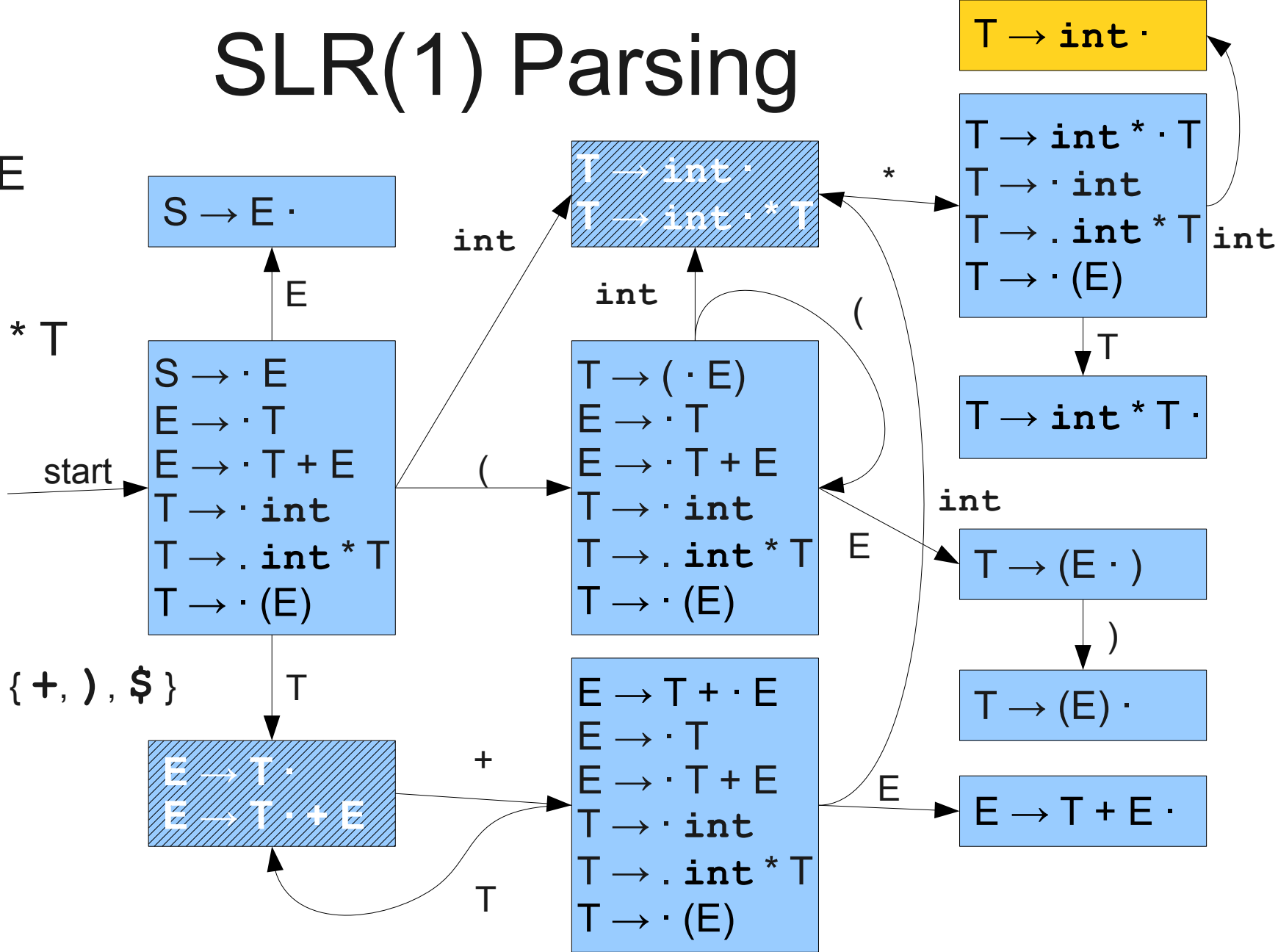
$\text{FOLLOW}(T) = \{ +,), \$ \}$



SLR(1) Parsing

$S \rightarrow E$
 $E \rightarrow T$
 $E \rightarrow T + E$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$
 $T \rightarrow \text{int} * T$

$\text{FOLLOW}(T) = \{ +,), \$ \}$

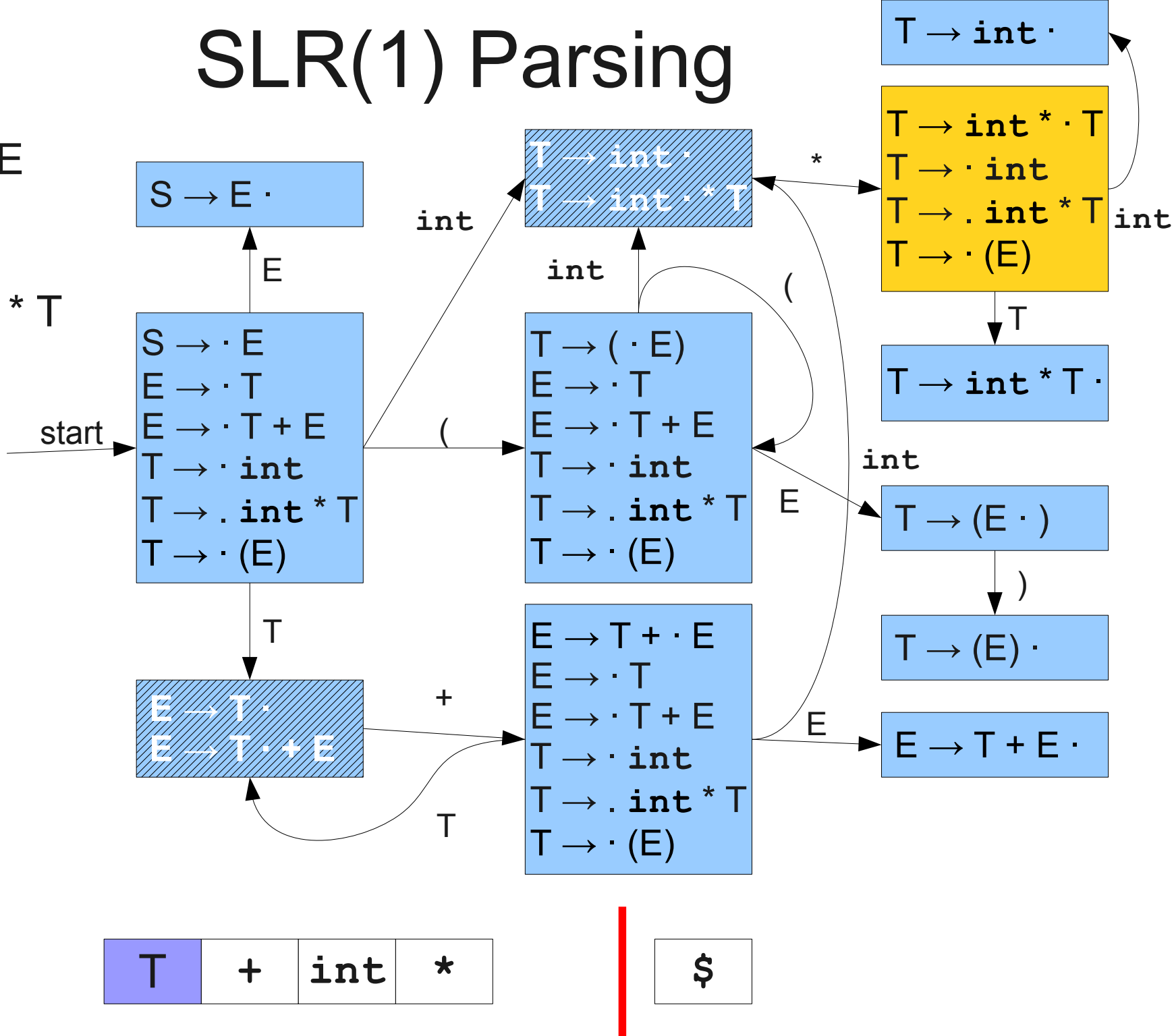


T + int * int

\$

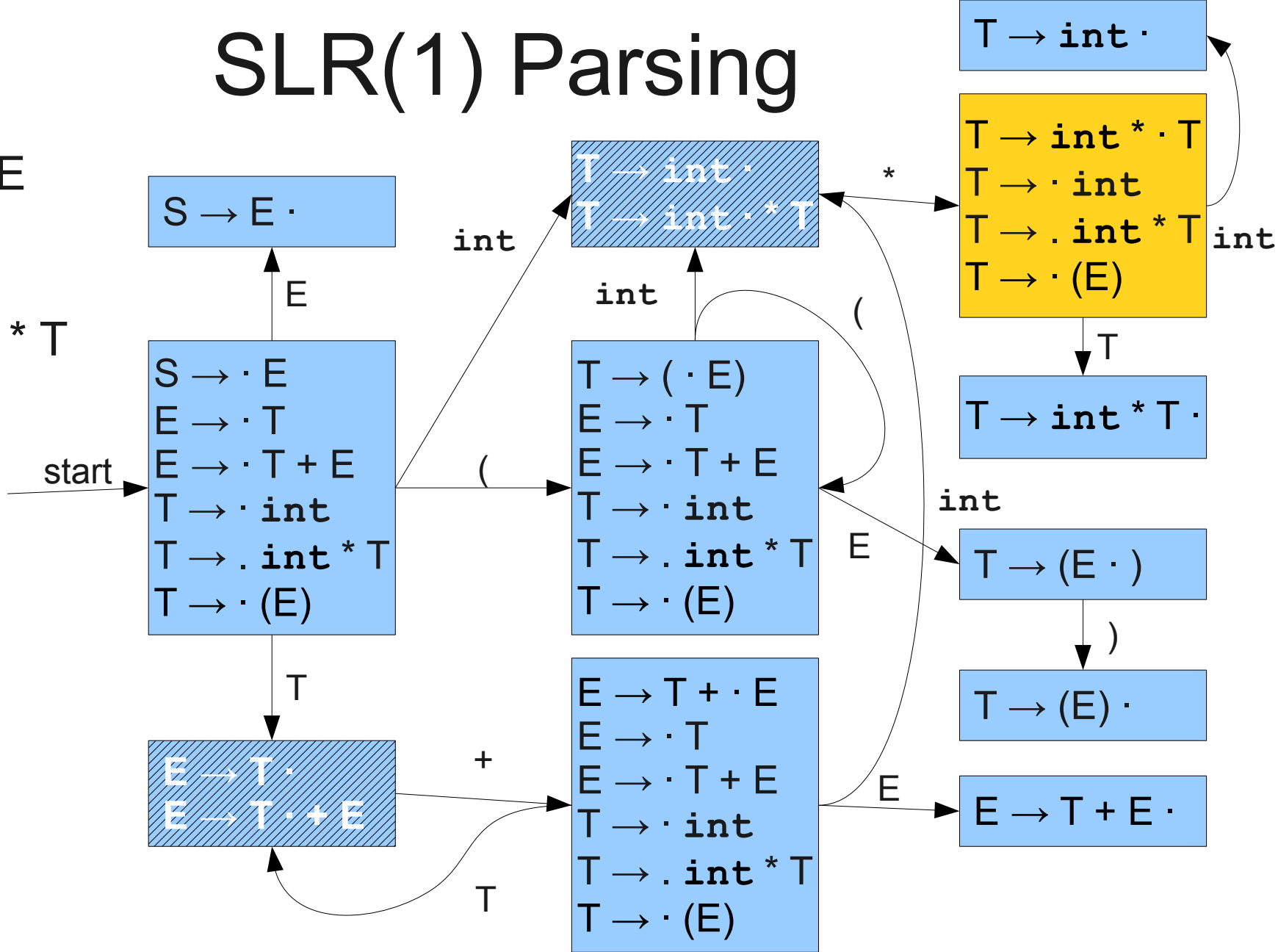
SLR(1) Parsing

$S \rightarrow E$
 $E \rightarrow T$
 $E \rightarrow T + E$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$
 $T \rightarrow \text{int} * T$



SLR(1) Parsing

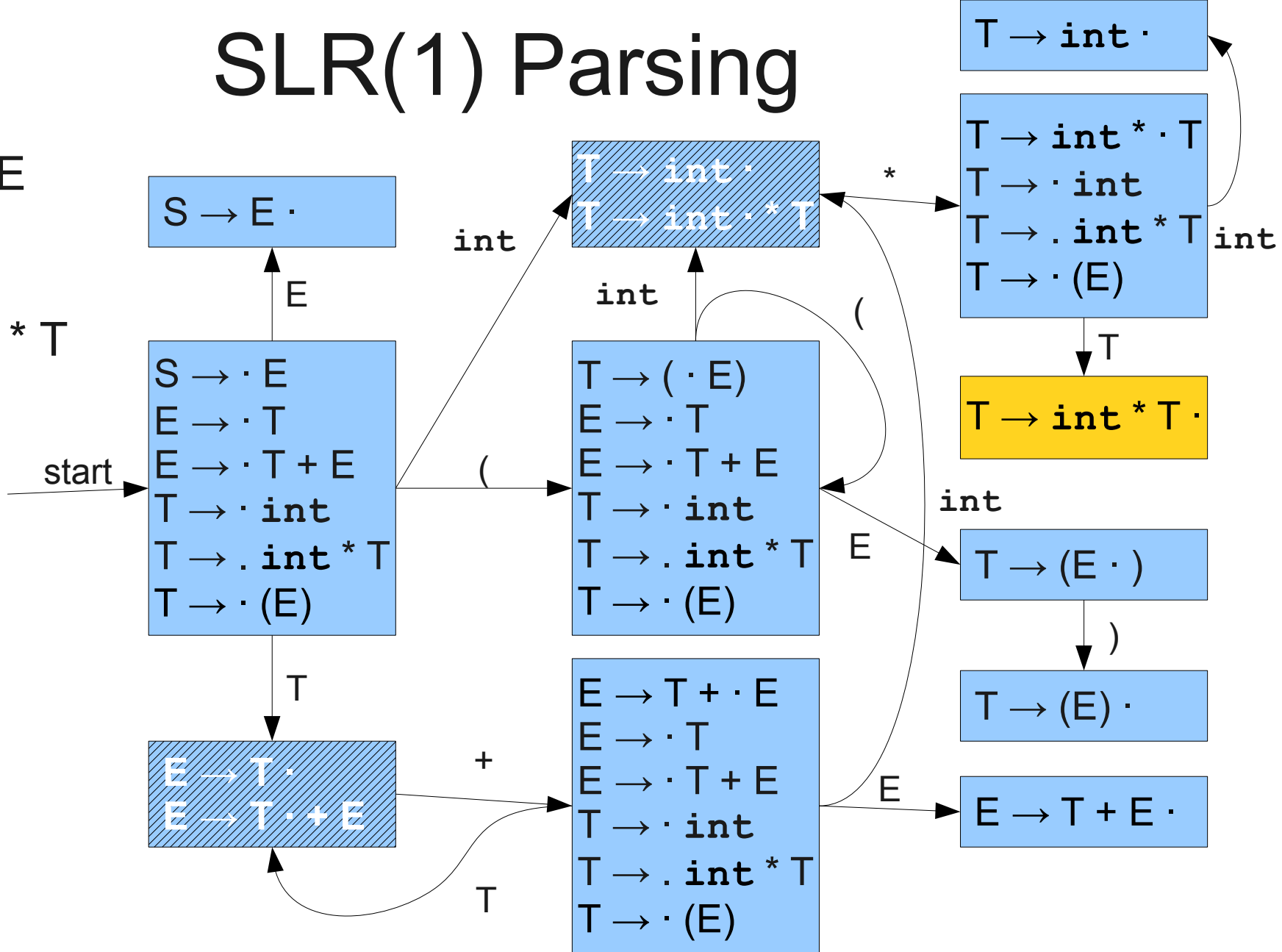
$S \rightarrow E$
 $E \rightarrow T$
 $E \rightarrow T + E$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$
 $T \rightarrow \text{int} * T$



T
+
int
*
T
|
\$

SLR(1) Parsing

$S \rightarrow E$
 $E \rightarrow T$
 $E \rightarrow T + E$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$
 $T \rightarrow \text{int} * T$

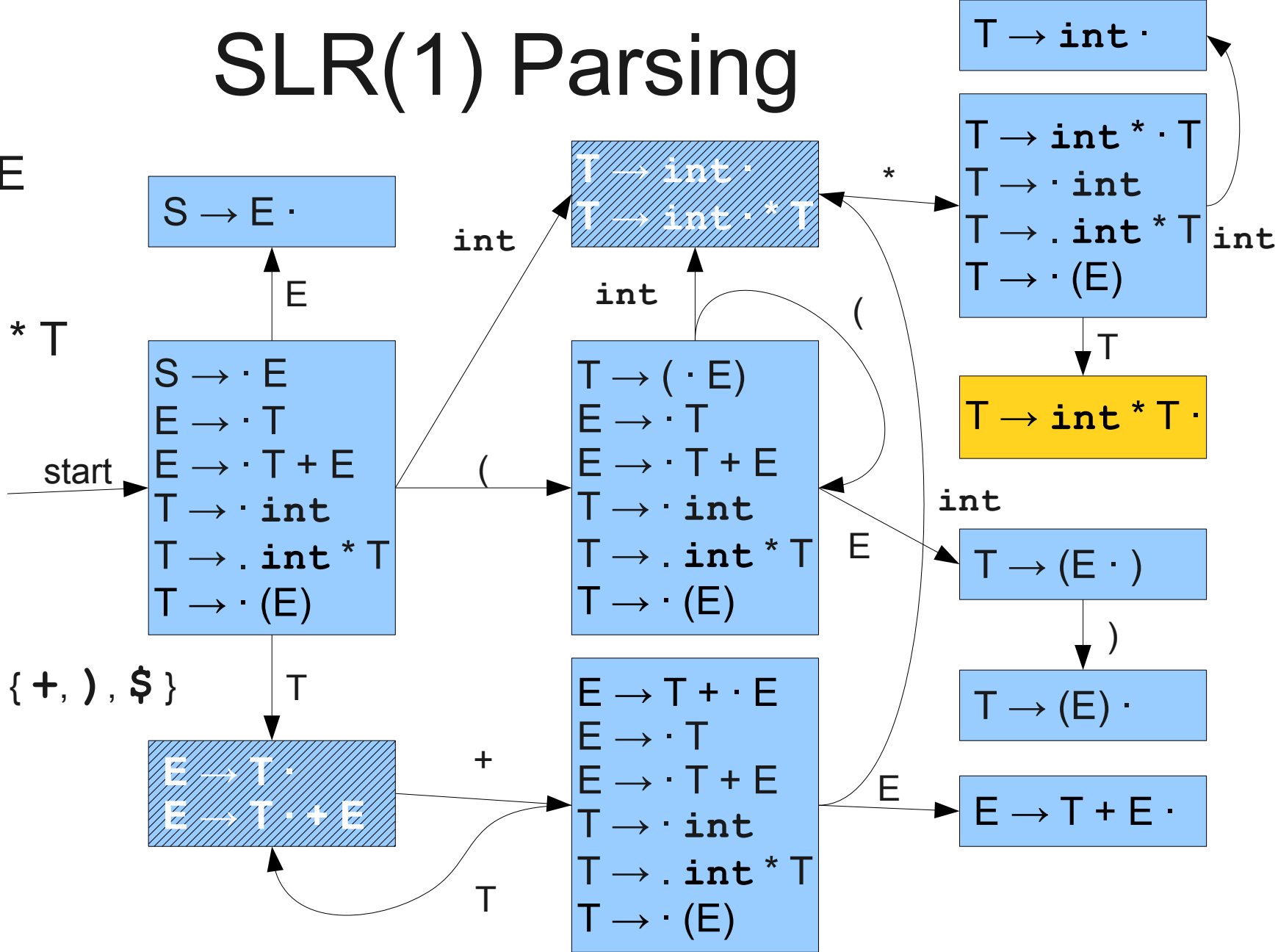


T
+
int
*
T
\$

SLR(1) Parsing

$S \rightarrow E$
 $E \rightarrow T$
 $E \rightarrow T + E$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$
 $T \rightarrow \text{int} * T$

$\text{FOLLOW}(T) = \{ +,), \$ \}$

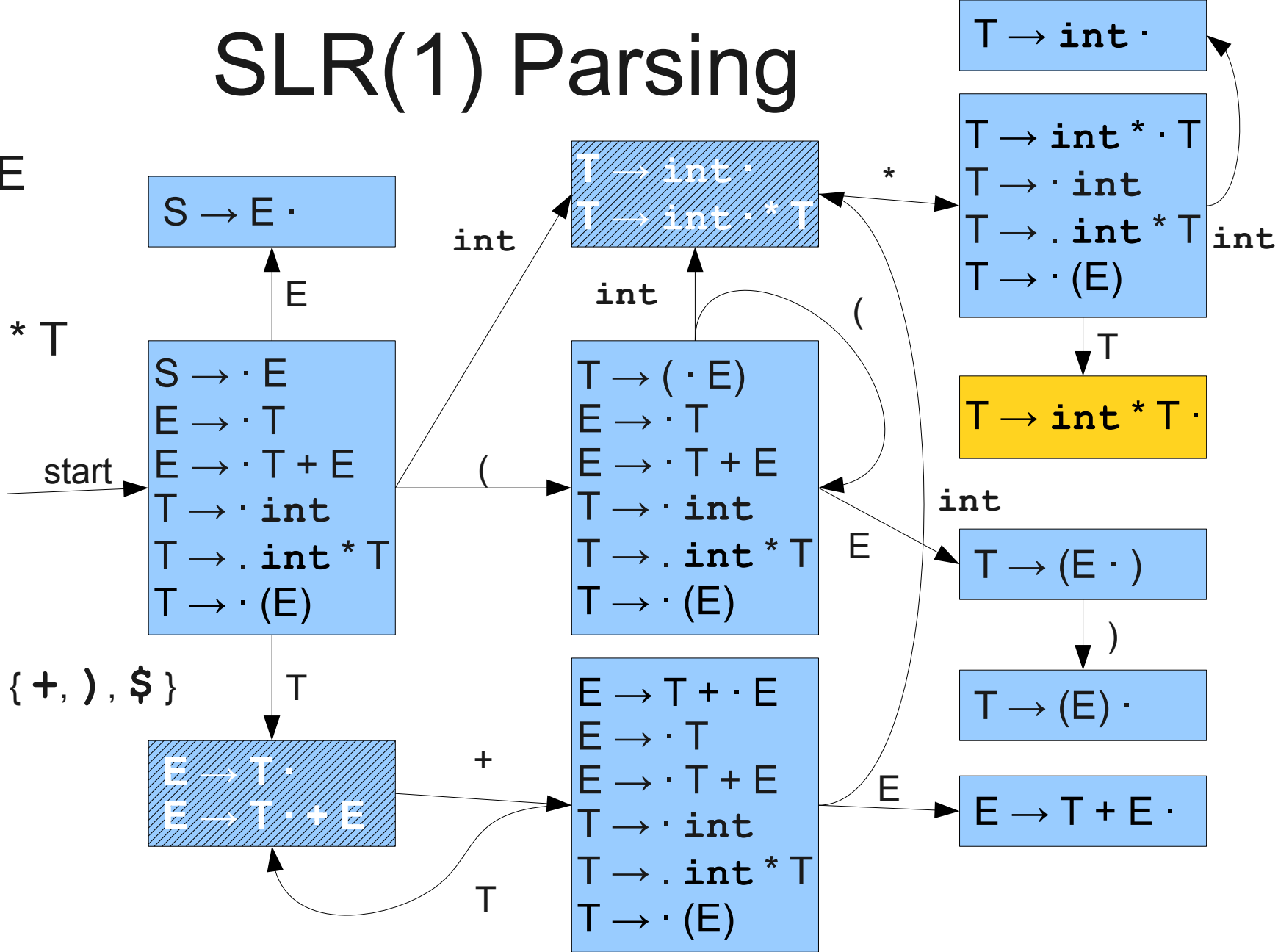


T
+
int
*
T
|
\$

SLR(1) Parsing

$S \rightarrow E$
 $E \rightarrow T$
 $E \rightarrow T + E$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$
 $T \rightarrow \text{int} * T$

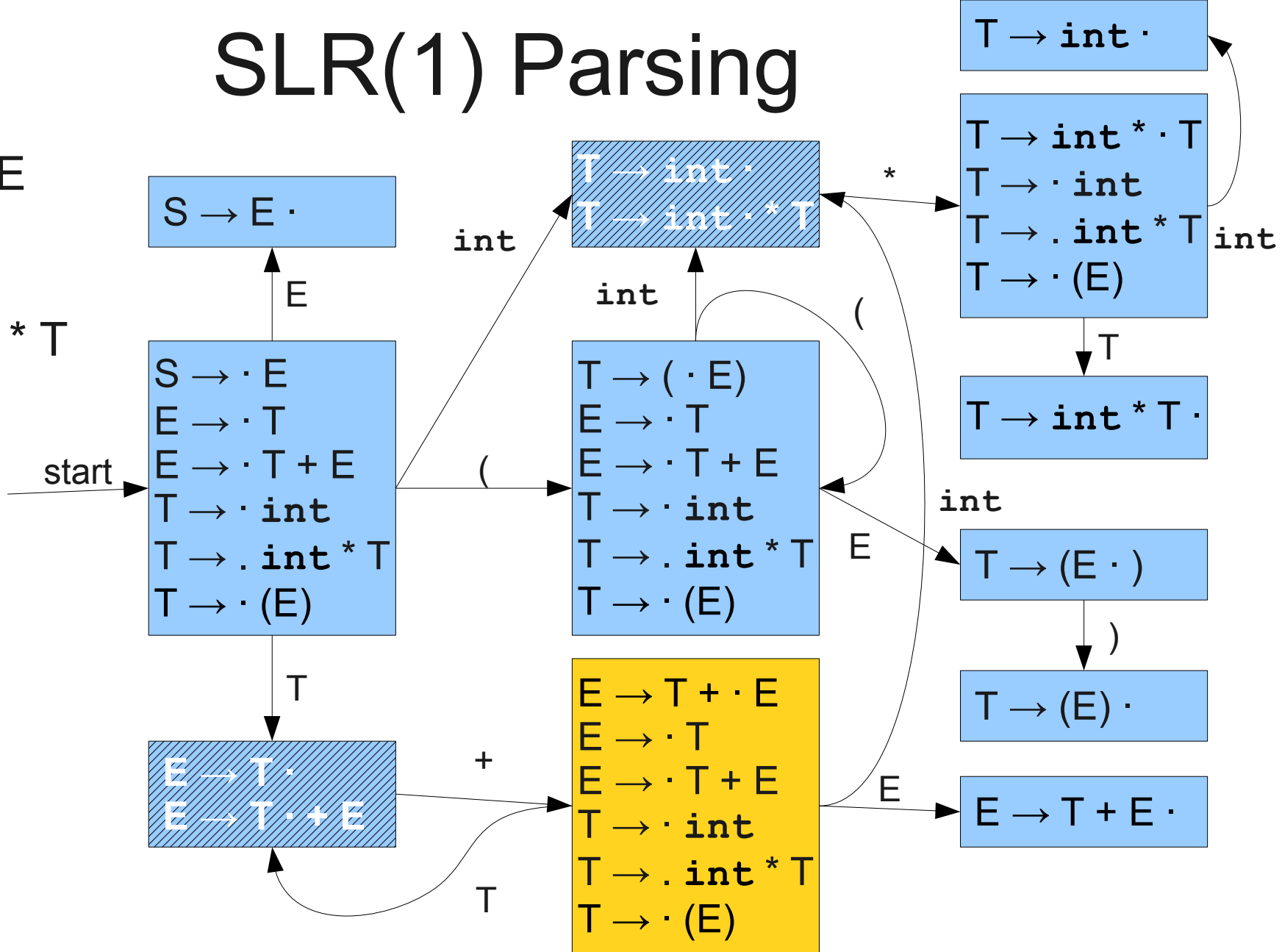
$\text{FOLLOW}(T) = \{ +,), \$ \}$



T
+
int
*
T
|
\$

SLR(1) Parsing

$S \rightarrow E$
 $E \rightarrow T$
 $E \rightarrow T + E$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$
 $T \rightarrow \text{int} * T$



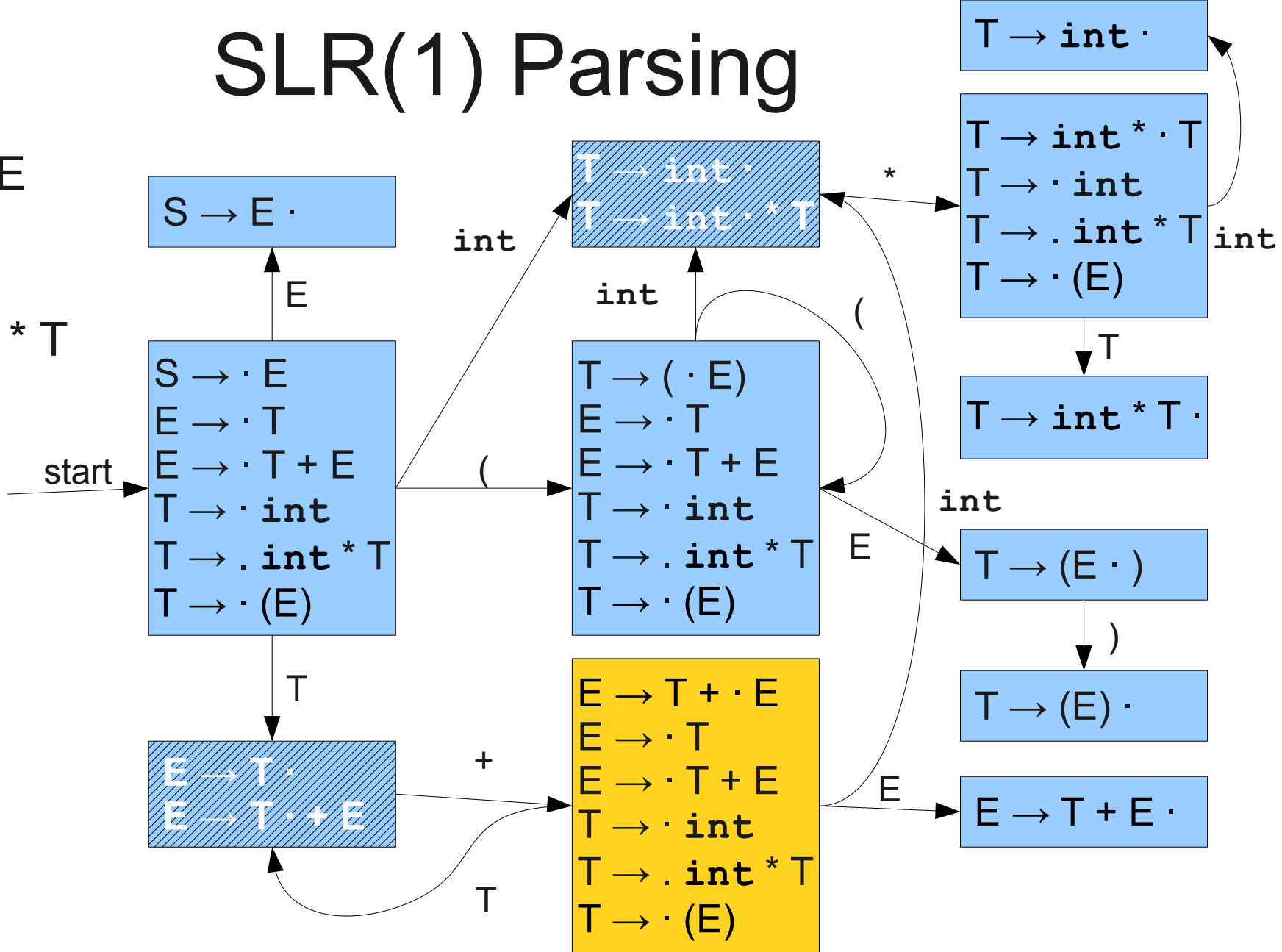
T +



\$

SLR(1) Parsing

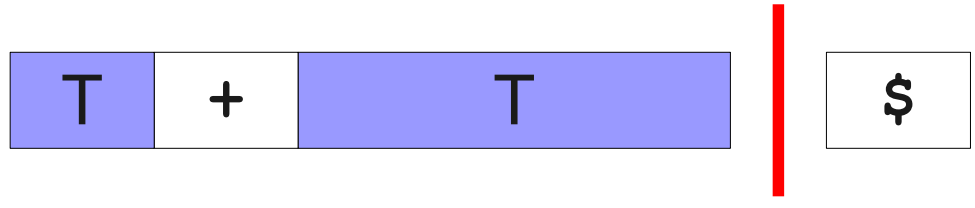
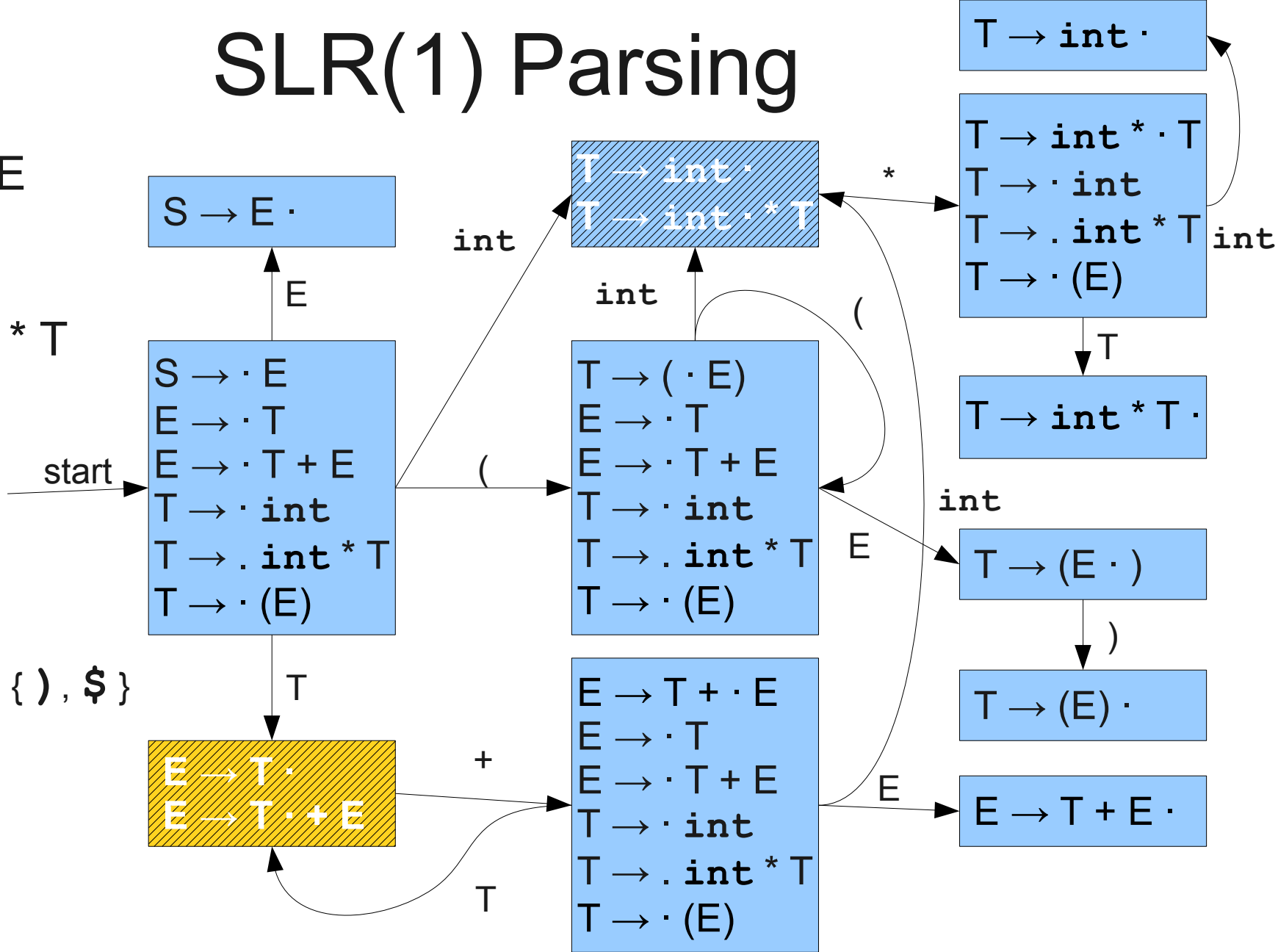
$S \rightarrow E$
 $E \rightarrow T$
 $E \rightarrow T + E$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$
 $T \rightarrow \text{int} * T$



SLR(1) Parsing

$S \rightarrow E$
 $E \rightarrow T$
 $E \rightarrow T + E$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$
 $T \rightarrow \text{int} * T$

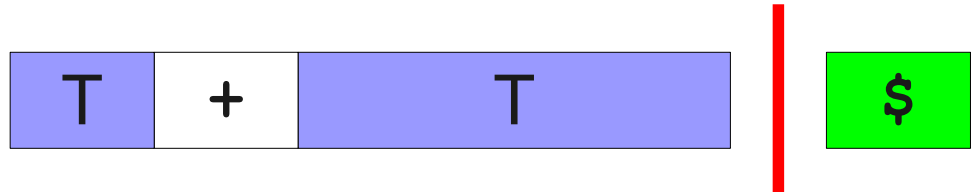
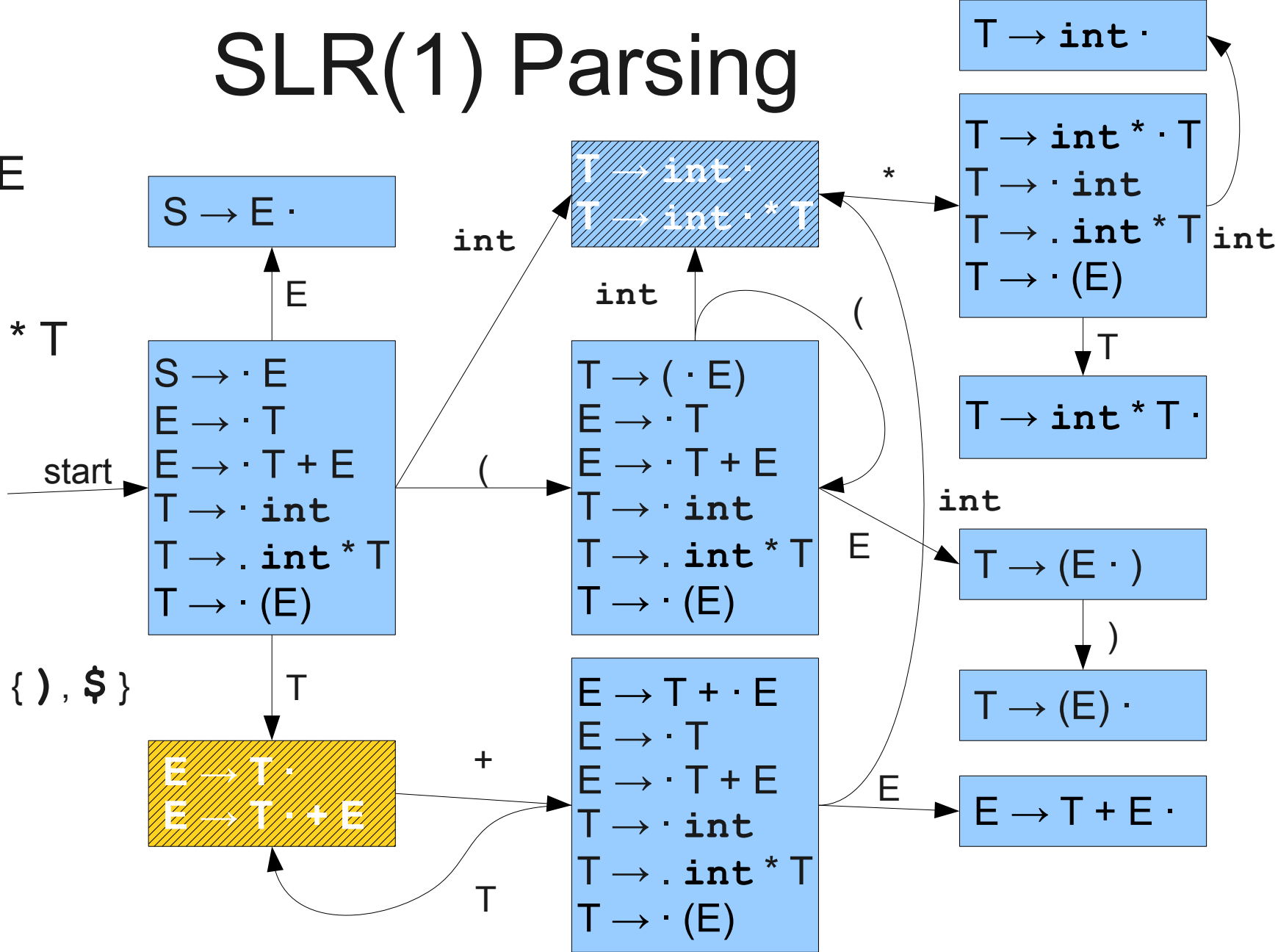
$\text{FOLLOW}(E) = \{), \$ \}$



SLR(1) Parsing

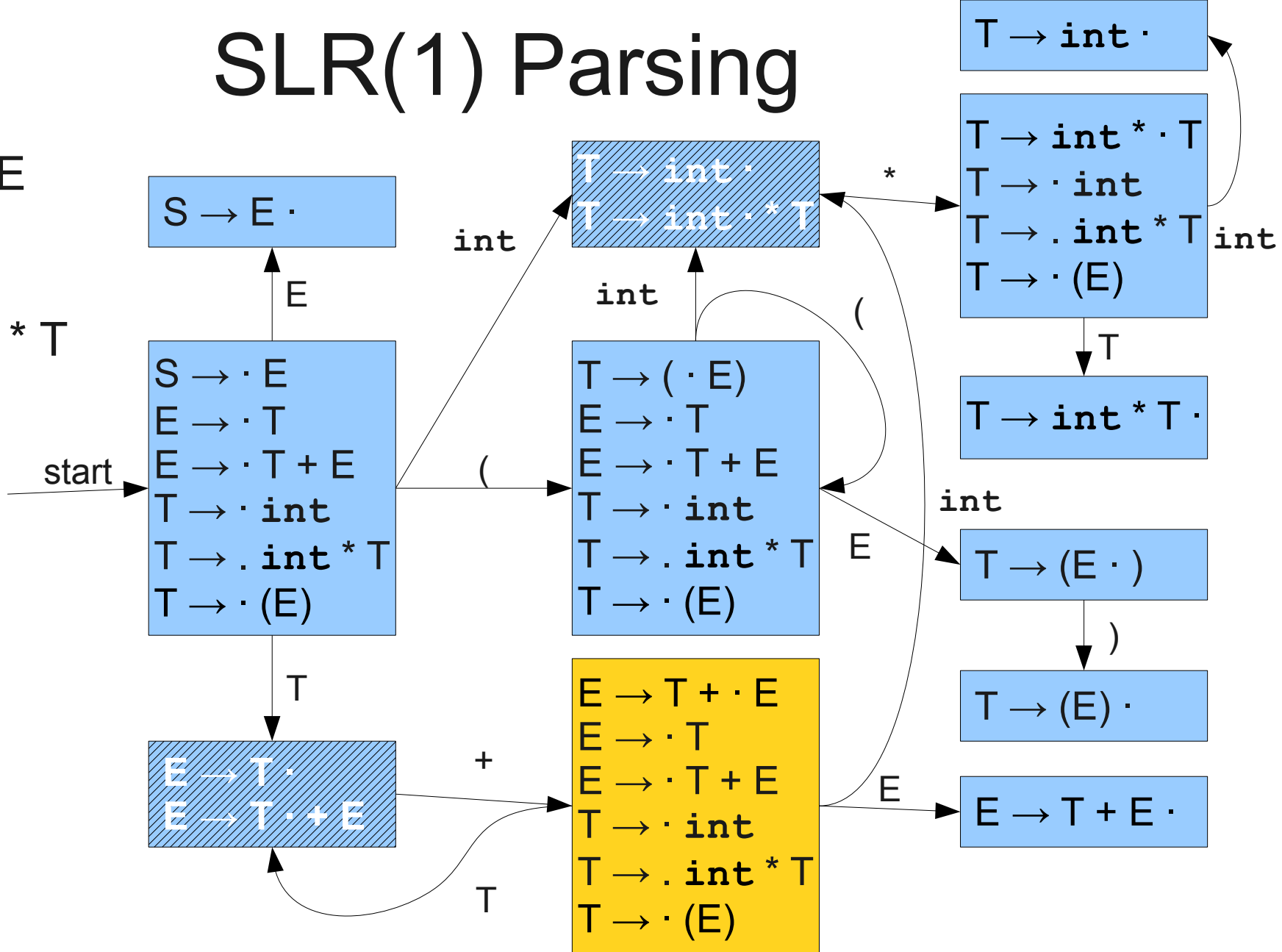
$S \rightarrow E$
 $E \rightarrow T$
 $E \rightarrow T + E$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$
 $T \rightarrow \text{int} * T$

$\text{FOLLOW}(E) = \{), \$ \}$



SLR(1) Parsing

$S \rightarrow E$
 $E \rightarrow T$
 $E \rightarrow T + E$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$
 $T \rightarrow \text{int} * T$



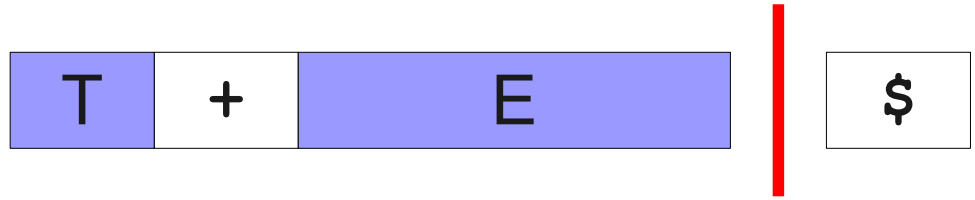
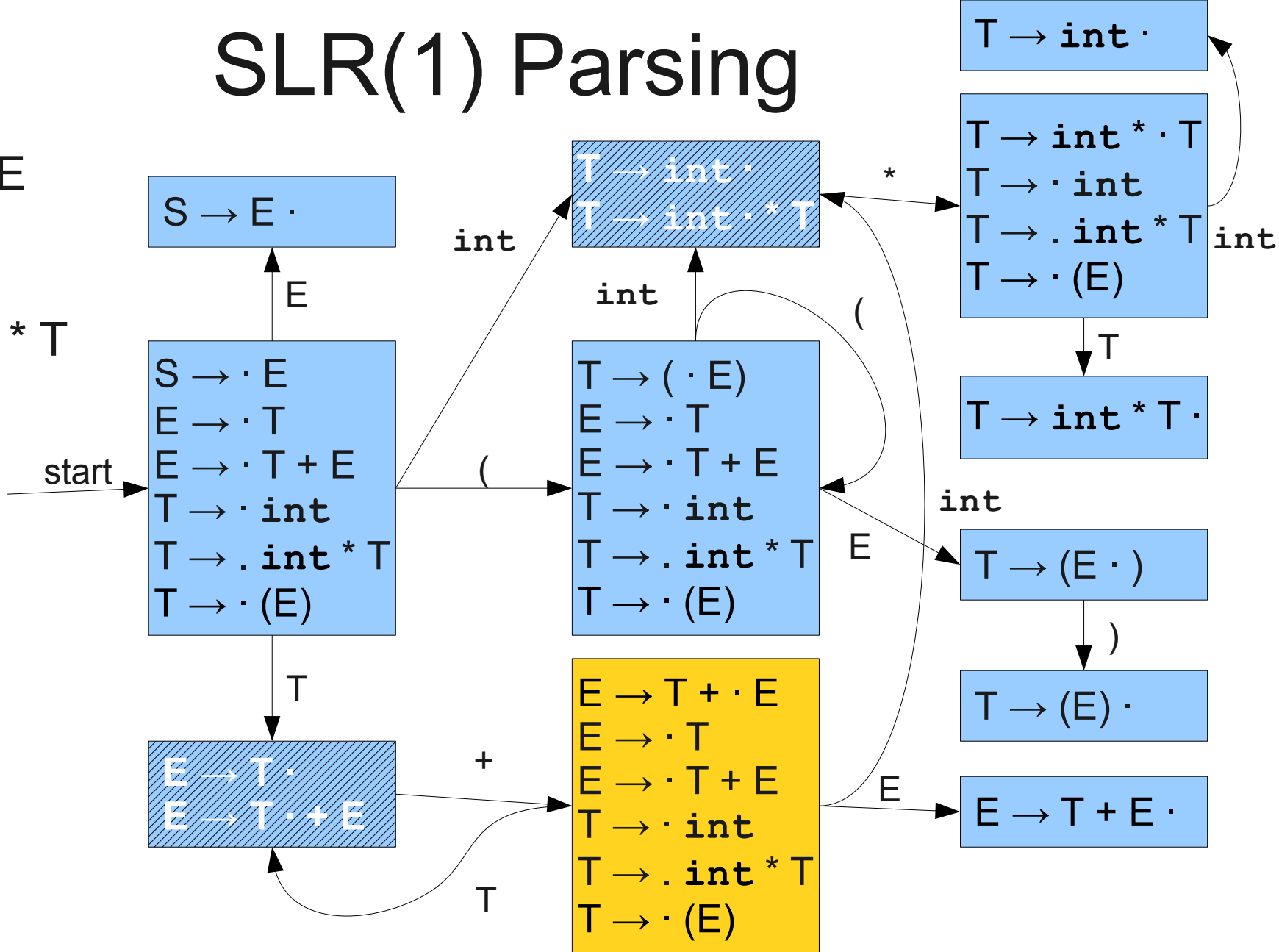
T +



\$

SLR(1) Parsing

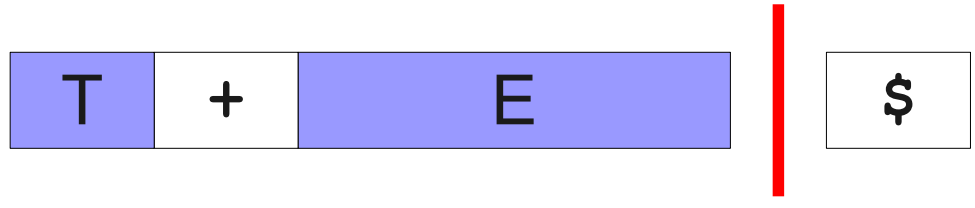
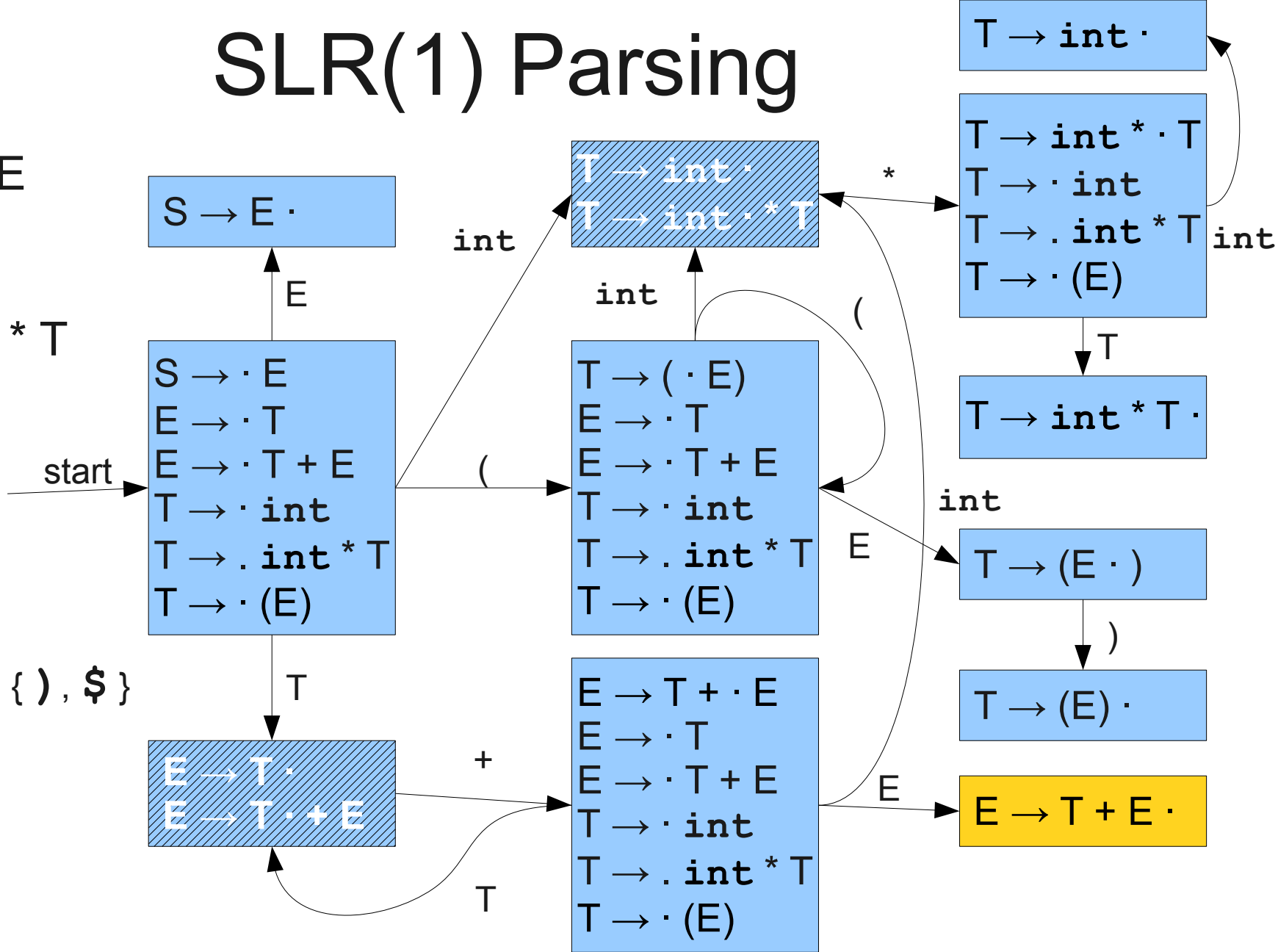
$S \rightarrow E$
 $E \rightarrow T$
 $E \rightarrow T + E$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$
 $T \rightarrow \text{int} * T$



SLR(1) Parsing

$S \rightarrow E$
 $E \rightarrow T$
 $E \rightarrow T + E$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$
 $T \rightarrow \text{int} * T$

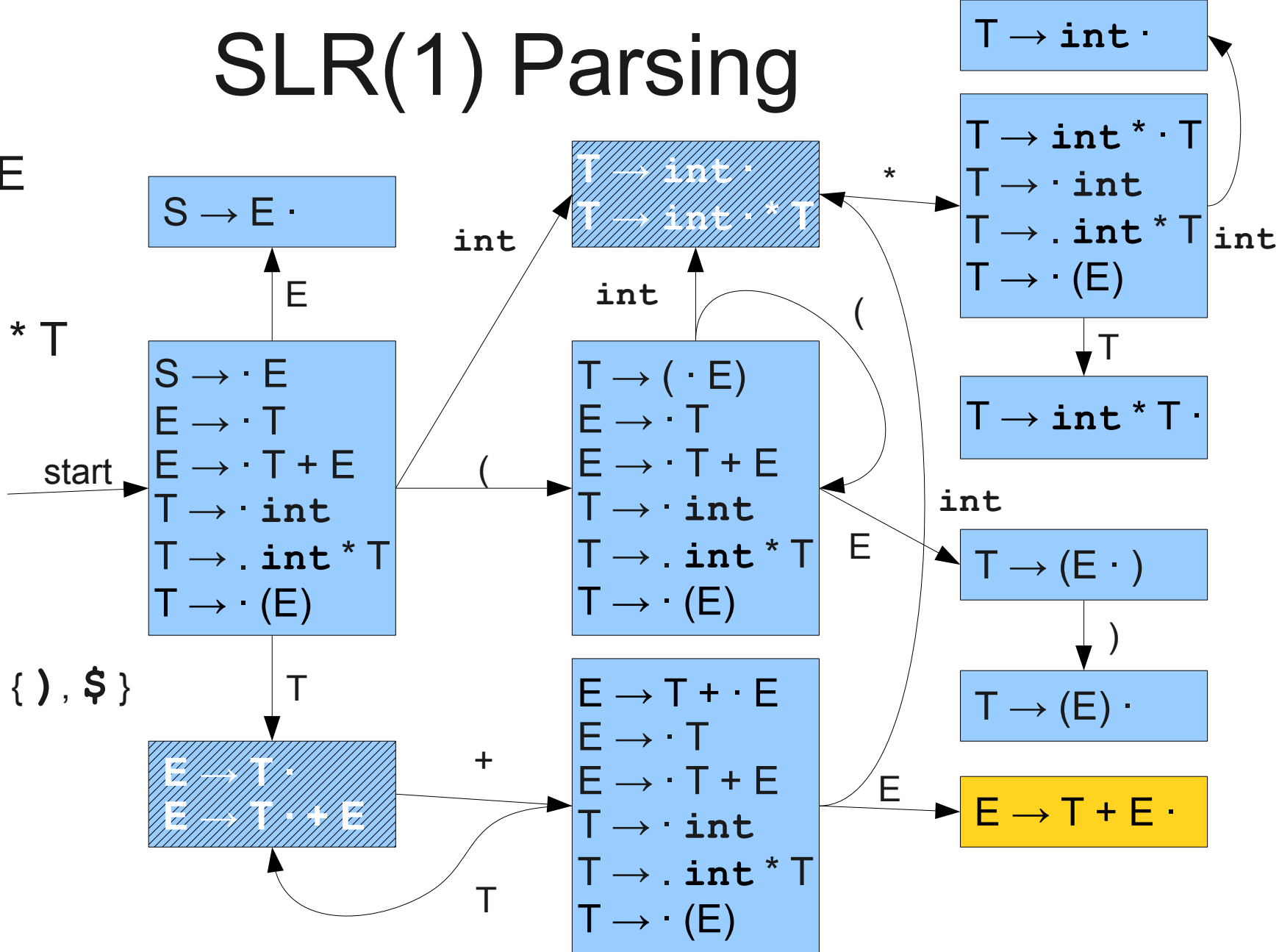
$\text{FOLLOW}(E) = \{), \$ \}$



SLR(1) Parsing

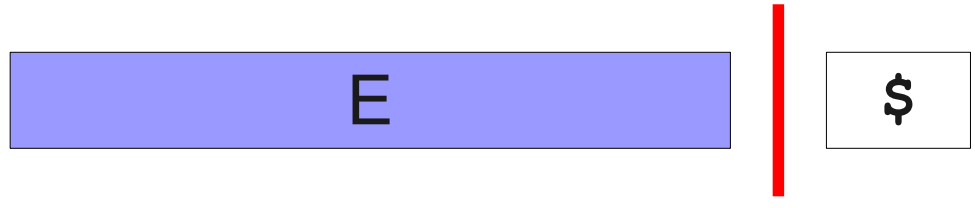
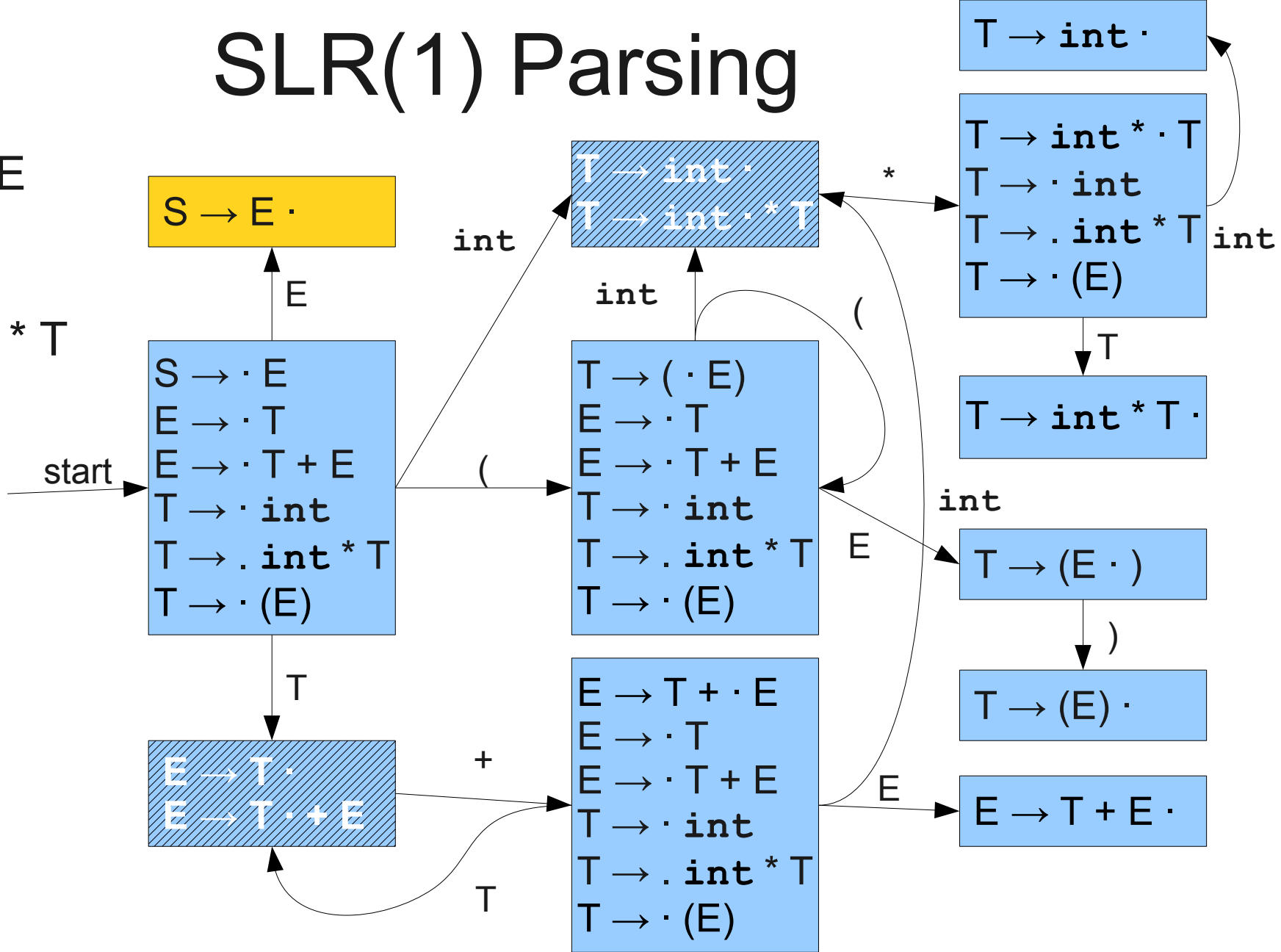
$S \rightarrow E$
 $E \rightarrow T$
 $E \rightarrow T + E$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$
 $T \rightarrow \text{int} * T$

$\text{FOLLOW}(E) = \{), \$ \}$



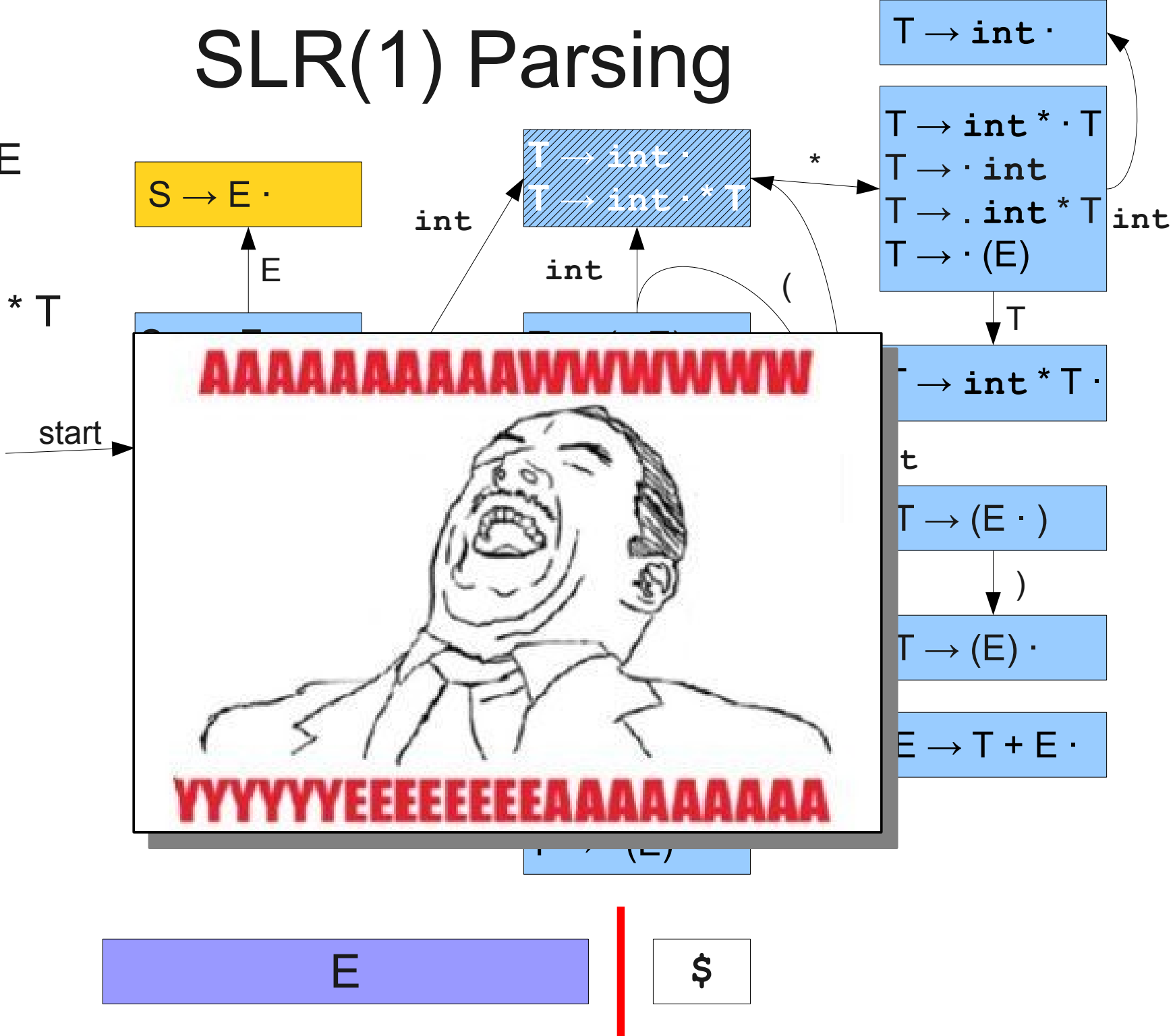
SLR(1) Parsing

$S \rightarrow E$
 $E \rightarrow T$
 $E \rightarrow T + E$
 $T \rightarrow \text{int}$
 $T \rightarrow (E)$
 $T \rightarrow \text{int} * T$



SLR(1) Parsing

- $S \rightarrow E$
- $E \rightarrow T$
- $E \rightarrow T + E$
- $T \rightarrow \text{int}$
- $T \rightarrow (E)$
- $T \rightarrow \text{int} * T$



Analysis of SLR(1)

- Exploits lookahead in a small space.
 - Small automaton – same automaton as LR(0).
 - Works on many more grammars than LR(0)
- Too weak for most grammars: lose context from not having extra states.
- Can do a lot better than SLR(1); we'll see how next time!

Next Time

- Very intelligent lookaheads: LALR(1)
- The **bison** Tool

This is the **400th** slide!

Congrats on making it so far!