Compiler Components & Generators

Traditional Parsing Algorithms

Guido Wachsmuth

Recap: Traditional Parsing Algorithms
lessons learned

How can we parse context-free languages effectively?

- predictive parsing

Which grammar classes are supported by these algorithms?

- LL(k) grammars, LL(k) languages

How can we generate compiler tools from that?

- implement automaton
- generate parse tables

Overview

today's lecture

Overview
today's lecture

efficient parsing algorithms

- LR parsing
- LR parse table generation
- SLR & LALR parse tables

Overview
today's lecture

efficient parsing algorithms

- LR parsing
- LR parse table generation
- SLR & LALR parse tables

I

LR parsing

LR parsing
idea

problems with LL parsing

- predicting right rule
- left recursion

LR parsing

- see whole left-hand side of a rule
- look ahead
- shift or reduce

LR parsing

example

LR parsing
example

$

7	*	3	+	7	*	3	$

LR parsing
example

$	7

*	3	+	7	*	3	$

LR parsing

example

$	7
$	E

*	3	+	7	*	3	$
*	3	+	7	*	3	$

LR parsing
example

$	7

$	E	*

*	3	+	7	*	3	$

3	+	7	*	3	$

LR parsing

example

$	7		
$	E	*	3

*	3	+	7	*	3	$
		+	7	*	3	$

LR parsing

example

$	7

$	E	*	3

$	E	*	E

*	3	+	7	*	3	$

+	7	*	3	$

+	7	*	3	$

LR parsing

example

$	7

$	E	*	3

$	E	*	E

$	E

*	3	+	7	*	3	$

+	7	*	3	$

+	7	*	3	$

+	7	*	3	$

LR parsing

example

$	7

$	E	*	3

$	E	*	E

$	E	+

*	3	+	7	*	3	$

+	7	*	3	$

+	7	*	3	$

7	*	3	$

LR parsing

example

$	7

$	E	*	3

$	E	*	E

$	E	+	7

*	3	+	7	*	3	$

+	7	*	3	$

+	7	*	3	$

*	3	$

LR parsing

example

$	7

$	E	*	3

$	E	*	E

$	E	+	7

$	E	+	E

*	3	+	7	*	3	$

+	7	*	3	$

+	7	*	3	$

*	3	$

*	3	$

LR parsing

example

$	7			
$	E	*	3	
$	E	*	E	
$	E	+	7	
$	E	+	E	*

*	3	+	7	*	3	$
		+	7	*	3	$
		+	7	*	3	$
		*	3	$		
			3	$		

LR parsing

example

$	7

$	E	*	3

$	E	*	E

$	E	+	7

$	E	+	E	*	3

*	3	+	7	*	3	$

+	7	*	3	$

+	7	*	3	$

*	3	$

$

LR parsing

example

$	7

$	E	*	3

$	E	*	E

$	E	+	7

$	E	+	E	*	3

$	E	+	E	*	E

*	3	+	7	*	3	$

+	7	*	3	$

+	7	*	3	$

*	3	$

$

$

LR parsing

example

$	7

$	E	*	3

$	E	*	E

$	E	+	7

$	E	+	E	*	3

$	E	+	E	*	E

$	E	+	E

*	3	+	7	*	3	$

+	7	*	3	$

+	7	*	3	$

*	3	$

$

$

$

LR parsing

example

$	7

$	E	*	3

$	E	*	E

$	E	+	7

$	E	+	E	*	3

$	E	+	E	*	E

$	E	+	E

$	E

*	3	+	7	*	3	$

+	7	*	3	$

+	7	*	3	$

*	3	$

$

$

$

$

LR parsing

example

$	7

$	E	*	3

$	E	*	E

$	E	+	7

$	E	+	E	*	3

$	E	+	E	*	E

$	E	+	E

$	E	$

*	3	+	7	*	3	$

+	7	*	3	$

+	7	*	3	$

*	3	$

$

$

$

LR parsing

example

$	7

$	E	*	3

$	E	*	E

$	E	+	7

$	E	+	E	*	3

$	E	+	E	*	E

$	E	+	E

$	E	$

$	S

*	3	+	7	*	3	$

+	7	*	3	$

+	7	*	3	$

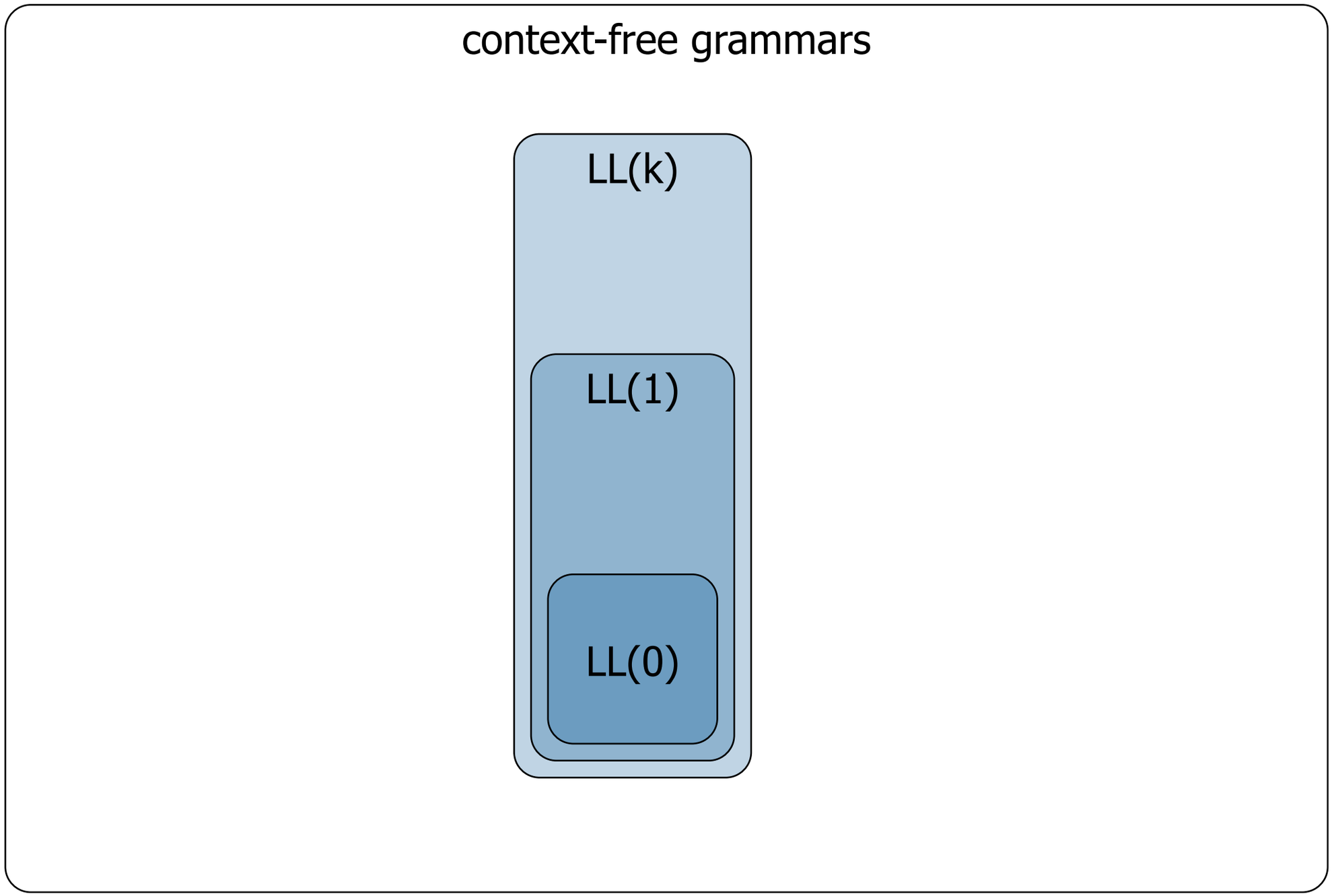
*	3	$

$

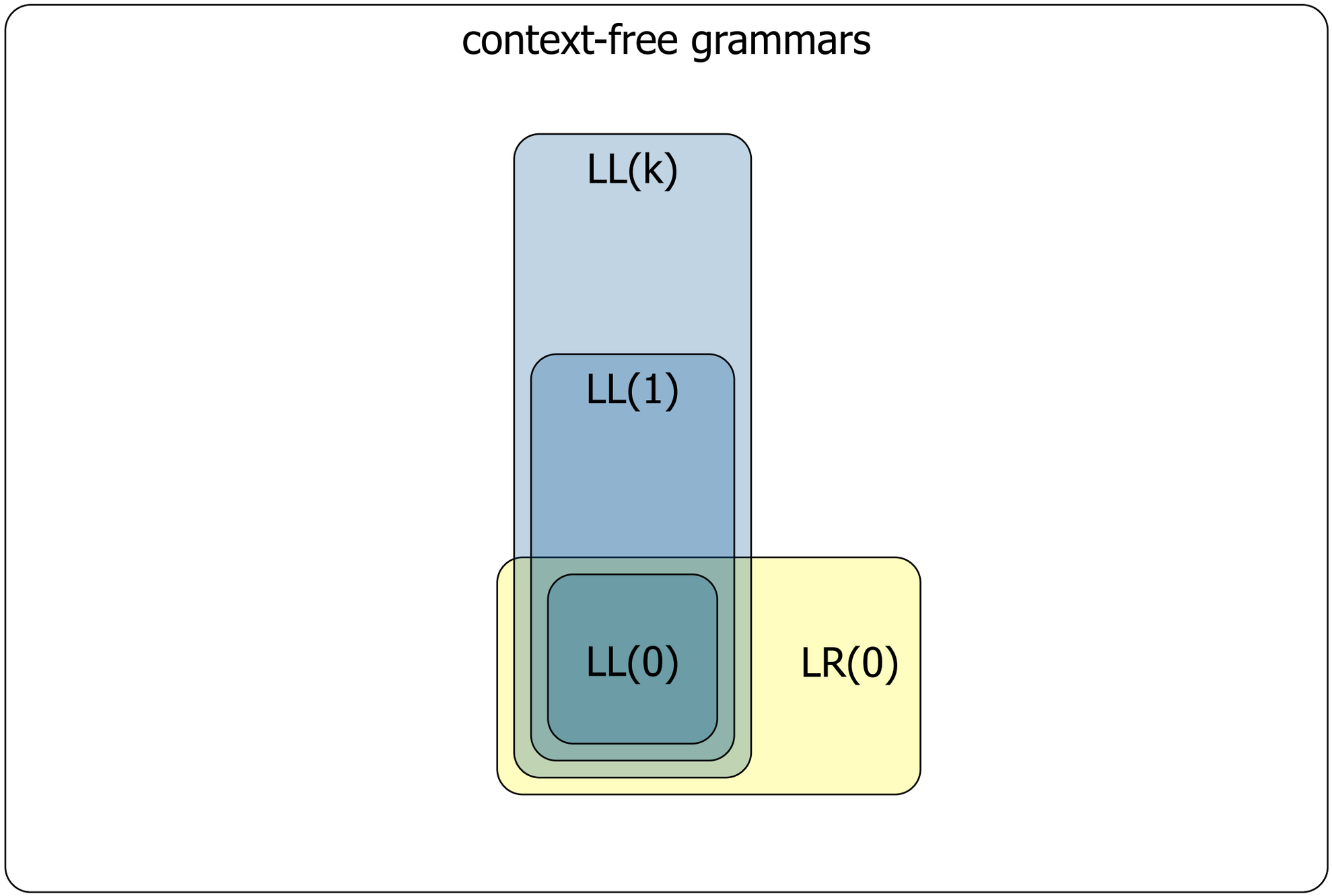
$

$

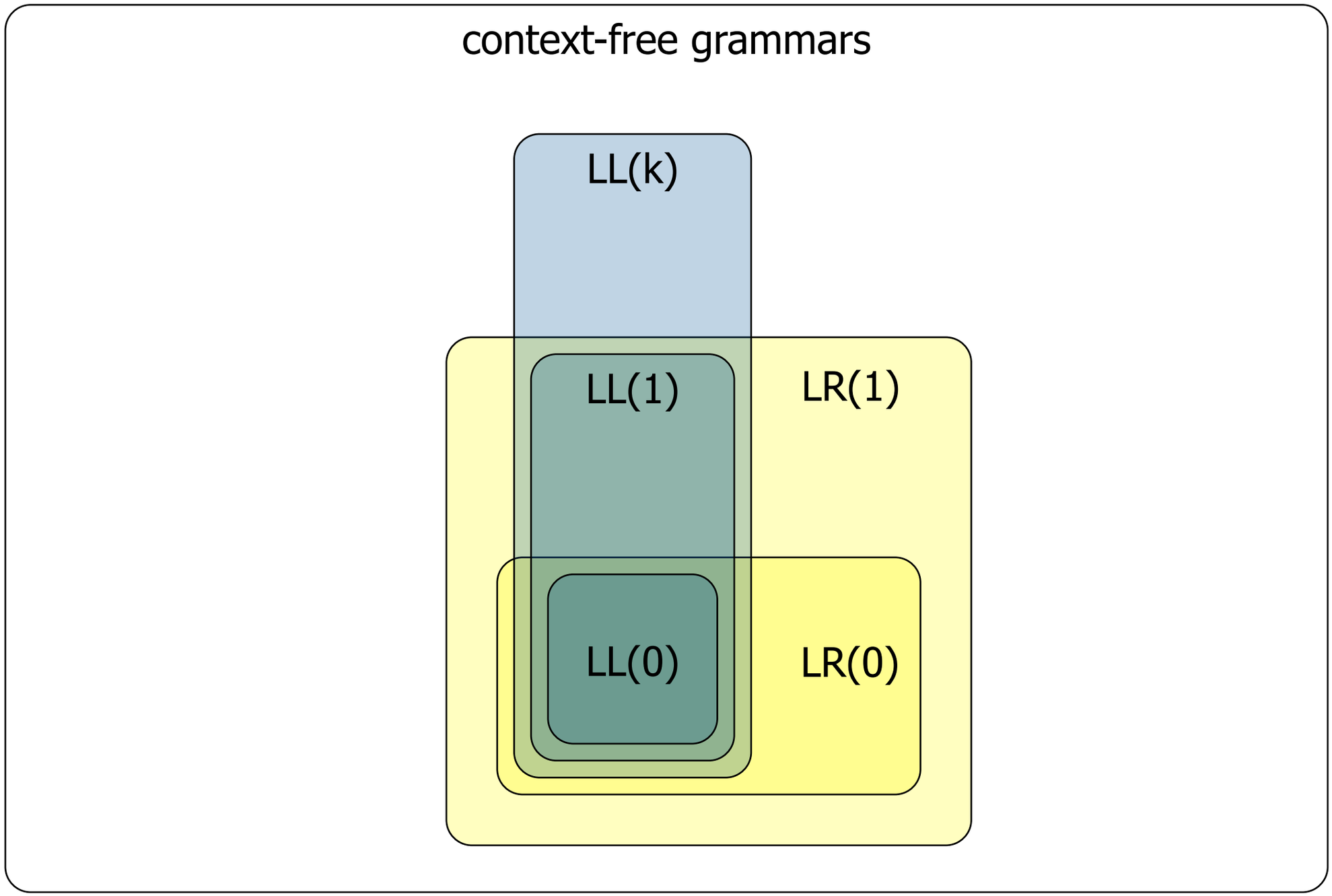
Grammar classes



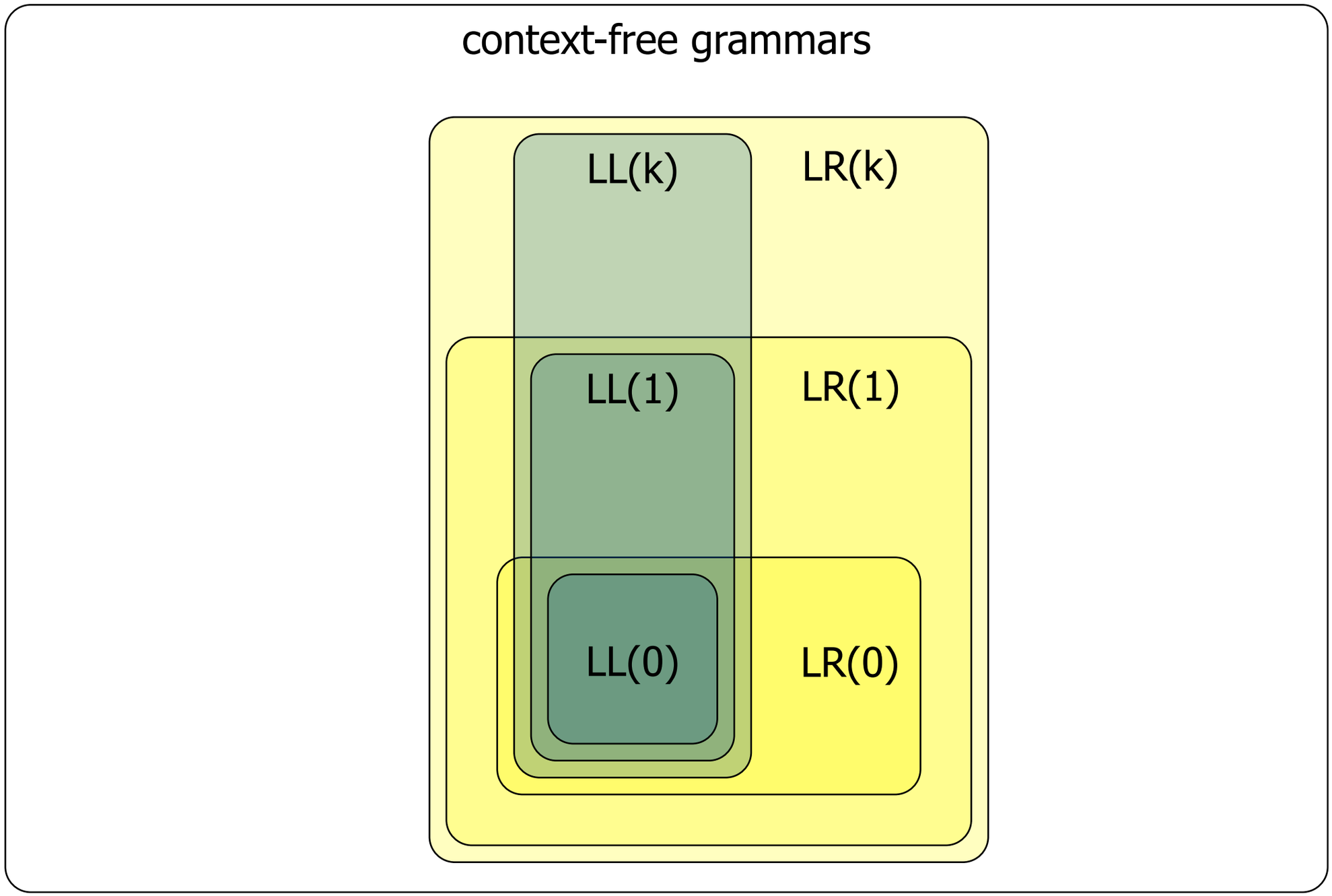
Grammar classes



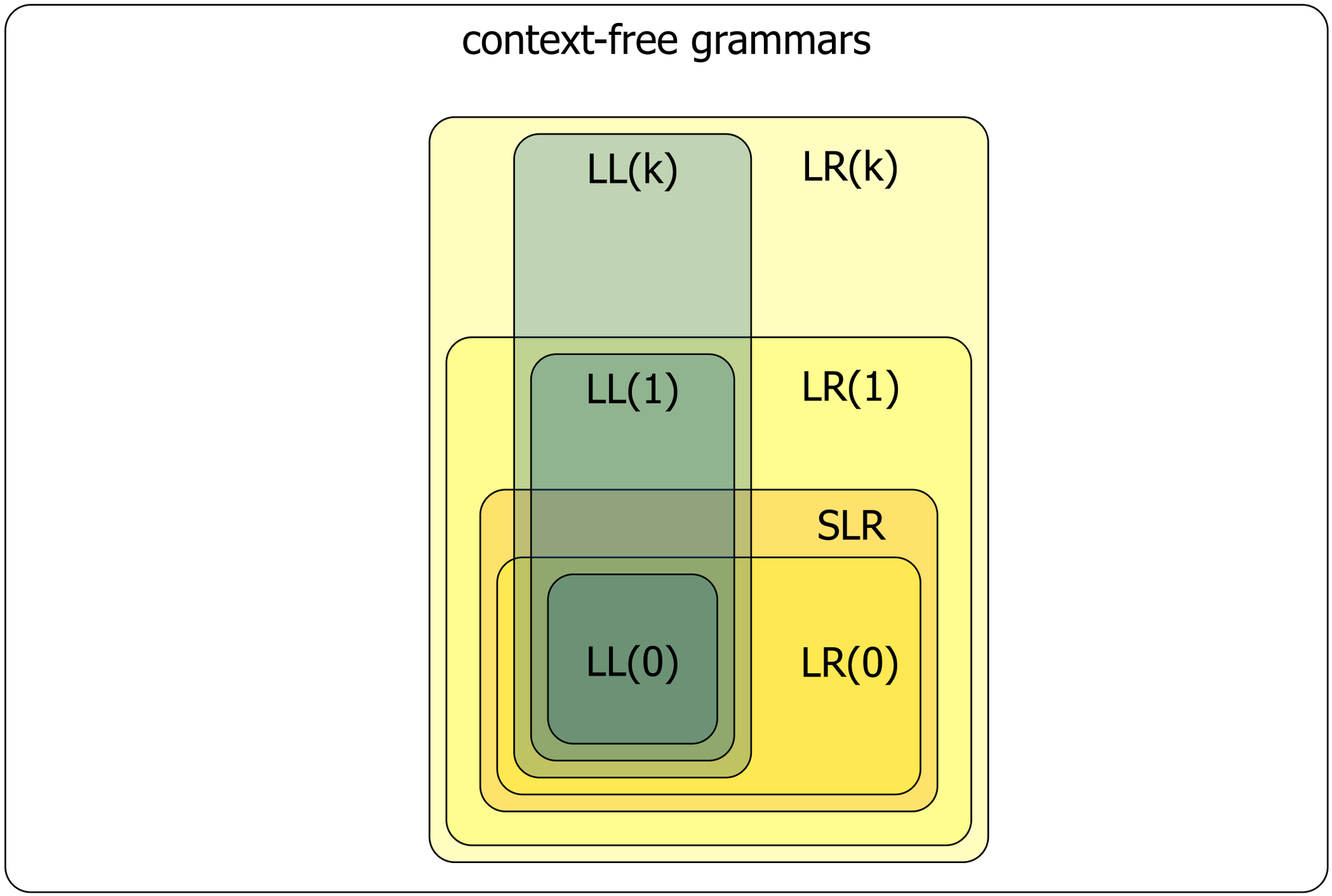
Grammar classes



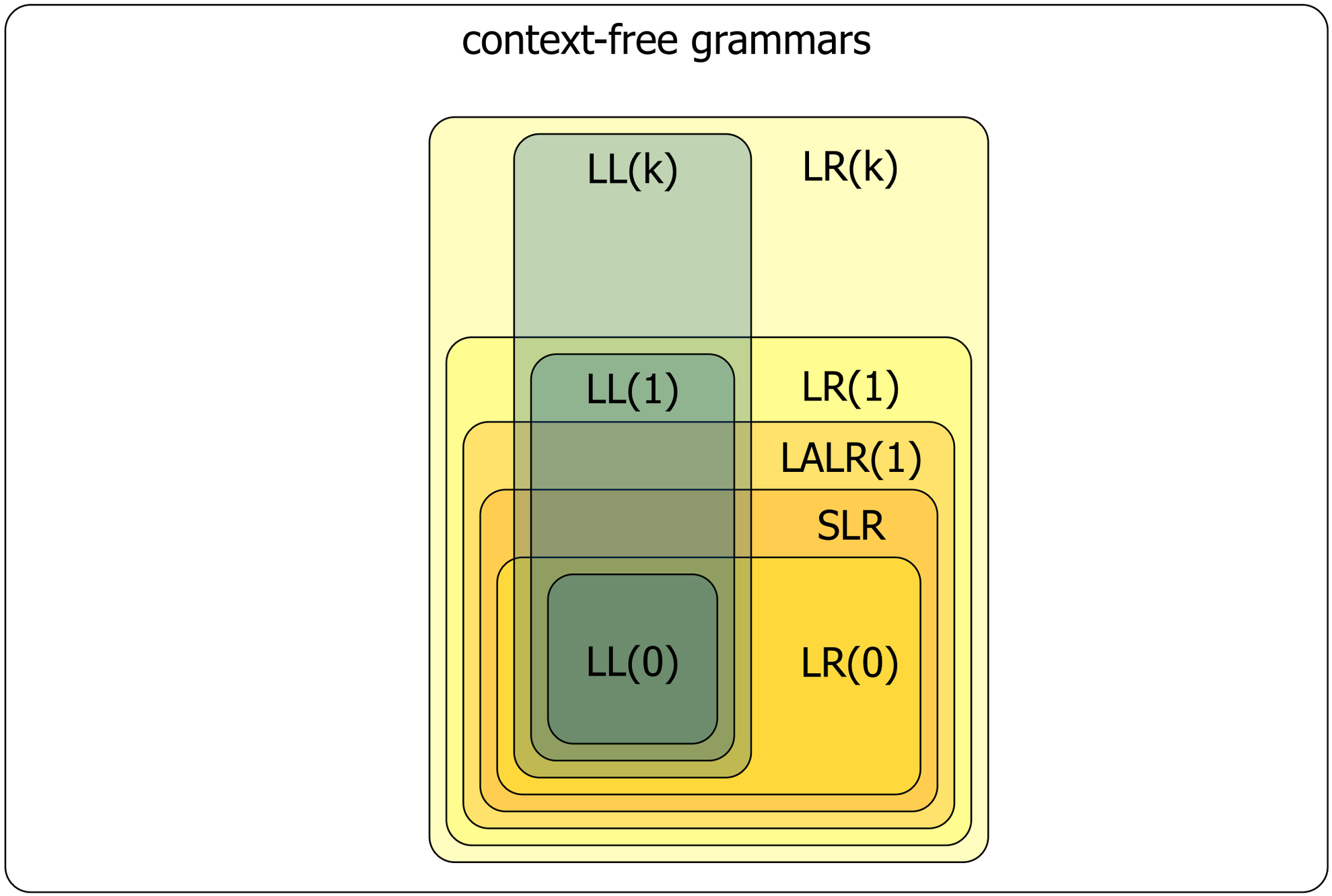
Grammar classes



Grammar classes



Grammar classes



II

LR parse tables

LR parsing
parse table

rows

- states of a DFA

columns

- topmost stack symbol
- Σ, N

entries

- reduce, goto state
- shift, goto state
- goto state

	T_1	...	N_1	...
1	s 3			
2			g 5	
3	r 1			
4	r 2			
5				
6			g 1	
7	s 1			
8				
...				

LR(0) parse tables

items, closure & goto

$S \rightarrow x$
$S \rightarrow (L)$
$L \rightarrow S$
$L \rightarrow L , S$

LR(0) parse tables

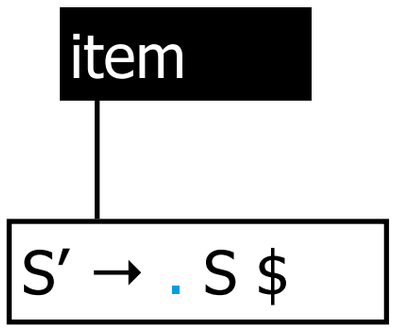
items, closure & goto

S′ → . S $

S → x
S → (L)
L → S
L → L , S

LR(0) parse tables

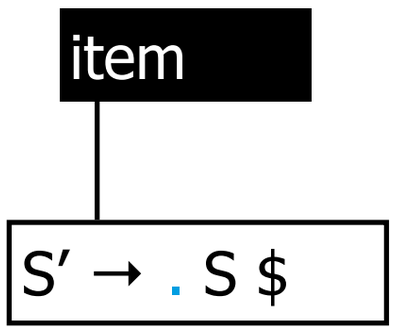
items, closure & goto



S → x
S → (L)
L → S
L → L , S

LR(0) parse tables

items, closure & goto



closure

- for every item A → α . X β
- for every rule X → γ
- add item X → . γ

S → x
S → (L)
L → S
L → L , S

LR(0) parse tables

items, closure & goto

S′ → . S $
S → . x
S → . (L)

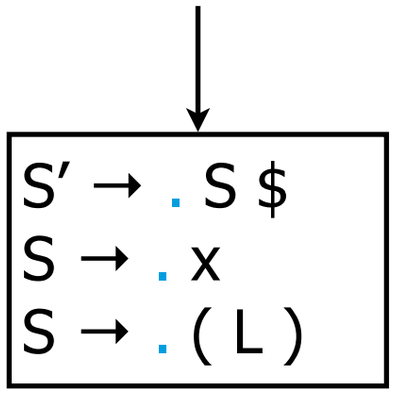
closure

- for every item A → α . X β
- for every rule X → γ
- add item X → . γ

S → x
S → (L)
L → S
L → L , S

LR(0) parse tables

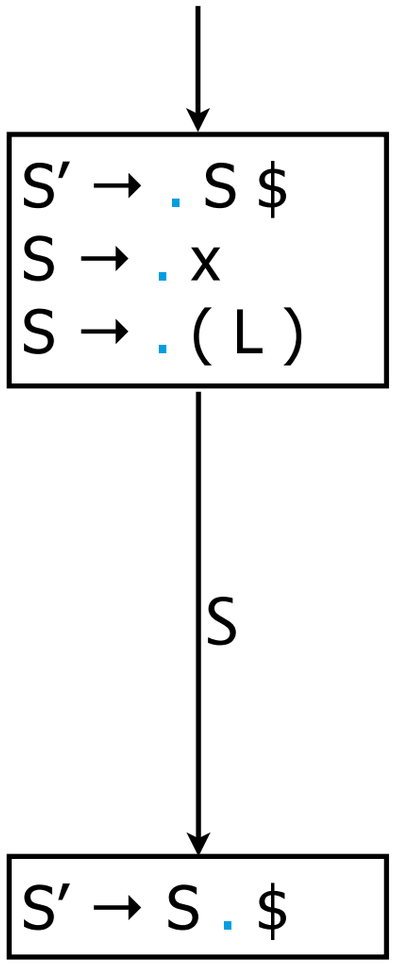
items, closure & goto



S → x
S → (L)
L → S
L → L , S

LR(0) parse tables

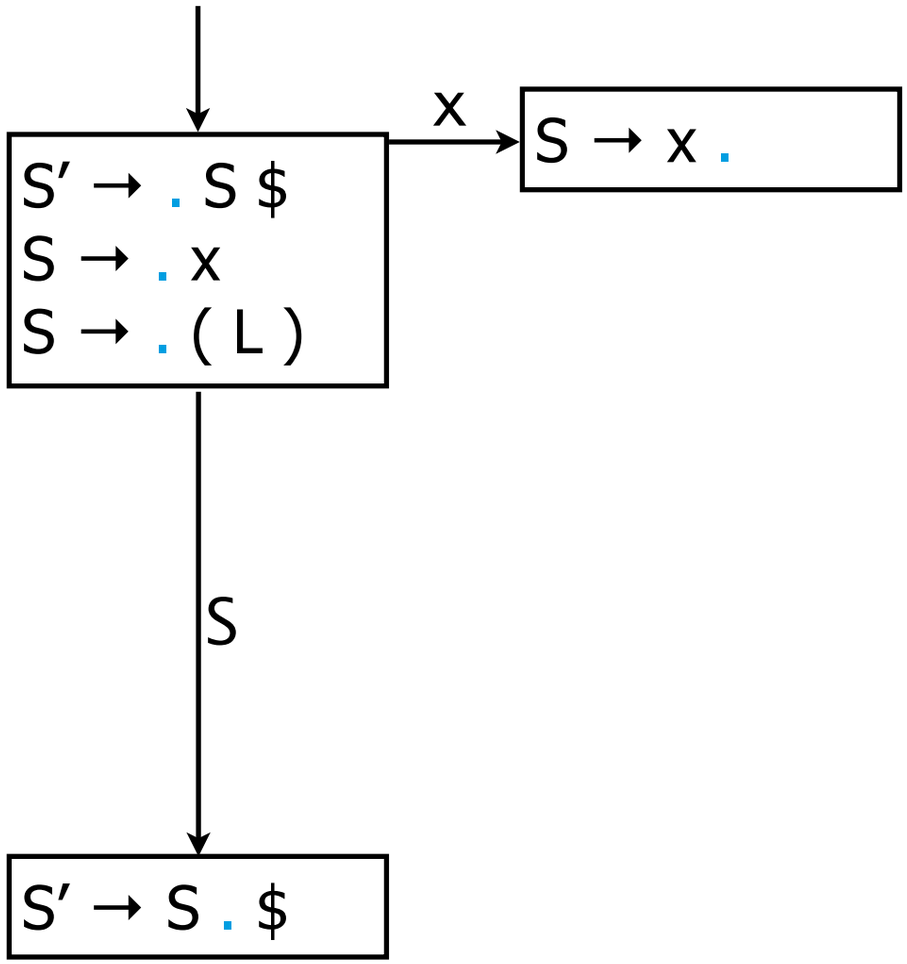
items, closure & goto



S → x
S → (L)
L → S
L → L , S

LR(0) parse tables

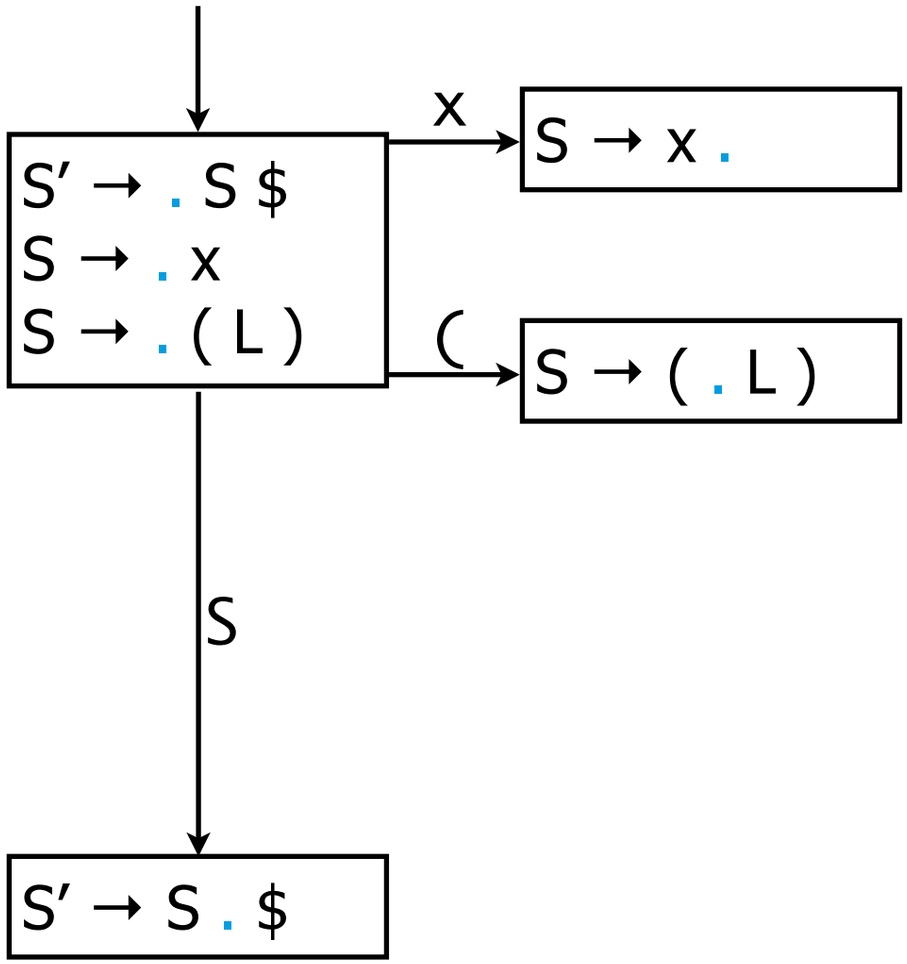
items, closure & goto



S → x
S → (L)
L → S
L → L , S

LR(0) parse tables

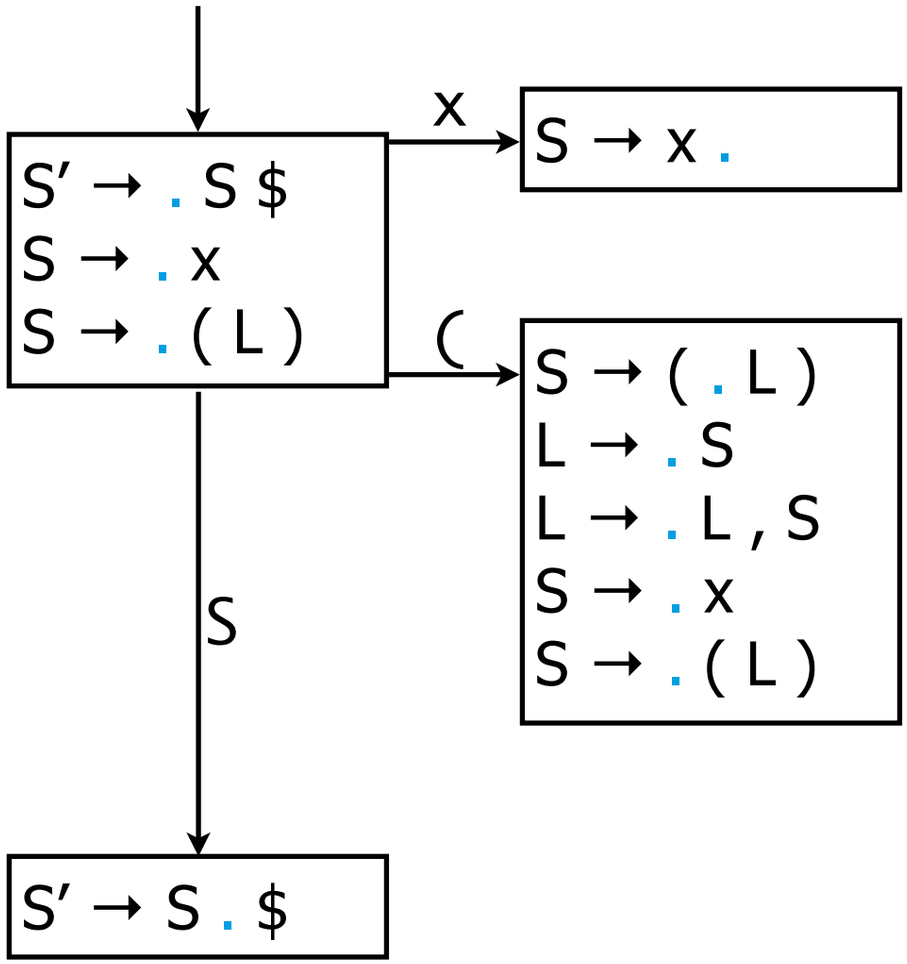
items, closure & goto



S → x
S → (L)
L → S
L → L , S

LR(0) parse tables

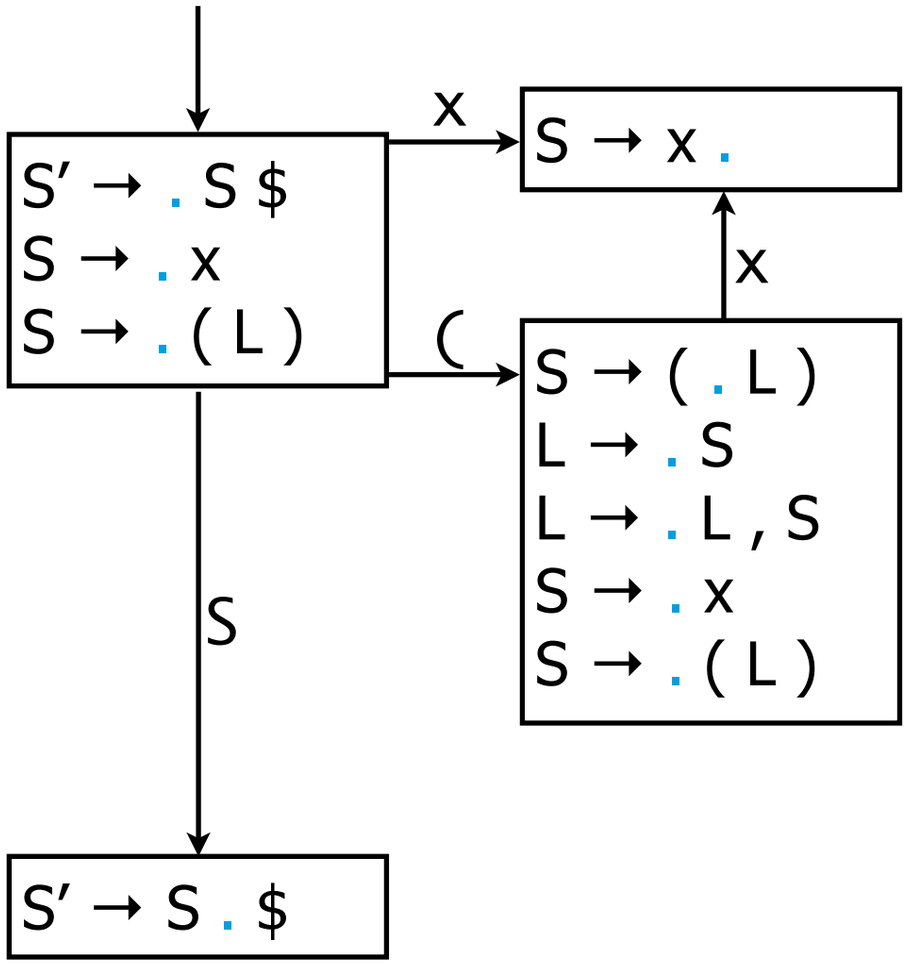
items, closure & goto



S → x
S → (L)
L → S
L → L , S

LR(0) parse tables

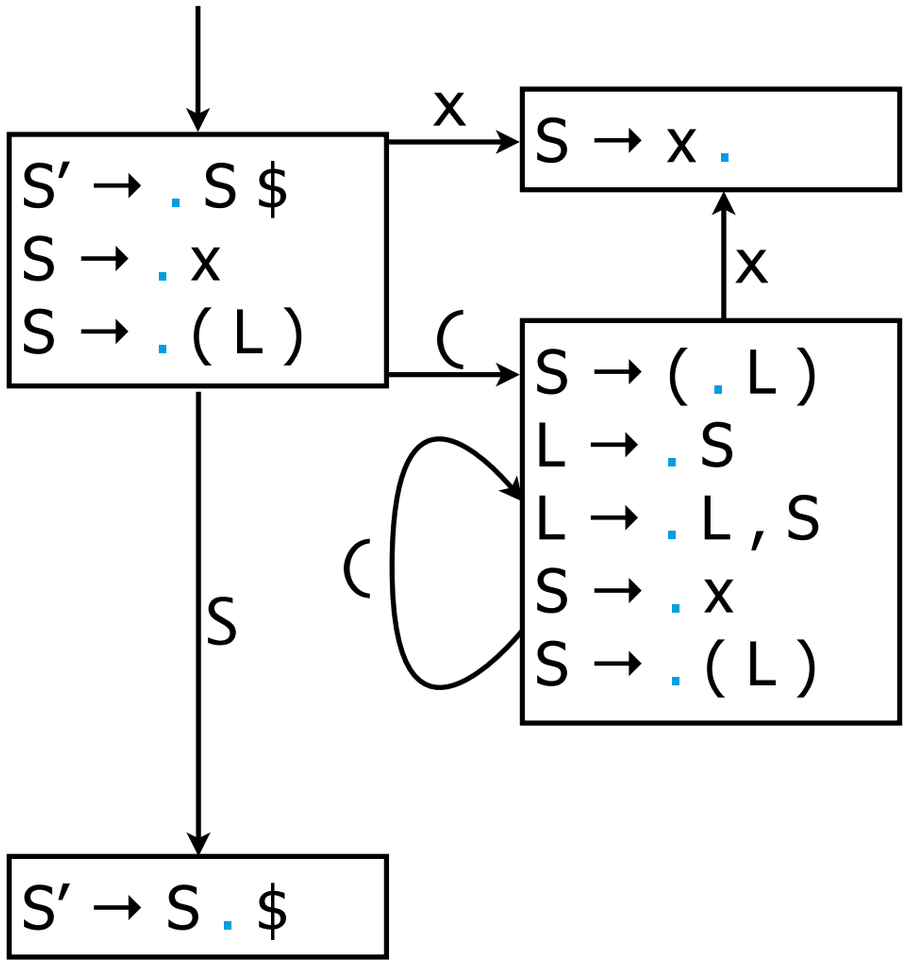
items, closure & goto



S → x
S → (L)
L → S
L → L , S

LR(0) parse tables

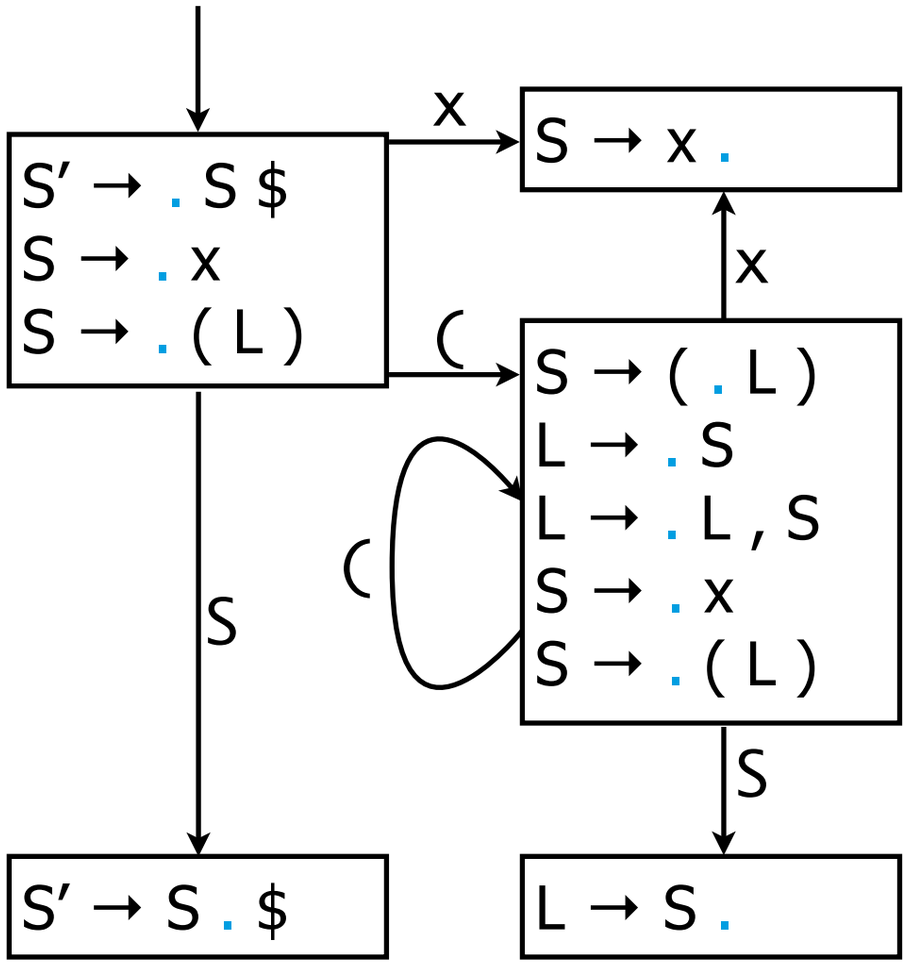
items, closure & goto



S → x
S → (L)
L → S
L → L , S

LR(0) parse tables

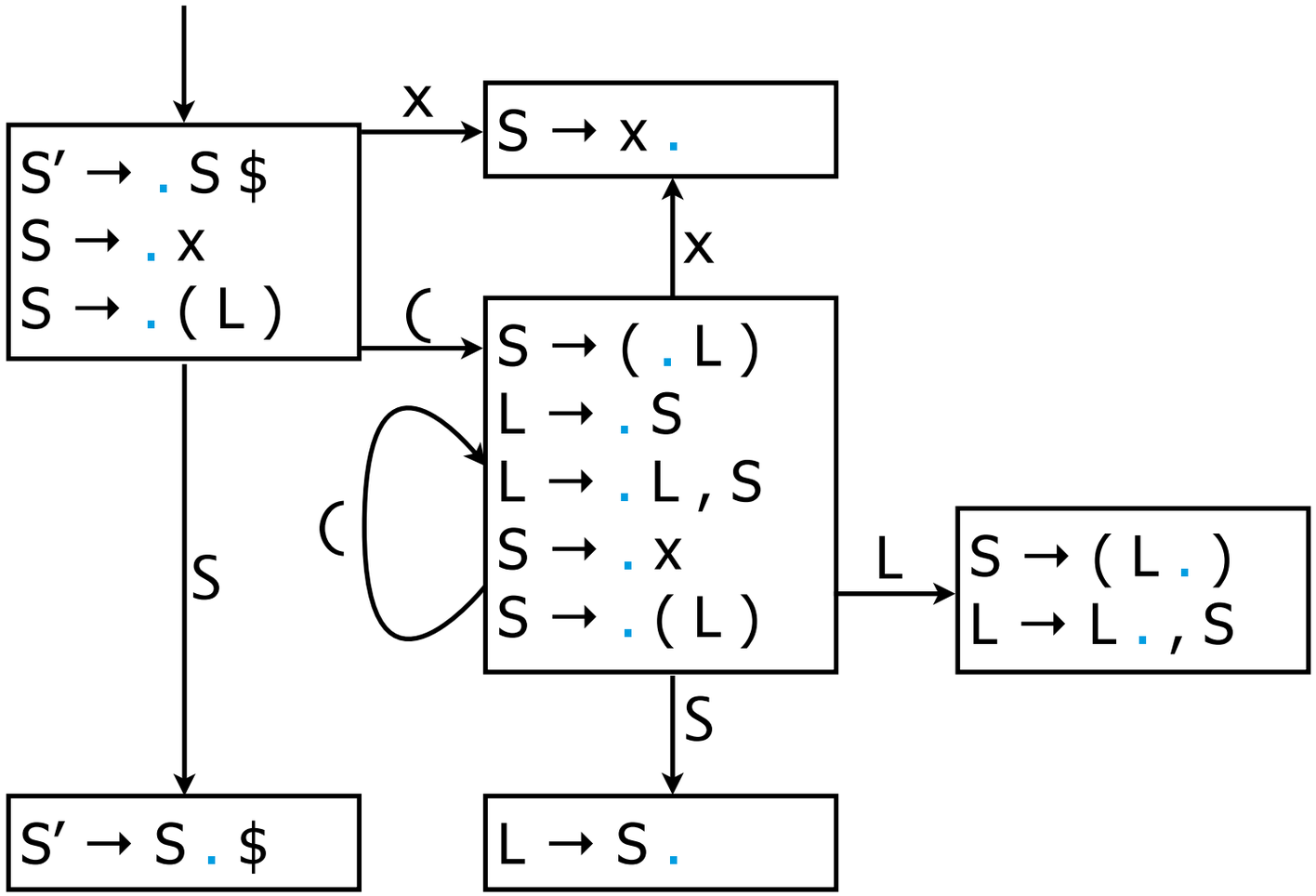
items, closure & goto



S → x
S → (L)
L → S
L → L , S

LR(0) parse tables

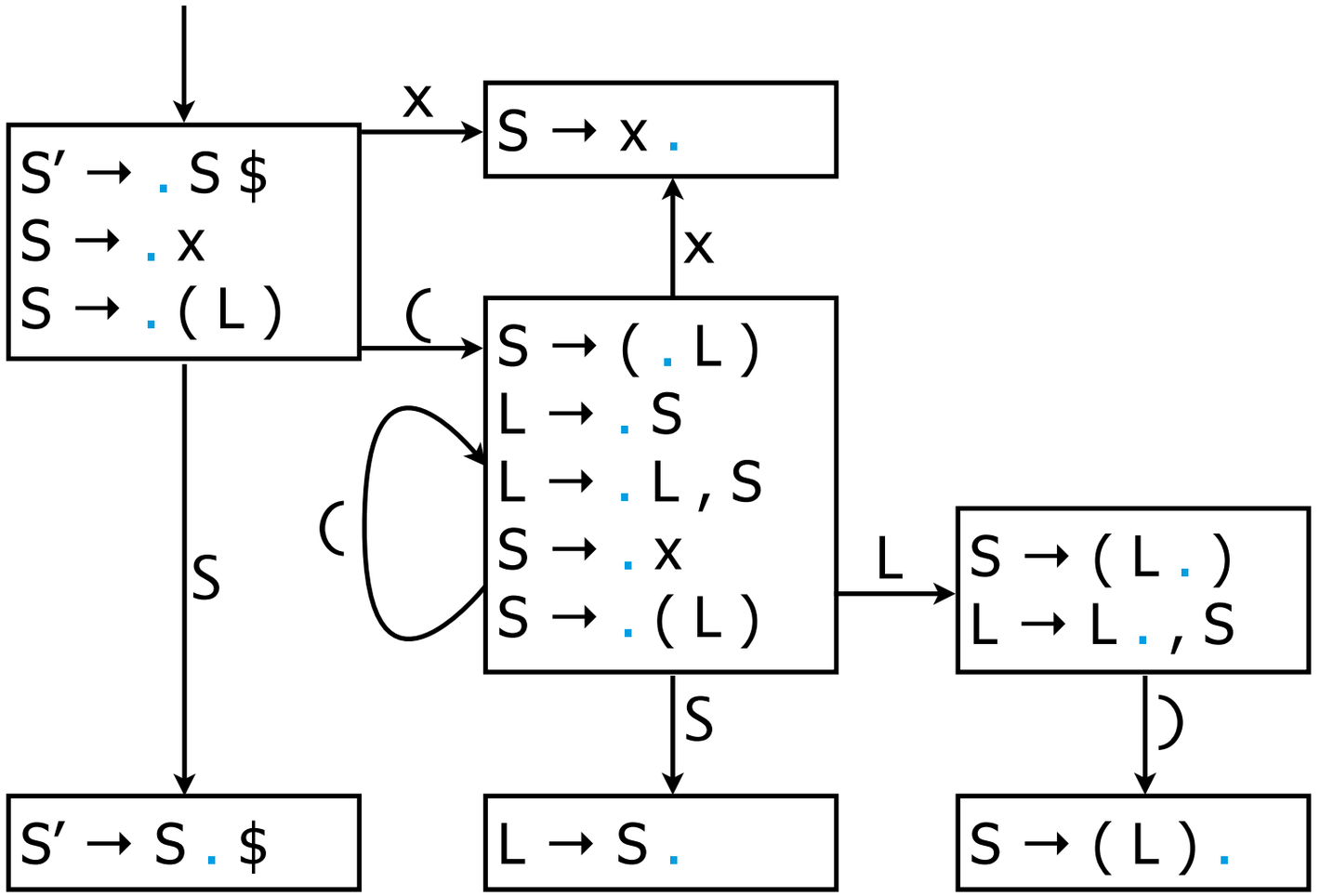
items, closure & goto



S → x
S → (L)
L → S
L → L , S

LR(0) parse tables

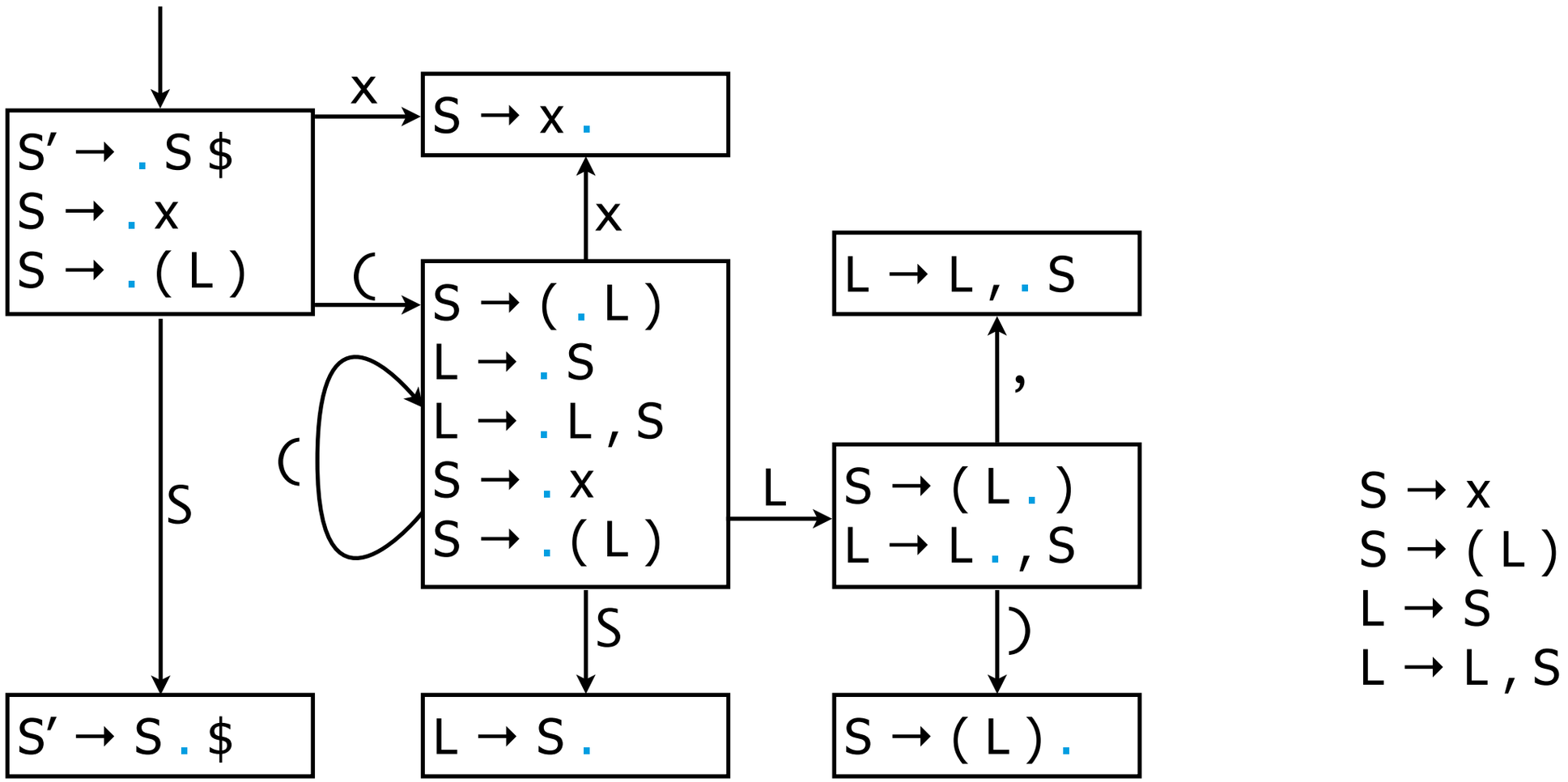
items, closure & goto



S → x
S → (L)
L → S
L → L , S

LR(0) parse tables

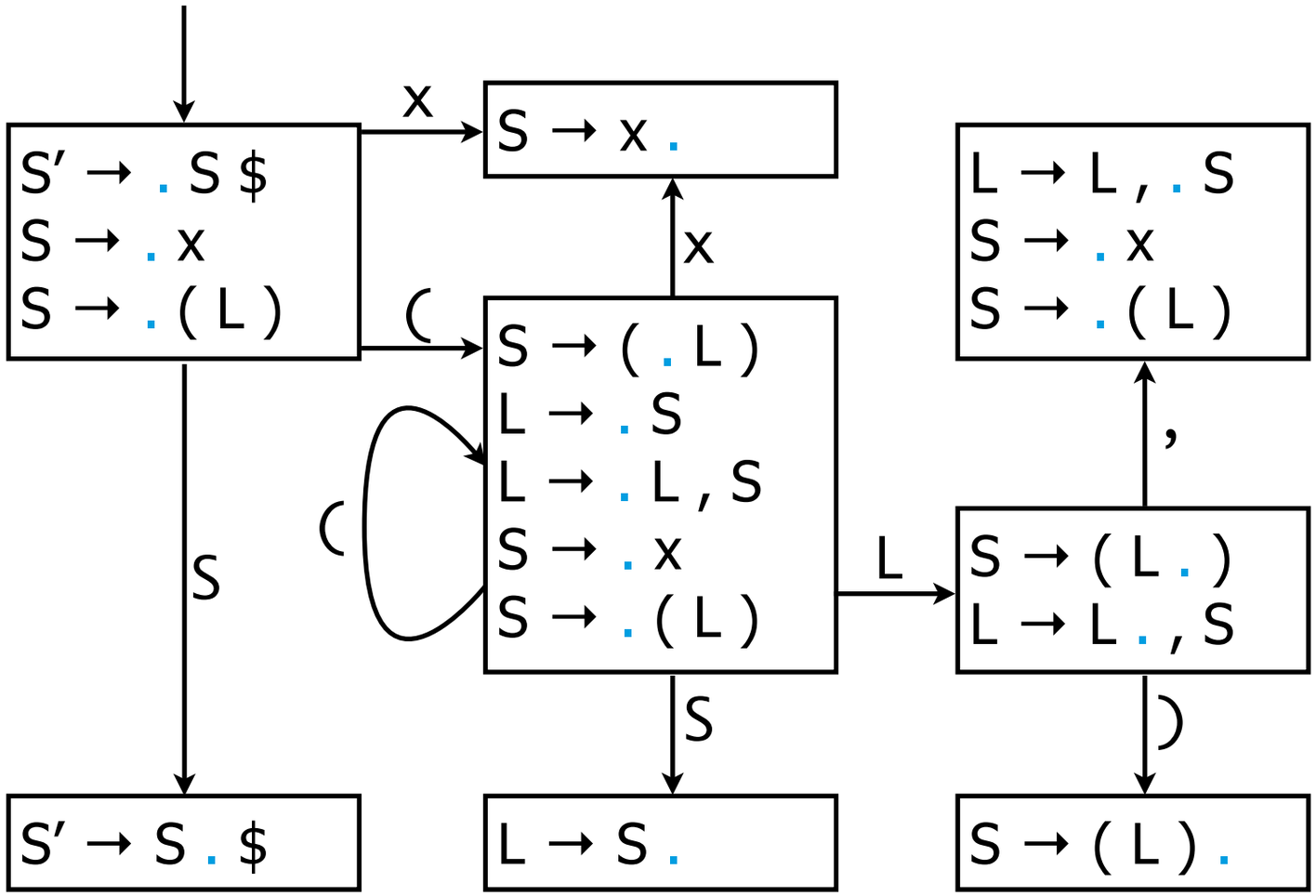
items, closure & goto



S → x
S → (L)
L → S
L → L , S

LR(0) parse tables

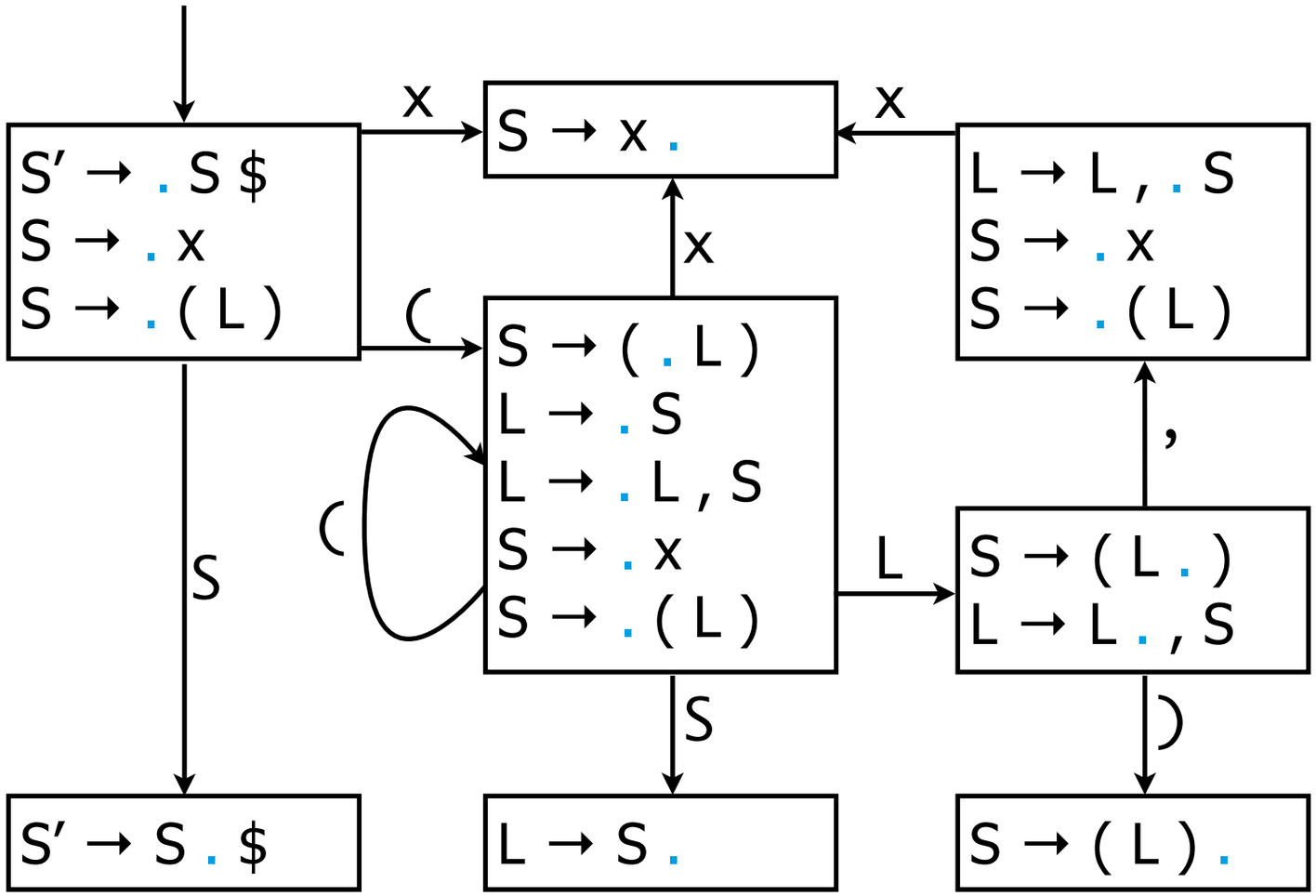
items, closure & goto



S → x
S → (L)
L → S
L → L , S

LR(0) parse tables

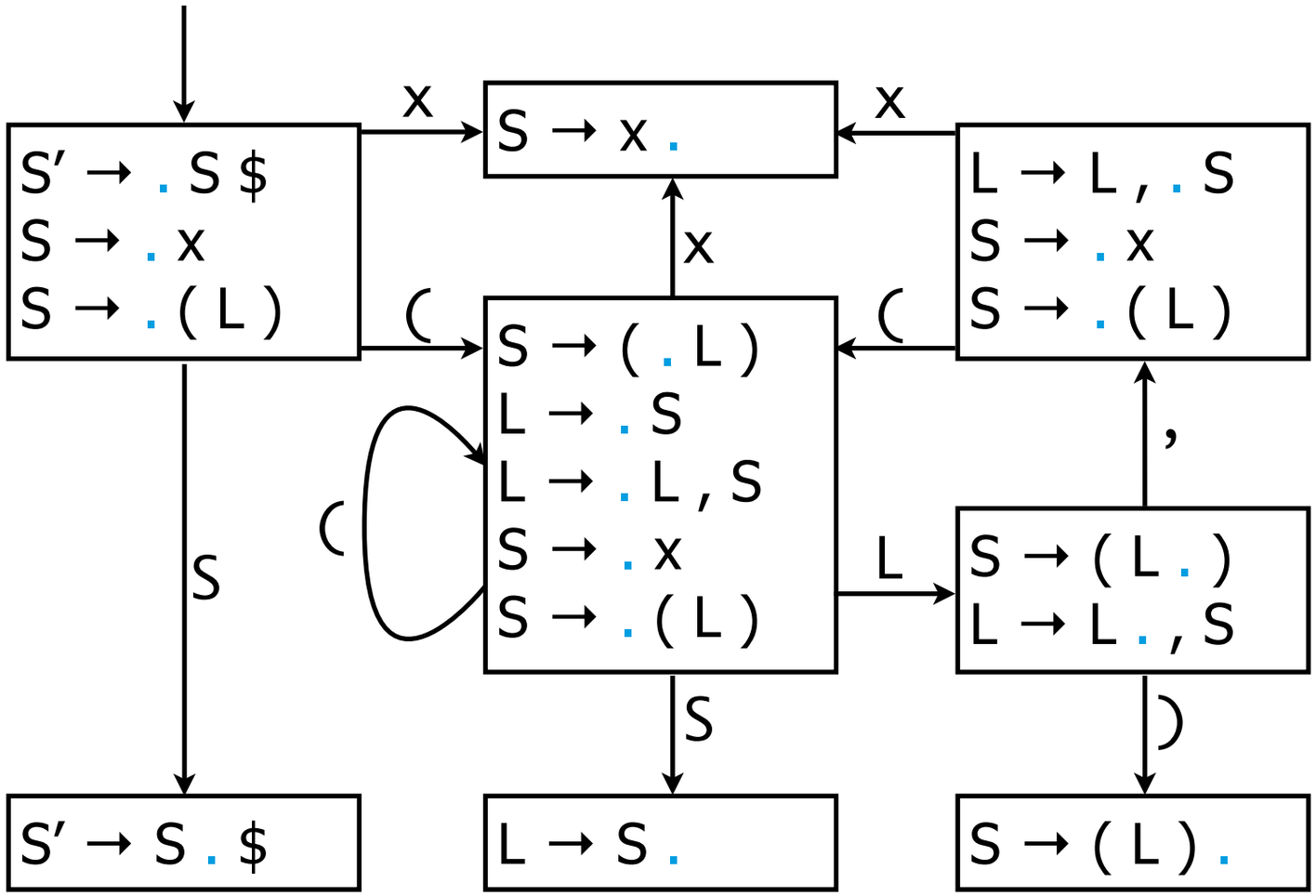
items, closure & goto



S → x
S → (L)
L → S
L → L , S

LR(0) parse tables

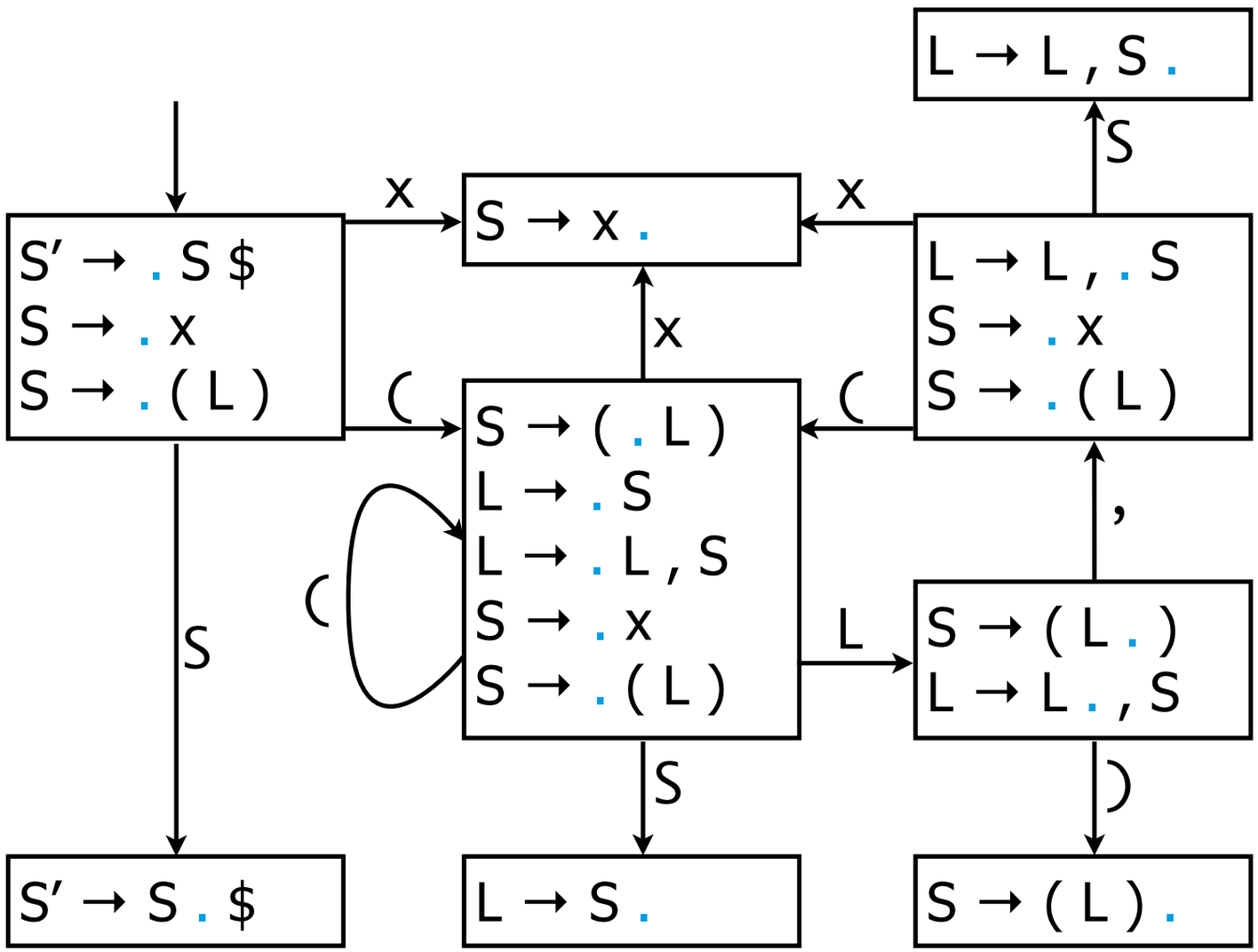
items, closure & goto



S → x
S → (L)
L → S
L → L , S

LR(0) parse tables

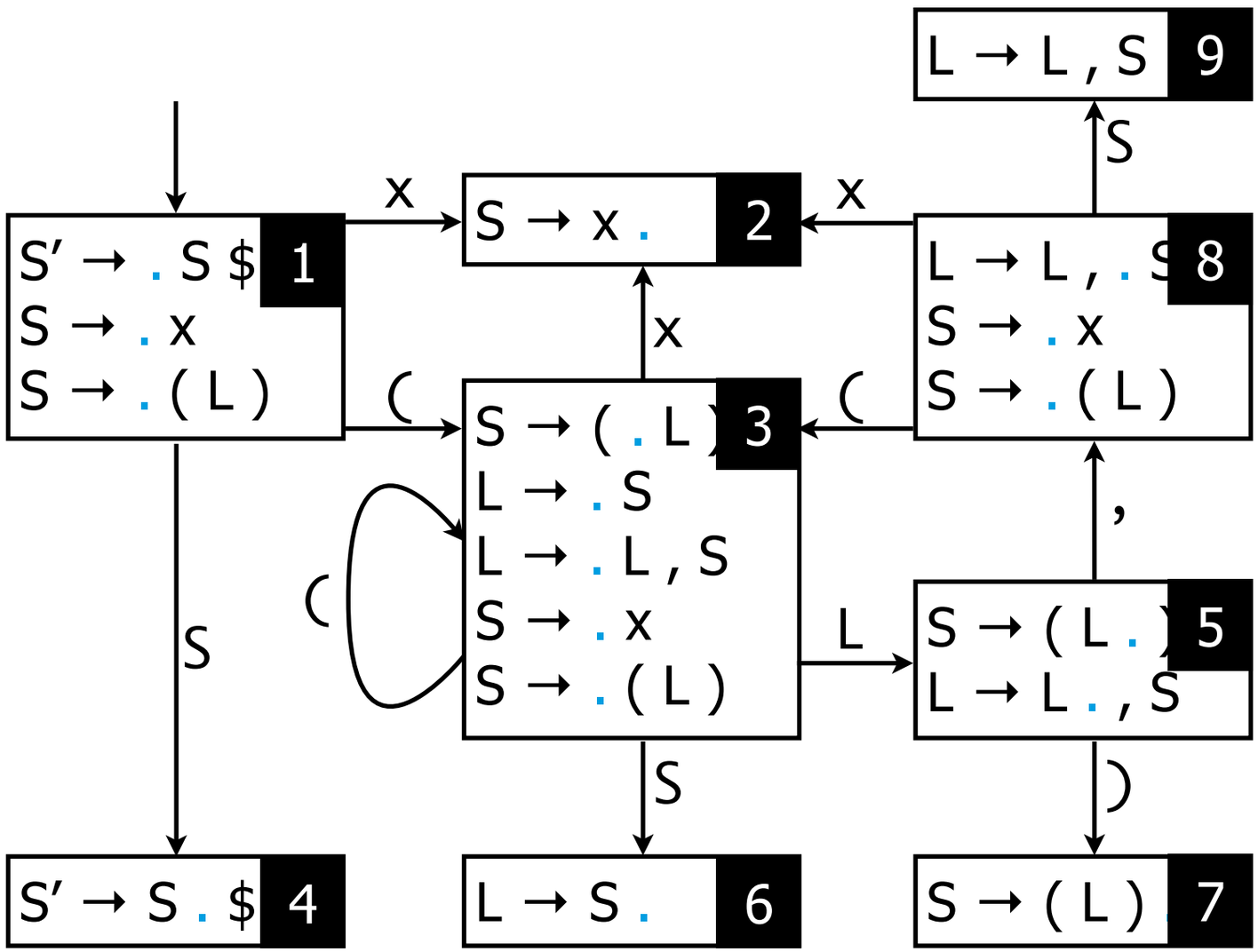
items, closure & goto



S → x
S → (L)
L → S
L → L , S

LR(0) parse tables

items, closure & goto



S → x
S → (L)
L → S
L → L , S

LR(0) parse tables

result

	(	)	x	,	$	S	L
1	s 3		s 2			g 4	
2	r 1	r 1	r 1	r 1	r 1		
3	s 3		s 2			g 6	g 5
4					a		
5		s 7		s 8			
6	r 3	r 3	r 3	r 3	r 3		
7	r 2	r 2	r 2	r 2	r 2		
8	s 3		s 2			g 9	
9	r 4	r 4	r 4	r 4	r 4		

S → x
S → (L)
L → S
L → L , S

LR(0) parse tables

result

(	x	,	x	)	$

	(	)	x	,	$	S	L
1	s 3		s 2			g 4	
2	r 1	r 1	r 1	r 1	r 1		
3	s 3		s 2			g 6	g 5
4					a		
5		s 7		s 8			
6	r 3	r 3	r 3	r 3	r 3		
7	r 2	r 2	r 2	r 2	r 2		
8	s 3		s 2			g 9	
9	r 4	r 4	r 4	r 4	r 4		

1

S → x
S → (L)
L → S
L → L , S

LR(0) parse tables

result

x	,	x	)	$

	(	)	x	,	$	S	L
1	s 3		s 2			g 4	
2	r 1	r 1	r 1	r 1	r 1		
3	s 3		s 2			g 6	g 5
4					a		
5		s 7		s 8			
6	r 3	r 3	r 3	r 3	r 3		
7	r 2	r 2	r 2	r 2	r 2		
8	s 3		s 2			g 9	
9	r 4	r 4	r 4	r 4	r 4		

3
1

S → x
S → (L)
L → S
L → L , S

LR(0) parse tables

result

,	x	)	$

	(	)	x	,	$	S	L
1	s 3		s 2			g 4	
2	r 1	r 1	r 1	r 1	r 1		
3	s 3		s 2			g 6	g 5
4					a		
5		s 7		s 8			
6	r 3	r 3	r 3	r 3	r 3		
7	r 2	r 2	r 2	r 2	r 2		
8	s 3		s 2			g 9	
9	r 4	r 4	r 4	r 4	r 4		

2
3
1

S → x
S → (L)
L → S
L → L , S

LR(0) parse tables

result

,	x	)	$

	(	)	x	,	$	S	L
1	s 3		s 2			g 4	
2	r 1	r 1	r 1	r 1	r 1		
3	s 3		s 2			g 6	g 5
4					a		
5		s 7		s 8			
6	r 3	r 3	r 3	r 3	r 3		
7	r 2	r 2	r 2	r 2	r 2		
8	s 3		s 2			g 9	
9	r 4	r 4	r 4	r 4	r 4		

3
1

S → x
S → (L)
L → S
L → L , S

LR(0) parse tables

result

,	x	)	$

	(	)	x	,	$	S	L
1	s 3		s 2			g 4	
2	r 1	r 1	r 1	r 1	r 1		
3	s 3		s 2			g 6	g 5
4					a		
5		s 7		s 8			
6	r 3	r 3	r 3	r 3	r 3		
7	r 2	r 2	r 2	r 2	r 2		
8	s 3		s 2			g 9	
9	r 4	r 4	r 4	r 4	r 4		

6
3
1

S → x
S → (L)
L → S
L → L , S

LR(0) parse tables

result

,	x	)	$

	(	)	x	,	$	S	L
1	s 3		s 2			g 4	
2	r 1	r 1	r 1	r 1	r 1		
3	s 3		s 2			g 6	g 5
4					a		
5		s 7		s 8			
6	r 3	r 3	r 3	r 3	r 3		
7	r 2	r 2	r 2	r 2	r 2		
8	s 3		s 2			g 9	
9	r 4	r 4	r 4	r 4	r 4		

3
1

S → x
S → (L)
L → S
L → L , S

LR(0) parse tables

result

,	x	)	$

	(	)	x	,	$	S	L
1	s 3		s 2			g 4	
2	r 1	r 1	r 1	r 1	r 1		
3	s 3		s 2			g 6	g 5
4					a		
5		s 7		s 8			
6	r 3	r 3	r 3	r 3	r 3		
7	r 2	r 2	r 2	r 2	r 2		
8	s 3		s 2			g 9	
9	r 4	r 4	r 4	r 4	r 4		

5
3
1

S → x
S → (L)
L → S
L → L , S

LR(0) parse tables

result

x	)	$

	(	)	x	,	$	S	L
1	s 3		s 2			g 4	
2	r 1	r 1	r 1	r 1	r 1		
3	s 3		s 2			g 6	g 5
4					a		
5		s 7		s 8			
6	r 3	r 3	r 3	r 3	r 3		
7	r 2	r 2	r 2	r 2	r 2		
8	s 3		s 2			g 9	
9	r 4	r 4	r 4	r 4	r 4		

8
5
3
1

S → x
S → (L)
L → S
L → L , S

LR(0) parse tables

result

)	$

	(	)	x	,	$	S	L
1	s 3		s 2			g 4	
2	r 1	r 1	r 1	r 1	r 1		
3	s 3		s 2			g 6	g 5
4					a		
5		s 7		s 8			
6	r 3	r 3	r 3	r 3	r 3		
7	r 2	r 2	r 2	r 2	r 2		
8	s 3		s 2			g 9	
9	r 4	r 4	r 4	r 4	r 4		

2
8
5
3
1

S → x
S → (L)
L → S
L → L , S

LR(0) parse tables

result

)	$

	(	)	x	,	$	S	L
1	s 3		s 2			g 4	
2	r 1	r 1	r 1	r 1	r 1		
3	s 3		s 2			g 6	g 5
4					a		
5		s 7		s 8			
6	r 3	r 3	r 3	r 3	r 3		
7	r 2	r 2	r 2	r 2	r 2		
8	s 3		s 2			g 9	
9	r 4	r 4	r 4	r 4	r 4		

8
5
3
1

S → x
S → (L)
L → S
L → L , S

LR(0) parse tables

result

)	$

	(	)	x	,	$	S	L
1	s 3		s 2			g 4	
2	r 1	r 1	r 1	r 1	r 1		
3	s 3		s 2			g 6	g 5
4					a		
5		s 7		s 8			
6	r 3	r 3	r 3	r 3	r 3		
7	r 2	r 2	r 2	r 2	r 2		
8	s 3		s 2			g 9	
9	r 4	r 4	r 4	r 4	r 4		

9
8
5
3
1

S → x
S → (L)
L → S
L → L , S

LR(0) parse tables

result

)	$

	(	)	x	,	$	S	L
1	s 3		s 2			g 4	
2	r 1	r 1	r 1	r 1	r 1		
3	s 3		s 2			g 6	g 5
4					a		
5		s 7		s 8			
6	r 3	r 3	r 3	r 3	r 3		
7	r 2	r 2	r 2	r 2	r 2		
8	s 3		s 2			g 9	
9	r 4	r 4	r 4	r 4	r 4		

3
1

S → x
S → (L)
L → S
L → L , S

LR(0) parse tables

result

)	$

	(	)	x	,	$	S	L
1	s 3		s 2			g 4	
2	r 1	r 1	r 1	r 1	r 1		
3	s 3		s 2			g 6	g 5
4					a		
5		s 7		s 8			
6	r 3	r 3	r 3	r 3	r 3		
7	r 2	r 2	r 2	r 2	r 2		
8	s 3		s 2			g 9	
9	r 4	r 4	r 4	r 4	r 4		

5
3
1

S → x
S → (L)
L → S
L → L , S

LR(0) parse tables

result

	(	)	x	,	$	S	L
1	s 3		s 2			g 4	
2	r 1	r 1	r 1	r 1	r 1		
3	s 3		s 2			g 6	g 5
4					a		
5		s 7		s 8			
6	r 3	r 3	r 3	r 3	r 3		
7	r 2	r 2	r 2	r 2	r 2		
8	s 3		s 2			g 9	
9	r 4	r 4	r 4	r 4	r 4		

$

7
5
3
1

S → x
S → (L)
L → S
L → L , S

LR(0) parse tables

result

$

	(	)	x	,	$	S	L
1	s 3		s 2			g 4	
2	r 1	r 1	r 1	r 1	r 1		
3	s 3		s 2			g 6	g 5
4					a		
5		s 7		s 8			
6	r 3	r 3	r 3	r 3	r 3		
7	r 2	r 2	r 2	r 2	r 2		
8	s 3		s 2			g 9	
9	r 4	r 4	r 4	r 4	r 4		

1

S → x
S → (L)
L → S
L → L , S

LR(0) parse tables

result

	(	)	x	,	$	S	L
1	s 3		s 2			g 4	
2	r 1	r 1	r 1	r 1	r 1		
3	s 3		s 2			g 6	g 5
4					a		
5		s 7		s 8			
6	r 3	r 3	r 3	r 3	r 3		
7	r 2	r 2	r 2	r 2	r 2		
8	s 3		s 2			g 9	
9	r 4	r 4	r 4	r 4	r 4		

4
1

$

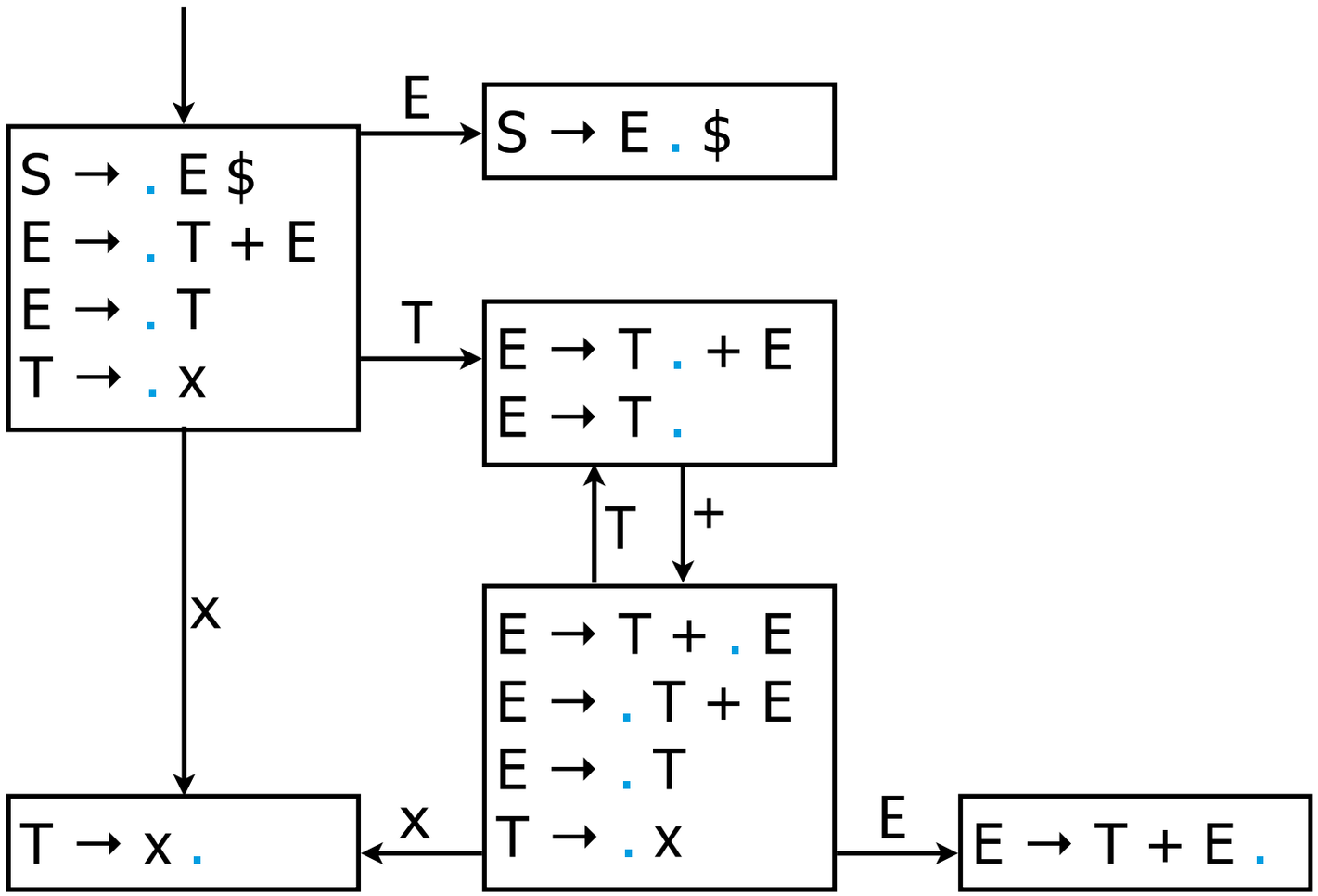
S → x
S → (L)
L → S
L → L , S

III

conflict resolution

SLR parse tables

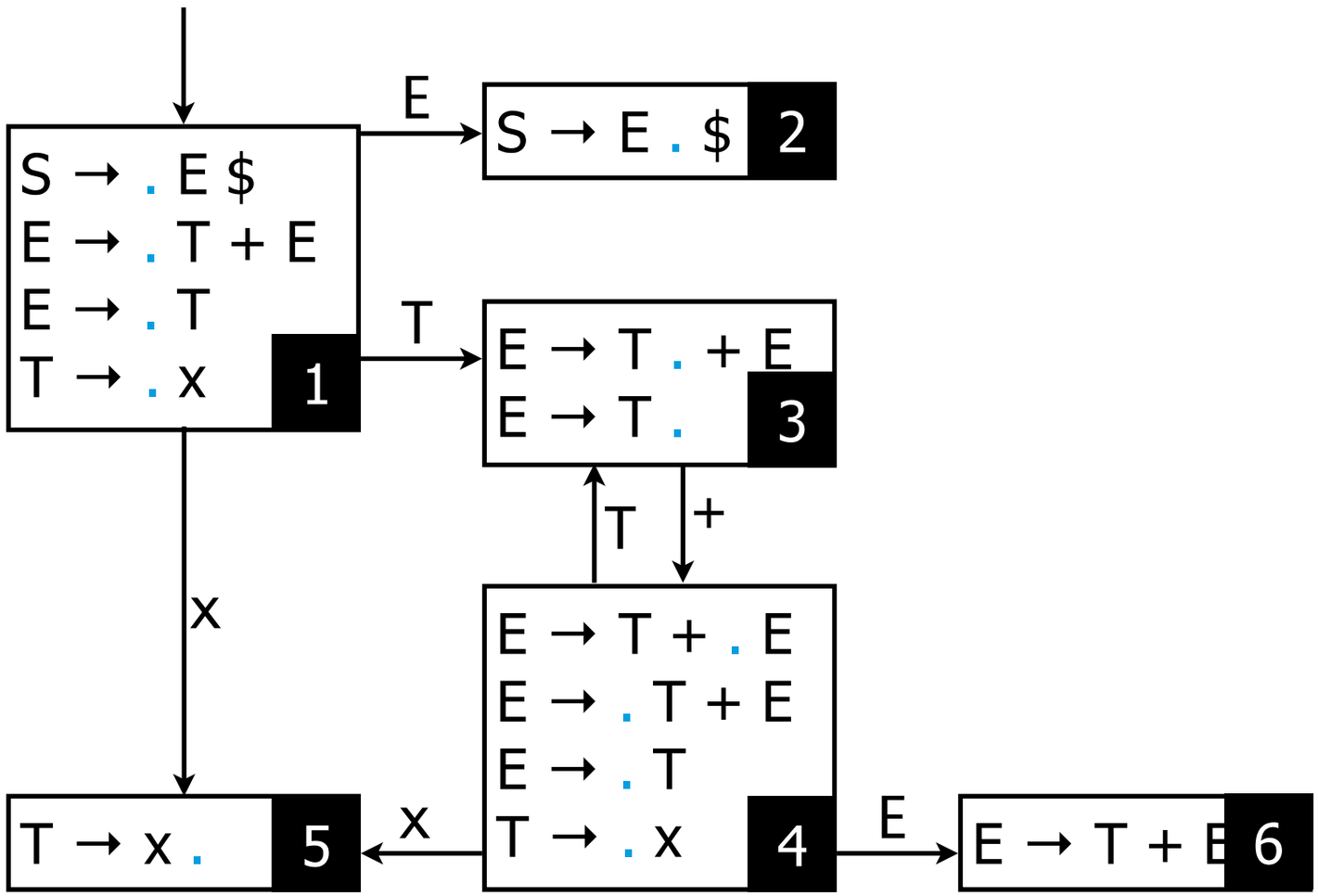
shift-reduce conflicts



E → T + E
E → T
T → x

SLR parse tables

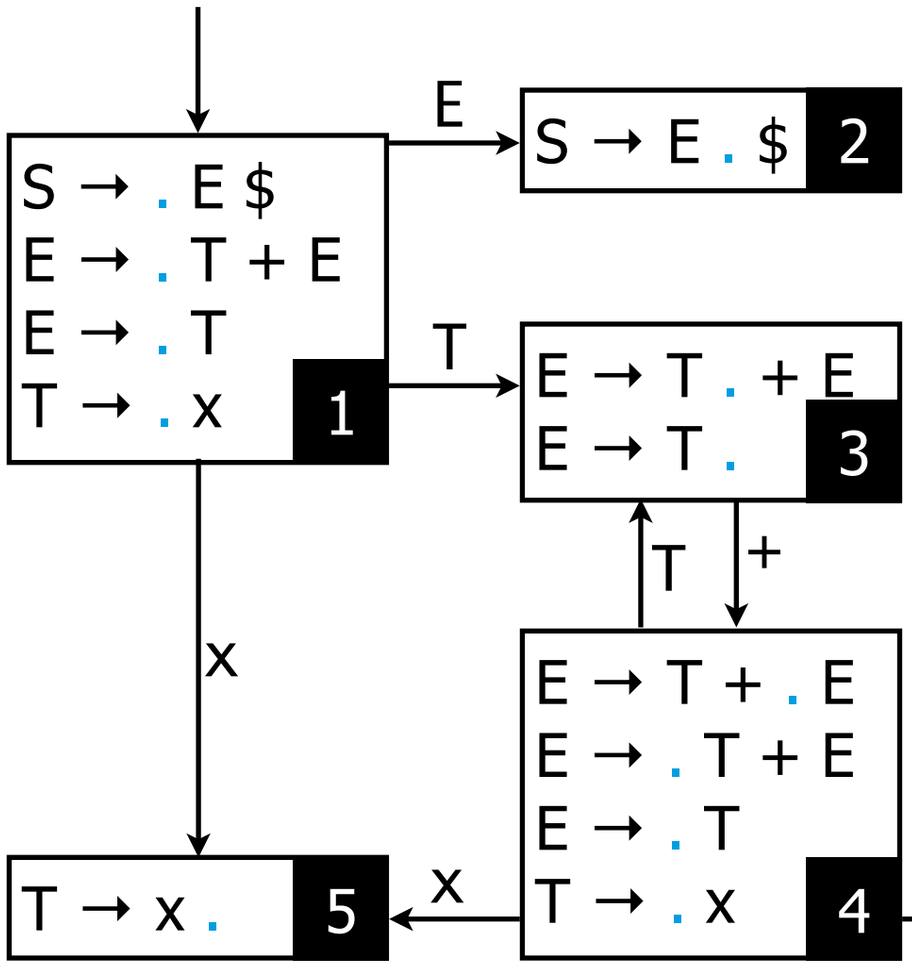
shift-reduce conflicts



E → T + E
E → T
T → x

SLR parse tables

shift-reduce conflicts

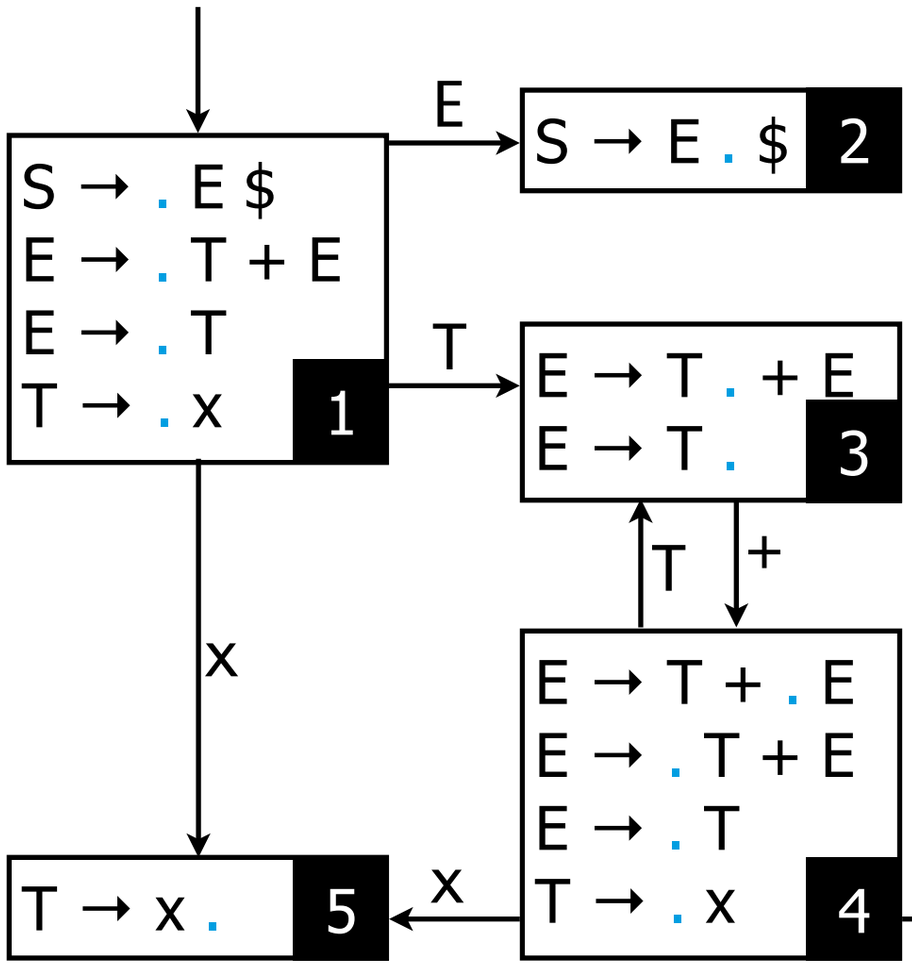


	x	+	$	E	T
1	s 5			g 2	g 3
2			a		
3	r 2	?	r 2		
4	s 5			g 6	g 3
5	r 3	r 3	r 3		
6	r 1	r 1	r 1		

E → T + E
E → T
T → x

SLR parse tables

shift-reduce conflicts

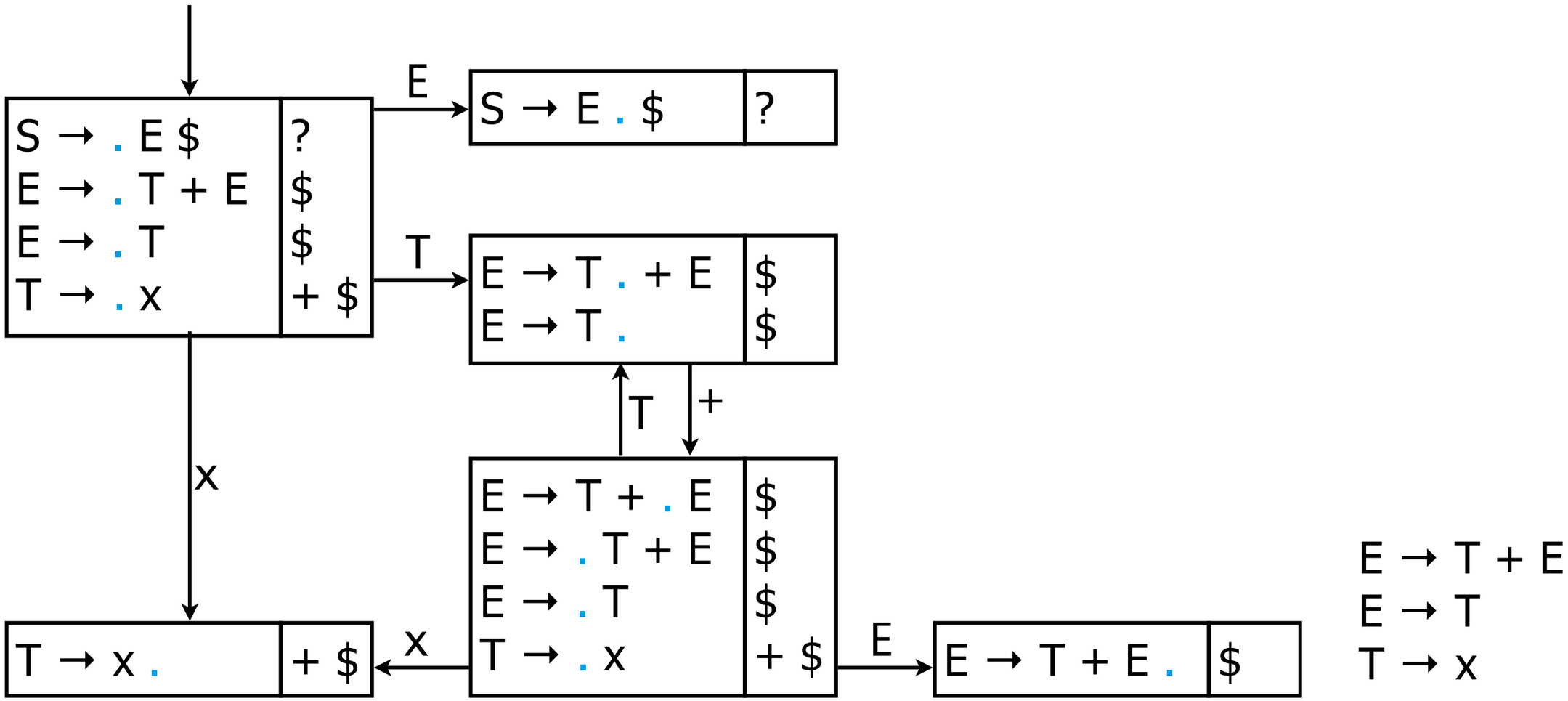


	x	+	$	E	T
1	s 5			g 2	g 3
2			a		
3		s 4	r 2		
4	s 5			g 6	g 3
5		r 3	r 3		
6			r 1		

E → T + E
E → T
T → x

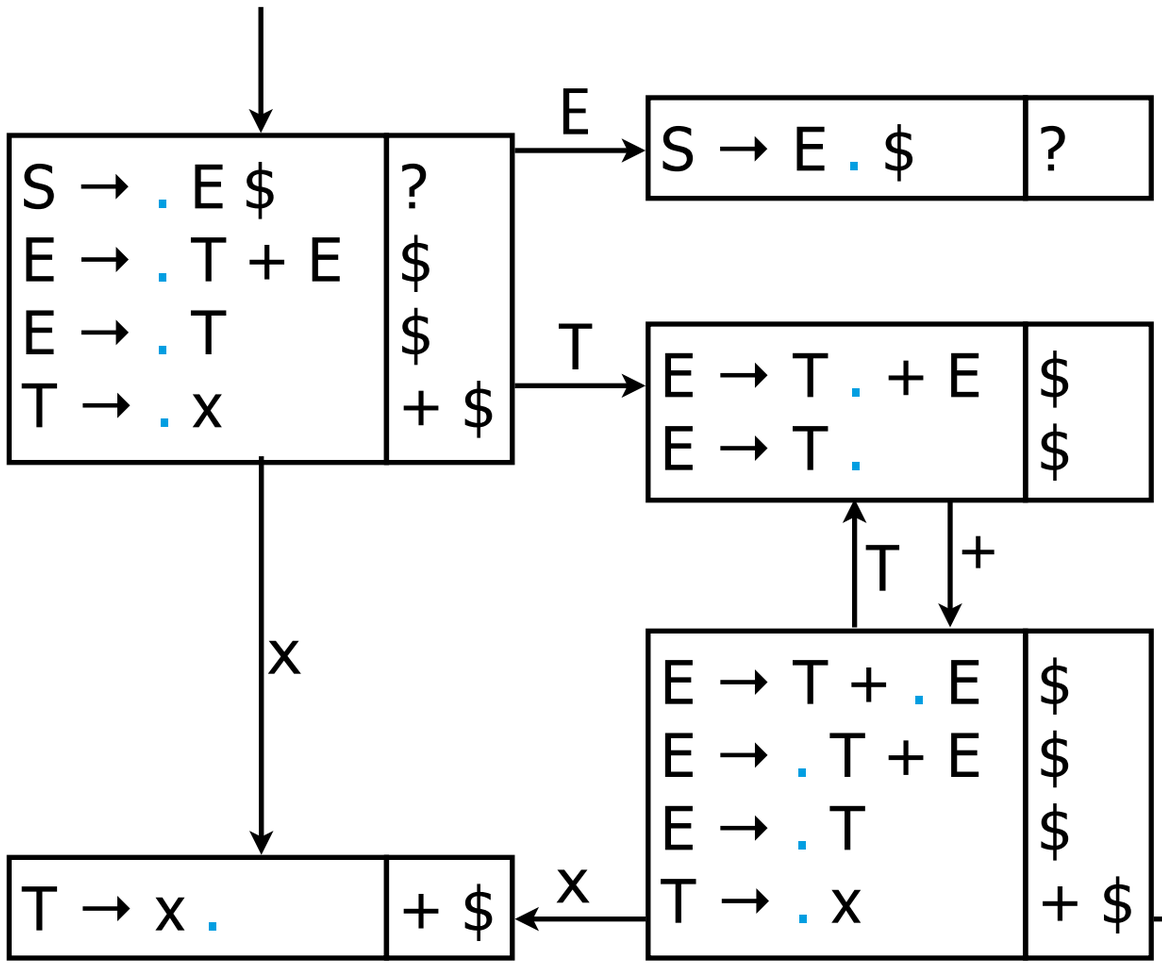
LR(1) parse tables

look-ahead



LR(1) parse tables

look-ahead



	x	+	$	E	T
1	s 5			g 2	g 3
2			a		
3		s 4	r 2		
4	s 5			g 6	g 3
5		r 3	r 3		
6			r 1		

E → T + E
E → T
T → x

LALR(1) parse tables
state space reduction

unify states

- with same items
- and same outgoing transitions
- but different look-ahead sets

might introduce new conflicts

IV

summary

Summary
lessons learned

Summary
lessons learned

How can we generate LR parse tables?

- items, closure, goto

Summary
lessons learned

How can we generate LR parse tables?

- items, closure, goto

How can we improve LR(0) parse table generation?

- SLR: consider FOLLOW sets to avoid shift-reduce conflicts
- LR(1): consider look-ahead in states
- LALR(1): unify LR(1) states to reduce state space

Summary
lessons learned

How can we generate LR parse tables?

- items, closure, goto

How can we improve LR(0) parse table generation?

- SLR: consider FOLLOW sets to avoid shift-reduce conflicts
- LR(1): consider look-ahead in states
- LALR(1): unify LR(1) states to reduce state space

Why are efficient parsing algorithms problematic?

- not longer pure, declarative, beautiful
- paradise lost: seven plagues
- paradise regained: scannerless generalised parsing

Summary
lessons learned

How can we generate LR parse tables?

- items, closure, goto

How can we improve LR(0) parse table generation?

- SLR: consider FOLLOW sets to avoid shift-reduce conflicts
- LR(1): consider look-ahead in states
- LALR(1): unify LR(1) states to reduce state space

Why are efficient parsing algorithms problematic?

- not longer pure, declarative, beautiful
- paradise lost: seven plagues
- paradise regained: scannerless generalised parsing

Literature

learn more

Literature

learn more

syntactical analysis

Andrew W. Appel, Jens Palsberg: Modern Compiler Implementation in Java, 2nd edition. 2002

Alfred V. Aho, Ravi Sethi, Jeffrey D. Ullman, Monica S. Lam: Compilers: Principles, Techniques, and Tools, 2nd edition. 2006

Literature

learn more

syntactical analysis

Andrew W. Appel, Jens Palsberg: Modern Compiler Implementation in Java, 2nd edition. 2002

Alfred V. Aho, Ravi Sethi, Jeffrey D. Ullman, Monica S. Lam: Compilers: Principles, Techniques, and Tools, 2nd edition. 2006

generalised parsing

Eelco Visser: Syntax Definition for Language Prototyping. PhD thesis 1997

M.G.J. van den Brand, J. Scheerder, J.J. Vinju, and E. Visser: Disambiguation Filters for Scannerless Generalized LR Parsers. CC 2002

Lennart C. L. Kats, Eelco Visser, Guido Wachsmuth: Pure and Declarative Syntax Definition - Paradise Lost and Regained. SPLASH 2010

Outlook
coming next

lectures

- guest lecture: DSLs

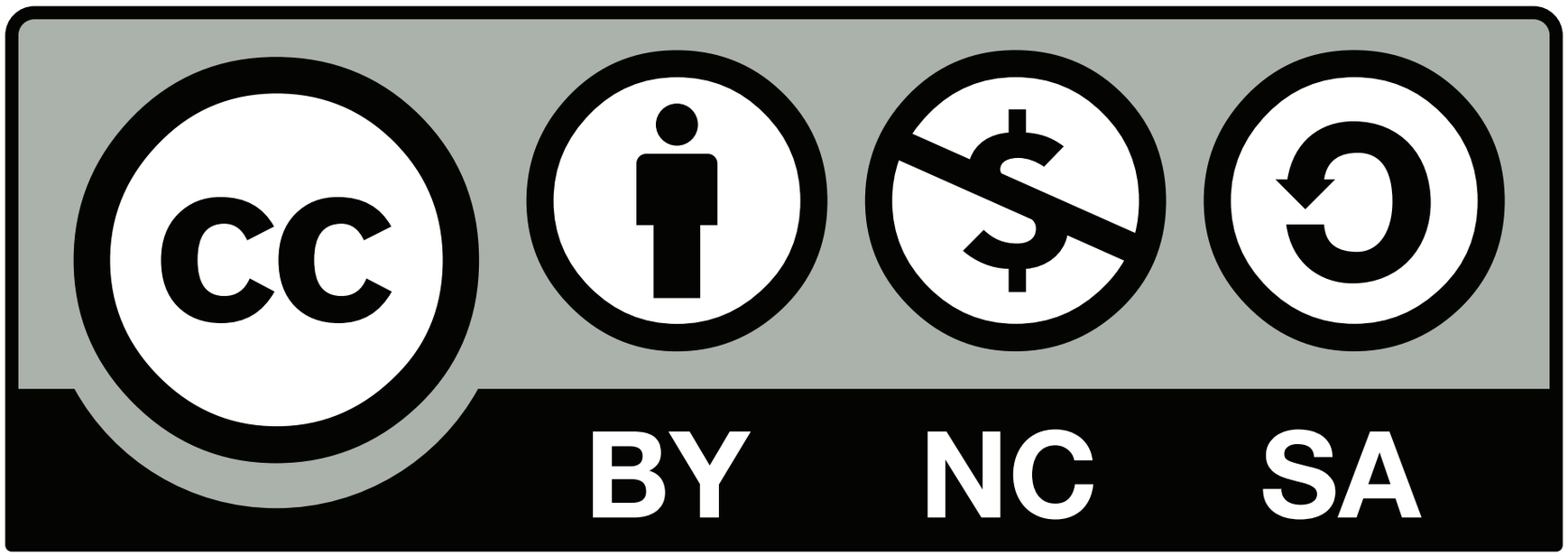
Question & Answer Jan 08

- 10 questions, submit & vote

Lab Dec 14

- translate fields & variables
- challenge: variable ranges

copyrights


BY
NC
SA

Pictures

copyrights

Slide 1:

Book Scanner by Ben Woosley, some rights reserved

Slide 19:

Ostsee by Mario Thiel, some rights reserved