Compiler Construction

LR

Rina Zviel-Girshin and Ohad Shacham
School of Computer Science
Tel-Aviv University

Administration

- PA1 grades are available
- Theoretical assignment regarding LR parsing
- Submit to Paz Grimberg's box
- Next recitation
  - Tomorrow at 9:00 and 13:00
  - Friday at 10:00

LR parsing

- LR(0)
  - Building an LR(0) parser
  - Parse an input
  - Is a grammar LR(0)?
- SLR(1)
  - Augmenting LR(0) parser to SLR(1)
  - Is a grammar SLR(1)?
- LR(1) – canonical LR
- LALR(1)

LR(0) parsing

1. Construct transition relation between states
   - Use algorithms **Initial item set** and **Next item set**
   - States are set of LR(0) items
     - Shift items of the form $P \rightarrow \alpha \bullet S \beta$
     - Reduce items of the form $P \rightarrow \alpha \bullet$
2. Construct parsing table
   - If every state contains no conflicts use LR(0) parsing algorithm
   - If states contain conflict
     - Rewrite grammar or
     - Use stronger parsing technique

LR(0) Example

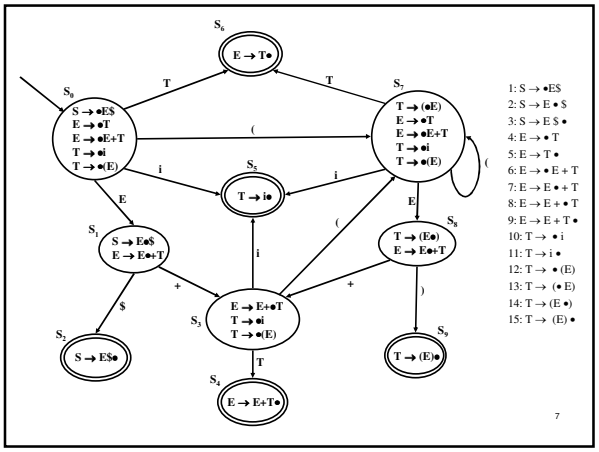
$S \rightarrow E\$$
$E \rightarrow T$
$E \rightarrow E + T$
$T \rightarrow i$
$T \rightarrow (E)$

- non-terminals denoted by upper-case letters
- terminals denoted by lower-case letters

LR(0) Example

Precomputed LR(0) items:

1: $S \rightarrow \bullet E\$$
2: $S \rightarrow E \bullet \$$
3: $S \rightarrow E \$ \bullet$
4: $E \rightarrow \bullet T$
5: $E \rightarrow T \bullet$
6: $E \rightarrow \bullet E + T$
7: $E \rightarrow E \bullet + T$
8: $E \rightarrow E + \bullet T$
9: $E \rightarrow E + T \bullet$
10: $T \rightarrow \bullet i$
11: $T \rightarrow i \bullet$
12: $T \rightarrow \bullet (E)$
13: $T \rightarrow (\bullet E)$
14: $T \rightarrow (E \bullet)$
15: $T \rightarrow (E) \bullet$



GOTO table symbol

ACTION table

state	i	+	(	)	$	E	T	
0	5		7			1	6	shift
1		3			2			shift
2	S→E$							reduce
3	5		7				4	shift
4	E→E+T							reduce
5	T→i							reduce
6	E→T							reduce
7	5		7			8	6	shift
8		3		9				shift
9	T→(E)							reduce

LR(0) example: i + (i + i)

state	i	+	(	)	$	E	T	
0	5		7			1	6	shift
1		3			2			shift
2	S→E$							reduce
3	5		7				4	shift
4	E→E+T							reduce
5	T→i							reduce
6	E→T							reduce
7	5		7			8	6	shift
8		3		9				shift
9	T→(E)							reduce

Stack	Input	Action
S_0	i + (i + i) $	shift
S_0 i S_5	+ (i + i) $	reduce by T→i
S_0 T S_6	+ (i + i) $	reduce by E→T
S_0 E S_1	+ (i + i) $	shift
S_0 E S_1 + S_3	(i + i) $	shift

LR(0) example: i + (i + i)

state	i	+	(	)	$	E	T	
0	5		7			1	6	shift
1		3			2			shift
2	S→E$							reduce
3	5		7				4	shift
4	E→E+T							reduce
5	T→i							reduce
6	E→T							reduce
7	5		7			8	6	shift
8		3		9				shift
9	T→(E)							reduce

Stack	Input	Action
S_0 E S_1 + S_3	(i + i) $	shift
S_0 E S_1 + S_3 (S_7	i + i) $	shift
S_0 E S_1 + S_3 (S_7i S_5	+ i)$	reduce by T→i
S_0 E S_1 + S_3 (S_7T S_6	+ i)$	reduce by E→T
S_0 E S_1 + S_3 (S_7E S_8	+ i)$	shift

LR(0) example: i + (i + i)

state	i	+	(	)	$	E	T	
0	5		7			1	6	shift
1		3			2			shift
2	S→E$							reduce
3	5		7				4	shift
4	E→E+T							reduce
5	T→i							reduce
6	E→T							reduce
7	5		7			8	6	shift
8		3		9				shift
9	T→(E)							reduce

Stack	Input	Action
S_0 E S_1 + S_3 (S_7E S_8	+ i)$	shift
S_0 E S_1 + S_3 (S_7E S_8+S_3	i)$	shift
S_0 E S_1 + S_3 (S_7E S_8+S_3i S_5	)$	reduce by T→i
S_0 E S_1 + S_3 (S_7E S_8+S_3T S_4	)$	reduce by E→E+T
S_0 E S_1 + S_3 (S_7E S_8	)$	shift

LR(0) example: i + (i + i)

state	i	+	(	)	$	E	T	
0	5		7			1	6	shift
1		3			2			shift
2	S→E$							reduce
3	5		7				4	shift
4	E→E+T							reduce
5	T→i							reduce
6	E→T							reduce
7	5		7			8	6	shift
8		3		9				shift
9	T→(E)							reduce

Stack	Input	Action
S_0 E S_1 + S_3 (S_7E S_8	)$	shift
S_0 E S_1 + S_3 (S_7E S_8)S_9	$	reduce by T→(E)
S_0 E S_1 + S_3T S_4	$	reduce by E→E+T
S_0 E S_1	$	shift
S_0 E $S_1$$S_2$		reduce by S→E$
S_0 S		accept

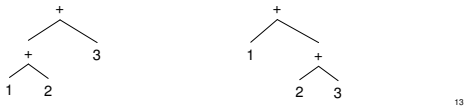
Example

$E \rightarrow E + E$
$E \rightarrow i$
$E \rightarrow (E)$

Ambiguous

Is the grammar LR(0) ?

1 + 2 + 3



13

Example

$E \rightarrow E + T$
$E \rightarrow (E)$
$E \rightarrow T$
$T \rightarrow i$
$E \rightarrow V = E$
$V \rightarrow i$

Reduce – Reduce conflict

Is the grammar LR(0) ?

$T \rightarrow i$
$V \rightarrow i$

14

Example

$E \rightarrow E + T$
$E \rightarrow (E)$
$T \rightarrow i[E]$
$T \rightarrow i$

Shift – Reduce conflict

Is the grammar LR(0) ?

$T \rightarrow i[E]$
$T \rightarrow i$

15

Example

$E \rightarrow E + E$
$\quad | E * E$
$\quad | num$

Is the grammar LR(0) ?

1 + 2 + 3

16

Example

E

Is the grammar LR(0) ?

$E \rightarrow E + T$
$\quad | T$

$T \rightarrow T * F$
$\quad | F$

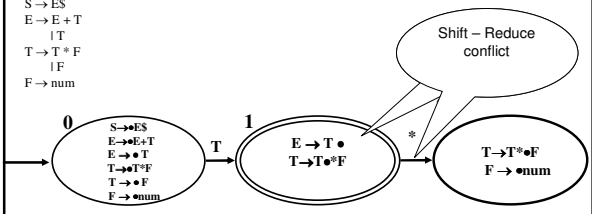
$F \rightarrow num$
$\quad | id$

17

Example

$S \rightarrow E\$$
$E \rightarrow E + T$
$\quad | T$
$T \rightarrow T * F$
$\quad | F$
$F \rightarrow num$

Shift – Reduce conflict



18

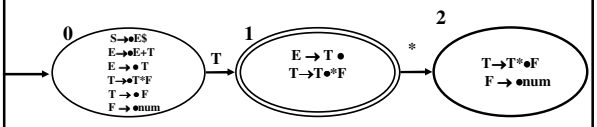
SLR(1)

- Simple LR(1) parser
- LR(0) + use of FOLLOW sets of non terminals
- FOLLOW(A) = {t | S $\Rightarrow^*$ βAtγ}
- Calculate FOLLOW(A) for each item A $\rightarrow\alpha$ •
- Use these sets to break conflicts

19

SLR(1) Example

S → E$
E → E + T
| T
T → T * F
| F
F → num



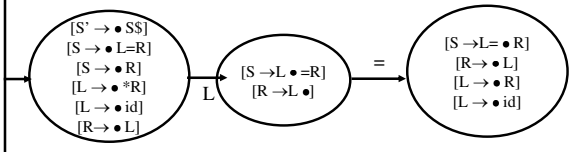
FOLLOW(E) = {$,+}

20

Example

S' → S$
S → L = R | R
L → *R | id
R → L

Is the grammar SLR(1) ?



Follow(R)= {$, =}

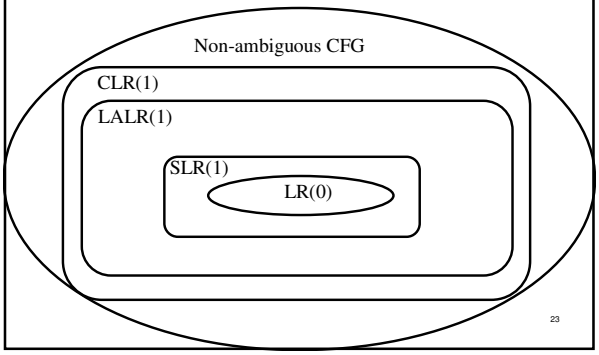
21

LR

- LR(0)
- Simple LR – LR(0) + Follow sets
- LR(1) – LR(0) + lookhead
  - Large parsing table
- LALR(1)
  - Merge LR(1) states with same "LR(0) items"
  - May have reduce – reduce conflict on LR(1) grammar
  - Most common in practic

22

Grammar Hierarchy



23