

Concept of Shift-reduce Parsing

- Bottom-up parsing is also known as **shift-reduce parsing** because its two main actions are shift and reduce.
- Data structures: input-string and stack
- Operations
 - At each **shift action**, the current symbol in the input string is pushed to a stack.
 - At each **reduction step**, the symbols at the top of the stack (this symbol sequence is the right side of a production) will be replaced by the non-terminal at the left side of that production.
 - **Accept**: Announce successful completion of parsing
 - **Error**: Discover a syntax error and call error recovery

Example 01

Grammar:

$E \rightarrow E + T \mid T$

$T \rightarrow T * F \mid F$

$F \rightarrow (E) \mid \text{id}$

Perform Shift Reduce Parsing for the string $\text{id1} * \text{id2}$

Grammar:

$E \rightarrow E + T \mid T$

$T \rightarrow T * F \mid F$

$F \rightarrow (E) \mid \text{id}$

String: id1 * id2

Stack	Input	Action
\$	id1 * id2 \$	Shift

Grammar:
E -> E + T | T
T -> T * F | F
F -> (E) | id

String: id1 * id2

Stack	Input	Action
\$	id1 * id2 \$	Shift
\$id1	*id2\$	Reduce F->id

Grammar:
E -> E + T | T
T -> T * F | F
F -> (E) | id

String: id1 * id2

Stack	Input	Action
\$	id1 * id2 \$	Shift
\$id1	*id2\$	Reduce F->id
\$F	*id2\$	Reduce T->F

Grammar:
E -> E + T | T
T -> T * F | F
F -> (E) | id

String: id1 * id2

Stack	Input	Action
\$	id1 * id2 \$	Shift
\$id1	*id2\$	Reduce F->id
\$F	*id2\$	Reduce T->F
\$ T	*id2\$	Shift

Grammar:

$E \rightarrow E + T \mid T$
 $T \rightarrow T * F \mid F$
 $F \rightarrow (E) \mid id$

String: id1 * id2

Stack	Input	Action
\$	id1 * id2 \$	Shift
\$id1	*id2\$	Reduce F->id
\$F	*id2\$	Reduce T->F
\$T	*id2\$	Shift
\$T *	id2\$	Shift

Grammar:

$E \rightarrow E + T \mid T$
 $T \rightarrow T * F \mid F$
 $F \rightarrow (E) \mid id$

String: id1 * id2

Stack	Input	Action
\$	id1 * id2 \$	Shift
\$id1	*id2\$	Reduce F->id
\$F	*id2\$	Reduce T->F
\$T	*id2\$	Shift
\$T *	id2\$	Shift
\$T*id	\$	Reduce F->id

Grammar:
$$E \rightarrow E + T \mid T$$
$$T \rightarrow T * F \mid F$$
$$F \rightarrow (E) \mid \text{id}$$
String: id1 * id2

Stack	Input	Action
\$	id1 * id2 \$	Shift
\$id1	*id2\$	Reduce $F \rightarrow \text{id}$
\$F	*id2\$	Reduce $T \rightarrow F$
\$T	*id2\$	Shift
\$T *	id2\$	Shift
\$T*id	\$	Reduce $F \rightarrow \text{id}$
\$T* F	\$	Reduce $T \rightarrow T * F$

Grammar:
$$E \rightarrow E + T \mid T$$
$$T \rightarrow T * F \mid F$$
$$F \rightarrow (E) \mid \text{id}$$
String: id1 * id2

Stack	Input	Action
\$	id1 * id2 \$	Shift
\$id1	*id2\$	Reduce $F \rightarrow \text{id}$
\$F	*id2\$	Reduce $T \rightarrow F$
\$T	*id2\$	Shift
\$T *	id2\$	Shift
\$T*id	\$	Reduce $F \rightarrow \text{id}$
\$T*F	\$	Reduce $T \rightarrow T*F$
\$ T	\$	Reduce $E \rightarrow T$

Grammar:
$$E \rightarrow E + T \mid T$$
$$T \rightarrow T * F \mid F$$
$$F \rightarrow (E) \mid \text{id}$$
String: id1 * id2

Stack	Input	Action
\$	id1 * id2 \$	Shift
\$id1	*id2\$	Reduce $F \rightarrow \text{id}$
\$F	*id2\$	Reduce $T \rightarrow F$
\$T	*id2\$	Shift
\$T *	id2\$	Shift
\$T*id	\$	Reduce $F \rightarrow \text{id}$
\$T*F	\$	Reduce $T \rightarrow T*F$
\$T	\$	Reduce $E \rightarrow T$
\$ E	\$	Accept

Example 02

Grammar:

$$E \rightarrow E + T \mid T$$
$$T \rightarrow T * F \mid F$$
$$F \rightarrow (E) \mid \text{id}$$

Perform Shift Reduce Parsing for the string (id)+id

Grammar:

$$E \rightarrow E + T \mid T$$
$$T \rightarrow T * F \mid F$$
$$F \rightarrow (E) \mid id$$

String: (id)+id

Stack	Input	Action
\$	(id)+id	Shift

Prepared By: Sukarna Sarker

Grammar:

$$E \rightarrow E + T \mid T$$
$$T \rightarrow T * F \mid F$$
$$F \rightarrow (E) \mid id$$

String: (id)+id

Stack	Input	Action
\$	(id)+id	Shift
\$((id)+id	Shift
	Prepared By: Sukarna Sarker	

Grammar:

$$E \rightarrow E + T \mid T$$
$$T \rightarrow T * F \mid F$$
$$F \rightarrow (E) \mid \text{id}$$

String: (id)+id

[illegible]

Grammar:

$$E \rightarrow E + T \mid T$$
$$T \rightarrow T * F \mid F$$
$$F \rightarrow (E) \mid \text{id}$$

String: (id)+id

Stack	Input	Action
\$	(id)+id\$	Shift
\$(id)+id\$	Shift
\$(id) + id \$	Reduce F->id
\$(F) + id \$	Reduce T->F
	Prepared By: Sukarna Sarker	

Grammar:

$$E \rightarrow E + T \mid T$$
$$T \rightarrow T * F \mid F$$
$$F \rightarrow (E) \mid \text{id}$$

String: (id)+id

Stack	Input	Action
\$	(id)+id\$	Shift
\$(id)+id\$	Shift
\$(id)+id\$	Reduce F->id
\$(F)+id\$	Reduce T->F
\$(T)+id\$	Reduce E->T

Prepared By: Sukarna Sarker

Grammar:

$$E \rightarrow E + T \mid T$$
$$T \rightarrow T * F \mid F$$
$$F \rightarrow (E) \mid \text{id}$$

String: (id)+id

Stack	Input	Action
\$	(id)+id\$	Shift
\$(id)+id\$	Shift
\$(id)+id\$	Reduce F->id
\$(F)+id\$	Reduce T->F
\$(T)+id\$	Reduce E->T
\$(E)+id\$	Shift

Prepared By: Sukarna Sarker

Grammar:

$$E \rightarrow E + T \mid T$$
$$T \rightarrow T * F \mid F$$
$$F \rightarrow (E) \mid \text{id}$$

String: (id)+id

Stack	Input	Action
\$	(id)+id\$	Shift
\$(id)+id\$	Shift
\$(id)+id\$	Reduce F->id
\$(F)+id\$	Reduce T->F
\$(T)+id\$	Reduce E->T
\$(E)+id\$	Shift
\$(E)	+id\$	Reduce F->(E)
	Prepared By: Sukarna Sarker	

Grammar:

$$E \rightarrow E + T \mid T$$
$$T \rightarrow T * F \mid F$$
$$F \rightarrow (E) \mid \text{id}$$

String: (id)+id

Stack	Input	Action
\$	(id)+id\$	Shift
\$(id)+id\$	Shift
\$(id)+id\$	Reduce F->id
\$(F)+id\$	Reduce T->F
\$(T)+id\$	Reduce E->T
\$(E)+id\$	Shift
\$(E)	+id\$	Reduce F->(E)
\$F	+id\$	Reduce T->F
	Prepared By: Sukarna Sarker	

Grammar:
E -> E + T | T
T -> T * F | F
F -> (E) | id

String: (id)+id

Stack	Input	Action
\$	(id)+id\$	Shift
\$(id)+id\$	Shift
\$(id)+id\$	Reduce F->id
\$(F)+id\$	Reduce T->F
\$(T)+id\$	Reduce E->T
\$(E)+id\$	Shift
\$(E)	+id\$	Reduce F->(E)
\$F	+id\$	Reduce T->F
\$T	+id\$	Reduce E->T
	Prepared By: Sukarna Sarker	

Grammar:
E -> E + T | T
T -> T * F | F
F -> (E) | id

String: (id)+id

Stack	Input	Action
\$	(id)+id\$	Shift
\$(id)+id\$	Shift
\$(id) +id\$	Reduce F->id
\$(F) +id\$	Reduce T->F
\$(T) +id\$	Reduce E->T
\$(E) +id\$	Shift
\$(E)	+id\$	Reduce F->(E)
\$F	+id\$	Reduce T->F
\$T	+id\$	Reduce E->T
\$E	+id\$	Shift
	Prepared By: Sukarna Sarker	

Grammar:
E -> E + T | T
T -> T * F | F
F -> (E) | id

String: (id)+id

Stack	Input	Action
\$	(id)+id\$	Shift
\$(id)+id\$	Shift
\$(id) +id\$	Reduce F->id
\$(F) +id\$	Reduce T->F
\$(T) +id\$	Reduce E->T
\$(E) +id\$	Shift
\$(E)	+id\$	Reduce F->(E)
\$F	+id\$	Reduce T->F
\$T	+id\$	Reduce E->T
\$E	+id\$	Shift
\$E+	id\$	Shift
	Prepared By: Sukarna Sarker	

Grammar:
E -> E + T | T
T -> T * F | F
F -> (E) | id

String: (id)+id

Stack	Input	Action
\$	(id)+id\$	Shift
\$(id)+id\$	Shift
\$(id) +id\$	Reduce F->id
\$(F) +id\$	Reduce T->F
\$(T) +id\$	Reduce E->T
\$(E) +id\$	Shift
\$(E)	+id\$	Reduce F->(E)
\$F	+id\$	Reduce T->F
\$T	+id\$	Reduce E->T
\$E	+id\$	Shift
\$E+	id\$	Shift
\$E+id	\$	Reduce F->id
	Prepared By: Sukarna Sarker	

Grammar:

$E \rightarrow E + T \mid T$
 $T \rightarrow T * F \mid F$
 $F \rightarrow (E) \mid id$

String: (id)+id

Stack	Input	Action
\$	(id)+id\$	Shift
\$(id)+id\$	Shift
\$(id) +id\$	Reduce F->id
\$(F) +id\$	Reduce T->F
\$(T) +id\$	Reduce E->T
\$(E) +id\$	Shift
\$(E)	+id\$	Reduce F->(E)
\$F	+id\$	Reduce T->F
\$T	+id\$	Reduce E->T
\$E	+id\$	Shift
\$E+	id\$	Shift
\$E+id	\$	Reduce F->id
\$E+F	\$	Reduce T->F
	Prepared By: Sukarna Sarker	

Grammar:

$E \rightarrow E + T \mid T$
 $T \rightarrow T * F \mid F$
 $F \rightarrow (E) \mid id$

String: (id)+id

Stack	Input	Action
\$	(id)+id\$	Shift
\$(id)+id\$	Shift
\$(id) +id\$	Reduce F->id
\$(F) +id\$	Reduce T->F
\$(T) +id\$	Reduce E->T
\$(E) +id\$	Shift
\$(E)	+id\$	Reduce F->(E)
\$F	+id\$	Reduce T->F
\$T	+id\$	Reduce E->T
\$E	+id\$	Shift
\$E+	id\$	Shift
\$E+id	\$	Reduce F->id
\$E+F	\$	Reduce T->F
\$E+T	\$	Reduce E->E+T

Grammar:

$E \rightarrow E + T \mid T$
 $T \rightarrow T * F \mid F$
 $F \rightarrow (E) \mid id$

String: (id)+id

Stack	Input	Action
\$	(id)+id\$	Shift
\$(id)+id\$	Shift
\$(id)+id\$	Reduce F->id
\$(F)+id\$	Reduce T->F
\$(T)+id\$	Reduce E->T
\$(E)+id\$	Shift
\$(E)	+id\$	Reduce F->(E)
\$F	+id\$	Reduce T->F
\$T	+id\$	Reduce E->T
\$E	+id\$	Shift
\$E+	id\$	Shift
\$E+id	\$	Reduce F->id
\$E+F	\$	Reduce T->F
\$E+T	\$	Reduce E->E+T
\$E	\$	Accept