

**1. Copyright.**

Copyright © Dave Bone 1998 - 2015

**2. *la\_expr\_src* Thread.**

Extract Grammar's lookahead expression chopped up into strings, literals, identifiers, operators, lr k constants, and keywords to be placed into a *tok\_can* container for post compilation. Apart from the keywords and lr k constants, the tokens are processed before the definition of a grammar's vocabulary — Terminals and Rules. It is only after the grammar is completely parsed that these tokens can be verified against the symbol table.

The parsed expression is placed into a token container of *T\_la\_expr\_src* by *parallel\_parser\_phrase\_th* for post evaluation by *la\_expr\_lexical* and *la\_expr* grammars.

**3. Fsm *Cla\_expr\_src* class.****4. *Cla\_expr\_src* constructor directive.**

⟨*Cla\_expr\_src* constructor directive 4⟩ ≡  
`exp_ = 0;`

**5. *Cla\_expr\_src* op directive.**

⟨*Cla\_expr\_src* op directive 5⟩ ≡  
`exp_ = new T_la_expr_src;  
 exp_>set_rc(*parser->start_token_, __FILE__, __LINE__);  
 parser->set_use_all_shift_on();`

**6. *Cla\_expr\_src* user-declaration directive.**

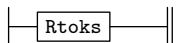
⟨*Cla\_expr\_src* user-declaration directive 6⟩ ≡  
**public:** *T\_la\_expr\_src* \* *exp\_;*

**7. *Cla\_expr\_src* user-prefix-declaration directive.**

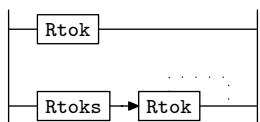
⟨*Cla\_expr\_src* user-prefix-declaration directive 7⟩ ≡  
`#include "c_string.h"  
#include "identifier.h"  
#include "c_literal.h"  
#include "lint_balls.h"  
#include "o2_code_end.h"  
#include "la_lrk_T.h"  
#include "yacco2_stbl.h"  
using namespace NS_yacco2_terminals;`

**8. *Rla\_expr\_src* rule.**

*Rla\_expr\_src*

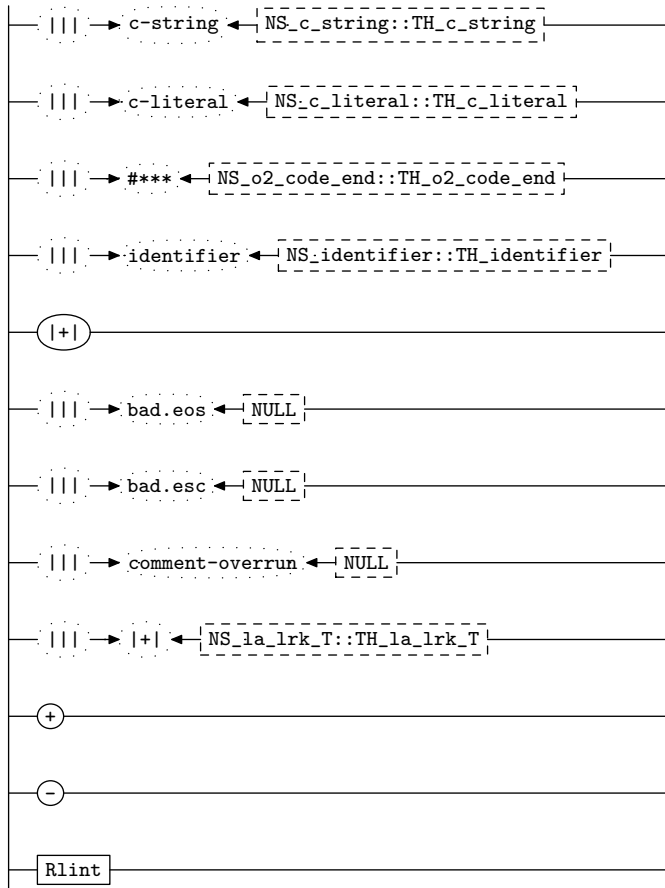
**9. *Rtoks* rule.**

*Rtoks*

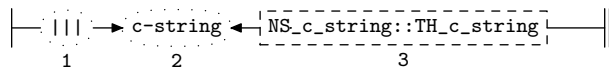


10. *Rtok* rule.

Rtok

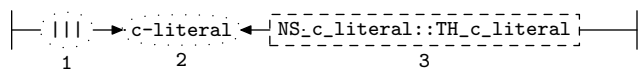


11. *Rtok*'s subrule 1.



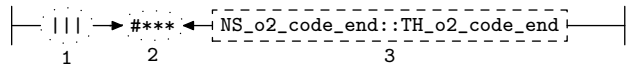
⟨Rtok subrule 1 op directive 11⟩ ≡  
*Cl*<sub>a</sub>*expr*<sub>src</sub> \* *fsm* = ( *Cl*<sub>a</sub>*expr*<sub>src</sub> \* ) *rule\_info*<sub>...</sub>*parser*<sub>...</sub>*fsm*<sub>tbl</sub><sub>...</sub>;  
*fsm*<sub>...</sub>*exp*<sub>...</sub>*la\_tok*<sub>can</sub>(*...*)*push\_back*(\**sf*<sub>...</sub>*p2*<sub>...</sub>);

12. *Rtok*'s subrule 2.



⟨Rtok subrule 2 op directive 12⟩ ≡  
*Cl*<sub>a</sub>*expr*<sub>src</sub> \* *fsm* = ( *Cl*<sub>a</sub>*expr*<sub>src</sub> \* ) *rule\_info*<sub>...</sub>*parser*<sub>...</sub>*fsm*<sub>tbl</sub><sub>...</sub>;  
*fsm*<sub>...</sub>*exp*<sub>...</sub>*la\_tok*<sub>can</sub>(*...*)*push\_back*(\**sf*<sub>...</sub>*p2*<sub>...</sub>);

## 13. Rtok's subrule 3.



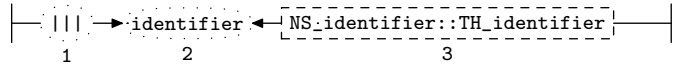
⟨Rtok subrule 3 op directive 13⟩ ≡

```

Cla_expr_src * fsm = ( Cla_expr_src * ) rule_info...parser--fsm_tbl_;
T_eocode * k = sf-p2--;
k->set_auto_delete(true);
rule_info...parser--set_use_all_shift_off();
fsm->exp->la_tok_can()->push_back(*yacco2::PTR_LR1_eog--);
fsm->exp->la_tok_can()->push_back(*yacco2::PTR_LR1_eog--);
RSVP(fsm->exp-);
fsm->exp_ = 0;
rule_info...parser--set_stop_parse(true);

```

## 14. Rtok's subrule 4.



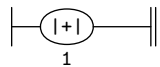
⟨Rtok subrule 4 op directive 14⟩ ≡

```

Cla_expr_src * fsm = ( Cla_expr_src * ) rule_info...parser--fsm_tbl_;
fsm->exp->la_tok_can()->push_back(*sf-p2--);

```

## 15. Rtok's subrule 5.



Watch out for overrun.

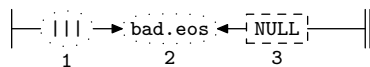
⟨Rtok subrule 5 op directive 15⟩ ≡

```

Cla_expr_src * fsm = ( Cla_expr_src * ) rule_info...parser--fsm_tbl_;
CAbs_lr1_sym * la_sym = sf-p1--;
using namespace NS_yacco2_T_enum;
using namespace NS_yacco2_terminals;
int id = la_sym->enumerated_id--;
if (id ≡ T_Enum::T_LR1_eog-) {
    CAbs_lr1_sym * sym = new Err_no_end_of_code;
    sym->set_rc(*la_sym, __FILE__, __LINE__);
    RSVP(sym);
    rule_info...parser--set_stop_parse(true);
    return;
}
fsm->exp->la_tok_can()->push_back(*la_sym);

```

## 16. Rtok's subrule 6.



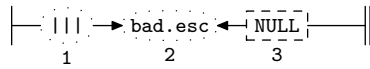
⟨Rtok subrule 6 op directive 16⟩ ≡

```

RSVP(sf-p2--);
rule_info...parser--set_stop_parse(true);

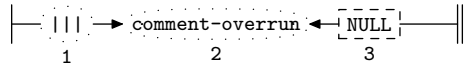
```

**17. Rtok's subrule 7.**



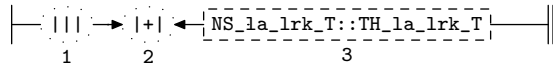
⟨Rtok subrule 7 op directive 17⟩ ≡  
 RSVP(sf-p2\_);  
 rule\_info...parser--set\_stop\_parse(true);

**18. Rtok's subrule 8.**



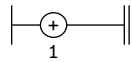
⟨Rtok subrule 8 op directive 18⟩ ≡  
 RSVP(sf-p2\_);  
 rule\_info...parser--set\_stop\_parse(true);

**19. Rtok's subrule 9.**



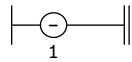
⟨Rtok subrule 9 op directive 19⟩ ≡  
 Cla\_expr\_src \* fsm = ( Cla\_expr\_src \* ) rule\_info...parser--fsm\_tbl\_;  
 fsm-exp-la\_tok\_can()-push\_back(\*sf-p2\_);

**20. Rtok's subrule 10.**



⟨Rtok subrule 10 op directive 20⟩ ≡  
 Cla\_expr\_src \* fsm = ( Cla\_expr\_src \* ) rule\_info...parser--fsm\_tbl\_;  
 fsm-exp-la\_tok\_can()-push\_back(\*sf-p1\_);

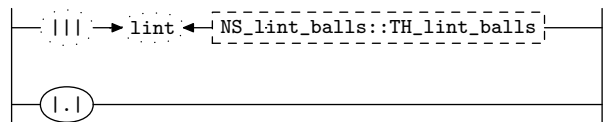
**21. Rtok's subrule 11.**



⟨Rtok subrule 11 op directive 21⟩ ≡  
 Cla\_expr\_src \* fsm = ( Cla\_expr\_src \* ) rule\_info...parser--fsm\_tbl\_;  
 fsm-exp-la\_tok\_can()-push\_back(\*sf-p1\_);

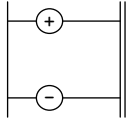
**22. Rlint rule.**

Rlint



**23.** *Rla\_minus* rule.

*Rla\_minus*



24. First Set Language for  $O_2^{linker}$ .

```

/*
  File: la_expr_src.fsc
  Date and Time: Fri Jan  2 15:33:40 2015
*/
transitive      y
grammar-name    "la_expr_src"
name-space     "NS_la_expr_src"
thread-name    "TH_la_expr_src"
monolithic     n
file-name      "la_expr_src.fsc"
no-of-T        569
list-of-native-first-set-terminals 4
  LR1_fset_transience_operator
  LR1_all_shift_operator
  raw_plus
  raw_minus
end-list-of-native-first-set-terminals
list-of-transitive-threads 6
  NS_c_literal::TH_c_literal
  NS_lint_balls::TH_lint_balls
  NS_c_string::TH_c_string
  NS_o2_code_end::TH_o2_code_end
  NS_la_lrk_T::TH_la_lrk_T
  NS_identifier::TH_identifier
end-list-of-transitive-threads
list-of-used-threads 6
  NS_c_literal::TH_c_literal
  NS_c_string::TH_c_string
  NS_identifier::TH_identifier
  NS_la_lrk_T::TH_la_lrk_T
  NS_lint_balls::TH_lint_balls
  NS_o2_code_end::TH_o2_code_end
end-list-of-used-threads
fsm-comments
"Lexer: 1st stage of Lookahead source expression\n converted into raw character tokens
for post evaluation."

```





t Rtok		3	3	3				1	0	7	1
$\Rightarrow$ lint					State: 8 state type: <i>r</i>						
← rule	→ R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA	
t Rlint		4	1	3			1	0	8	1	
$\Rightarrow$ badeos					State: 9 state type: <i>r</i>						
← rule	→ R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA	
t Rtok		3	6	3			1	0	9	1	
$\Rightarrow$ badesc					State: 10 state type: <i>r</i>						
← rule	→ R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA	
t Rtok		3	7	3			1	0	10	1	
$\Rightarrow$ comment-overrun					State: 11 state type: <i>r</i>						
← rule	→ R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA	
t Rtok		3	8	3			1	0	11	1	
$\Rightarrow$  .					State: 12 state type: <i>r</i>						
← rule	→ R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA	
t Rlint		4	2	2			1	0	12	1	
$\Rightarrow$  +					State: 13 state type: <i>r</i>						
← rule	→ R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA	
t Rtok		3	5	2			1	0	13	1	
$\Rightarrow$ +					State: 14 state type: <i>r</i>						
← rule	→ R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA	
t Rtok		3	10	2			1	0	14	1	
$\Rightarrow$ -					State: 15 state type: <i>r</i>						
← rule	→ R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA	
t Rtok		3	11	2			1	0	15	1	
$\Rightarrow$ Rtoks					State: 16 state type: <i>s/r</i>						
← rule	→ R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA	
t Rla_expr_src		1	1	2			1	0	16	2	
c Rtok		3	2	1	c-literal NS_c_literal::TH_c_literal		16	2	4		
c Rtok		3	6	1	bad eos NULL		16	2	9		
c Rtok		3	7	1	bad esc NULL		16	2	10		
c Rtok		3	9	1	+ NS_la_lrk_T::TH_la_lrk_T		16	2	3		
c Rtok		3	1	1	c-string NS_c_string::TH_c_string		16	2	5		
c Rtok		3	3	1	# *** NS_o2_code_end::TH_o2_code_end		16	2	7		
c Rtok		3	4	1	identifier NS_identifier::TH_identifier		16	2	6		
c Rtok		3	8	1	comment-overrun NULL		16	2	11		
c Rlint		4	1	1	lint NS_lint_balls::TH_lint_balls		16	2	8		
c Rlint		4	2	1	.		16	12	12		
c Rtok		3	5	1	+		16	13	13		
c Rtok		3	10	1	+		16	14	14		
c Rtok		3	11	1	-		16	15	15		
t Rtoks		2	2	2	Rtok		1	17	17		
c Rtok		3	12	1	Rlint		16	18	18		

$\Rightarrow$ <i>Rtok</i>					State: 17 state type: <i>r</i>		
←	rule	→	R#	sr# Po	←	subrule element	→ Brn Gto Red LA
t	Rtoks		2	2 3			1 0 17 1
$\Rightarrow$ <i>Rlint</i>					State: 18 state type: <i>r</i>		
←	rule	→	R#	sr# Po	←	subrule element	→ Brn Gto Red LA
t	Rtok		3	12 2			16 0 18 1
$\Rightarrow$ <i>Rtok</i>					State: 19 state type: <i>r</i>		
←	rule	→	R#	sr# Po	←	subrule element	→ Brn Gto Red LA
t	Rtoks		2	1 2			1 0 19 1

**26. Index.**

|+|: 10.  
 |.|: 22.  
 # \*\*\*: 10.  
 |||: 10, 22.  
 \_\_FILE\_\_: 5, 15.  
 \_\_LINE\_\_: 5, 15.  
 bad eos: 10.  
 bad esc: 10.  
 c-literal: 10.  
 c-string: 10.  
 CAbs\_lr1\_sym: 15.  
 Cla\_expr\_src: 11, 12, 13, 14, 15, 19, 20, 21.  
 comment-overrun: 10.  
 enumerated\_id\_: 15.  
 Err\_no\_end\_of\_code: 15.  
 exp: 4, 5, 6, 11, 12, 13, 14, 15, 19, 20, 21.  
 fsm: 11, 12, 13, 14, 15, 19, 20, 21.  
 fsm\_tbl\_: 11, 12, 13, 14, 15, 19, 20, 21.  
 id: 15.  
 identifier: 10.  
 la\_expr: 2.  
 la\_expr\_lexical: 2.  
 la\_expr\_src: 2.  
 la\_sym: 15.  
 la\_tok\_can: 11, 12, 13, 14, 15, 19, 20, 21.  
 lint: 22.  
 NS\_c\_literal::TH\_c\_literal: 10.  
 NS\_c\_string::TH\_c\_string: 10.  
 NS\_identifier::TH\_identifier: 10.  
 NS\_la\_lrk\_T::TH\_la\_lrk\_T: 10.  
 NS\_lint\_balls::TH\_lint\_balls: 22.  
 NS\_o2\_code\_end::TH\_o2\_code\_end: 10.  
 NS\_yacco2\_T\_enum: 15.  
 NS\_yacco2\_terminals: 7, 15.  
 NULL: 10.  
 parallel\_parser\_phrase\_th: 2.  
 parser\_: 5, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21.  
 PTR\_LR1\_eog\_: 13.  
 push\_back: 11, 12, 13, 14, 15, 19, 20, 21.  
 p1\_: 15, 20, 21.  
 p2\_: 11, 12, 13, 14, 16, 17, 18, 19.  
 Rla\_expr\_src: 8.  
 Rla\_minus: 23.  
 Rlint: 22.  
 Rlint: 10.  
 RSVP: 13, 15, 16, 17, 18.  
 Rtok: 9.  
 Rtok: 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21.  
 Rtoks: 9.  
 Rtoks: 8, 9.  
 rule\_info\_: 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21.  
 set\_auto\_delete: 13.  
 set\_rc: 5, 15.  
 set\_stop\_parse: 13, 15, 16, 17, 18.  
 set\_use\_all\_shift\_off: 13.  
 set\_use\_all\_shift\_on: 5.  
 sf: 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21.  
 start\_token\_: 5.  
 sym: 15.  
 T\_Enum: 15.  
 T\_eocode: 13.  
 T\_la\_expr\_src: 2, 5, 6.  
 T\_LR1\_eog\_: 15.  
 tok\_can: 2.  
 true: 13, 15, 16, 17, 18.  
 yacco2: 13.

- ⟨ Cla\_expr\_src constructor directive 4 ⟩
- ⟨ Cla\_expr\_src op directive 5 ⟩
- ⟨ Cla\_expr\_src user-declaration directive 6 ⟩
- ⟨ Cla\_expr\_src user-prefix-declaration directive 7 ⟩
- ⟨ Rtok subrule 1 op directive 11 ⟩
- ⟨ Rtok subrule 10 op directive 20 ⟩
- ⟨ Rtok subrule 11 op directive 21 ⟩
- ⟨ Rtok subrule 2 op directive 12 ⟩
- ⟨ Rtok subrule 3 op directive 13 ⟩
- ⟨ Rtok subrule 4 op directive 14 ⟩
- ⟨ Rtok subrule 5 op directive 15 ⟩
- ⟨ Rtok subrule 6 op directive 16 ⟩
- ⟨ Rtok subrule 7 op directive 17 ⟩
- ⟨ Rtok subrule 8 op directive 18 ⟩
- ⟨ Rtok subrule 9 op directive 19 ⟩

# la\_expr\_src Grammar

Date: January 2, 2015 at 15:36

File: la\_expr\_src.lex

Ns: NS\_la\_expr\_src

Version: 1.0

Debug: false

Grammar Comments:

Type: Thread

Lexer: 1st stage of Lookahead source expression converted into raw character tokens for post evaluation.

1 element(s) in Lookahead Expression below

eog

	Section	Page
<b>Copyright</b> .....	1	1
<i>la_expr_src</i> <b>Thread</b> .....	2	2
Fsm Cla_expr_src class .....	3	2
Cla_expr_src constructor directive .....	4	2
Cla_expr_src op directive .....	5	2
Cla_expr_src user-declaration directive .....	6	2
Cla_expr_src user-prefix-declaration directive .....	7	2
<i>Rla_expr_src</i> rule .....	8	2
<i>Rtoks</i> rule .....	9	2
<i>Rtok</i> rule .....	10	3
<i>Rtok</i> 's subrule 1 .....	11	3
<i>Rtok</i> 's subrule 2 .....	12	3
<i>Rtok</i> 's subrule 3 .....	13	4
<i>Rtok</i> 's subrule 4 .....	14	4
<i>Rtok</i> 's subrule 5 .....	15	4
<i>Rtok</i> 's subrule 6 .....	16	4
<i>Rtok</i> 's subrule 7 .....	17	5
<i>Rtok</i> 's subrule 8 .....	18	5
<i>Rtok</i> 's subrule 9 .....	19	5
<i>Rtok</i> 's subrule 10 .....	20	5
<i>Rtok</i> 's subrule 11 .....	21	5
<i>Rlint</i> rule .....	22	5
<i>Rla_minus</i> rule .....	23	6
<b>First Set Language for <math>O_2^{linker}</math></b> .....	24	7
<b>Lr1 State Network</b> .....	25	8
<b>Index</b> .....	26	11