1. Copyright.

2. *rc_phrase_th* **Thread.**
Parse raw-char-symbols phrase. The *Rc_phrase* terminal keeps its own local symbol table: map and list. It calls the common backend *term_def_ph* thread extractor of symbol definition.
Example of a raw chaacter vocabulary to parse:

```
/*
file: rcsym.txt
Why: raw character symbols vocabulary to parse.
Note: raw-characters is shown for clarity. It is the keyword trigger
      that calls this grammar. Parsed is comments onwards.
*/
raw-characters
// comments
(file-name o2_chars,name-space NS_o2_chars) {
// 7 bit char set
"x00"        (sym-class raw_nul{
  user-declaration
    public:
    raw_nul(INT Ext_file,size_t Pos);
  ***
  user-implementation
    raw_nul::raw_nul(INT Ext_file,size_t Pos){
      T_CTOR_RW("x00",T_raw_nul_,false,false,Ext_file,Pos)}
  ***
  }
)
              ....
"!" (sym-class raw_exclam{
  user-declaration
    public:
    raw_exclam(INT Ext_file,size_t Pos);
  ***
  user-implementation
    raw_exclam::raw_exclam(INT Ext_file,size_t Pos){
      T_CTOR_RW("!",T_raw_exclam_,false,false,Ext_file,Pos)}
  ***
  }
)
              ....
// 8 bit char set
"xff" (sym-class raw_xff{
  user-declaration
    public:
    raw_xff(INT Ext_file,size_t Pos);
  ***
  user-implementation
    raw_xff::raw_xff(INT Ext_file,size_t Pos){
      T_CTOR_RW("xff",T_raw_xff_,false,false,Ext_file,Pos)}
  ***
  }
}
```

3. Fsm Crc_phrase_th class.

4. Crc_phrase_th constructor directive.

⟨ Crc_phrase_th constructor directive 4 ⟩ ≡
```
rc_phrase_ = 0;
```

5. Crc_phrase_th op directive.

⟨ Crc_phrase_th op directive 5 ⟩ ≡
```
if (rc_phrase_ ≠ 0) {
  delete rc_phrase_;
  rc_phrase_ = 0;
}
rc_phrase_ = new T_rc_phrase;
rc_phrase_→set_rc(*parser__→start_token__, __FILE__, __LINE__);
AST * t = new AST(*rc_phrase_);
rc_phrase_→phrase_tree(t);
```

6. Crc_phrase_th user-declaration directive.

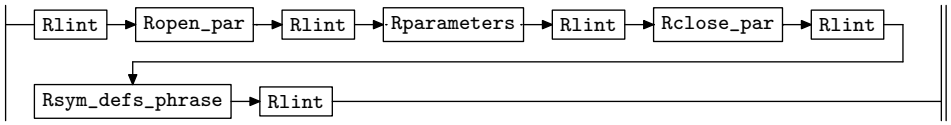
⟨ Crc_phrase_th user-declaration directive 6 ⟩ ≡
```
public: T_rc_phrase * rc_phrase_;
```

7. Crc_phrase_th user-prefix-declaration directive.

⟨ Crc_phrase_th user-prefix-declaration directive 7 ⟩ ≡
```
#include "lint_balls.h"
#include "identifier.h"
#include "term_def_ph.h"
#include "c_string.h"
```

8. *Rrc_phrase* rule.

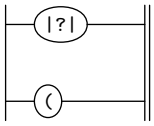
Rrc_phrase



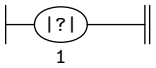
⟨ Rrc_phrase subrule 1 op directive 8 ⟩ ≡
```
Crc_phrase_th * fsm = ( Crc_phrase_th * ) rule_info__.parser__→fsm_tbl__;
RSVP(fsm→rc_phrase_);
fsm→rc_phrase_ = 0;
```

9. *Ropen_par* rule.

Ropen_par



10. *Ropen_par*'**s subrule 1.**

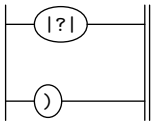


⟨ Ropen_par subrule 1 op directive 10 ⟩ ≡

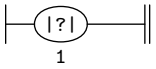
```
CAbs_lr1_sym * sym = new Err_no_open_parenthesis;
sym->set_rc(*rule_info__.parser__->current_token( ), __FILE__, __LINE__);
RSVP(sym);
rule_info__.parser__->set_stop_parse(true);
```

11. *Rclose_par* **rule.**

Rclose_par



12. *Rclose_par*'**s subrule 1.**

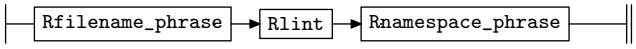


⟨ Rclose_par subrule 1 op directive 12 ⟩ ≡

```
CAbs_lr1_sym * sym = new Err_no_close_parenthesis;
sym->set_rc(*rule_info__.parser__->current_token( ), __FILE__, __LINE__);
RSVP(sym);
rule_info__.parser__->set_stop_parse(true);
```

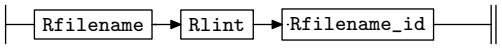
13. *Rparameters* **rule.**

Rparameters



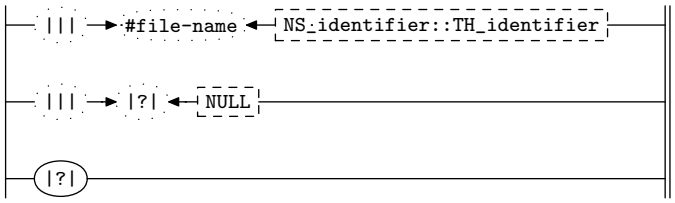
14. *Rfilename_phrase* **rule.**

Rfilename_phrase

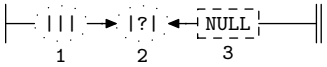


15. *Rfilename* **rule.**

Rfilename



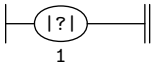
16. *Rfilename*'s subrule 2.



⟨ Rfilename subrule 2 op directive 16 ⟩ ≡

```
sf→p2__→set_auto_delete(true);
CAbs_lr1_sym * sym = new Err_no_filename_present;
sym→set_rc(*sf→p2__, __FILE__, __LINE__);
RSVP(sym);
rule_info__.parser__→set_stop_parse(true);
```

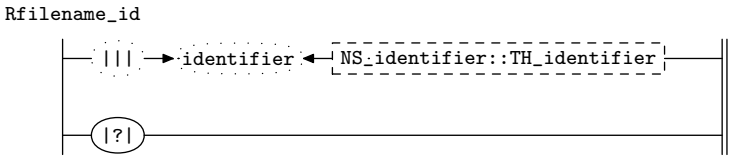
17. *Rfilename*'s subrule 3.



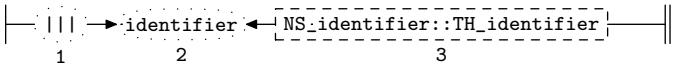
⟨ Rfilename subrule 3 op directive 17 ⟩ ≡

```
CAbs_lr1_sym * sym = new Err_no_filename_present;
sym→set_rc(*rule_info__.parser__→current_token( ), __FILE__, __LINE__);
RSVP(sym);
rule_info__.parser__→set_stop_parse(true);
```

18. *Rfilename_id* rule.



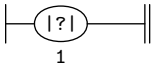
19. *Rfilename_id*'s subrule 1.



⟨ Rfilename_id subrule 1 op directive 19 ⟩ ≡

```
Crc_phrase_th * fsm = ( Crc_phrase_th * ) rule_info__.parser__→fsm_tbl__;
fsm→rc_phrase_→filename_id(sf→p2__);
```

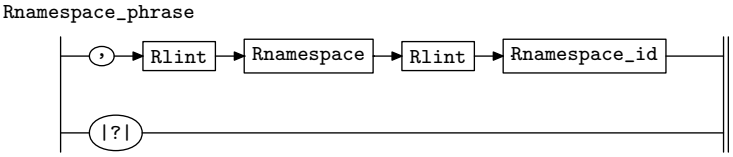
20. *Rfilename_id*'s subrule 2.



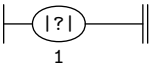
⟨ Rfilename_id subrule 2 op directive 20 ⟩ ≡

```
CAbs_lr1_sym * sym = new Err_no_filename_id_present;
sym→set_rc(*rule_info__.parser__→current_token( ), __FILE__, __LINE__);
RSVP(sym);
rule_info__.parser__→set_stop_parse(true);
```

21. *Rnamespace_phrase* **rule.**



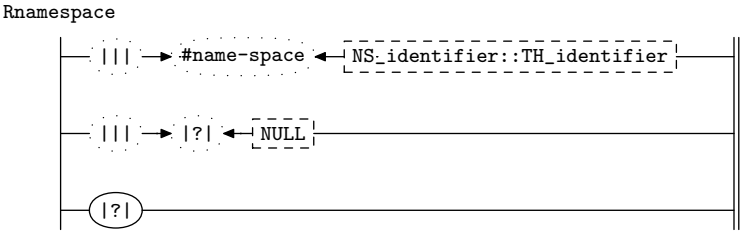
22. *Rnamespace_phrase*'**s subrule 2.**



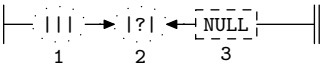
⟨ Rnamespace_phrase subrule 2 op directive 22 ⟩ ≡

```
CAbs_lr1_sym * sym = new Err_no_comma_present;
sym→set_rc(*rule_info__.parser__→current_token( ), __FILE__, __LINE__);
RSVP(sym);
rule_info__.parser__→set_stop_parse(true);
```

23. *Rnamespace* **rule.**



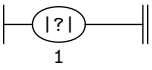
24. *Rnamespace*'**s subrule 2.**



⟨ Rnamespace subrule 2 op directive 24 ⟩ ≡

```
sf→p2__→set_auto_delete(true);
CAbs_lr1_sym * sym = new Err_no_namespace_present;
sym→set_rc(*sf→p2__, __FILE__, __LINE__);
RSVP(sym);
rule_info__.parser__→set_stop_parse(true);
```

25. *Rnamespace*'**s subrule 3.**

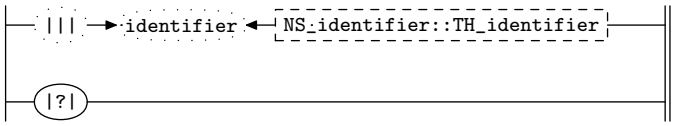


⟨ Rnamespace subrule 3 op directive 25 ⟩ ≡

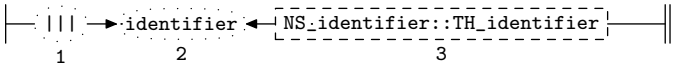
```
CAbs_lr1_sym * sym = new Err_no_namespace_present;
sym→set_rc(*rule_info__.parser__→current_token( ), __FILE__, __LINE__);
RSVP(sym);
rule_info__.parser__→set_stop_parse(true);
```

26. *Rnamespace_id* **rule.**

Rnamespace_id



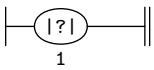
27. *Rnamespace_id*'s **subrule 1.**



⟨ Rnamespace_id subrule 1 op directive 27 ⟩ ≡

```
Crc_phrase_th * fsm = ( Crc_phrase_th * ) rule_info__.parser__→fsm_tbl__;
fsm→rc_phrase_→namespace_id(sf→p2__);
```

28. *Rnamespace_id*'s **subrule 2.**

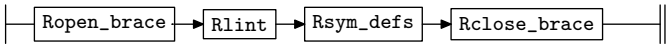


⟨ Rnamespace_id subrule 2 op directive 28 ⟩ ≡

```
CAbs_lr1_sym * sym = new Err_no_namespace_id_present;
sym→set_rc(*rule_info__.parser__→current_token( ), __FILE__, __LINE__);
RSVP(sym);
rule_info__.parser__→set_stop_parse(true);
```

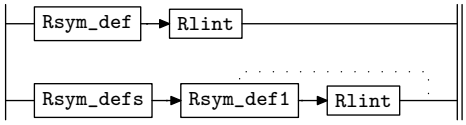
29. *Rsym_defs_phrase* **rule.**

Rsym_defs_phrase

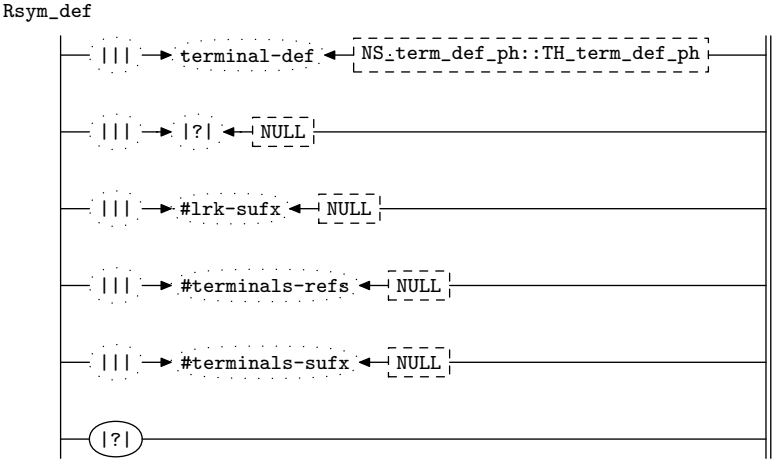


30. *Rsym_defs* **rule.**

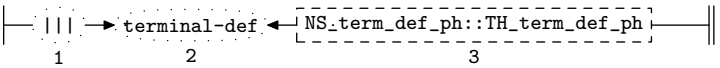
Rsym_defs



31. *Rsym_def* **rule.**



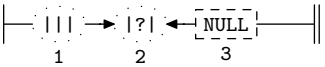
32. *Rsym_def* **'s subrule 1.**



⟨ Rsym_def subrule 1 op directive 32 ⟩ ≡

```
Crc_phrase_th * fsm = ( Crc_phrase_th * ) rule_info__.parser__→fsm_tbl__;
sf→p2__→classification(T_terminal_def :: rc);
CAbs_lr1_sym * r = fsm→rc_phrase_→add_t_to_alphabet(sf→p2__, rule_info__.parser__);
if (r ≡ 0) return;
RSVP(r);
rule_info__.parser__→set_stop_parse(true);
```

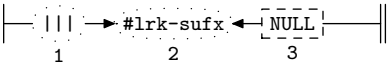
33. *Rsym_def* **'s subrule 2.**



⟨ Rsym_def subrule 2 op directive 33 ⟩ ≡

```
RSVP(sf→p2__);
rule_info__.parser__→set_stop_parse(true);
```

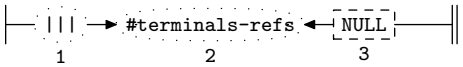
34. *Rsym_def* **'s subrule 3.**



⟨ Rsym_def subrule 3 op directive 34 ⟩ ≡

```
sf→p2__→set_auto_delete(true);
CAbs_lr1_sym * sym = new Err_improper_directive;
sym→set_rc(*sf→p2__, __FILE__, __LINE__);
RSVP(sym);
rule_info__.parser__→set_stop_parse(true);
```

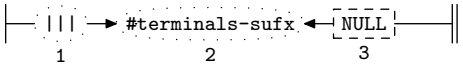
35. *Rsym_def* **'s subrule 4.**



⟨ Rsym_def subrule 4 op directive 35 ⟩ ≡

```
sf→p2__→set_auto_delete(true);
CAbs_lr1_sym * sym = new Err_improper_directive;
sym→set_rc(*sf→p2__, __FILE__, __LINE__);
RSVP(sym);
rule_info__.parser__→set_stop_parse(true);
```

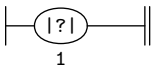
36. *Rsym_def* **'s subrule 5.**



⟨ Rsym_def subrule 5 op directive 36 ⟩ ≡

```
sf→p2__→set_auto_delete(true);
CAbs_lr1_sym * sym = new Err_improper_directive;
sym→set_rc(*sf→p2__, __FILE__, __LINE__);
RSVP(sym);
rule_info__.parser__→set_stop_parse(true);
```

37. *Rsym_def* **'s subrule 6.**

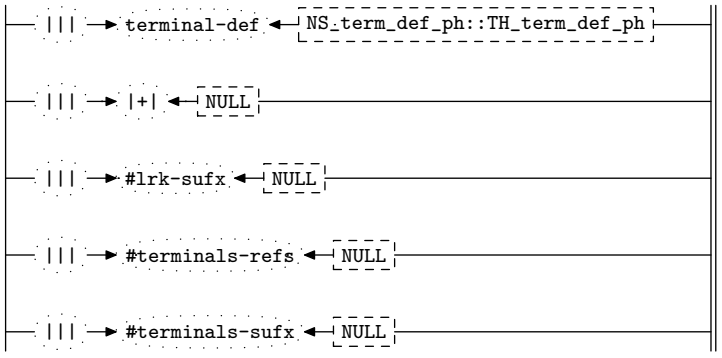


⟨ Rsym_def subrule 6 op directive 37 ⟩ ≡

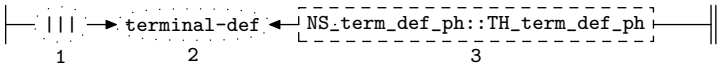
```
CAbs_lr1_sym * sym = new Err_no_sym_defs_present;
sym→set_rc(*rule_info__.parser__→current_token( ), __FILE__, __LINE__);
RSVP(sym);
rule_info__.parser__→set_stop_parse(true);
```

38. *Rsym_def1* **rule.**

Rsym_def1

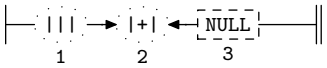


39. *Rsym_def1*'**s subrule 1.**



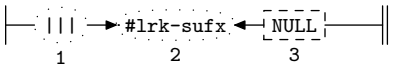
⟨ Rsym_def1 subrule 1 op directive 39 ⟩ ≡
Crc_phrase_th ∗ *fsm* = (*Crc_phrase_th* ∗) *rule_info__.parser__→fsm_tbl__*;
sf→p2__→classification(*T_terminal_def* :: *rc*);
CAbs_lr1_sym ∗ *r* = *fsm→rc_phrase_→add_t_to_alphabet*(*sf→p2__*, *rule_info__.parser__*);
if (*r* ≡ 0) **return**;
RSVP(*r*);
rule_info__.parser__→set_stop_parse(*true*);

40. *Rsym_def1*'**s subrule 2.**



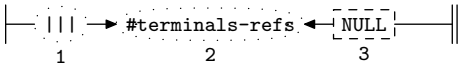
⟨ Rsym_def1 subrule 2 op directive 40 ⟩ ≡
RSVP(*sf→p2__*);
rule_info__.parser__→set_stop_parse(*true*);

41. *Rsym_def1*'**s subrule 3.**



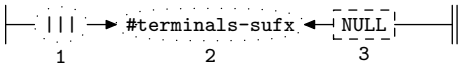
⟨ Rsym_def1 subrule 3 op directive 41 ⟩ ≡
sf→p2__→set_auto_delete(*true*);
CAbs_lr1_sym ∗ *sym* = **new** *Err_improper_directive*;
sym→set_rc(∗*sf→p2__*, __FILE__, __LINE__);
RSVP(*sym*);
rule_info__.parser__→set_stop_parse(*true*);

42. *Rsym_def1*'**s subrule 4.**



⟨ Rsym_def1 subrule 4 op directive 42 ⟩ ≡
sf→p2__→set_auto_delete(*true*);
CAbs_lr1_sym ∗ *sym* = **new** *Err_improper_directive*;
sym→set_rc(∗*sf→p2__*, __FILE__, __LINE__);
RSVP(*sym*);
rule_info__.parser__→set_stop_parse(*true*);

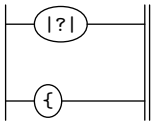
43. *Rsym_def1*'**s subrule 5.**



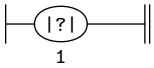
⟨ Rsym_def1 subrule 5 op directive 43 ⟩ ≡
sf→p2__→set_auto_delete(*true*);
CAbs_lr1_sym ∗ *sym* = **new** *Err_improper_directive*;
sym→set_rc(∗*sf→p2__*, __FILE__, __LINE__);
RSVP(*sym*);
rule_info__.parser__→set_stop_parse(*true*);

44. *Ropen_brace* **rule.**

Ropen_brace



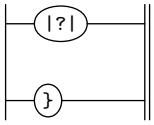
45. *Ropen_brace*'**s subrule 1.**



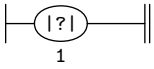
⟨ Ropen_brace subrule 1 op directive 45 ⟩ ≡
CAbs_lr1_sym ∗ *sym* = **new** *Err_no_open_brace*;
sym→set_rc(∗*rule_info__.parser__→current_token*( ), __FILE__, __LINE__);
RSVP(*sym*);
rule_info__.parser__→set_stop_parse(*true*);

46. *Rclose_brace* **rule.**

Rclose_brace



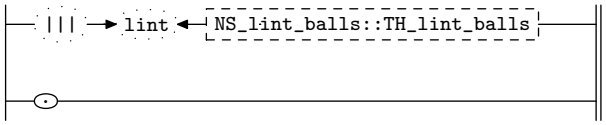
47. *Rclose_brace*'**s subrule 1.**



⟨ Rclose_brace subrule 1 op directive 47 ⟩ ≡
CAbs_lr1_sym ∗ *sym* = **new** *Err_no_close_brace*;
sym→set_rc(∗*rule_info__.parser__→current_token*( ), __FILE__, __LINE__);
RSVP(*sym*);
rule_info__.parser__→set_stop_parse(*true*);

48. *Rlint* **rule.**

Rlint



49. First Set Language for O_2^{linker}.

```
/*
 File: rc_phrase_th.fsc
 Date and Time: Fri Jan  2 15:33:51 2015
*/
transitive   y
grammar-name "rc_phrase_th"
name-space   "NS_rc_phrase_th"
thread-name  "TH_rc_phrase_th"
monolithic   n
file-name    "rc_phrase_th.fsc"
no-of-T      569
list-of-native-first-set-terminals 2
   LR1_questionable_shift_operator
   raw_open_bracket
end-list-of-native-first-set-terminals
list-of-transitive-threads 1
  NS_lint_balls::TH_lint_balls
end-list-of-transitive-threads
list-of-used-threads 3
  NS_identifier::TH_identifier
  NS_lint_balls::TH_lint_balls
  NS_term_def_ph::TH_term_def_ph
end-list-of-used-threads
fsm-comments
"Raw character vocabulary parser."
```

50. Lr1 State Network.

⇒ State: 1 state type: $^{s/r}$

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
c	Rlint			17	2	1	ϵ			1	0	1	1
c	Rlint			17	1	1	||| lint NS_lint_balls::TH_lint_balls			1	2	3	
c	Rrc_phrase			1	1	1	Rlint *Ropen_par*			1	4	14	

⇒\|\|\| *arbitration−code*: ϵ State: 2 state type: ^{s}

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rlint			17	1	2	lint			1	3	3	

⇒^{lint} State: 3 state type: ^{r}

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rlint			17	1	3				1	0	3	1

⇒^{Rlint} State: 4 state type: ^{s}

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
c	Ropen_par			2	1	1	|?|			4	27	27	
c	Ropen_par			2	2	1	(			4	28	28	
t	Rrc_phrase			1	1	2	Ropen_par *Rlint*$^{\epsilon}$ *Rparameters*			1	5	14	

⇒$^{Ropen\_par}$ State: 5 state type: $^{s/r}$

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
c	Rlint			17	2	1	ϵ			5	0	5	2
c	Rlint			17	1	1	||| lint NS_lint_balls::TH_lint_balls			5	2	3	
t	Rrc_phrase			1	1	3	Rlint *Rparameters*			1	6	14	

⇒^{Rlint} State: 6 state type: ^{s}

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
c	Rfilename			6	3	1	|?|			6	29	29	
c	Rfilename			6	1	1	||| # file-name NS_identifier::TH_identifier			6	30	32	
c	Rfilename			6	2	1	||| |?| NULL			6	30	31	
t	Rrc_phrase			1	1	4	Rparameters *Rlint*$^{\epsilon}$ *Rclose_par*			1	7	14	
c	Rparameters			4	1	1	Rfilename_phrase *Rlint*$^{\epsilon}$ *Rnamespace_phrase*			6	33	44	
c	Rfilename_phrase			5	1	1	Rfilename *Rlint*$^{\epsilon}$ *Rfilename_id*			6	45	50	

⇒^{Rparameters} State: 7 state type: $^{s/r}$

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
c	Rlint			17	2	1	ϵ			7	0	7	3
c	Rlint			17	1	1	||| lint NS_lint_balls::TH_lint_balls			7	2	3	
t	Rrc_phrase			1	1	5	Rlint *Rclose_par*			1	8	14	

⇒^{Rlint} State: 8 state type: ^{s}

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
c	Rclose_par			3	1	1	|?|			8	51	51	
c	Rclose_par			3	2	1	)			8	52	52	
t	Rrc_phrase			1	1	6	Rclose_par *Rlint*$^{\epsilon}$ *Rsym_defs_phrase*			1	9	14	

⇒$^{Rclose\_par}$ State: 9 state type: $^{s/r}$

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
c	Rlint			17	2	1	ϵ			9	0	9	4

c Rlint	17	1	1	||| lint NS_lint_balls::TH_lint_balls	9	2	3	
t Rrc_phrase	1	1	7	Rlint *Rsym_defs_phrase*	1	10	14	

⇒*Rlint*

State: 10 state type: ^{s}

← rule	→ R#	sr#	Po	← subrule element	→ Brn	Gto	Red	LA
c Ropen_brace	15	1	1	|?|	10	11	11	
c Ropen_brace	15	2	1	{	10	12	12	
t Rrc_phrase	1	1	8	Rsym_defs_phrase *Rlint*$^{\epsilon}$	1	13	14	
c Rsym_defs_phrase	11	1	1	Ropen_brace *Rlint*$^{\epsilon}$ *Rsym_defs*	10	15	26	

⇒|?|

State: 11 state type: ^{r}

← rule	→ R#	sr#	Po	← subrule element	→ Brn	Gto	Red	LA
t Ropen_brace	15	1	2		10	0	11	2

⇒{

State: 12 state type: ^{r}

← rule	→ R#	sr#	Po	← subrule element	→ Brn	Gto	Red	LA
t Ropen_brace	15	2	2		10	0	12	2

⇒*Rsym_defs_phrase*

State: 13 state type: $^{s/r}$

← rule	→ R#	sr#	Po	← subrule element	→ Brn	Gto	Red	LA
c Rlint	17	2	1	ϵ	13	0	13	1
c Rlint	17	1	1	||| lint NS_lint_balls::TH_lint_balls	13	2	3	
t Rrc_phrase	1	1	9	Rlint	1	14	14	

⇒*Rlint*

State: 14 state type: ^{r}

← rule	→ R#	sr#	Po	← subrule element	→ Brn	Gto	Red	LA
t Rrc_phrase	1	1	10		1	0	14	1

⇒*Ropen_brace*

State: 15 state type: $^{s/r}$

← rule	→ R#	sr#	Po	← subrule element	→ Brn	Gto	Red	LA
c Rlint	17	2	1	ϵ	15	0	15	2
c Rlint	17	1	1	||| lint NS_lint_balls::TH_lint_balls	15	2	3	
t Rsym_defs_phrase	11	1	2	Rlint *Rsym_defs*	10	16	26	

⇒*Rlint*

State: 16 state type: ^{s}

← rule	→ R#	sr#	Po	← subrule element	→ Brn	Gto	Red	LA
c Rsym_def	13	6	1	|?|	16	53	53	
c Rsym_def	13	1	1	||| terminal-def NS_term_def_ph::TH_term_def_ph	16	54	59	
c Rsym_def	13	3	1	||| # lrk-sufx NULL	16	54	58	
c Rsym_def	13	4	1	||| # terminals-refs NULL	16	54	56	
c Rsym_def	13	2	1	||| |?| NULL	16	54	55	
c Rsym_def	13	5	1	||| # terminals-sufx NULL	16	54	57	
t Rsym_defs_phrase	11	1	3	Rsym_defs *Rclose_brace*	10	17	26	
c Rsym_defs	12	2	1	Rsym_defs *Rsym_def1*	16	17	61	
c Rsym_defs	12	1	1	Rsym_def *Rlint*$^{\epsilon}$	16	62	63	

⇒*Rsym_defs*

State: 17 state type: ^{s}

← rule	→ R#	sr#	Po	← subrule element	→ Brn	Gto	Red	LA
c Rclose_brace	16	1	1	|?|	17	18	18	
c Rsym_def1	14	1	1	||| terminal-def NS_term_def_ph::TH_term_def_ph	17	19	24	
c Rsym_def1	14	3	1	||| # lrk-sufx NULL	17	19	23	
c Rsym_def1	14	5	1	||| # terminals-sufx NULL	17	19	22	

c	Rsym_def1	14	2	1	||| |+| NULL	17	19	20
c	Rsym_def1	14	4	1	||| # terminals-refs NULL	17	19	21
c	Rclose_brace	16	2	1	}	17	25	25	
t	Rsym_defs	12	2	2	Rsym_def1 $\underline{Rlint^{\epsilon}}$	16	60	61	
t	Rsym_defs_phrase	11	1	4	Rclose_brace	10	26	26	

⇒|?| State: 18 state type: ^{r}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rclose_brace		16	1	2				17	0	18	1

⇒||| *arbitration−code*: ϵ State: 19 state type: ^{s}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rsym_def1		14	2	2	|+|		17	20	20	
t	Rsym_def1		14	4	2		# terminals-refs		17	21	21	
t	Rsym_def1		14	5	2		# terminals-sufx		17	22	22	
t	Rsym_def1		14	3	2		# lrk-sufx		17	23	23	
t	Rsym_def1		14	1	2		terminal-def		17	24	24	

⇒|+| State: 20 state type: ^{r}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rsym_def1		14	2	3				17	0	20	5

⇒*#terminals−refs* State: 21 state type: ^{r}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rsym_def1		14	4	3				17	0	21	5

⇒*#terminals−sufx* State: 22 state type: ^{r}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rsym_def1		14	5	3				17	0	22	5

⇒*#lrk−sufx* State: 23 state type: ^{r}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rsym_def1		14	3	3				17	0	23	5

⇒*terminal−def* State: 24 state type: ^{r}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rsym_def1		14	1	3				17	0	24	5

⇒} State: 25 state type: ^{r}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rclose_brace		16	2	2				17	0	25	1

⇒*Rclose_brace* State: 26 state type: ^{r}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rsym_defs_phrase		11	1	5				10	0	26	1

⇒|?| State: 27 state type: ^{r}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Ropen_par		2	1	2				4	0	27	2

⇒(State: 28 state type: ^{r}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA

t	Ropen_par		2	2	2				4	0	28	2

⇒`|?|` State: 29 state type: ^{r}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rfilename		6	3	2				6	0	29	2

⇒`|||` *arbitration−code*: ϵ State: 30 state type: ^{s}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rfilename		6	2	2	`|?|`			6	31	31	
t	Rfilename		6	1	2	`#` file-name			6	32	32	

⇒`|?|` State: 31 state type: ^{r}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rfilename		6	2	3				6	0	31	2

⇒*#file−name* State: 32 state type: ^{r}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rfilename		6	1	3				6	0	32	2

⇒*Rfilename_phrase* State: 33 state type: $^{s/r}$

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
c	Rlint		17	2	1	ϵ			33	0	33	6
c	Rlint		17	1	1	`|||` lint NS_lint_balls::TH_lint_balls			33	2	3	
t	Rparameters		4	1	2	Rlint *Rnamespace_phrase*			6	34	44	

⇒*Rlint* State: 34 state type: ^{s}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
c	Rnamespace_phrase		8	2	1	`|?|`			34	35	35	
c	Rnamespace_phrase		8	1	1	,			34	36	43	
t	Rparameters		4	1	3	Rnamespace_phrase			6	44	44	

⇒`|?|` State: 35 state type: ^{r}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rnamespace_phrase		8	2	2				34	0	35	7

⇒, State: 36 state type: $^{s/r}$

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
c	Rlint		17	2	1	ϵ			36	0	36	2
c	Rlint		17	1	1	`|||` lint NS_lint_balls::TH_lint_balls			36	2	3	
t	Rnamespace_phrase		8	1	2	Rlint *Rnamespace*			34	37	43	

⇒*Rlint* State: 37 state type: ^{s}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
c	Rnamespace		9	3	1	`|?|`			37	64	64	
c	Rnamespace		9	1	1	`|||` `#` name-space NS_identifier::TH_identifier			37	65	67	
c	Rnamespace		9	2	1	`|||` `|?|` NULL			37	65	66	
t	Rnamespace_phrase		8	1	3	Rnamespace *Rlint*$^{\epsilon}$ *Rnamespace_id*			34	38	43	

⇒*Rnamespace* State: 38 state type: $^{s/r}$

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
c	Rlint		17	2	1	ϵ			38	0	38	2
c	Rlint		17	1	1	`|||` lint NS_lint_balls::TH_lint_balls			38	2	3	

t	Rnamespace_phrase	8	1	4	Rlint *Rnamespace_id*	34	39	43	

⇒*Rlint* State: 39 state type: *s*

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rnamespace_id		10	2	1	|?|		39	40	40	
c	Rnamespace_id		10	1	1	||| identifier NS_identifier::TH_identifier		39	41	42	
t	Rnamespace_phrase		8	1	5	Rnamespace_id		34	43	43	

⇒\|?\| State: 40 state type: *r*

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rnamespace_id		10	2	2			39	0	40	7

⇒\|\|\| *arbitration−code*: ϵ State: 41 state type: *s*

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rnamespace_id		10	1	2	identifier		39	42	42	

⇒*identifier* State: 42 state type: *r*

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rnamespace_id		10	1	3			39	0	42	7

⇒*Rnamespace_id* State: 43 state type: *r*

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rnamespace_phrase		8	1	6			34	0	43	7

⇒*Rnamespace_phrase* State: 44 state type: *r*

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rparameters		4	1	4			6	0	44	7

⇒*Rfilename* State: 45 state type: *s/r*

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rlint		17	2	1	ϵ		45	0	45	2
c	Rlint		17	1	1	||| lint NS_lint_balls::TH_lint_balls		45	2	3	
t	Rfilename_phrase		5	1	2	Rlint *Rfilename_id*		6	46	50	

⇒*Rlint* State: 46 state type: *s*

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rfilename_id		7	2	1	|?|		46	47	47	
c	Rfilename_id		7	1	1	||| identifier NS_identifier::TH_identifier		46	48	49	
t	Rfilename_phrase		5	1	3	Rfilename_id		6	50	50	

⇒\|?\| State: 47 state type: *r*

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rfilename_id		7	2	2			46	0	47	8

⇒\|\|\| *arbitration−code*: ϵ State: 48 state type: *s*

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rfilename_id		7	1	2	identifier		46	49	49	

⇒*identifier* State: 49 state type: *r*

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rfilename_id		7	1	3			46	0	49	8

⇒*Rfilename_id* State: 50 state type: *r*

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rfilename_phrase		5	1	4			6	0	50	8

⇒|?| State: 51 state type: *r*

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rclose_par		3	1	2			8	0	51	9

⇒) State: 52 state type: *r*

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rclose_par		3	2	2			8	0	52	9

⇒|?| State: 53 state type: *r*

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rsym_def		13	6	2			16	0	53	5

⇒||| *arbitration−code*: ϵ State: 54 state type: *s*

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rsym_def		13	2	2	|?|		16	55	55	
t	Rsym_def		13	4	2	# terminals-refs		16	56	56	
t	Rsym_def		13	5	2	# terminals-sufx		16	57	57	
t	Rsym_def		13	3	2	# lrk-sufx		16	58	58	
t	Rsym_def		13	1	2	terminal-def		16	59	59	

⇒|?| State: 55 state type: *r*

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rsym_def		13	2	3			16	0	55	5

⇒*#terminals−refs* State: 56 state type: *r*

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rsym_def		13	4	3			16	0	56	5

⇒*#terminals−sufx* State: 57 state type: *r*

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rsym_def		13	5	3			16	0	57	5

⇒*#lrk−sufx* State: 58 state type: *r*

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rsym_def		13	3	3			16	0	58	5

⇒*terminal−def* State: 59 state type: *r*

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rsym_def		13	1	3			16	0	59	5

⇒*Rsym_def*1 State: 60 state type: *s/r*

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rlint		17	2	1	ϵ		60	0	60	5
c	Rlint		17	1	1	||| lint NS_lint_balls::TH_lint_balls		60	2	3	
t	Rsym_defs		12	2	3	Rlint		16	61	61	

⇒*Rlint* State: 61 state type: *r*

	← rule	→ R#	sr#	Po	← subrule element	→ Brn	Gto	Red	LA
t	Rsym_defs	12	2	4		16	0	61	5

⇒*Rsym_def* State: 62 state type: $^{s/r}$

	← rule	→ R#	sr#	Po	← subrule element	→ Brn	Gto	Red	LA
c	Rlint	17	2	1	ϵ	62	0	62	5
c	Rlint	17	1	1	`|||` lint NS_lint_balls::TH_lint_balls	62	2	3	
t	Rsym_defs	12	1	2	Rlint	16	63	63	

⇒*Rlint* State: 63 state type: ^{r}

	← rule	→ R#	sr#	Po	← subrule element	→ Brn	Gto	Red	LA
t	Rsym_defs	12	1	3		16	0	63	5

⇒`|?|` State: 64 state type: ^{r}

	← rule	→ R#	sr#	Po	← subrule element	→ Brn	Gto	Red	LA
t	Rnamespace	9	3	2		37	0	64	2

⇒`|||` *arbitration−code*: ϵ State: 65 state type: ^{s}

	← rule	→ R#	sr#	Po	← subrule element	→ Brn	Gto	Red	LA
t	Rnamespace	9	2	2	`|?|`	37	66	66	
t	Rnamespace	9	1	2	`#` name-space	37	67	67	

⇒`|?|` State: 66 state type: ^{r}

	← rule	→ R#	sr#	Po	← subrule element	→ Brn	Gto	Red	LA
t	Rnamespace	9	2	3		37	0	66	2

⇒*#name−space* State: 67 state type: ^{r}

	← rule	→ R#	sr#	Po	← subrule element	→ Brn	Gto	Red	LA
t	Rnamespace	9	1	3		37	0	67	2

51. Index.

rc_phrase_th Grammar

Date: January 2, 2015 at 15:38

File: rc_phrase_th.lex Ns: NS_rc_phrase_th

Version: 1.0 Debug: false

Grammar Comments: Type: Thread

Raw character vocabulary parser.

1 element(s) in Lookahead Expression below

eolr