1. Copyright.

2. *terminals_phrase_th* **thread.**
Parse terminals phrase. It goes thru the terminal definitions and keeps its own local symbol table: map and list. The *terminal_def_phrase* thread adds each symbol to O_2's global symbol table. It calls the common backend *term_def_ph* thread extractor of symbol definition.

The only wrinkle is in the handling of pre / post amble directives $\#terminal-refs$ and $\#terminals-sufx$. These directives are outside the normal containment within a rule, subrule, or "fsm" entities. They come before or after the terminal definitions. Why the wrinkle? *cweb* introductory comments for these directives come before the directive declaration: just like syntax directed code directives. They could stymie the grammar writer by applying this approach to introductory comments for rules which is an error; *cweb* introductory comments for a rule or terminals of all colours comes between the key / name and just before their class constructs.

Example of a terminals vocabulary to parse:

```
/*
file: Tsym.txt
Why:  terminals symbols vocabulary to parse.
Note: terminals is shown for clarity. It is the keyword trigger
      that calls this grammar. Parsed is comments onwards.
*/
terminals
// comments
(file-name  o2_terminals,name-space NS_o2_terminals){
/@
Makesure c++ forward references compile.
@/
  terminals-refs
   using namespace std;
   struct T_c_literal;
         ....
   struct refered_T;
  ***
"option-t"    AD
  /@
   Command line option for \O2 indicating generate T vocabulary code.
   Returned from |yacco2_lcl_option| thread.
  @/
 (sym-class  T_option_t)
         ....
 "lint" (sym-class T_lint)
 terminals-sufx
   static int* NS_o2_terminals::example_(0);
 ***
}
```

3. Fsm Cterminals_phrase_th class.

4. Cterminals_phrase_th constructor directive.

⟨ Cterminals_phrase_th constructor directive 4 ⟩ ≡
*terminals_phrase_* = 0;

5. Cterminals_phrase_th op directive.

⟨ Cterminals_phrase_th op directive 5 ⟩ ≡

```
if (terminals_phrase_ ≠ 0) {
  delete terminals_phrase_;
  terminals_phrase_ = 0;
}
terminals_phrase_ = new T_terminals_phrase;
terminals_phrase_→set_rc(*parser__→start_token__, __FILE__, __LINE__);
AST * t = new AST(*terminals_phrase_);
terminals_phrase_→phrase_tree(t);
```

6. Cterminals_phrase_th user-declaration directive.

⟨ Cterminals_phrase_th user-declaration directive 6 ⟩ ≡

```
public: T_terminals_phrase * terminals_phrase_;
```

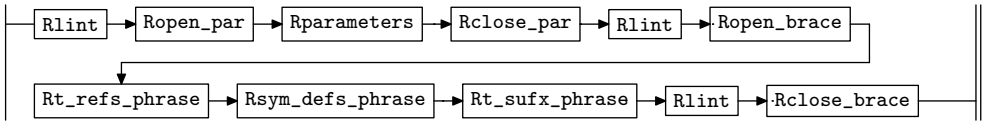
7. Cterminals_phrase_th user-prefix-declaration directive.

⟨ Cterminals_phrase_th user-prefix-declaration directive 7 ⟩ ≡

```
#include "lint_balls.h"
#include "cweb_or_c_k.h"
#include "identifier.h"
#include "term_def_ph.h"
#include "c_string.h"
#include "o2_sdc.h"
```

8. *Rterminals_phrase* rule.

Rterminals_phrase

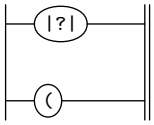


⟨ Rterminals_phrase subrule 1 op directive 8 ⟩ ≡

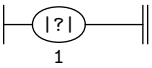
```
Cterminals_phrase_th * fsm = ( Cterminals_phrase_th * ) rule_info__.parser__→fsm_tbl__;
RSVP(fsm→terminals_phrase_);
fsm→terminals_phrase_ = 0;
```

9. *Ropen_par* rule.

Ropen_par



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

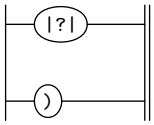


⟨ Ropen_par subrule 1 op directive 10 ⟩ ≡

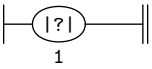
```
CAbs_lr1_sym * sym = new Err_no_open_parenthesis;
sym->set_rc(*rule_info__.parser__->current_token( ), __FILE__, __LINE__);
RSVP(sym);
rule_info__.parser__->set_stop_parse(true);
```

11. *Rclose_par* **rule.**

Rclose_par



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

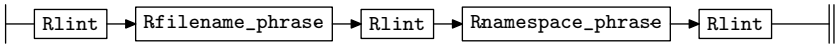


⟨ Rclose_par subrule 1 op directive 12 ⟩ ≡

```
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);
```

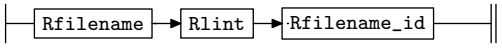
13. *Rparameters* **rule.**

Rparameters



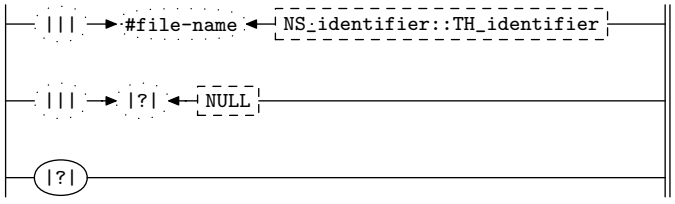
14. *Rfilename_phrase* **rule.**

Rfilename_phrase

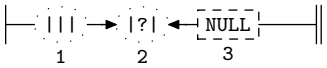


15. *Rfilename* **rule.**

Rfilename

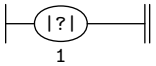


16. *Rfilename*'s subrule 2.



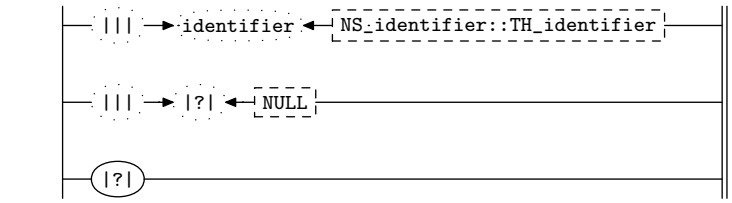
⟨ Rfilename subrule 2 op directive 16 ⟩ ≡
```
sf→p2__→set_auto_delete(true);
CAbs_lr1_sym * sym = new Err_no_filename_present;
sym→set_rc(*sf→p2__, __FILE__, __LINE__);
RSVP(sym);
rule_info__.parser__→set_stop_parse(true);
```

17. *Rfilename*'s subrule 3.

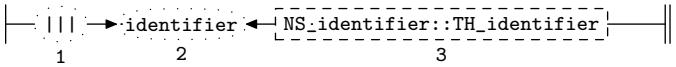


⟨ Rfilename subrule 3 op directive 17 ⟩ ≡
```
CAbs_lr1_sym * sym = new Err_no_filename_present;
sym→set_rc(*rule_info__.parser__→current_token( ), __FILE__, __LINE__);
RSVP(sym);
rule_info__.parser__→set_stop_parse(true);
```

18. *Rfilename_id* rule.

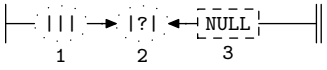


19. *Rfilename_id*'s subrule 1.



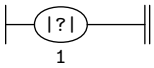
⟨ Rfilename_id subrule 1 op directive 19 ⟩ ≡
```
Cterminals_phrase_th * fsm = ( Cterminals_phrase_th * ) rule_info__.parser__→fsm_tbl__;
fsm→terminals_phrase_→filename_id(sf→p2__);
```

20. *Rfilename_id*'s subrule 2.



⟨ Rfilename_id subrule 2 op directive 20 ⟩ ≡
```
sf→p2__→set_auto_delete(true);
CAbs_lr1_sym * sym = new Err_no_filename_id_present;
sym→set_rc(*sf→p2__, __FILE__, __LINE__);
RSVP(sym);
rule_info__.parser__→set_stop_parse(true);
```

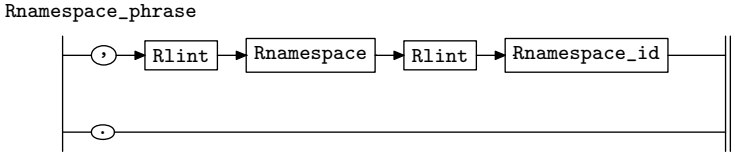
21. *Rfilename_id*'**s subrule 3.**



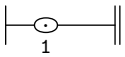
⟨ Rfilename_id subrule 3 op directive 21 ⟩ ≡

```
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);
```

22. *Rnamespace_phrase* **rule.**



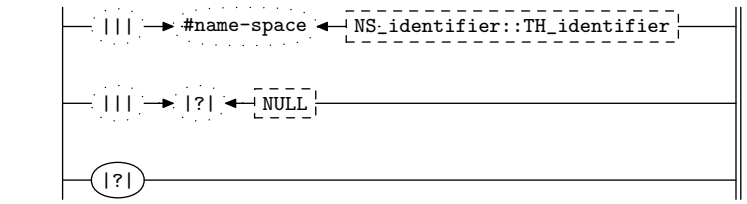
23. *Rnamespace_phrase*'**s subrule 2.**



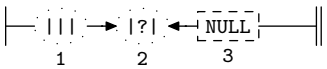
⟨ Rnamespace_phrase subrule 2 op directive 23 ⟩ ≡

```
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);
```

24. *Rnamespace* **rule.**



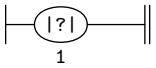
25. *Rnamespace*'**s subrule 2.**



⟨ Rnamespace subrule 2 op directive 25 ⟩ ≡

```
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);
```

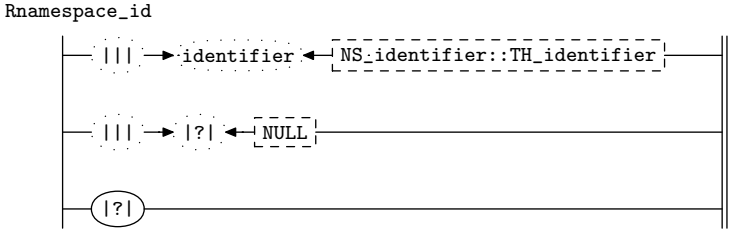
26. *Rnamespace*'s subrule 3.



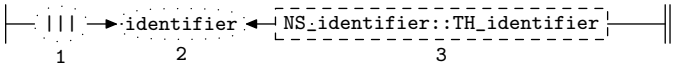
⟨ Rnamespace subrule 3 op directive 26 ⟩ ≡

```
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);
```

27. *Rnamespace_id* rule.



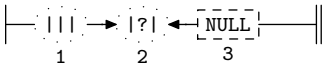
28. *Rnamespace_id*'s subrule 1.



⟨ Rnamespace_id subrule 1 op directive 28 ⟩ ≡

```
Cterminals_phrase_th * fsm = ( Cterminals_phrase_th * ) rule_info__.parser__->fsm_tbl__;
fsm->terminals_phrase_->namespace_id(sf->p2__);
```

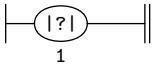
29. *Rnamespace_id*'s subrule 2.



⟨ Rnamespace_id subrule 2 op directive 29 ⟩ ≡

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

30. *Rnamespace_id*'s subrule 3.

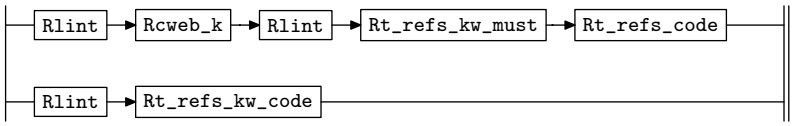


⟨ Rnamespace_id subrule 3 op directive 30 ⟩ ≡

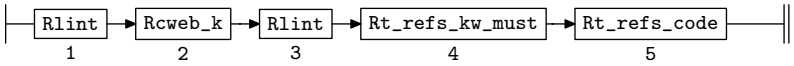
```
CAbs_lr1_sym * sym = new Err_no_namespace_id_present;
sym->set_rc(*rule_info__.parser__->current_token( ), __FILE__, __LINE__);
RSVP(sym);
rule_info__.parser__->set_stop_parse(true);
```

31. *Rt_refs_phrase* **rule.**

Rt_refs_phrase



32. *Rt_refs_phrase*'**s subrule 1.**

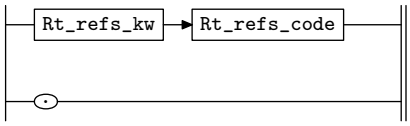


⟨ Rt_refs_phrase subrule 1 op directive 32 ⟩ ≡

```
Cterminals_phrase_th * fsm = ( Cterminals_phrase_th * ) rule_info__.parser__->fsm_tbl__;
fsm->terminals_phrase_->terminals_refs_code( )->add_cweb_marker(sf->p2__->cweb_t_);
```

33. *Rt_refs_kw_code* **rule.**

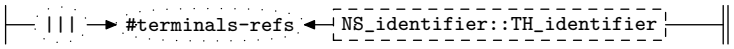
Rt_refs_kw_code



34. *Rt_refs_kw* **rule.**

Drop the keyword as it is not needed. So delete at pop time. Mark the directive as seen just in case it is repeated and so it's an error situation.

Rt_refs_kw



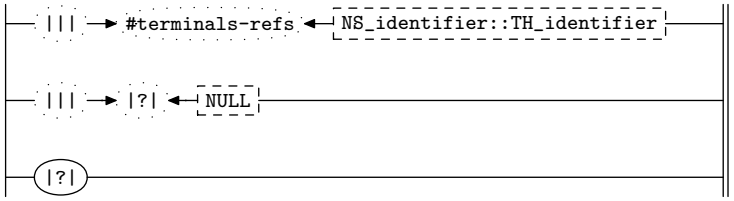
⟨ Rt_refs_kw subrule 1 op directive 34 ⟩ ≡

```
sf->p2__->set_auto_delete(true);
```

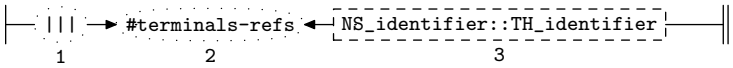
35. *Rt_refs_kw_must* **rule.**

Drop the keyword as it is not needed. So delete at pop time. Mark the directive as seen just in case it is repeated and so it's an error situation.

Rt_refs_kw_must

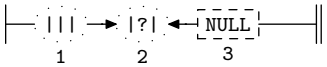


36. *Rt_refs_kw_must*'s subrule 1.



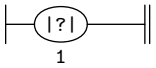
⟨ Rt_refs_kw_must subrule 1 op directive 36 ⟩ ≡
sf→p2__→set_auto_delete(*true*);

37. *Rt_refs_kw_must*'s subrule 2.



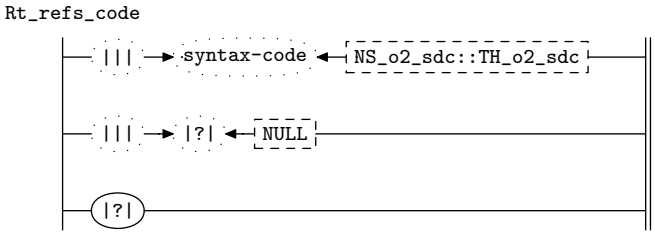
⟨ Rt_refs_kw_must subrule 2 op directive 37 ⟩ ≡
CAbs_lr1_sym ∗ *sym* = **new** *Err_missing_terminals_refs_kw*;
sym→set_rc(∗*sf→p2__*, __FILE__, __LINE__);
RSVP(*sym*);
sf→p2__→set_auto_delete(*true*);
rule_info__.parser__→set_stop_parse(*true*);

38. *Rt_refs_kw_must*'s subrule 3.

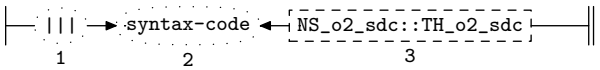


⟨ Rt_refs_kw_must subrule 3 op directive 38 ⟩ ≡
CAbs_lr1_sym ∗ *sym* = **new** *Err_missing_terminals_refs_kw*;
sym→set_rc(∗*sf→p1__*, __FILE__, __LINE__);
RSVP(*sym*);
rule_info__.parser__→set_stop_parse(*true*);

39. *Rt_refs_code* **rule.**

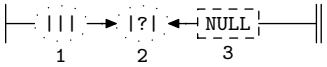


40. *Rt_refs_code*'s subrule 1.



⟨ Rt_refs_code subrule 1 op directive 40 ⟩ ≡
Cterminals_phrase_th ∗ *fsm* = (*Cterminals_phrase_th* ∗) *rule_info__.parser__→fsm_tbl__*;
fsm→terminals_phrase_→terminals_refs_code(*sf→p2__*);

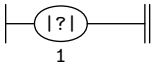
41. *Rt_refs_code*'s subrule 2.



⟨ Rt_refs_code subrule 2 op directive 41 ⟩ ≡

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

42. *Rt_refs_code*'s subrule 3.

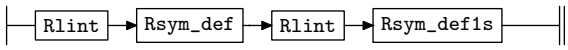


⟨ Rt_refs_code subrule 3 op directive 42 ⟩ ≡

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

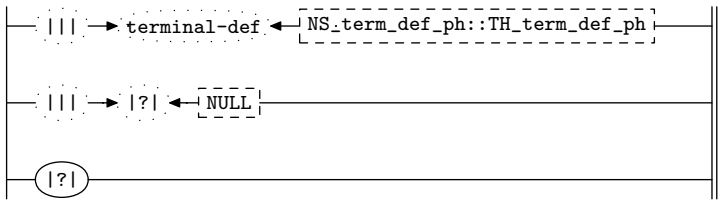
43. *Rsym_defs_phrase* rule.

Rsym_defs_phrase

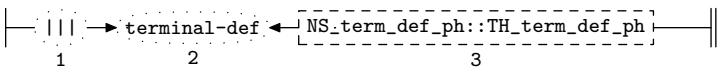


44. *Rsym_def* rule.

Rsym_def



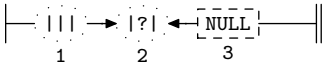
45. *Rsym_def*'s subrule 1.



⟨ Rsym_def subrule 1 op directive 45 ⟩ ≡

```
Cterminals_phrase_th * fsm = ( Cterminals_phrase_th * ) rule_info__.parser__->fsm_tbl__;
sf->p2__->classification(T_terminal_def::t);
CAbs_lr1_sym * r = fsm->terminals_phrase_->add_t_to_alphabet(sf->p2__, rule_info__.parser__);
if (r ≡ 0) return;
RSVP(r);
rule_info__.parser__->set_stop_parse(true);
```

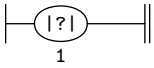
46. *Rsym_def* **'s subrule 2.**



⟨ Rsym_def subrule 2 op directive 46 ⟩ ≡

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

47. *Rsym_def* **'s subrule 3.**

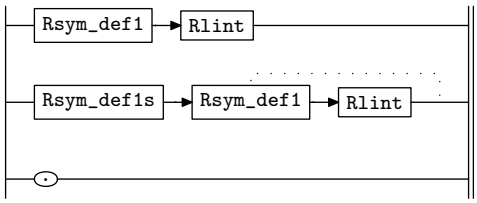


⟨ Rsym_def subrule 3 op directive 47 ⟩ ≡

```
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);
```

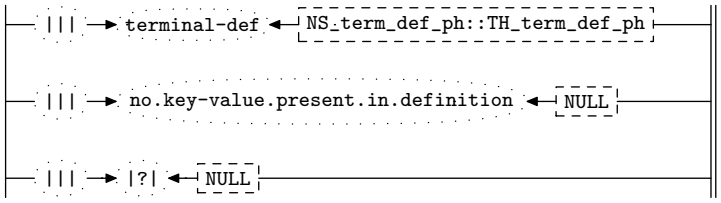
48. *Rsym_def1s* **rule.**

Rsym_def1s

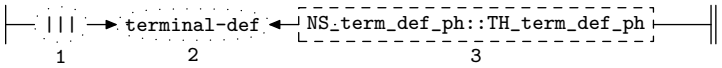


49. *Rsym_def1* **rule.**

Rsym_def1



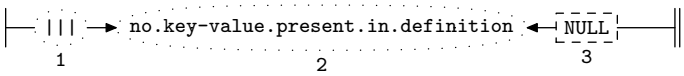
50. *Rsym_def1*'s subrule 1.



⟨ Rsym_def1 subrule 1 op directive 50 ⟩ ≡

```
Cterminals_phrase_th * fsm = ( Cterminals_phrase_th * ) rule_info__.parser__->fsm_tbl__;
sf->p2__->classification(T_terminal_def::t);
CAbs_lr1_sym * r = fsm->terminals_phrase_->add_t_to_alphabet(sf->p2__, rule_info__.parser__);
if (r ≠ 0) {
  RSVP(r);
  rule_info__.parser__->set_stop_parse(true);
  return;
}
```

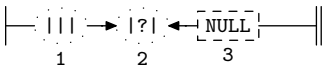
51. *Rsym_def1*'s subrule 2.



⟨ Rsym_def1 subrule 2 op directive 51 ⟩ ≡

```
Cterminals_phrase_th * fsm = ( Cterminals_phrase_th * ) rule_info__.parser__->fsm_tbl__;
RSVP(fsm->terminals_phrase_);
fsm->terminals_phrase_ = 0;
rule_info__.parser__->set_stop_parse(true);
```

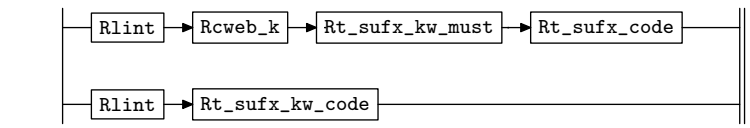
52. *Rsym_def1*'s subrule 3.



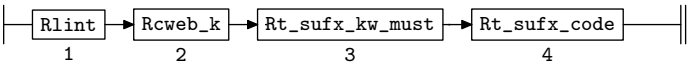
⟨ Rsym_def1 subrule 3 op directive 52 ⟩ ≡

```
using namespace NS_yacco2_T_enum;
int id = sf->p2__->enumerated_id__;
if (id ≥ T_Enum::start_ERR ∧ id ≤ T_Enum::end_ERR) {
  RSVP(sf->p2__);
  rule_info__.parser__->set_stop_parse(true);
}
else {
  CAbs_lr1_sym * sym = new Err_not_a_terminal_definition;
  sym->set_rc(*sf->p2__, __FILE__, __LINE__);
  RSVP(sym);
  rule_info__.parser__->set_stop_parse(true);
}
```

53. *Rt_sufx_phrase* **rule.**



54. *Rt_sufx_phrase*'s subrule 1.

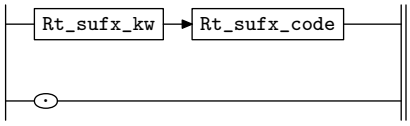


⟨ Rt_sufx_phrase subrule 1 op directive 54 ⟩ ≡
Cterminals_phrase_th ∗ *fsm* = (*Cterminals_phrase_th* ∗) *rule_info__.parser__→fsm_tbl__*;
fsm→terminals_phrase_→terminals_sufx_code( )*→add_cweb_marker*(*sf→p2__→cweb_t_*);

55. *Rt_sufx_kw_code* **rule.**

If a *cweb* comment is seen, deposit it in the directive's syntax directed code.

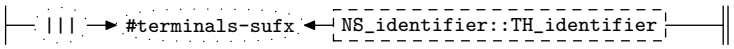
Rt_sufx_kw_code



56. *Rt_sufx_kw* **rule.**

Drop keyword as it's presence is indicated elsewhere. So pop the cork and let the good times roll.

Rt_sufx_kw

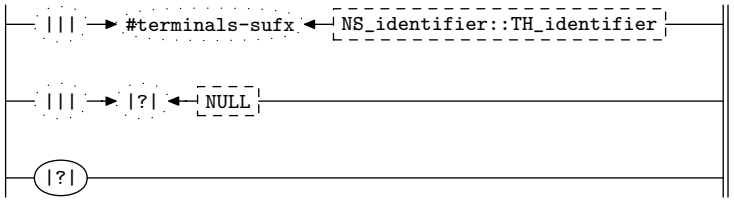


⟨ Rt_sufx_kw subrule 1 op directive 56 ⟩ ≡
sf→p2__→set_auto_delete(*true*);

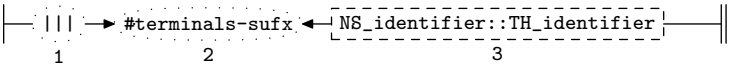
57. *Rt_sufx_kw_must* **rule.**

Drop keyword as it's presence is indicated elsewhere. So pop the cork and let the good times roll.

Rt_sufx_kw_must

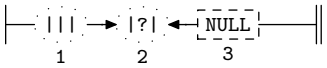


58. *Rt_sufx_kw_must*'s subrule 1.



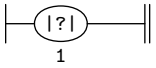
⟨ Rt_sufx_kw_must subrule 1 op directive 58 ⟩ ≡
sf→p2__→set_auto_delete(*true*);

59. *Rt_sufx_kw_must*'s subrule 2.



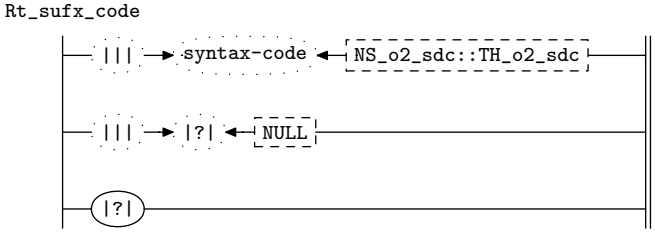
⟨ Rt_sufx_kw_must subrule 2 op directive 59 ⟩ ≡
CAbs_lr1_sym ∗ *sym* = **new** *Err_missing_terminals_sufx_kw*;
sym→*set_rc*(∗*sf*→*p2__*, __FILE__, __LINE__);
RSVP(*sym*);
sf→*p2__*→*set_auto_delete*(*true*);
rule_info__.parser__→*set_stop_parse*(*true*);

60. *Rt_sufx_kw_must*'s subrule 3.

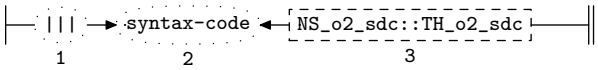


⟨ Rt_sufx_kw_must subrule 3 op directive 60 ⟩ ≡
CAbs_lr1_sym ∗ *sym* = **new** *Err_missing_terminals_sufx_kw*;
sym→*set_rc*(∗*sf*→*p1__*, __FILE__, __LINE__);
RSVP(*sym*);
rule_info__.parser__→*set_stop_parse*(*true*);
return;

61. *Rt_sufx_code* rule.

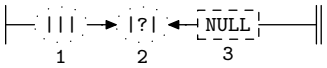


62. *Rt_sufx_code*'s subrule 1.



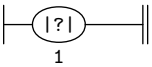
⟨ Rt_sufx_code subrule 1 op directive 62 ⟩ ≡
Cterminals_phrase_th ∗ *fsm* = (*Cterminals_phrase_th* ∗) *rule_info__.parser__*→*fsm_tbl__*;
fsm→*terminals_phrase_*→*terminals_sufx_code*(*sf*→*p2__*);

63. *Rt_sufx_code*'s subrule 2.



⟨ Rt_sufx_code subrule 2 op directive 63 ⟩ ≡
CAbs_lr1_sym ∗ *sym* = **new** *Err_no_syntax_code_present*;
sym→*set_rc*(∗*sf*→*p2__*, __FILE__, __LINE__);
RSVP(*sym*);
sf→*p2__*→*set_auto_delete*(*true*);
rule_info__.parser__→*set_stop_parse*(*true*);

64. *Rt_sufx_code*'s subrule 3.

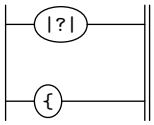


⟨ Rt_sufx_code subrule 3 op directive 64 ⟩ ≡

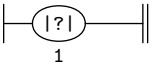
```
CAbs_lr1_sym * sym = new Err_no_syntax_code_present;
sym->set_rc(*rule_info__.parser__->start_token( ), __FILE__, __LINE__);
RSVP(sym);
rule_info__.parser__->set_stop_parse(true);
```

65. *Ropen_brace* rule.

Ropen_brace



66. *Ropen_brace*'s subrule 1.

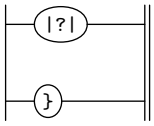


⟨ Ropen_brace subrule 1 op directive 66 ⟩ ≡

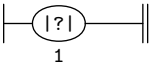
```
CAbs_lr1_sym * sym = new Err_no_open_brace;
sym->set_rc(*rule_info__.parser__->current_token( ), __FILE__, __LINE__);
RSVP(sym);
rule_info__.parser__->set_stop_parse(true);
```

67. *Rclose_brace* rule.

Rclose_brace



68. *Rclose_brace*'s subrule 1.

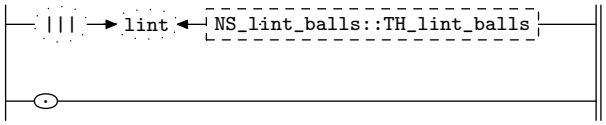


⟨ Rclose_brace subrule 1 op directive 68 ⟩ ≡

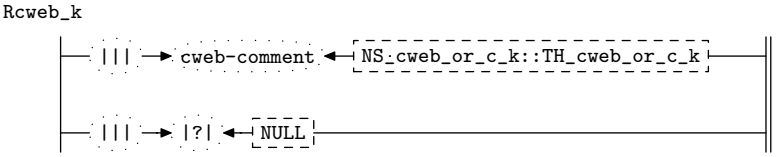
```
CAbs_lr1_sym * sym = new Err_no_close_brace;
sym->set_rc(*rule_info__.parser__->current_token( ), __FILE__, __LINE__);
RSVP(sym);
rule_info__.parser__->set_stop_parse(true);
```

69. *Rlint* rule.

Rlint



70. *Rcweb_k* **rule.**



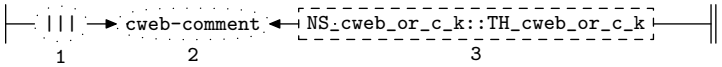
71. Rcweb_k constructor directive.

⟨ Rcweb_k constructor directive 71 ⟩ ≡
```
cweb_t_ = 0;
```

72. Rcweb_k user-declaration directive.

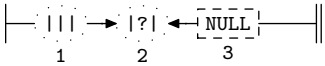
⟨ Rcweb_k user-declaration directive 72 ⟩ ≡
```
AST * cweb_t_;
```

73. *Rcweb_k*'**s subrule 1.**



⟨ Rcweb_k subrule 1 op directive 73 ⟩ ≡
```
Cterminals_phrase_th * fsm = ( Cterminals_phrase_th * ) rule_info__.parser__→fsm_tbl__;
T_cweb_comment * k = sf→p2__;
AST * cwebk_t_ = new AST(*k);
cweb_t_ = new AST( );
T_cweb_marker * cw = new T_cweb_marker(cweb_t_);
cw→set_rc(*k, __FILE__, __LINE__);
cweb_t_ = cw→ast( );
AST::set_content(*cweb_t_, *cw);
AST::join_pts(*cweb_t_, *cwebk_t_);
```

74. *Rcweb_k*'**s subrule 2.**



⟨ Rcweb_k subrule 2 op directive 74 ⟩ ≡
```
RSVP(sf→p2__);
rule_info__.parser__→set_stop_parse(true);
```

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

```
/*
 File: terminals_phrase_th.fsc
 Date and Time: Fri Jan  2 15:33:59 2015
*/
transitive   y
grammar-name "terminals_phrase_th"
name-space   "NS_terminals_phrase_th"
thread-name  "TH_terminals_phrase_th"
monolithic   n
file-name    "terminals_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 5
  NS_cweb_or_c_k::TH_cweb_or_c_k
  NS_identifier::TH_identifier
  NS_lint_balls::TH_lint_balls
  NS_o2_sdc::TH_o2_sdc
  NS_term_def_ph::TH_term_def_ph
end-list-of-used-threads
fsm-comments
"Parse terminals vocabulary."
```

76. Lr1 State Network.

⇒ State: 1 state type: $^{s/r}$

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rlint		27	2	1	ϵ		1	0	1	1
c	Rlint		27	1	1	||| lint NS_lint_balls::TH_lint_balls		1	2	3	
c	Rterminals_phrase		1	1	1	Rlint *Ropen_par*		1	4	16	

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

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rlint		27	1	2	lint		1	3	3	

⇒*lint* State: 3 state type: ^{r}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rlint		27	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	17	17	
c	Ropen_par		2	2	1	(		4	18	18	
t	Rterminals_phrase		1	1	2	Ropen_par *Rparameters*		1	5	16	

⇒*Ropen_par* State: 5 state type: $^{s/r}$

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rlint		27	2	1	ϵ		5	0	5	2
c	Rlint		27	1	1	||| lint NS_lint_balls::TH_lint_balls		5	2	3	
t	Rterminals_phrase		1	1	3	Rparameters *Rclose_par*		1	6	16	
c	Rparameters		4	1	1	Rlint *Rfilename_phrase*		5	19	36	

⇒*Rparameters* State: 6 state type: ^{s}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rclose_par		3	1	1	|?|		6	44	44	
c	Rclose_par		3	2	1	)		6	45	45	
t	Rterminals_phrase		1	1	4	Rclose_par *Rlint*$^{\epsilon}$ *Ropen_brace*		1	7	16	

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

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rlint		27	2	1	ϵ		7	0	7	3
c	Rlint		27	1	1	||| lint NS_lint_balls::TH_lint_balls		7	2	3	
t	Rterminals_phrase		1	1	5	Rlint *Ropen_brace*		1	8	16	

⇒*Rlint* State: 8 state type: ^{s}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Ropen_brace		25	1	1	|?|		8	46	46	
c	Ropen_brace		25	2	1	{		8	47	47	
t	Rterminals_phrase		1	1	6	Ropen_brace *Rt_refs_phrase*$^{\epsilon}$ *Rsym_defs_phrase*		1	9	16	

⇒*Ropen_brace* State: 9 state type: $^{s/r}$

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rlint		27	2	1	ϵ		9	0	9	2
c	Rlint		27	1	1	||| lint NS_lint_balls::TH_lint_balls		9	2	3	
t	Rterminals_phrase		1	1	7	Rt_refs_phrase *Rsym_defs_phrase*		1	10	16	

c	Rt_refs_phrase	11	2	1	Rlint $\underline{Rt\_refs\_kw\_code^{\epsilon}}$	9	48	53
c	Rt_refs_phrase	11	1	1	Rlint $\underline{Rcweb\_k}$	9	48	63

$\Rightarrow^{Rt\_refs\_phrase}$ State: 10 state type: $^{s/r}$

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rlint		27	2	1	ϵ		10	0	10	2
c	Rlint		27	1	1	||| lint NS_lint_balls::TH_lint_balls		10	2	3	
t	Rterminals_phrase		1	1	8	Rsym_defs_phrase $\underline{Rt\_sufx\_phrase^{\epsilon}}$ $\underline{Rlint^{\epsilon}}$ ...		1	11	16	
c	Rsym_defs_phrase		16	1	1	Rlint $\underline{Rsym\_def}$		10	64	75	

$\Rightarrow^{Rsym\_defs\_phrase}$ State: 11 state type: $^{s/r}$

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rlint		27	2	1	ϵ		11	0	11	4
c	Rlint		27	1	1	||| lint NS_lint_balls::TH_lint_balls		11	2	3	
t	Rterminals_phrase		1	1	9	Rt_sufx_phrase $\underline{Rlint^{\epsilon}}$ $\underline{Rclose\_brace}$		1	12	16	
c	Rt_sufx_phrase		20	1	1	Rlint $\underline{Rcweb\_k}$		11	80	92	
c	Rt_sufx_phrase		20	2	1	Rlint $\underline{Rt\_sufx\_kw\_code^{\epsilon}}$		11	80	83	

$\Rightarrow^{Rt\_sufx\_phrase}$ State: 12 state type: $^{s/r}$

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rlint		27	2	1	ϵ		12	0	12	5
c	Rlint		27	1	1	||| lint NS_lint_balls::TH_lint_balls		12	2	3	
t	Rterminals_phrase		1	1	10	Rlint $\underline{Rclose\_brace}$		1	13	16	

$\Rightarrow^{Rlint}$ State: 13 state type: ^{s}

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rclose_brace		26	1	1	|?|		13	14	14	
c	Rclose_brace		26	2	1	}		13	15	15	
t	Rterminals_phrase		1	1	11	Rclose_brace		1	16	16	

$\Rightarrow^{|?|}$ State: 14 state type: ^{r}

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rclose_brace		26	1	2			13	0	14	6

$\Rightarrow^{\}}$ State: 15 state type: ^{r}

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rclose_brace		26	2	2			13	0	15	6

$\Rightarrow^{Rclose\_brace}$ State: 16 state type: ^{r}

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rterminals_phrase		1	1	12			1	0	16	6

$\Rightarrow^{|?|}$ State: 17 state type: ^{r}

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Ropen_par		2	1	2			4	0	17	2

$\Rightarrow^{(}$ State: 18 state type: ^{r}

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Ropen_par		2	2	2			4	0	18	2

$\Rightarrow^{Rlint}$ State: 19 state type: ^{s}

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

c Rfilename	6	3	1	|?|	19	20	20	
c Rfilename	6	1	1	||| # file-name NS_identifier::TH_identifier	19	21	23	
c Rfilename	6	2	1	||| |?| NULL	19	21	22	
t Rparameters	4	1	2	Rfilename_phrase *Rlint*$^{\epsilon}$ *Rnamespace_phrase*$^{\epsilon}$ ...	5	24	36	
c Rfilename_phrase	5	1	1	Rfilename *Rlint*$^{\epsilon}$ *Rfilename_id*	19	37	43	

⇒\|?\| State: 20 state type: ^{r}

← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t Rfilename		6	3	2			19	0	20	2

⇒\|\|\| *arbitration−code*: ϵ State: 21 state type: ^{s}

← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t Rfilename		6	2	2	|?|		19	22	22	
t Rfilename		6	1	2	# file-name		19	23	23	

⇒\|?\| State: 22 state type: ^{r}

← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t Rfilename		6	2	3			19	0	22	2

⇒*#file−name* State: 23 state type: ^{r}

← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t Rfilename		6	1	3			19	0	23	2

⇒*Rfilename_phrase* State: 24 state type: $^{s/r}$

← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c Rlint		27	2	1	ϵ		24	0	24	7
c Rlint		27	1	1	||| lint NS_lint_balls::TH_lint_balls		24	2	3	
t Rparameters		4	1	3	Rlint *Rnamespace_phrase*$^{\epsilon}$ *Rlint*$^{\epsilon}$ ...		5	25	36	

⇒*Rlint* State: 25 state type: $^{s/r}$

← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c Rnamespace_phrase		8	2	1	ϵ		25	0	25	8
c Rnamespace_phrase		8	1	1	,		25	26	34	
t Rparameters		4	1	4	Rnamespace_phrase *Rlint*$^{\epsilon}$		5	35	36	

⇒, State: 26 state type: $^{s/r}$

← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c Rlint		27	2	1	ϵ		26	0	26	2
c Rlint		27	1	1	||| lint NS_lint_balls::TH_lint_balls		26	2	3	
t Rnamespace_phrase		8	1	2	Rlint *Rnamespace*		25	27	34	

⇒*Rlint* State: 27 state type: ^{s}

← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c Rnamespace		9	3	1	|?|		27	93	93	
c Rnamespace		9	1	1	||| # name-space NS_identifier::TH_identifier		27	94	96	
c Rnamespace		9	2	1	||| |?| NULL		27	94	95	
t Rnamespace_phrase		8	1	3	Rnamespace *Rlint*$^{\epsilon}$ *Rnamespace_id*		25	28	34	

⇒*Rnamespace* State: 28 state type: $^{s/r}$

← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c Rlint		27	2	1	ϵ		28	0	28	2
c Rlint		27	1	1	||| lint NS_lint_balls::TH_lint_balls		28	2	3	

t	Rnamespace_phrase	8	1	4	Rlint *Rnamespace_id*	25	29	34	

⇒*Rlint*

State: 29 state type: *s*

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rnamespace_id		10	3	1	|?|		29	30	30	
c	Rnamespace_id		10	1	1	||| identifier NS_identifier::TH_identifier		29	31	33	
c	Rnamespace_id		10	2	1	||| |?| NULL		29	31	32	
t	Rnamespace_phrase		8	1	5	Rnamespace_id		25	34	34	

⇒\|?\|

State: 30 state type: *r*

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rnamespace_id		10	3	2			29	0	30	8

⇒\|\|\| *arbitration−code*: ϵ

State: 31 state type: *s*

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rnamespace_id		10	2	2	|?|		29	32	32	
t	Rnamespace_id		10	1	2	identifier		29	33	33	

⇒\|?\|

State: 32 state type: *r*

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rnamespace_id		10	2	3			29	0	32	8

⇒*identifier*

State: 33 state type: *r*

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rnamespace_id		10	1	3			29	0	33	8

⇒*Rnamespace_id*

State: 34 state type: *r*

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rnamespace_phrase		8	1	6			25	0	34	8

⇒*Rnamespace_phrase*

State: 35 state type: *s/r*

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rlint		27	2	1	ϵ		35	0	35	9
c	Rlint		27	1	1	||| lint NS_lint_balls::TH_lint_balls		35	2	3	
t	Rparameters		4	1	5	Rlint		5	36	36	

⇒*Rlint*

State: 36 state type: *r*

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rparameters		4	1	6			5	0	36	9

⇒*Rfilename*

State: 37 state type: *s/r*

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rlint		27	2	1	ϵ		37	0	37	2
c	Rlint		27	1	1	||| lint NS_lint_balls::TH_lint_balls		37	2	3	
t	Rfilename_phrase		5	1	2	Rlint *Rfilename_id*		19	38	43	

⇒*Rlint*

State: 38 state type: *s*

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rfilename_id		7	3	1	|?|		38	39	39	
c	Rfilename_id		7	2	1	||| |?| NULL		38	40	41	
c	Rfilename_id		7	1	1	||| identifier NS_identifier::TH_identifier		38	40	42	
t	Rfilename_phrase		5	1	3	Rfilename_id		19	43	43	

⇒|?| State: 39 state type: ^{r}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rfilename_id		7	3	2			38	0	39	7

⇒||| *arbitration−code*: ϵ State: 40 state type: ^{s}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rfilename_id		7	2	2	|?|		38	41	41	
t	Rfilename_id		7	1	2	identifier		38	42	42	

⇒|?| State: 41 state type: ^{r}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rfilename_id		7	2	3			38	0	41	7

⇒*identifier* State: 42 state type: ^{r}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rfilename_id		7	1	3			38	0	42	7

⇒*Rfilename_id* State: 43 state type: ^{r}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rfilename_phrase		5	1	4			19	0	43	7

⇒|?| State: 44 state type: ^{r}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rclose_par		3	1	2			6	0	44	10

⇒) State: 45 state type: ^{r}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rclose_par		3	2	2			6	0	45	10

⇒|?| State: 46 state type: ^{r}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Ropen_brace		25	1	2			8	0	46	2

⇒{ State: 47 state type: ^{r}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Ropen_brace		25	2	2			8	0	47	2

⇒*Rlint* State: 48 state type: $^{s/r}$

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rt_refs_kw_code		12	2	1	ϵ		48	0	48	2
c	Rcweb_k		28	2	1	||| |?| NULL		48	49	50	
c	Rt_refs_kw		13	1	1	||| # terminals-refs NS_identifier::TH_identifier		48	49	51	
c	Rcweb_k		28	1	1	||| cweb-comment NS_cweb_or_c_k::TH_cweb_or_c_k		48	49	52	
t	Rt_refs_phrase		11	2	2	Rt_refs_kw_code		9	53	53	
c	Rt_refs_kw_code		12	1	1	Rt_refs_kw *Rt_refs_code*		48	54	59	
t	Rt_refs_phrase		11	1	2	Rcweb_k *Rlint*$^{\epsilon}$ *Rt_refs_kw_must*		9	60	63	

⇒||| *arbitration−code*: ϵ State: 49 state type: ^{s}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rcweb_k		28	2	2	|?|		48	50	50	
t	Rt_refs_kw		13	1	2	# terminals-refs		48	51	51	

t	Rcweb_k	28	1	2	cweb-comment			48	52	52	

⇒`|?|` State: 50 state type: ^{r}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rcweb_k		28	2	3				48	0	50	2

⇒$^{\#terminals-refs}$ State: 51 state type: ^{r}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rt_refs_kw		13	1	3				48	0	51	2

⇒$^{cweb-comment}$ State: 52 state type: ^{r}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rcweb_k		28	1	3				48	0	52	2

⇒$^{Rt\_refs\_kw\_code}$ State: 53 state type: ^{r}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rt_refs_phrase		11	2	3				9	0	53	2

⇒$^{Rt\_refs\_kw}$ State: 54 state type: ^{s}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
c	Rt_refs_code		15	3	1	`|?|`			54	55	55	
c	Rt_refs_code		15	2	1	`|||` `|?|` NULL			54	56	57	
c	Rt_refs_code		15	1	1	`|||` syntax-code NS_o2_sdc::TH_o2_sdc			54	56	58	
t	Rt_refs_kw_code		12	1	2	Rt_refs_code			48	59	59	

⇒`|?|` State: 55 state type: ^{r}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rt_refs_code		15	3	2				54	0	55	2

⇒`|||` $^{arbitration-code:\ \epsilon}$ State: 56 state type: ^{s}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rt_refs_code		15	2	2	`|?|`			54	57	57	
t	Rt_refs_code		15	1	2	syntax-code			54	58	58	

⇒`|?|` State: 57 state type: ^{r}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rt_refs_code		15	2	3				54	0	57	2

⇒$^{syntax-code}$ State: 58 state type: ^{r}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rt_refs_code		15	1	3				54	0	58	2

⇒$^{Rt\_refs\_code}$ State: 59 state type: ^{r}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rt_refs_kw_code		12	1	3				48	0	59	2

⇒$^{Rcweb\_k}$ State: 60 state type: $^{s/r}$

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
c	Rlint		27	2	1	ϵ			60	0	60	2
c	Rlint		27	1	1	`|||` lint NS_lint_balls::TH_lint_balls			60	2	3	
t	Rt_refs_phrase		11	1	3	Rlint $\underline{Rt\_refs\_kw\_must}$			9	61	63	

⇒*Rlint* State: 61 state type: *s*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rt_refs_kw_must		14	3	1	|?|		61	97	97	
c	Rt_refs_kw_must		14	1	1	||| # terminals-refs NS_identifier::TH_identifier		61	98	100	
c	Rt_refs_kw_must		14	2	1	||| |?| NULL		61	98	99	
t	Rt_refs_phrase		11	1	4	Rt_refs_kw_must *Rt_refs_code*		9	62	63	

⇒*Rt_refs_kw_must* State: 62 state type: *s*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rt_refs_code		15	3	1	|?|		62	55	55	
c	Rt_refs_code		15	2	1	||| |?| NULL		62	56	57	
c	Rt_refs_code		15	1	1	||| syntax-code NS_o2_sdc::TH_o2_sdc		62	56	58	
t	Rt_refs_phrase		11	1	5	Rt_refs_code		9	63	63	

⇒*Rt_refs_code* State: 63 state type: *r*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rt_refs_phrase		11	1	6			9	0	63	2

⇒*Rlint* State: 64 state type: *s*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rsym_def		17	3	1	|?|		64	65	65	
c	Rsym_def		17	1	1	||| terminal-def NS_term_def_ph::TH_term_def_ph		64	66	68	
c	Rsym_def		17	2	1	||| |?| NULL		64	66	67	
t	Rsym_defs_phrase		16	1	2	Rsym_def *Rlint*$^{\epsilon}$ *Rsym_def1s*$^{\epsilon}$ ...		10	69	75	

⇒\|?\| State: 65 state type: *r*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rsym_def		17	3	2			64	0	65	4

⇒\|\|\| *arbitration−code*: ϵ State: 66 state type: *s*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rsym_def		17	2	2	|?|		64	67	67	
t	Rsym_def		17	1	2	terminal-def		64	68	68	

⇒\|?\| State: 67 state type: *r*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rsym_def		17	2	3			64	0	67	4

⇒*terminal−def* State: 68 state type: *r*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rsym_def		17	1	3			64	0	68	4

⇒*Rsym_def* State: 69 state type: *s/r*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rlint		27	2	1	ϵ		69	0	69	4
c	Rlint		27	1	1	||| lint NS_lint_balls::TH_lint_balls		69	2	3	
t	Rsym_defs_phrase		16	1	3	Rlint *Rsym_def1s*$^{\epsilon}$		10	70	75	

⇒*Rlint* State: 70 state type: *s/r*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rsym_def1s		18	3	1	ϵ		70	0	70	4

c Rsym_def1	19	2	1	||| no key-value present in definition NULL	70	71	74	
c Rsym_def1	19	1	1	||| terminal-def NS_term_def_ph::TH_term_def_ph	70	71	73	
c Rsym_def1	19	3	1	||| |?| NULL	70	71	72	
t Rsym_defs_phrase	16	1	4	Rsym_def1s	10	75	75	
c Rsym_def1s	18	2	1	Rsym_def1s $\underline{Rsym\_def1}$	70	75	77	
c Rsym_def1s	18	1	1	Rsym_def1 $\underline{Rlint^{\epsilon}}$	70	78	79	

$\Rightarrow$\|\|\| *arbitration−code*: *AR_Rsym_def1* State: 71 state type: ^{s}

← rule	→ R#	sr#	Po	← subrule element	→ Brn	Gto	Red	LA
t Rsym_def1	19	3	2	|?|	70	72	72	
t Rsym_def1	19	1	2	terminal-def	70	73	73	
t Rsym_def1	19	2	2	no key-value present in definition	70	74	74	

$\Rightarrow$\|?\| State: 72 state type: ^{r}

← rule	→ R#	sr#	Po	← subrule element	→ Brn	Gto	Red	LA
t Rsym_def1	19	3	3		70	0	72	4

$\Rightarrow^{terminal-def}$ State: 73 state type: ^{r}

← rule	→ R#	sr#	Po	← subrule element	→ Brn	Gto	Red	LA
t Rsym_def1	19	1	3		70	0	73	4

$\Rightarrow^{nokey-valuepresentindefinition}$ State: 74 state type: ^{r}

← rule	→ R#	sr#	Po	← subrule element	→ Brn	Gto	Red	LA
t Rsym_def1	19	2	3		70	0	74	4

$\Rightarrow^{Rsym\_def1s}$ State: 75 state type: $^{s/r}$

← rule	→ R#	sr#	Po	← subrule element	→ Brn	Gto	Red	LA
t Rsym_defs_phrase	16	1	5		10	0	75	4
c Rsym_def1	19	2	1	||| no key-value present in definition NULL	75	71	74	
c Rsym_def1	19	1	1	||| terminal-def NS_term_def_ph::TH_term_def_ph	75	71	73	
c Rsym_def1	19	3	1	||| |?| NULL	75	71	72	
t Rsym_def1s	18	2	2	Rsym_def1 $\underline{Rlint^{\epsilon}}$	70	76	77	

$\Rightarrow^{Rsym\_def1}$ State: 76 state type: $^{s/r}$

← rule	→ R#	sr#	Po	← subrule element	→ Brn	Gto	Red	LA
c Rlint	27	2	1	ϵ	76	0	76	4
c Rlint	27	1	1	||| lint NS_lint_balls::TH_lint_balls	76	2	3	
t Rsym_def1s	18	2	3	Rlint	70	77	77	

$\Rightarrow^{Rlint}$ State: 77 state type: ^{r}

← rule	→ R#	sr#	Po	← subrule element	→ Brn	Gto	Red	LA
t Rsym_def1s	18	2	4		70	0	77	4

$\Rightarrow^{Rsym\_def1}$ State: 78 state type: $^{s/r}$

← rule	→ R#	sr#	Po	← subrule element	→ Brn	Gto	Red	LA
c Rlint	27	2	1	ϵ	78	0	78	4
c Rlint	27	1	1	||| lint NS_lint_balls::TH_lint_balls	78	2	3	
t Rsym_def1s	18	1	2	Rlint	70	79	79	

$\Rightarrow^{Rlint}$ State: 79 state type: ^{r}

← rule	→ R#	sr#	Po	← subrule element	→ Brn	Gto	Red	LA
t Rsym_def1s	18	1	3		70	0	79	4

⇒*Rlint* State: 80 state type: *s/r*

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
c	Rt_sufx_kw_code		21	2	1	ϵ			80	0	80	4
c	Rcweb_k		28	2	1	||| |?| NULL			80	81	50	
c	Rt_sufx_kw		22	1	1	||| # terminals-sufx NS_identifier::TH_identifier			80	81	82	
c	Rcweb_k		28	1	1	||| cweb-comment NS_cweb_or_c_k::TH_cweb_or_c_k			80	81	52	
t	Rt_sufx_phrase		20	2	2	Rt_sufx_kw_code			11	83	83	
c	Rt_sufx_kw_code		21	1	1	Rt_sufx_kw *Rt_sufx_code*			80	84	89	
t	Rt_sufx_phrase		20	1	2	Rcweb_k *Rt_sufx_kw_must*			11	90	92	

⇒\|\|\| *arbitration−code*: ϵ State: 81 state type: *s*

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rcweb_k		28	2	2	|?|			80	50	50	
t	Rt_sufx_kw		22	1	2	# terminals-sufx			80	82	82	
t	Rcweb_k		28	1	2	cweb-comment			80	52	52	

⇒*#terminals−sufx* State: 82 state type: *r*

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rt_sufx_kw		22	1	3				80	0	82	2

⇒*Rt_sufx_kw_code* State: 83 state type: *r*

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rt_sufx_phrase		20	2	3				11	0	83	4

⇒*Rt_sufx_kw* State: 84 state type: *s*

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
c	Rt_sufx_code		24	3	1	|?|			84	85	85	
c	Rt_sufx_code		24	2	1	||| |?| NULL			84	86	87	
c	Rt_sufx_code		24	1	1	||| syntax-code NS_o2_sdc::TH_o2_sdc			84	86	88	
t	Rt_sufx_kw_code		21	1	2	Rt_sufx_code			80	89	89	

⇒\|?\| State: 85 state type: *r*

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rt_sufx_code		24	3	2				84	0	85	4

⇒\|\|\| *arbitration−code*: ϵ State: 86 state type: *s*

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rt_sufx_code		24	2	2	|?|			84	87	87	
t	Rt_sufx_code		24	1	2	syntax-code			84	88	88	

⇒\|?\| State: 87 state type: *r*

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rt_sufx_code		24	2	3				84	0	87	4

⇒*syntax−code* State: 88 state type: *r*

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rt_sufx_code		24	1	3				84	0	88	4

⇒*Rt_sufx_code* State: 89 state type: *r*

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rt_sufx_kw_code		21	1	3				80	0	89	4

⇒*Rcweb_k* State: 90 state type: *s*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rt_sufx_kw_must		23	3	1	|?|		90	101	101	
c	Rt_sufx_kw_must		23	1	1	||| # terminals-sufx NS_identifier::TH_identifier		90	102	104	
c	Rt_sufx_kw_must		23	2	1	||| |?| NULL		90	102	103	
t	Rt_sufx_phrase		20	1	3	Rt_sufx_kw_must *Rt_sufx_code*		11	91	92	

⇒*Rt_sufx_kw_must* State: 91 state type: *s*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rt_sufx_code		24	3	1	|?|		91	85	85	
c	Rt_sufx_code		24	2	1	||| |?| NULL		91	86	87	
c	Rt_sufx_code		24	1	1	||| syntax-code NS_o2_sdc::TH_o2_sdc		91	86	88	
t	Rt_sufx_phrase		20	1	4	Rt_sufx_code		11	92	92	

⇒*Rt_sufx_code* State: 92 state type: *r*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rt_sufx_phrase		20	1	5			11	0	92	4

⇒\|?\| State: 93 state type: *r*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rnamespace		9	3	2			27	0	93	2

⇒\|\|\| *arbitration−code*: ϵ State: 94 state type: *s*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rnamespace		9	2	2	|?|		27	95	95	
t	Rnamespace		9	1	2	# name-space		27	96	96	

⇒\|?\| State: 95 state type: *r*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rnamespace		9	2	3			27	0	95	2

⇒*#name−space* State: 96 state type: *r*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rnamespace		9	1	3			27	0	96	2

⇒\|?\| State: 97 state type: *r*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rt_refs_kw_must		14	3	2			61	0	97	2

⇒\|\|\| *arbitration−code*: ϵ State: 98 state type: *s*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rt_refs_kw_must		14	2	2	|?|		61	99	99	
t	Rt_refs_kw_must		14	1	2	# terminals-refs		61	100	100	

⇒\|?\| State: 99 state type: *r*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rt_refs_kw_must		14	2	3			61	0	99	2

⇒*#terminals−refs* State: 100 state type: *r*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rt_refs_kw_must		14	1	3			61	0	100	2

⇒`|?|`

State: 101 state type: ^{r}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rt_sufx_kw_must		23	3	2			90	0	101	2

⇒`|||` *arbitration−code*: ϵ

State: 102 state type: ^{s}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rt_sufx_kw_must		23	2	2	`|?|`		90	103	103	
t	Rt_sufx_kw_must		23	1	2	`#` terminals-sufx		90	104	104	

⇒`|?|`

State: 103 state type: ^{r}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rt_sufx_kw_must		23	2	3			90	0	103	2

⇒*#terminals−sufx*

State: 104 state type: ^{r}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rt_sufx_kw_must		23	1	3			90	0	104	2

77. Index.

terminals_phrase_th Grammar

Date: January 2, 2015 at 15:39

File: terminals_phrase_th.lex Ns: NS_terminals_phrase_th

Version: 1.0 Debug: false

Grammar Comments: Type: Thread

Parse terminals vocabulary.

1 element(s) in Lookahead Expression below

eolr