1. Copyright.

2. *T_enum_phrase_th* **Thread.**
parse T-enumeration phrase.

3. Fsm CT_enum_phrase_th class.

4. CT_enum_phrase_th constructor directive.

⟨ CT_enum_phrase_th constructor directive 4 ⟩ ≡
```
enum_phrase_ = 0;
```

5. CT_enum_phrase_th op directive.

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

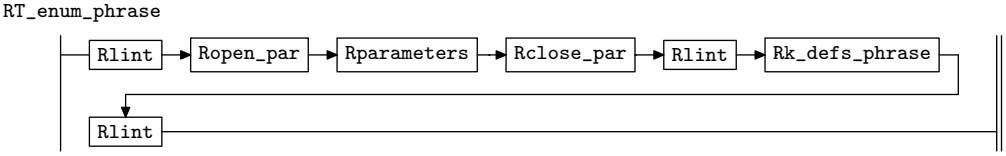
6. CT_enum_phrase_th user-declaration directive.

⟨ CT_enum_phrase_th user-declaration directive 6 ⟩ ≡
```
public: T_enum_phrase * enum_phrase_;
```

7. CT_enum_phrase_th user-prefix-declaration directive.

⟨ CT_enum_phrase_th user-prefix-declaration directive 7 ⟩ ≡
```
#include "lint_balls.h"
#include "eol.h"
#include "c_comments.h"
#include "identifier.h"
#include "c_string.h"
#include "o2_sdc.h"
  using namespace NS_yacco2_terminals;
```

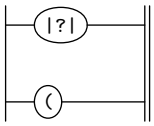
8. *RT_enum_phrase* **rule.**



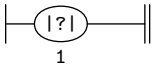
⟨ RT_enum_phrase subrule 1 op directive 8 ⟩ ≡
```
CT_enum_phrase_th * fsm = ( CT_enum_phrase_th * ) rule_info__.parser__→fsm_tbl__;
RSVP(fsm→enum_phrase_);
fsm→enum_phrase_ = 0;
```

9. *Ropen_par* **rule.**

Ropen_par



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

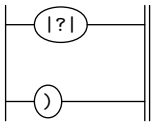


⟨ Ropen_par subrule 1 op directive 10 ⟩ ≡

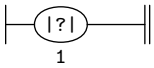
```
CAbs_lr1_sym * sym = new Err_no_open_parenthesis;
sym->set_rc(*rule_info__.parser__->current_token( ), __FILE__, __LINE__);
RSVP(sym);
rule_info__.parser__->set_stop_parse(true);
```

11. *Rclose_par* **rule.**

Rclose_par



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

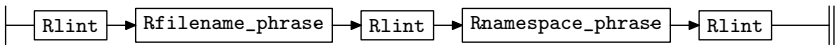


⟨ Rclose_par subrule 1 op directive 12 ⟩ ≡

```
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);
```

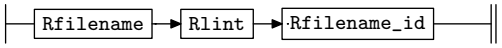
13. *Rparameters* **rule.**

Rparameters

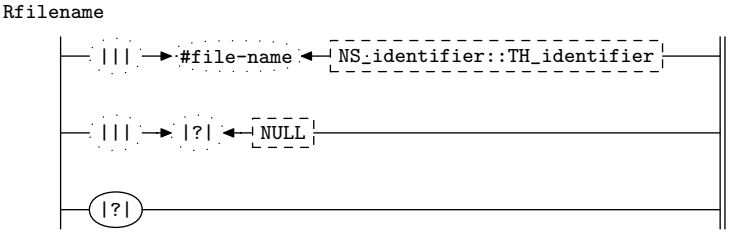


14. *Rfilename_phrase* **rule.**

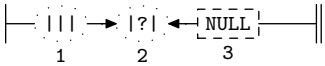
Rfilename_phrase



15. *Rfilename* **rule.**



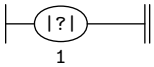
16. *Rfilename*'**s subrule 2.**



⟨ 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);
```

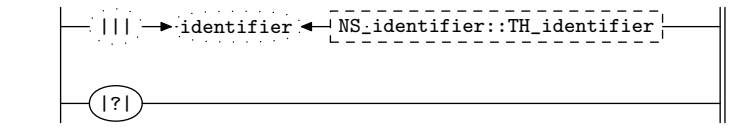
17. *Rfilename*'**s subrule 3.**



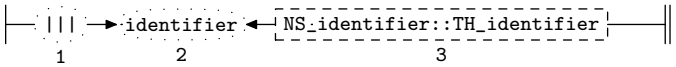
⟨ Rfilename subrule 3 op directive 17 ⟩ ≡

```
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.**



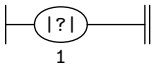
19. *Rfilename_id*'**s subrule 1.**



⟨ Rfilename_id subrule 1 op directive 19 ⟩ ≡

```
CT_enum_phrase_th * fsm = ( CT_enum_phrase_th * ) rule_info__.parser__→fsm_tbl__;
fsm→enum_phrase_→filename_id(sf→p2__);
```

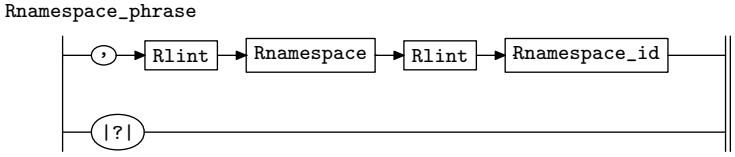
20. *Rfilename_id*'**s subrule 2.**



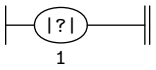
⟨ Rfilename_id subrule 2 op directive 20 ⟩ ≡

```
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.**



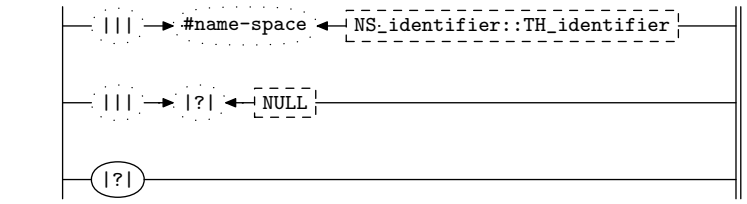
22. *Rnamespace_phrase*'**s subrule 2.**



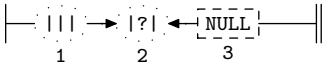
⟨ Rnamespace_phrase subrule 2 op directive 22 ⟩ ≡

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

23. *Rnamespace* **rule.**



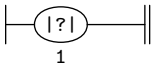
24. *Rnamespace*'**s subrule 2.**



⟨ 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.**

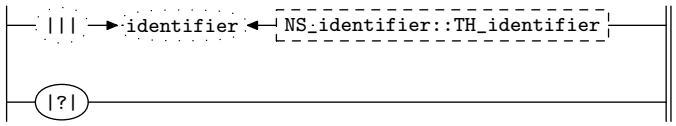


⟨ Rnamespace subrule 3 op directive 25 ⟩ ≡

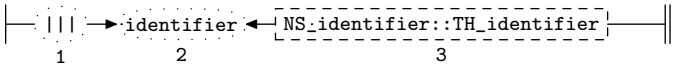
```
CAbs_lr1_sym * sym = new Err_no_namespace_present;
sym->set_rc(*rule_info__.parser__->current_token( ), __FILE__, __LINE__);
RSVP(sym);
rule_info__.parser__->set_stop_parse(true);
```

26. *Rnamespace_id* **rule.**

Rnamespace_id



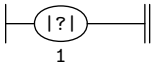
27. *Rnamespace_id*'**s subrule 1.**



⟨ Rnamespace_id subrule 1 op directive 27 ⟩ ≡

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

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

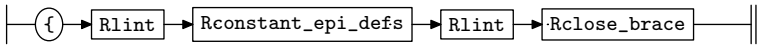


⟨ Rnamespace_id subrule 2 op directive 28 ⟩ ≡

```
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);
```

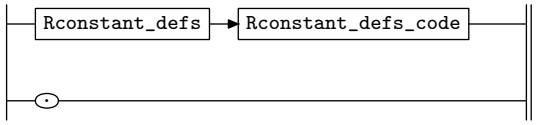
29. *Rk_defs_phrase* **rule.**

Rk_defs_phrase



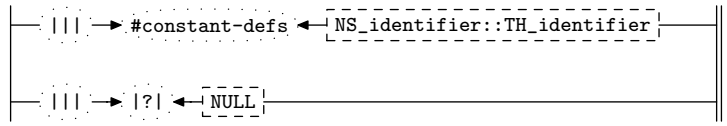
30. *Rconstant_epi_defs* **rule.**

Rconstant_epi_defs

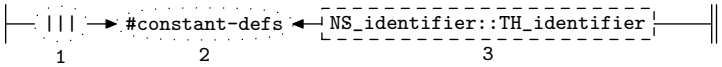


31. *Rconstant_defs* **rule.**

Rconstant_defs

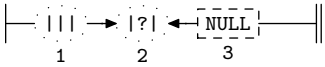


32. *Rconstant_defs*'s **subrule 1.**



⟨ Rconstant_defs subrule 1 op directive 32 ⟩ ≡
sf→p2__→set_auto_delete(*true*);

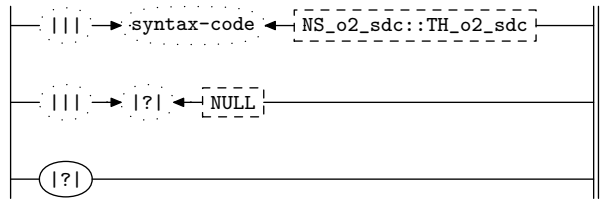
33. *Rconstant_defs*'s **subrule 2.**



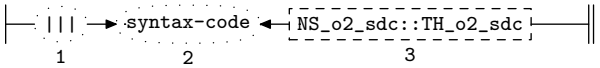
⟨ Rconstant_defs subrule 2 op directive 33 ⟩ ≡
RSVP(*sf→p2__*);
rule_info__.parser__→set_stop_parse(*true*);

34. *Rconstant_defs_code* **rule.**

Rconstant_defs_code

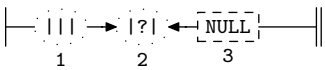


35. *Rconstant_defs_code*'s **subrule 1.**



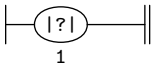
⟨ Rconstant_defs_code subrule 1 op directive 35 ⟩ ≡
CT_enum_phrase_th ∗ *fsm* = (*CT_enum_phrase_th* ∗) *rule_info__.parser__→fsm_tbl__*;
fsm→enum_phrase_→kdefs(*sf→p2__*);

36. *Rconstant_defs_code*'s **subrule 2.**



⟨ Rconstant_defs_code subrule 2 op directive 36 ⟩ ≡
CAbs_lr1_sym ∗ *sym* = **new** *Err_no_kdefs_code_present*;
sym→set_rc(∗*sf→p2__*, __FILE__, __LINE__);
sf→p2__→set_auto_delete(*true*);
RSVP(*sym*);
rule_info__.parser__→set_stop_parse(*true*);

37. *Rconstant_defs_code*'**s subrule 3.**

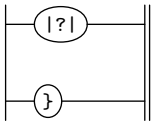


⟨ Rconstant_defs_code subrule 3 op directive 37 ⟩ ≡

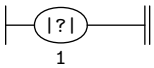
```
CAbs_lr1_sym * sym = new Err_no_kdefs_code_present;
sym->set_rc(*rule_info__.parser__->current_token( ), __FILE__, __LINE__);
RSVP(sym);
rule_info__.parser__->set_stop_parse(true);
```

38. *Rclose_brace* **rule.**

Rclose_brace



39. *Rclose_brace*'**s subrule 1.**

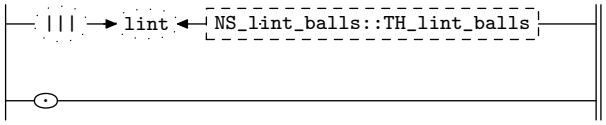


⟨ Rclose_brace subrule 1 op directive 39 ⟩ ≡

```
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);
```

40. *Rlint* **rule.**

Rlint



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

```
/*
 File: T_enum_phrase_th.fsc
 Date and Time: Sun May 11 09:59:33 2014
*/
transitive   y
grammar-name "T_enum_phrase_th"
name-space   "NS_T_enum_phrase_th"
thread-name  "TH_T_enum_phrase_th"
monolithic   n
file-name    "T_enum_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_o2_sdc::TH_o2_sdc
end-list-of-used-threads
fsm-comments
"Parse ‘‘T-enumeration’’ construct: Time out smell the tullips."
```

42. Lr1 State Network.

⇒ State: 1 state type: $^{s/r}$

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rlint		16	2	1	ϵ		1	0	1	1
c	Rlint		16	1	1	||| lint NS_lint_balls::TH_lint_balls		1	2	3	
c	RT_enum_phrase		1	1	1	Rlint *Ropen_par*		1	4	17	

⇒||| State: 2 state type: ^{s}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rlint		16	1	2	lint		1	3	3	

⇒^{lint} State: 3 state type: ^{r}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rlint		16	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	18	18	
c	Ropen_par		2	2	1	(		4	19	19	
t	RT_enum_phrase		1	1	2	Ropen_par *Rparameters*		1	5	17	

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

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rlint		16	2	1	ϵ		5	0	5	2
c	Rlint		16	1	1	||| lint NS_lint_balls::TH_lint_balls		5	2	3	
t	RT_enum_phrase		1	1	3	Rparameters *Rclose_par*		1	6	17	
c	Rparameters		4	1	1	Rlint *Rfilename_phrase*		5	20	33	

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

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rclose_par		3	1	1	|?|		6	34	34	
c	Rclose_par		3	2	1	)		6	35	35	
t	RT_enum_phrase		1	1	4	Rclose_par *Rlint*$^{\epsilon}$ *Rk_defs_phrase*		1	7	17	

⇒$^{Rclose\_par}$ State: 7 state type: $^{s/r}$

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rlint		16	2	1	ϵ		7	0	7	3
c	Rlint		16	1	1	||| lint NS_lint_balls::TH_lint_balls		7	2	3	
t	RT_enum_phrase		1	1	5	Rlint *Rk_defs_phrase*		1	8	17	

⇒^{Rlint} State: 8 state type: ^{s}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rk_defs_phrase		11	1	1	{		8	9	15	
t	RT_enum_phrase		1	1	6	Rk_defs_phrase *Rlint*$^{\epsilon}$		1	16	17	

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

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rlint		16	2	1	ϵ		9	0	9	4
c	Rlint		16	1	1	||| lint NS_lint_balls::TH_lint_balls		9	2	3	
t	Rk_defs_phrase		11	1	2	Rlint *Rconstant_epi_defs*$^{\epsilon}$ *Rlint*$^{\epsilon}$ ...		8	10	15	

⇒*Rlint* State: 10 state type: *s/r*

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
c	Rconstant_epi_defs		12	2	1	ϵ			10	0	10	4
c	Rconstant_defs		13	1	1	|||	# constant-defs NS_identifier::TH_identifier		10	36	38	
c	Rconstant_defs		13	2	1	||| |?|	NULL		10	36	37	
t	Rk_defs_phrase		11	1	3	Rconstant_epi_defs	*Rlint*$^{\epsilon}$ *Rclose_brace*		8	11	15	
c	Rconstant_epi_defs		12	1	1	Rconstant_defs	*Rconstant_defs_code*		10	39	44	

⇒*Rconstant_epi_defs* State: 11 state type: *s/r*

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
c	Rlint		16	2	1	ϵ			11	0	11	5
c	Rlint		16	1	1	|||	lint NS_lint_balls::TH_lint_balls		11	2	3	
t	Rk_defs_phrase		11	1	4	Rlint	*Rclose_brace*		8	12	15	

⇒*Rlint* State: 12 state type: *s*

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
c	Rclose_brace		15	1	1	|?|			12	13	13	
c	Rclose_brace		15	2	1	}			12	14	14	
t	Rk_defs_phrase		11	1	5	Rclose_brace			8	15	15	

⇒\|?\| State: 13 state type: *r*

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rclose_brace		15	1	2				12	0	13	1

⇒} State: 14 state type: *r*

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rclose_brace		15	2	2				12	0	14	1

⇒*Rclose_brace* State: 15 state type: *r*

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rk_defs_phrase		11	1	6				8	0	15	1

⇒*Rk_defs_phrase* State: 16 state type: *s/r*

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
c	Rlint		16	2	1	ϵ			16	0	16	1
c	Rlint		16	1	1	|||	lint NS_lint_balls::TH_lint_balls		16	2	3	
t	RT_enum_phrase		1	1	7	Rlint			1	17	17	

⇒*Rlint* State: 17 state type: *r*

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	RT_enum_phrase		1	1	8				1	0	17	1

⇒\|?\| State: 18 state type: *r*

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Ropen_par		2	1	2				4	0	18	2

⇒(State: 19 state type: *r*

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Ropen_par		2	2	2				4	0	19	2

⇒*Rlint* State: 20 state type: *s*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rfilename		6	3	1	|?|		20	45	45	
c	Rfilename		6	2	1	||| |?| NULL		20	46	47	
c	Rfilename		6	1	1	||| # file-name NS_identifier::TH_identifier		20	46	48	
t	Rparameters		4	1	2	Rfilename_phrase *Rlint*$^{\epsilon}$ *Rnamespace_phrase*		5	21	33	
c	Rfilename_phrase		5	1	1	Rfilename *Rlint*$^{\epsilon}$ *Rfilename_id*		20	49	54	

⇒*Rfilename_phrase* State: 21 state type: $^{s/r}$

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rlint		16	2	1	ϵ		21	0	21	6
c	Rlint		16	1	1	||| lint NS_lint_balls::TH_lint_balls		21	2	3	
t	Rparameters		4	1	3	Rlint *Rnamespace_phrase*		5	22	33	

⇒*Rlint* State: 22 state type: ^{s}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rnamespace_phrase		8	2	1	|?|		22	23	23	
c	Rnamespace_phrase		8	1	1	,		22	24	31	
t	Rparameters		4	1	4	Rnamespace_phrase *Rlint*$^{\epsilon}$		5	32	33	

⇒\|?\| State: 23 state type: ^{r}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rnamespace_phrase		8	2	2			22	0	23	7

⇒, State: 24 state type: $^{s/r}$

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rlint		16	2	1	ϵ		24	0	24	2
c	Rlint		16	1	1	||| lint NS_lint_balls::TH_lint_balls		24	2	3	
t	Rnamespace_phrase		8	1	2	Rlint *Rnamespace*		22	25	31	

⇒*Rlint* State: 25 state type: ^{s}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rnamespace		9	3	1	|?|		25	55	55	
c	Rnamespace		9	1	1	||| # name-space NS_identifier::TH_identifier		25	56	58	
c	Rnamespace		9	2	1	||| |?| NULL		25	56	57	
t	Rnamespace_phrase		8	1	3	Rnamespace *Rlint*$^{\epsilon}$ *Rnamespace_id*		22	26	31	

⇒*Rnamespace* State: 26 state type: $^{s/r}$

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rlint		16	2	1	ϵ		26	0	26	2
c	Rlint		16	1	1	||| lint NS_lint_balls::TH_lint_balls		26	2	3	
t	Rnamespace_phrase		8	1	4	Rlint *Rnamespace_id*		22	27	31	

⇒*Rlint* State: 27 state type: ^{s}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rnamespace_id		10	2	1	|?|		27	28	28	
c	Rnamespace_id		10	1	1	||| identifier NS_identifier::TH_identifier		27	29	30	
t	Rnamespace_phrase		8	1	5	Rnamespace_id		22	31	31	

⇒\|?\| State: 28 state type: ^{r}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rnamespace_id		10	2	2			27	0	28	7

⇒||| State: 29 state type: ^{s}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rnamespace_id		10	1	2	identifier			27	30	30	

⇒*identifier* State: 30 state type: ^{r}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rnamespace_id		10	1	3				27	0	30	7

⇒*Rnamespace_id* State: 31 state type: ^{r}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rnamespace_phrase		8	1	6				22	0	31	7

⇒*Rnamespace_phrase* State: 32 state type: $^{s/r}$

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
c	Rlint		16	2	1	ϵ			32	0	32	8
c	Rlint		16	1	1	||| lint NS_lint_balls::TH_lint_balls		32	2	3	
t	Rparameters		4	1	5	Rlint			5	33	33	

⇒*Rlint* State: 33 state type: ^{r}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rparameters		4	1	6				5	0	33	8

⇒|?| State: 34 state type: ^{r}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rclose_par		3	1	2				6	0	34	9

⇒) State: 35 state type: ^{r}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rclose_par		3	2	2				6	0	35	9

⇒||| State: 36 state type: ^{s}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rconstant_defs		13	2	2	|?|			10	37	37	
t	Rconstant_defs		13	1	2	# constant-defs			10	38	38	

⇒|?| State: 37 state type: ^{r}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rconstant_defs		13	2	3				10	0	37	2

⇒*#constant−defs* State: 38 state type: ^{r}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rconstant_defs		13	1	3				10	0	38	2

⇒*Rconstant_defs* State: 39 state type: ^{s}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
c	Rconstant_defs_code		14	3	1	|?|			39	40	40	
c	Rconstant_defs_code		14	2	1	||| |?| NULL		39	41	42	
c	Rconstant_defs_code		14	1	1	||| syntax-code NS_o2_sdc::TH_o2_sdc		39	41	43	
t	Rconstant_epi_defs		12	1	2	Rconstant_defs_code			10	44	44	

⇒|?| State: 40 state type: ^{r}

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rconstant_defs_code			14	3	2				39	0	40	4

⇒|||

State: 41 state type: ^{s}

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rconstant_defs_code			14	2	2	|?|			39	42	42	
t	Rconstant_defs_code			14	1	2	syntax-code			39	43	43	

⇒|?|

State: 42 state type: ^{r}

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rconstant_defs_code			14	2	3				39	0	42	4

⇒$^{syntax-code}$

State: 43 state type: ^{r}

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rconstant_defs_code			14	1	3				39	0	43	4

⇒$^{Rconstant\_defs\_code}$

State: 44 state type: ^{r}

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rconstant_epi_defs			12	1	3				10	0	44	4

⇒|?|

State: 45 state type: ^{r}

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rfilename			6	3	2				20	0	45	2

⇒|||

State: 46 state type: ^{s}

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rfilename			6	2	2	|?|			20	47	47	
t	Rfilename			6	1	2	# file-name			20	48	48	

⇒|?|

State: 47 state type: ^{r}

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rfilename			6	2	3				20	0	47	2

⇒$^{\#file-name}$

State: 48 state type: ^{r}

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rfilename			6	1	3				20	0	48	2

⇒^{Rfilename}

State: 49 state type: $^{s/r}$

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
c	Rlint			16	2	1	ϵ			49	0	49	2
c	Rlint			16	1	1	||| lint NS_lint_balls::TH_lint_balls			49	2	3	
t	Rfilename_phrase			5	1	2	Rlint $\underline{Rfilename\_id}$			20	50	54	

⇒^{Rlint}

State: 50 state type: ^{s}

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
c	Rfilename_id			7	2	1	|?|			50	51	51	
c	Rfilename_id			7	1	1	||| identifier NS_identifier::TH_identifier			50	52	53	
t	Rfilename_phrase			5	1	3	Rfilename_id			20	54	54	

⇒|?|

State: 51 state type: ^{r}

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rfilename_id			7	2	2				50	0	51	10

⇒`|||` State: 52 state type: ^{s}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rfilename_id		7	1	2	identifier		50	53	53	

⇒^{identifier} State: 53 state type: ^{r}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rfilename_id		7	1	3			50	0	53	10

⇒$^{Rfilename\_id}$ State: 54 state type: ^{r}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rfilename_phrase		5	1	4			20	0	54	10

⇒`|?|` State: 55 state type: ^{r}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rnamespace		9	3	2			25	0	55	2

⇒`|||` State: 56 state type: ^{s}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rnamespace		9	2	2	`|?|`		25	57	57	
t	Rnamespace		9	1	2	`#` name-space		25	58	58	

⇒`|?|` State: 57 state type: ^{r}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rnamespace		9	2	3			25	0	57	2

⇒$^{\#name-space}$ State: 58 state type: ^{r}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rnamespace		9	1	3			25	0	58	2

43. Index.

T_enum_phrase_th Grammar

Date: May 17, 2014 at 13:33

File: T_enum_phrase_th.lex Ns: NS_T_enum_phrase_th

Version: 1.0 Debug: false

Grammar Comments: Type: Thread

Parse “T-enumeration” construct: Time out smell the tullips.

1 element(s) in Lookahead Expression below

eolr