1. Copyright.

2. *subrule_def* **Thread.**
Parse a rule's subrule expression.
Example of a parallel-parser construct to parse:

```
/*
file: subrdef.txt
Why: some text examples of sub rule construct.
*/

  ->  ||| "#NULL" NS_identifier::TH_identifier {
    op
      Csubrule_def* fsm = (Csubrule_def*)parser()->fsm_tbl();
      fsm->subrule_def_->add_elem_to_subrule_vector(sf->p2__);
    ***
    }
  ->  |?| {
      op
        CAbs_lr1_sym* sym = new LR1_err_no_open_brace;
        sym->set_rc(*parser()->current_token(),*parser());
        RSVP(sym);
        parser()->set_stop_parse(true);
      ***
      }
 ->  "{"
 -> Rvector Rlint Rrhs_expr
```

3. Fsm Csubrule_def class.

4. Csubrule_def constructor directive.

⟨ Csubrule_def constructor directive 4 ⟩ ≡
*subrule_def_* = 0;

5. Csubrule_def op directive.

⟨ Csubrule_def op directive 5 ⟩ ≡
if (*subrule_def_* ≠ 0) {
delete *subrule_def_*;
*subrule_def_* = 0;
}
*subrule_def_* = **new** *T_subrule_def*;
*subrule_def_*→*set_rc*(∗*parser__*→*start_token__*, __FILE__, __LINE__);

6. Csubrule_def user-declaration directive.

⟨ Csubrule_def user-declaration directive 6 ⟩ ≡
public: **void** *add_sdc_to_directive*(*yacco2* :: *CAbs_lr1_sym* ∗ *Dir*, *T_syntax_code* ∗ *Sdc*);
T_subrule_def ∗ *subrule_def_*;

7. Csubrule_def user-implementation directive.

⟨ Csubrule_def user-implementation directive 7 ⟩ ≡

```
void Csubrule_def::add_sdc_to_directive(yacco2::CAbs_lr1_sym * Dir, T_syntax_code * Sdc){ using
        namespace NS_yacco2_T_enum;
    using namespace NS_yacco2_terminals;
    yacco2::INT eid = Dir→enumerated_id__; switch (eid) { case T_Enum::T_T_op_: { T_op * k = (
        T_op * ) Dir;
    k→syntax_code(Sdc);
    break; }
  default:
    {
      CAbs_lr1_sym * sym = new Err_improper_directive;
      sym→set_rc(*Dir, __FILE__, __LINE__);
      RSVP_FSM(sym);
      parser__→set_stop_parse(true);
    }
    } }
```

8. Csubrule_def user-prefix-declaration directive.

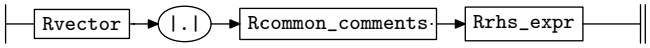
⟨ Csubrule_def user-prefix-declaration directive 8 ⟩ ≡

```
  using namespace NS_yacco2_terminals;
#include "lint_balls.h"
#include "cweb_or_c_k.h"
#include "subrule_vector.h"
#include "o2_sdc.h"
#include "identifier.h"
#include "dbl_colon.h"
#include "rhs_component.h"
#include "rtn_component.h"
#include "rhs_bnd.h"
#include "parallel_oper.h"
```

9. *Rsubrule_def* rule.

Use of `|.|` to make grammar lr(1) due to s/r conflict on `|||`.

Rsubrule_def

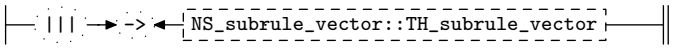


⟨ Rsubrule_def subrule 1 op directive 9 ⟩ ≡

```
Csubrule_def * fsm = ( Csubrule_def * ) rule_info__.parser__→fsm_tbl__;
fsm→subrule_def_→bld_its_tree( );
RSVP(fsm→subrule_def_);
fsm→subrule_def_ = 0;
```

10. *Rvector* rule.

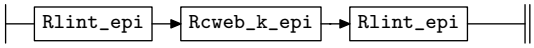
Rvector



11. *Rcommon_comments* **rule.**

Basic possible lint balls and a single cweb comment fronting the sub rule definition.

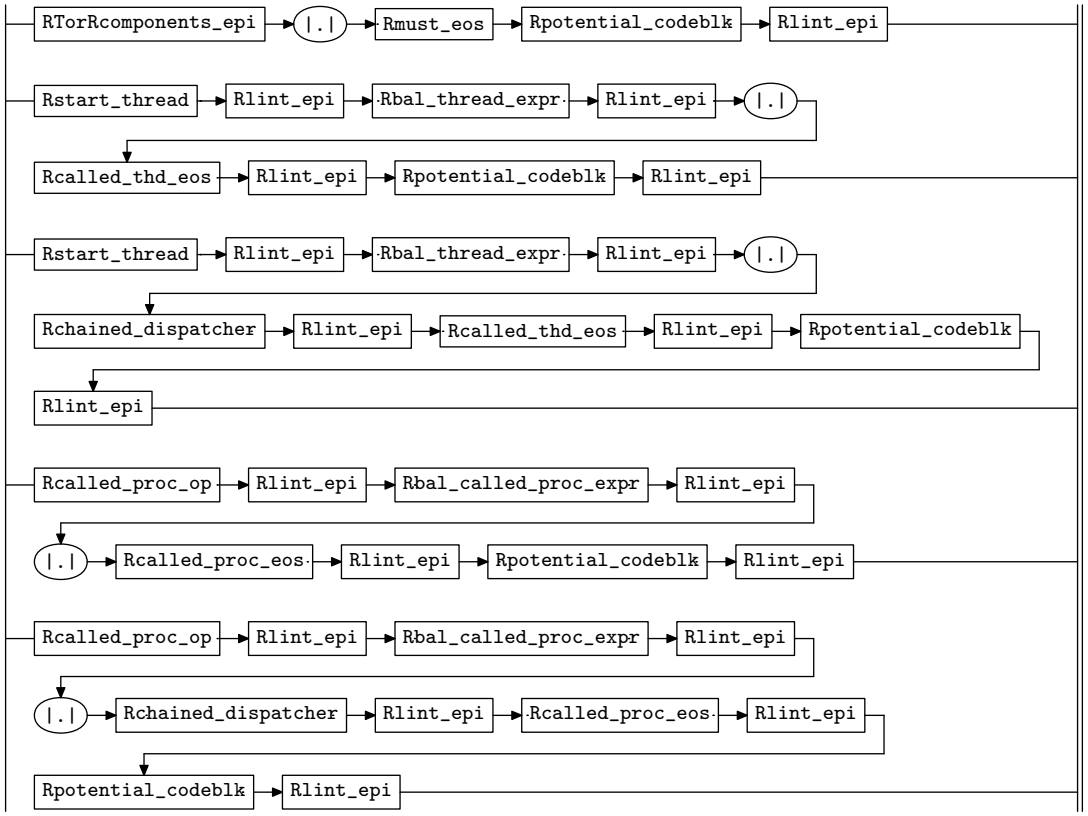
Rcommon_comments



12. *Rrhs_expr* **rule.**

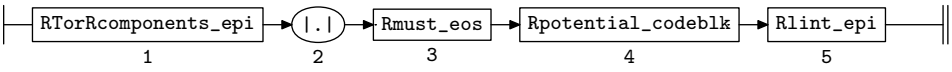
Use of |.| to make grammar lr(1) due to s/r conflict on |||.

Rrhs_expr



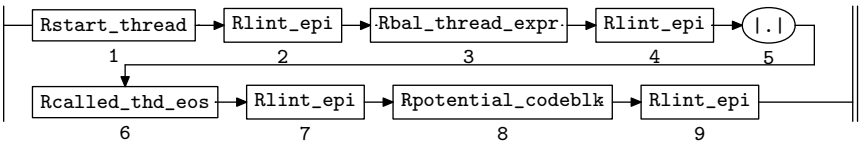
13. *Rrhs_expr*'**s subrule 1.**

A regular grammar expression containing either ϵ or an α string of symbols drawn from the T or R vocabularies. Why the |.| symbol following *RTorRcomponents_epi*? There is a shift / reduce conflict caused by the ||| so let the ϵ reduce on this meta terminal.



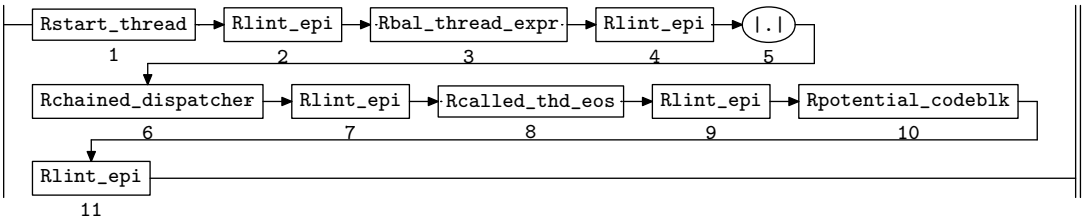
14. *Rrhs_expr*'**s subrule 2.**

Your standard thread call expression.



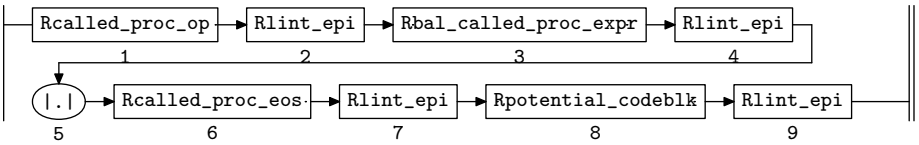
15. *Rrhs_expr*'**s subrule 3.**

The new back-to-back chained thread call dispatcher expression. *Rchained_dispatcher* rule supports nested chained calls. The returned output from the `|t|` expression becomes the input to the next *Rchained_dispatcher* `|t|` expression.



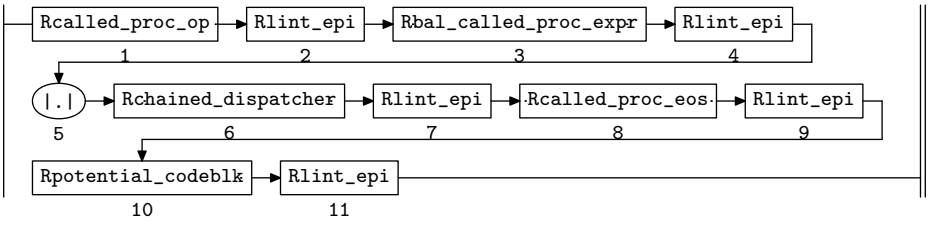
16. *Rrhs_expr*'**s subrule 4.**

Your chained procedure call expression. It can be used against a previous T element. Note: a R element is not allowed as the chained procedure call uses the stacked T of the caller parser as its First set. So what does *Rxxx* mean to it? nada.



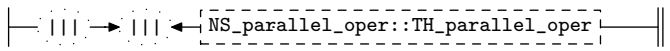
17. *Rrhs_expr*'**s subrule 5.**

The new back-to-back chained thread call dispatcher expression.



18. *Rstart_thread* **rule.**

Rstart_thread



⟨ Rstart_thread subrule 1 op directive 18 ⟩ ≡

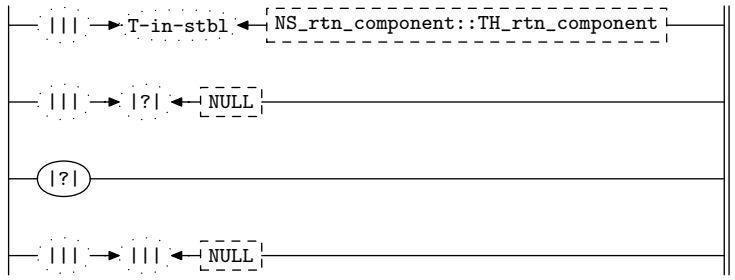
```
Csubrule_def * fsm = ( Csubrule_def * ) rule_info__.parser__→fsm_tbl__;
T_sym_tbl_report_card report_card;
using namespace yacco2_stbl;
find_sym_in_stbl(report_card, *sf→p2__→id__);
if (report_card.action_ ≡ T_sym_tbl_report_card::not_fnd) {
  CAbs_lr1_sym * sym = new Err_subrule_use_undefined_T;
  sym→set_rc(*rule_info__.parser__→start_token__, __FILE__, __LINE__);
  RSVP(sym);
  rule_info__.parser__→set_stop_parse(true);
  return;
}
CAbs_lr1_sym * sym = report_card.tbl_entry_→symbol_; T_in_stbl * t = ( T_in_stbl * ) sym;
refered_T * rT = *(t→xref_t( )→rbegin( ));
fsm→subrule_def_→add_elem_to_subrule_vector(rT);
rT→its_subrule_def(fsm→subrule_def_);
```

19. *Rbal_thread_expr* **rule.**

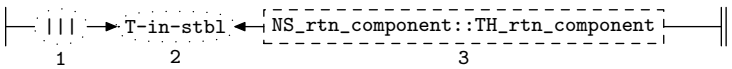
Rbal_thread_expr

20. *Rrtned_T* **rule.**

Rrtned_T



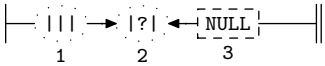
21. *Rrtned_T*'**s subrule 1.**



⟨ Rrtned_T subrule 1 op directive 21 ⟩ ≡

```
Csubrule_def * fsm = ( Csubrule_def * ) rule_info__.parser__→fsm_tbl__;
refered_T * reft = *(sf→p2__→xref_t( )→rbegin( ));
fsm→subrule_def_→add_elem_to_subrule_vector(reft);
reft→its_subrule_def(fsm→subrule_def_);
```

22. *Rrtned_T*'**s subrule 2.**

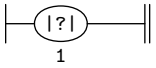


Other returned types are errors.

⟨ Rrtned_T subrule 2 op directive 22 ⟩ ≡

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

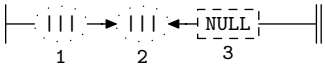
23. *Rrtned_T*'**s subrule 3.**



⟨ Rrtned_T subrule 3 op directive 23 ⟩ ≡

```
CAbs_lr1_sym * sym = new Err_not_T_for_rtned_token_from_th( );
sym→set_rc(*sf→p1__, __FILE__, __LINE__);
RSVP(sym);
rule_info__.parser__→set_stop_parse(true);
```

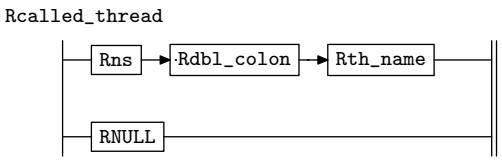
24. *Rrtned_T*'**s subrule 4.**



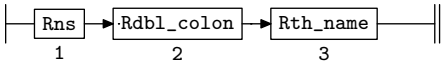
⟨ Rrtned_T subrule 4 op directive 24 ⟩ ≡

```
Csubrule_def * fsm = ( Csubrule_def * ) rule_info__.parser__→fsm_tbl__;
fsm→subrule_def_→add_elem_to_subrule_vector(sf→p2__);
```

25. *Rcalled_thread* **rule.**



26. *Rcalled_thread*'**s subrule 1.**

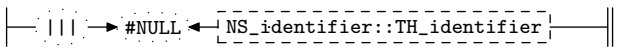


⟨ Rcalled_thread subrule 1 op directive 26 ⟩ ≡

```
Csubrule_def * fsm = ( Csubrule_def * ) rule_info__.parser__→fsm_tbl__;
T_called_thread_eosubrule * sym = new T_called_thread_eosubrule(sf→p1__→ns_, sf→p3__→thd_nm_);
sym→set_rc(*sf→p1__→ns_, __FILE__, __LINE__);
sym→its_subrule_def(fsm→subrule_def_);
fsm→subrule_def_→add_elem_to_subrule_vector(sym);
```

27. RNULL rule.

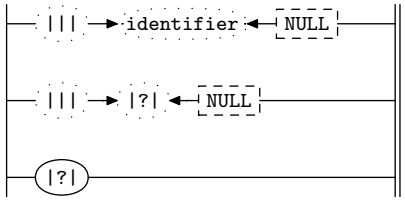
RNULL



⟨ RNULL subrule 1 op directive 27 ⟩ ≡
Csubrule_def ∗ *fsm* = (*Csubrule_def* ∗) *rule_info__.parser__→fsm_tbl__*;
T_null_call_thread_eosubrule ∗ *sym* = **new** *T_null_call_thread_eosubrule*;
sym→set_rc(∗*sf→p2__*, __FILE__, __LINE__);
sym→its_subrule_def(*fsm→subrule_def_*);
fsm→subrule_def_→add_elem_to_subrule_vector(*sym*);
sf→p2__→set_auto_delete(*true*);

28. *Rns* rule.

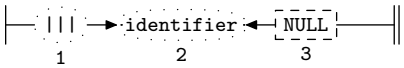
Rns



29. Rns user-declaration directive.

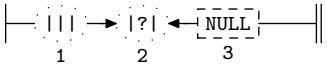
⟨ Rns user-declaration directive 29 ⟩ ≡
public: *T_identifier* ∗ *ns_*;

30. *Rns*'s subrule 1.



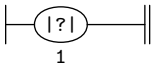
⟨ Rns subrule 1 op directive 30 ⟩ ≡
*ns_* = *sf→p2__*;

31. *Rns*'s subrule 2.



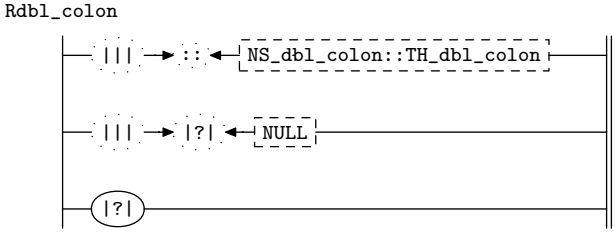
⟨ Rns subrule 2 op directive 31 ⟩ ≡
CAbs_lr1_sym ∗ *sym* = **new** *Err_not_id_for_ns_in_th_stmt*( );
sym→set_rc(∗*sf→p2__*, __FILE__, __LINE__);
RSVP(*sym*);
rule_info__.parser__→set_stop_parse(*true*);

32. *Rns*'**s subrule 3.**

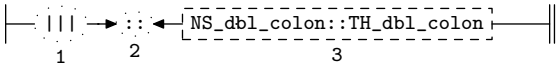


⟨ Rns subrule 3 op directive 32 ⟩ ≡
```
CAbs_lr1_sym * sym = new Err_not_id_for_ns_in_th_stmt( );
sym→set_rc(*sf→p1__, __FILE__, __LINE__);
RSVP(sym);
rule_info__.parser__→set_stop_parse(true);
```

33. *Rdbl_colon* **rule.**

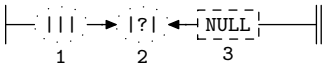


34. *Rdbl_colon*'**s subrule 1.**



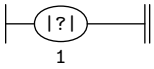
⟨ Rdbl_colon subrule 1 op directive 34 ⟩ ≡
```
sf→p2__→set_auto_delete(true);
```

35. *Rdbl_colon*'**s subrule 2.**



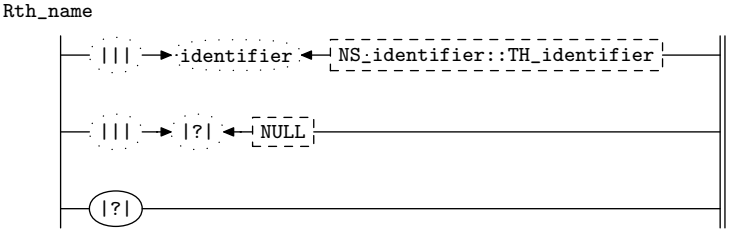
⟨ Rdbl_colon subrule 2 op directive 35 ⟩ ≡
```
CAbs_lr1_sym * sym = new Err_not_dbl_colon_in_th_stmt( );
sym→set_rc(*sf→p2__, __FILE__, __LINE__);
RSVP(sym);
rule_info__.parser__→set_stop_parse(true);
```

36. *Rdbl_colon*'**s subrule 3.**



⟨ Rdbl_colon subrule 3 op directive 36 ⟩ ≡
```
CAbs_lr1_sym * sym = new Err_not_dbl_colon_in_th_stmt( );
sym→set_rc(*sf→p1__, __FILE__, __LINE__);
RSVP(sym);
rule_info__.parser__→set_stop_parse(true);
```

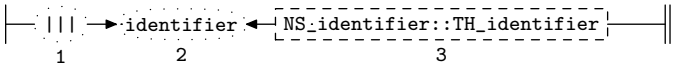
37. *Rth_name* **rule.**



38. Rth_name user-declaration directive.

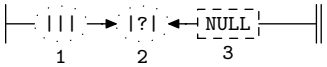
⟨ Rth_name user-declaration directive 38 ⟩ ≡
public: *T_identifier* ∗ *thd_nm_*;

39. *Rth_name*'**s subrule 1.**



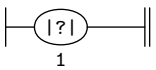
⟨ Rth_name subrule 1 op directive 39 ⟩ ≡
*thd_nm_* = *sf*→*p2__*;

40. *Rth_name*'**s subrule 2.**



⟨ Rth_name subrule 2 op directive 40 ⟩ ≡
CAbs_lr1_sym ∗ *sym* = **new** *Err_not_id_for_th_name_in_th_stmt*( );
sym→*set_rc*(∗*sf*→*p2__*, __FILE__, __LINE__);
RSVP(*sym*);
rule_info__.parser__→*set_stop_parse*(*true*);

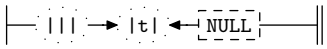
41. *Rth_name*'**s subrule 3.**



⟨ Rth_name subrule 3 op directive 41 ⟩ ≡
CAbs_lr1_sym ∗ *sym* = **new** *Err_not_id_for_th_name_in_th_stmt*( );
sym→*set_rc*(∗*sf*→*p1__*, __FILE__, __LINE__);
RSVP(*sym*);
rule_info__.parser__→*set_stop_parse*(*true*);

42. *Rcalled_proc_op* **rule.**

Rcalled_proc_op

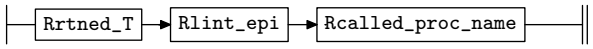


⟨ Rcalled_proc_op subrule 1 op directive 42 ⟩ ≡

```
Csubrule_def * fsm = ( Csubrule_def * ) rule_info__.parser__→fsm_tbl__;
T_sym_tbl_report_card report_card;
using namespace yacco2_stbl;
find_sym_in_stbl(report_card, *sf→p2__→id__);
if (report_card.action_ ≡ T_sym_tbl_report_card::not_fnd) {
  CAbs_lr1_sym * sym = new Err_subrule_use_undefined_T;
  sym→set_rc(*rule_info__.parser__→start_token__, __FILE__, __LINE__);
  RSVP(sym);
  rule_info__.parser__→set_stop_parse(true);
  return;
}
CAbs_lr1_sym * sym = report_card.tbl_entry_→symbol_; T_in_stbl * t = ( T_in_stbl * ) sym;
refered_T * rT = *(t→xref_t( )→rbegin( ));
fsm→subrule_def_→add_elem_to_subrule_vector(rT);
rT→its_subrule_def(fsm→subrule_def_);
```

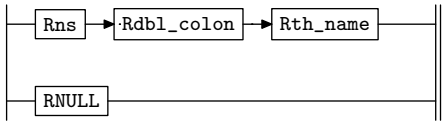
43. *Rbal_called_proc_expr* **rule.**

Rbal_called_proc_expr

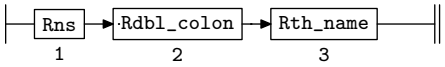


44. *Rcalled_proc_name* **rule.**

Rcalled_proc_name



45. *Rcalled_proc_name*'**s subrule 1.**

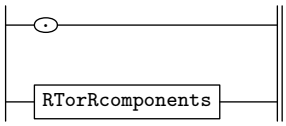


⟨ Rcalled_proc_name subrule 1 op directive 45 ⟩ ≡

```
Csubrule_def * fsm = ( Csubrule_def * ) rule_info__.parser__→fsm_tbl__;
T_called_thread_eosubrule * sym = new T_called_thread_eosubrule(sf→p1__→ns_, sf→p3__→thd_nm_);
sym→set_rc(*sf→p1__→ns_, __FILE__, __LINE__);
sym→its_subrule_def(fsm→subrule_def_);
fsm→subrule_def_→add_elem_to_subrule_vector(sym);
```

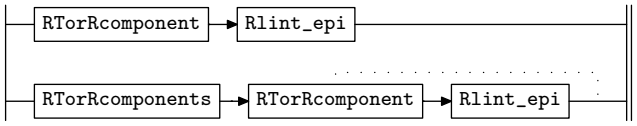
46. *RTorRcomponents_epi* **rule.**

RTorRcomponents_epi



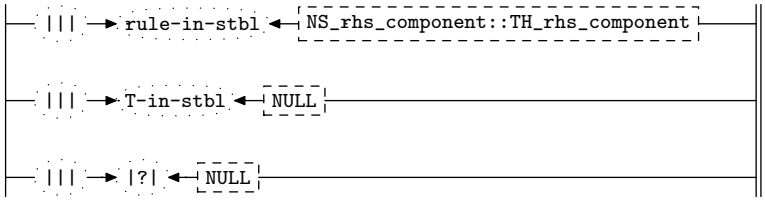
47. *RTorRcomponents* **rule.**

RTorRcomponents

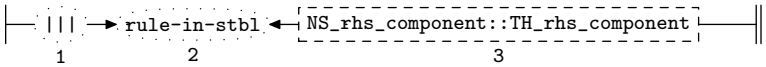


48. *RTorRcomponent* **rule.**

RTorRcomponent



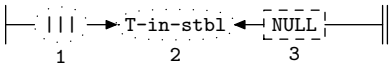
49. *RTorRcomponent*'**s subrule 1.**



⟨ RTorRcomponent subrule 1 op directive 49 ⟩ ≡

```
Csubrule_def * fsm = ( Csubrule_def * ) rule_info__.parser__→fsm_tbl__;
refered_rule * refr = *(sf→p2__→xref_r( )→rbegin( ));
fsm→subrule_def_→add_elem_to_subrule_vector(refr);
refr→its_subrule_def(fsm→subrule_def_);
```

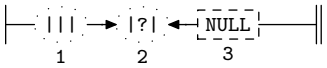
50. *RTorRcomponent*'**s subrule 2.**



⟨ RTorRcomponent subrule 2 op directive 50 ⟩ ≡

```
Csubrule_def * fsm = ( Csubrule_def * ) rule_info__.parser__→fsm_tbl__;
refered_T * reft = *(sf→p2__→xref_t( )→rbegin( ));
fsm→subrule_def_→add_elem_to_subrule_vector(reft);
reft→its_subrule_def(fsm→subrule_def_);
```

51. *RTorRcomponent*'s subrule 3.

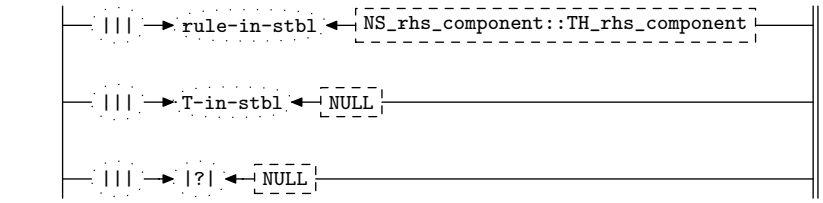


⟨ RTorRcomponent subrule 3 op directive 51 ⟩ ≡

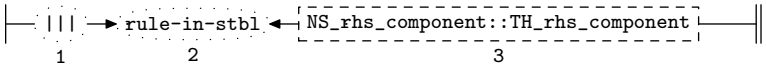
```
CAbs_lr1_sym * sym = new Err_not_T_or_R_in_subrule_expr( );
sym→set_rc(*sf→p2__, __FILE__, __LINE__);
RSVP(sym);
rule_info__.parser__→set_stop_parse(true);
```

52. *Rchained_dispatcher* rule.

Rchained_dispatcher



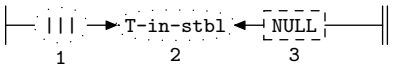
53. *Rchained_dispatcher*'s subrule 1.



⟨ Rchained_dispatcher subrule 1 op directive 53 ⟩ ≡

```
Csubrule_def * fsm = ( Csubrule_def * ) rule_info__.parser__→fsm_tbl__;
refered_rule * refr = *(sf→p2__→xref_r( )→rbegin( ));
fsm→subrule_def_→add_elem_to_subrule_vector(refr);
refr→its_subrule_def(fsm→subrule_def_);
```

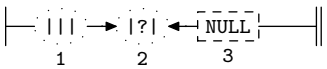
54. *Rchained_dispatcher*'s subrule 2.



⟨ Rchained_dispatcher subrule 2 op directive 54 ⟩ ≡

```
Csubrule_def * fsm = ( Csubrule_def * ) rule_info__.parser__→fsm_tbl__;
CAbs_lr1_sym * sym = new Err_not_a_Rule( );
sym→set_rc(*sf→p2__, __FILE__, __LINE__);
RSVP(sym);
rule_info__.parser__→set_stop_parse(true);
```

55. *Rchained_dispatcher*'s subrule 3.

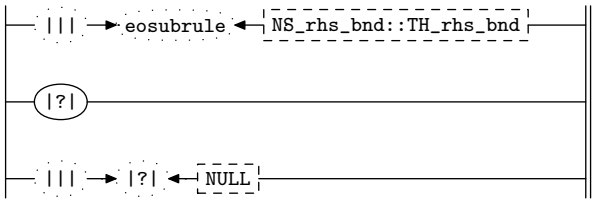


⟨ Rchained_dispatcher subrule 3 op directive 55 ⟩ ≡

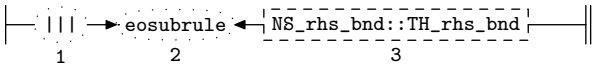
```
CAbs_lr1_sym * sym = new Err_not_a_Rule( );
sym→set_rc(*sf→p2__, __FILE__, __LINE__);
RSVP(sym);
rule_info__.parser__→set_stop_parse(true);
```

56. *Rmust_eos* **rule.**

Rmust_eos



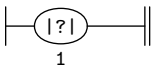
57. *Rmust_eos*'**s subrule 1.**



⟨ Rmust_eos subrule 1 op directive 57 ⟩ ≡

```
Csubrule_def * fsm = ( Csubrule_def * ) rule_info__.parser__→fsm_tbl__;
fsm→subrule_def_→add_elem_to_subrule_vector(sf→p2__);
sf→p2__→its_subrule_def(fsm→subrule_def_);
```

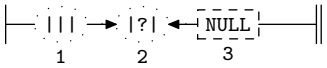
58. *Rmust_eos*'**s subrule 2.**



⟨ Rmust_eos subrule 2 op directive 58 ⟩ ≡

```
CAbs_lr1_sym * sym = new Err_not_eos_in_subrule_expr( );
sym→set_rc(*sf→p1__, __FILE__, __LINE__);
RSVP(sym);
rule_info__.parser__→set_stop_parse(true);
```

59. *Rmust_eos*'**s subrule 3.**

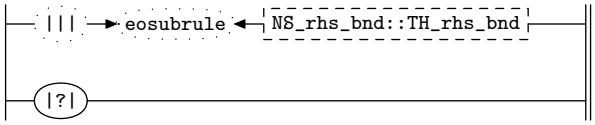


⟨ Rmust_eos subrule 3 op directive 59 ⟩ ≡

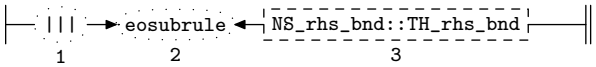
```
CAbs_lr1_sym * sym = new Err_not_eos_in_subrule_expr( );
sym→set_rc(*sf→p2__, __FILE__, __LINE__);
RSVP(sym);
rule_info__.parser__→set_stop_parse(true);
sf→p2__→set_auto_delete(true);
```

60. *Rcalled_thd_eos* **rule.**

Rcalled_thd_eos



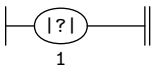
61. *Rcalled_thd_eos*'s subrule 1.



⟨ Rcalled_thd_eos subrule 1 op directive 61 ⟩ ≡

```
Csubrule_def * fsm = ( Csubrule_def * ) rule_info__.parser__→fsm_tbl__;
fsm→subrule_def_→add_elem_to_subrule_vector(sf→p2__);
sf→p2__→its_subrule_def(fsm→subrule_def_);
```

62. *Rcalled_thd_eos*'s subrule 2.

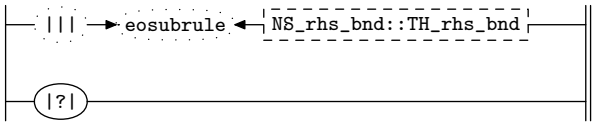


⟨ Rcalled_thd_eos subrule 2 op directive 62 ⟩ ≡

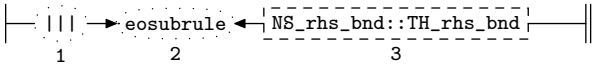
```
CAbs_lr1_sym * sym = new Err_not_eos_in_subrule_expr( );
sym→set_rc(*sf→p1__, __FILE__, __LINE__);
RSVP(sym);
rule_info__.parser__→set_stop_parse(true);
```

63. *Rcalled_proc_eos* rule.

Rcalled_proc_eos



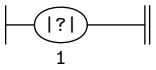
64. *Rcalled_proc_eos*'s subrule 1.



⟨ Rcalled_proc_eos subrule 1 op directive 64 ⟩ ≡

```
Csubrule_def * fsm = ( Csubrule_def * ) rule_info__.parser__→fsm_tbl__;
fsm→subrule_def_→add_elem_to_subrule_vector(sf→p2__);
sf→p2__→its_subrule_def(fsm→subrule_def_);
```

65. *Rcalled_proc_eos*'s subrule 2.

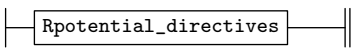


⟨ Rcalled_proc_eos subrule 2 op directive 65 ⟩ ≡

```
CAbs_lr1_sym * sym = new Err_not_eos_in_subrule_expr( );
sym→set_rc(*sf→p1__, __FILE__, __LINE__);
RSVP(sym);
rule_info__.parser__→set_stop_parse(true);
```

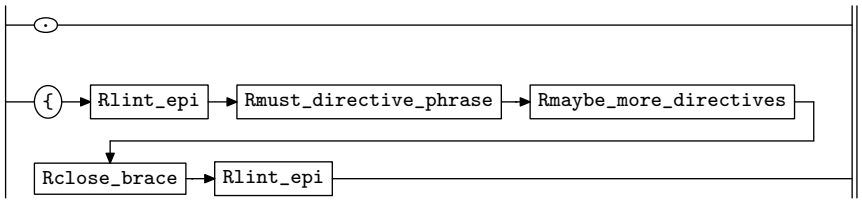
66. *Rpotential_codeblk* rule.

Rpotential_codeblk



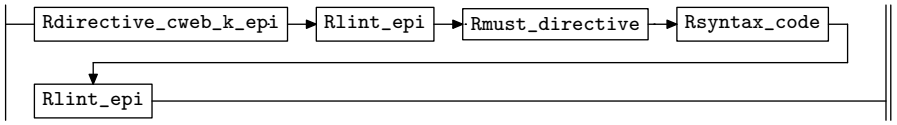
67. *Rpotential_directives* **rule.**

Rpotential_directives



68. *Rmust_directive_phrase* **rule.**

Rmust_directive_phrase

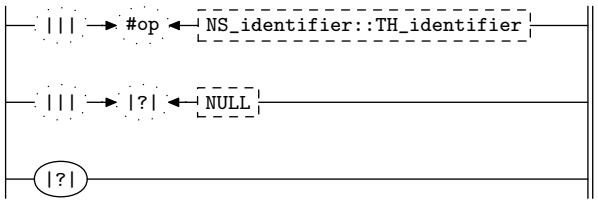


⟨ Rmust_directive_phrase subrule 1 op directive 68 ⟩ ≡

```
AST * cwebt = sf→p1__→cweb_t_;
Rmust_directive * dir = sf→p3__;
Rsyntax_code * sdc = sf→p4__;
if (cwebt ≠ 0) {
  sdc→syntax_code_→add_cweb_marker(cwebt);
}
Csubrule_def * fsm = ( Csubrule_def * ) rule_info__.parser__→fsm_tbl__;
fsm→add_sdc_to_directive(dir→directive_, sdc→syntax_code_);
```

69. *Rmust_directive* **rule.**

Rmust_directive



70. Rmust_directive constructor directive.

⟨ Rmust_directive constructor directive 70 ⟩ ≡

```
directive_ = 0;
```

71. Rmust_directive op directive.

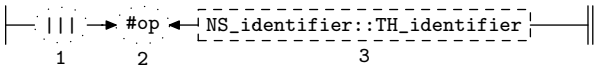
⟨ Rmust_directive op directive 71 ⟩ ≡

```
if (directive_ ≡ 0) return;
Csubrule_def * fsm = ( Csubrule_def * ) rule_info__.parser__→fsm_tbl__;
CAbs_lr1_sym * result = fsm→subrule_def_→add_directive_to_subrule(directive_, rule_info__.parser__);
if (result ≡ 0) return;      /* ok added */
directive_→set_auto_delete(true);      /* dup: delete when popped from stack */
RSVP(result);
rule_info__.parser__→set_stop_parse(true);
```

72. Rmust_directive user-declaration directive.

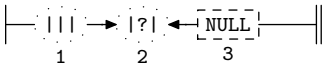
⟨ Rmust_directive user-declaration directive 72 ⟩ ≡
CAbs_lr1_sym ∗ *directive_*;

73. *Rmust_directive*'**s subrule 1.**



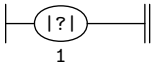
⟨ Rmust_directive subrule 1 op directive 73 ⟩ ≡
*directive_* = *sf*→*p2__*;

74. *Rmust_directive*'**s subrule 2.**



⟨ Rmust_directive subrule 2 op directive 74 ⟩ ≡
*directive_* = 0;
sf→*p2__*→*set_auto_delete*(*true*);
CAbs_lr1_sym ∗ *sym* = **new** *Err_bad_directive*;
sym→*set_rc*(∗*sf*→*p2__*, __FILE__, __LINE__);
RSVP(*sym*);
rule_info__.parser__→*set_stop_parse*(*true*);

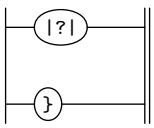
75. *Rmust_directive*'**s subrule 3.**



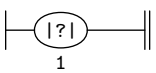
⟨ Rmust_directive subrule 3 op directive 75 ⟩ ≡
CAbs_lr1_sym ∗ *sym* = **new** *Err_no_directive_present*;
sym→*set_rc*(∗*rule_info__.parser__*→*current_token*( ), __FILE__, __LINE__);
RSVP(*sym*);
rule_info__.parser__→*set_stop_parse*(*true*);

76. *Rclose_brace* **rule.**

Rclose_brace



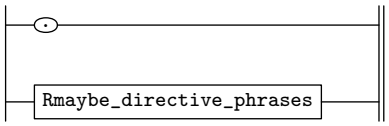
77. *Rclose_brace*'**s subrule 1.**



⟨ Rclose_brace subrule 1 op directive 77 ⟩ ≡
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*);

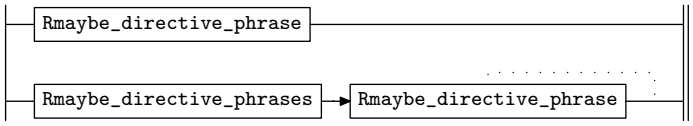
78. *Rmaybe_more_directives* **rule.**

Rmaybe_more_directives



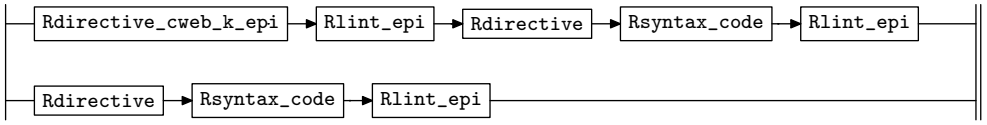
79. *Rmaybe_directive_phrases* **rule.**

Rmaybe_directive_phrases



80. *Rmaybe_directive_phrase* **rule.**

Rmaybe_directive_phrase

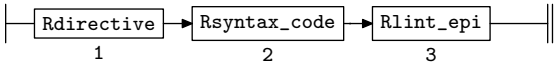


81. *Rmaybe_directive_phrase*'**s subrule 1.**

⟨ Rmaybe_directive_phrase subrule 1 op directive 81 ⟩ ≡

```
AST * cwebt = sf→p1__→cweb_t_;
Rdirective * dir = sf→p3__;
Rsyntax_code * sdc = sf→p4__;
if (cwebt ≠ 0) {
  sdc→syntax_code_→add_cweb_marker(cwebt);
}
Csubrule_def * fsm = ( Csubrule_def * ) rule_info__.parser__→fsm_tbl__;
fsm→add_sdc_to_directive(dir→directive_, sdc→syntax_code_);
```

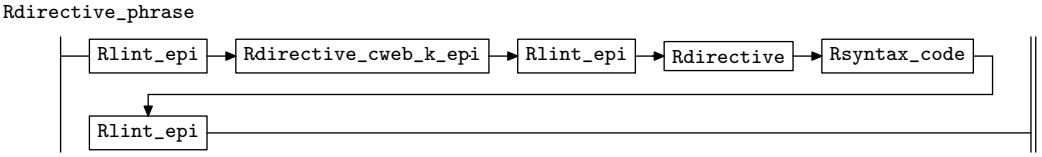
82. *Rmaybe_directive_phrase*'**s subrule 2.**



⟨ Rmaybe_directive_phrase subrule 2 op directive 82 ⟩ ≡

```
Rdirective * dir = sf→p1__;
Rsyntax_code * sdc = sf→p2__; Csubrule_def * fsm = ( Csubrule_def * ) rule_info__.parser__→fsm_tbl__;
fsm→add_sdc_to_directive(dir→directive_, sdc→syntax_code_);
```

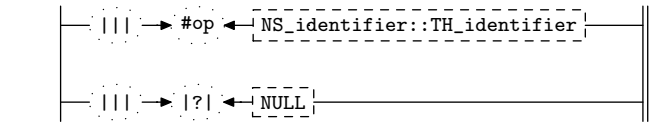
83. *Rdirective_phrase* **rule.**



⟨ Rdirective_phrase subrule 1 op directive 83 ⟩ ≡

```
AST * cwebt = sf→p2__→cweb_t_;
Rdirective * dir = sf→p4__;
Rsyntax_code * sdc = sf→p5__;
if (cwebt ≠ 0) {
  sdc→syntax_code_→add_cweb_marker(cwebt);
}
Csubrule_def * fsm = ( Csubrule_def * ) rule_info__.parser__→fsm_tbl__;
fsm→add_sdc_to_directive(dir→directive_, sdc→syntax_code_);
```

84. *Rdirective* **rule.**



85. Rdirective constructor directive.

⟨ Rdirective constructor directive 85 ⟩ ≡

```
directive_ = 0;
```

86. Rdirective op directive.

⟨ Rdirective op directive 86 ⟩ ≡

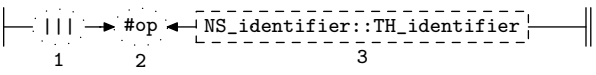
```
if (directive_ ≡ 0) return;
Csubrule_def * fsm = ( Csubrule_def * ) rule_info__.parser__→fsm_tbl__;
CAbs_lr1_sym * result = fsm→subrule_def_→add_directive_to_subrule(directive_, rule_info__.parser__);
if (result ≡ 0) return;      /* ok added */
directive_→set_auto_delete(true);      /* dup: delete when popped from stack */
RSVP(result);
rule_info__.parser__→set_stop_parse(true);
```

87. Rdirective user-declaration directive.

⟨ Rdirective user-declaration directive 87 ⟩ ≡

```
CAbs_lr1_sym * directive_;
```

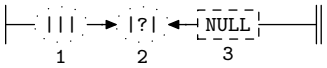
88. *Rdirective*'**s subrule 1.**



⟨ Rdirective subrule 1 op directive 88 ⟩ ≡

```
directive_ = sf→p2__;
```

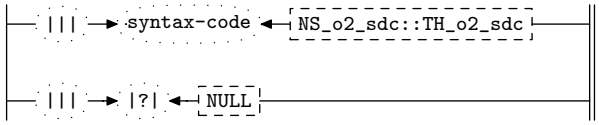
89. *Rdirective*'s subrule 2.



⟨ Rdirective subrule 2 op directive 89 ⟩ ≡
```
directive_ = 0;
sf→p2__→set_auto_delete(true);
CAbs_lr1_sym * sym = new Err_bad_directive;
sym→set_rc(*sf→p2__, __FILE__, __LINE__);
RSVP(sym);
rule_info__.parser__→set_stop_parse(true);
```

90. *Rsyntax_code* **rule.**

Rsyntax_code



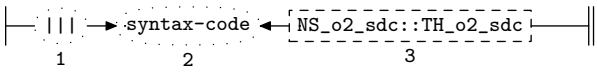
91. Rsyntax_code constructor directive.

⟨ Rsyntax_code constructor directive 91 ⟩ ≡
```
syntax_code_ = 0;
```

92. Rsyntax_code user-declaration directive.

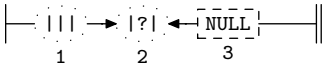
⟨ Rsyntax_code user-declaration directive 92 ⟩ ≡
```
T_syntax_code * syntax_code_;
```

93. *Rsyntax_code*'s subrule 1.



⟨ Rsyntax_code subrule 1 op directive 93 ⟩ ≡
```
syntax_code_ = sf→p2__;
```

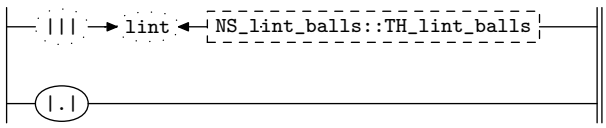
94. *Rsyntax_code*'s subrule 2.



⟨ Rsyntax_code subrule 2 op directive 94 ⟩ ≡
```
RSVP(sf→p2__);
rule_info__.parser__→set_stop_parse(true);
```

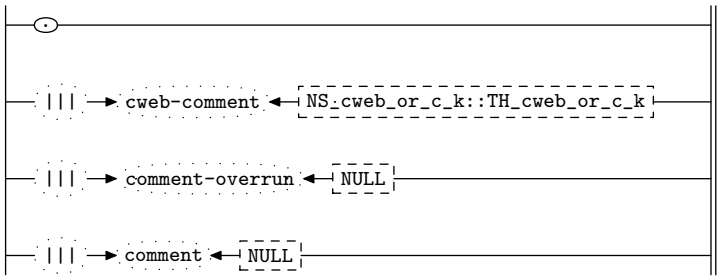
95. *Rlint_epi* **rule.**

Rlint_epi



96. *Rdirective_cweb_k_epi* **rule.**

Rdirective_cweb_k_epi



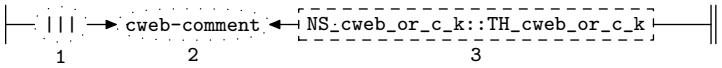
97. Rdirective_cweb_k_epi constructor directive.

⟨ Rdirective_cweb_k_epi constructor directive 97 ⟩ ≡
*cweb_t_* = 0;

98. Rdirective_cweb_k_epi user-declaration directive.

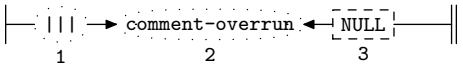
⟨ Rdirective_cweb_k_epi user-declaration directive 98 ⟩ ≡
AST ∗ *cweb_t_*;

99. *Rdirective_cweb_k_epi*'**s subrule 2.**



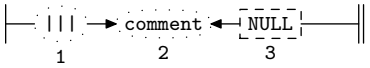
⟨ Rdirective_cweb_k_epi subrule 2 op directive 99 ⟩ ≡
T_cweb_comment ∗ *k* = *sf*→*p2__*;
AST ∗ *cwebk_t_* = **new** AST(∗*k*);
*cweb_t_* = **new** AST( );
T_cweb_marker ∗ *cw* = **new** *T_cweb_marker*(*cweb_t_*);
cw→*set_rc*(∗*k*, __FILE__, __LINE__);
AST::*join_pts*(∗*cweb_t_*, ∗*cwebk_t_*);
AST::*set_content*(∗*cweb_t_*, ∗*cw*);

100. *Rdirective_cweb_k_epi*'**s subrule 3.**



⟨ Rdirective_cweb_k_epi subrule 3 op directive 100 ⟩ ≡
RSVP(*sf*→*p2__*);
rule_info__.parser__→*set_stop_parse*(*true*);

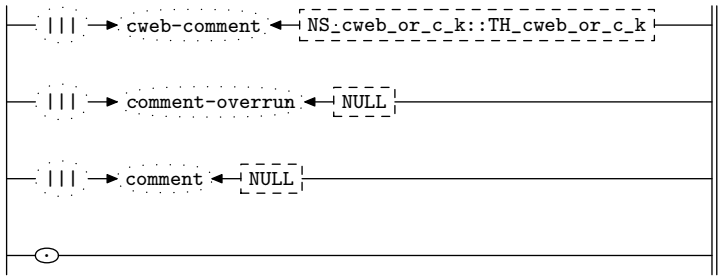
101. *Rdirective_cweb_k_epi*'**s subrule 4.**



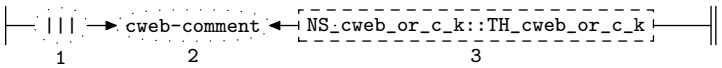
⟨ Rdirective_cweb_k_epi subrule 4 op directive 101 ⟩ ≡
sf→*p2__*→*set_auto_delete*(*true*);

102. *Rcweb_k_epi* **rule.**

Rcweb_k_epi



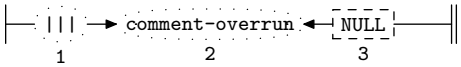
103. *Rcweb_k_epi*'**s subrule 1.**



⟨ Rcweb_k_epi subrule 1 op directive 103 ⟩ ≡

```
Csubrule_def * fsm = ( Csubrule_def * ) rule_info__.parser__→fsm_tbl__;
T_cweb_comment * k = sf→p2__;
AST * cwebk_t_ = new AST(*k);
AST * cweb_t_ = new AST( );
T_cweb_marker * cw = new T_cweb_marker(cweb_t_);
cw→set_rc(*k, __FILE__, __LINE__);
AST::join_pts(*cweb_t_, *cwebk_t_);
AST::set_content(*cweb_t_, *cw);
fsm→subrule_def_→add_cweb_marker(cweb_t_);
```

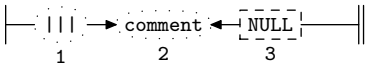
104. *Rcweb_k_epi*'**s subrule 2.**



⟨ Rcweb_k_epi subrule 2 op directive 104 ⟩ ≡

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

105. *Rcweb_k_epi*'**s subrule 3.**



⟨ Rcweb_k_epi subrule 3 op directive 105 ⟩ ≡

```
sf→p2__→set_auto_delete(true);
```

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

```
/*
 File: subrule_def.fsc
 Date and Time: Fri Jan  2 15:33:55 2015
*/
transitive   y
grammar-name "subrule_def"
name-space   "NS_subrule_def"
thread-name  "TH_subrule_def"
monolithic   n
file-name    "subrule_def.fsc"
no-of-T      569
list-of-native-first-set-terminals 0
end-list-of-native-first-set-terminals
list-of-transitive-threads 1
  NS_subrule_vector::TH_subrule_vector
end-list-of-transitive-threads
list-of-used-threads 10
  NS_cweb_or_c_k::TH_cweb_or_c_k
  NS_dbl_colon::TH_dbl_colon
  NS_identifier::TH_identifier
  NS_lint_balls::TH_lint_balls
  NS_o2_sdc::TH_o2_sdc
  NS_parallel_oper::TH_parallel_oper
  NS_rhs_bnd::TH_rhs_bnd
  NS_rhs_component::TH_rhs_component
  NS_rtn_component::TH_rtn_component
  NS_subrule_vector::TH_subrule_vector
end-list-of-used-threads
fsm-comments
"Parse a subrule: into the valley of sin..."
```

107. Lr1 State Network.

⇒ State: 1 state type: ^{s}

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
c	Rvector			2	1	1	||| - >	NS_subrule_vector::TH_subrule_vector		1	2	3	
c	Rsubrule_def			1	1	1	Rvector	|.|		1	4	13	

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

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rvector			2	1	2	- >			1	3	3	

⇒$^{->}$ State: 3 state type: ^{r}

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

⇒*Rvector* State: 4 state type: ^{s}

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rsubrule_def			1	1	2	|.|			1	5	13	

⇒\|.\| State: 5 state type: ^{s}

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
c	Rlint_epi			34	1	1	||| lint	NS_lint_balls::TH_lint_balls		5	24	25	
c	Rlint_epi			34	2	1	|.|			5	26	26	
t	Rsubrule_def			1	1	3	Rcommon_comments	*Rrhs_expr*		1	6	13	
c	Rcommon_comments			3	1	1	Rlint_epi	*Rcweb_k_epi*$^{\epsilon}$ *Rlint_epi*		5	57	59	

⇒*Rcommon_comments* State: 6 state type: $^{s/r}$

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
c	RTorRcomponents_epi			16	1	1	ϵ			6	0	6	1
c	Rstart_thread			5	1	1	||| |||	NS_parallel_oper::TH_parallel_oper		6	7	9	
c	Rcalled_proc_op			13	1	1	||| |t|	NULL		6	7	10	
c	RTorRcomponent			18	1	1	||| rule-in-stbl	NS_rhs_component::TH_rhs_component		6	7	12	
c	RTorRcomponent			18	2	1	||| T-in-stbl	NULL		6	7	11	
c	RTorRcomponent			18	3	1	||| |?|	NULL		6	7	8	
t	Rsubrule_def			1	1	4	Rrhs_expr			1	13	13	
c	Rrhs_expr			4	3	1	Rstart_thread	*Rlint_epi*		6	14	27	
c	Rrhs_expr			4	2	1	Rstart_thread	*Rlint_epi*		6	14	31	
c	Rrhs_expr			4	4	1	Rcalled_proc_op	*Rlint_epi*		6	32	46	
c	Rrhs_expr			4	5	1	Rcalled_proc_op	*Rlint_epi*		6	32	42	
c	Rrhs_expr			4	1	1	RTorRcomponents_epi	|.|		6	47	51	
c	RTorRcomponents_epi			16	2	1	RTorRcomponents			6	52	52	
c	RTorRcomponents			17	2	1	RTorRcomponents	*RTorRcomponent*		6	52	54	
c	RTorRcomponents			17	1	1	RTorRcomponent	*Rlint_epi*		6	55	56	

⇒\|\|\| *arbitration−code*: *AR_Rstart_thread* State: 7 state type: ^{s}

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	RTorRcomponent			18	3	2	|?|			6	8	8	
t	Rstart_thread			5	1	2	|||			6	9	9	
t	Rcalled_proc_op			13	1	2	|t|			6	10	10	
t	RTorRcomponent			18	2	2	T-in-stbl			6	11	11	
t	RTorRcomponent			18	1	2	rule-in-stbl			6	12	12	

⇒`|?|` State: 8 state type: ^{r}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	RTorRcomponent		18	3	3				6	0	8	2

⇒`|||` *arbitration−code*: ϵ State: 9 state type: ^{r}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rstart_thread		5	1	3				6	0	9	2

⇒`|t|` State: 10 state type: ^{r}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rcalled_proc_op		13	1	3				6	0	10	2

⇒*T−in−stbl* State: 11 state type: ^{r}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	RTorRcomponent		18	2	3				6	0	11	2

⇒*rule−in−stbl* State: 12 state type: ^{r}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	RTorRcomponent		18	1	3				6	0	12	2

⇒*Rrhs_expr* State: 13 state type: ^{r}

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

⇒*Rstart_thread* State: 14 state type: ^{s}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
c	Rlint_epi		34	1	1	`|||` lint NS_lint_balls::TH_lint_balls			14	24	25	
c	Rlint_epi		34	2	1	`|.|`			14	26	26	
t	Rrhs_expr		4	3	2	Rlint_epi *Rbal_thread_expr*			6	15	27	
t	Rrhs_expr		4	2	2	Rlint_epi *Rbal_thread_expr*			6	15	31	

⇒*Rlint_epi* State: 15 state type: ^{s}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
c	Rrtned_T		7	3	1	`|?|`			15	60	60	
c	Rrtned_T		7	1	1	`|||` T-in-stbl NS_rtn_component::TH_rtn_component			15	61	64	
c	Rrtned_T		7	2	1	`|||` `|?|` NULL			15	61	62	
c	Rrtned_T		7	4	1	`|||` `|||` NULL			15	61	63	
t	Rrhs_expr		4	3	3	Rbal_thread_expr *Rlint_epi*			6	16	27	
t	Rrhs_expr		4	2	3	Rbal_thread_expr *Rlint_epi*			6	16	31	
c	Rbal_thread_expr		6	1	1	Rrtned_T *Rlint_epi*			15	65	72	

⇒*Rbal_thread_expr* State: 16 state type: ^{s}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
c	Rlint_epi		34	1	1	`|||` lint NS_lint_balls::TH_lint_balls			16	24	25	
c	Rlint_epi		34	2	1	`|.|`			16	26	26	
t	Rrhs_expr		4	3	4	Rlint_epi `|.|`			6	17	27	
t	Rrhs_expr		4	2	4	Rlint_epi `|.|`			6	17	31	

⇒*Rlint_epi* State: 17 state type: ^{s}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rrhs_expr		4	3	5	`|.|`			6	18	27	

t	Rrhs_expr	4	2	5	`|.|`	6	18	31		

⇒`|.|` State: 18 state type: ^{s}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
c	Rcalled_thd_eos		21	2	1	`|?|`			18	81	81	
c	Rchained_dispatcher		19	1	1	`|||`	rule-in-stbl NS_rhs_component::TH_rhs_component		18	82	85	
c	Rchained_dispatcher		19	3	1	`|||` `|?|`	NULL		18	82	83	
c	Rchained_dispatcher		19	2	1	`|||`	T-in-stbl NULL		18	82	84	
c	Rcalled_thd_eos		21	1	1	`|||`	eosubrule NS_rhs_bnd::TH_rhs_bnd		18	82	86	
t	Rrhs_expr		4	3	6	Rchained_dispatcher	*Rlint_epi*		6	19	27	
t	Rrhs_expr		4	2	6	Rcalled_thd_eos	*Rlint_epi*		6	28	31	

⇒*Rchained_dispatcher* State: 19 state type: ^{s}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
c	Rlint_epi		34	1	1	`|||`	lint NS_lint_balls::TH_lint_balls		19	24	25	
c	Rlint_epi		34	2	1	`|.|`			19	26	26	
t	Rrhs_expr		4	3	7	Rlint_epi	*Rcalled_thd_eos*		6	20	27	

⇒*Rlint_epi* State: 20 state type: ^{s}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
c	Rcalled_thd_eos		21	2	1	`|?|`			20	81	81	
c	Rcalled_thd_eos		21	1	1	`|||`	eosubrule NS_rhs_bnd::TH_rhs_bnd		20	87	86	
t	Rrhs_expr		4	3	8	Rcalled_thd_eos	*Rlint_epi*		6	21	27	

⇒*Rcalled_thd_eos* State: 21 state type: ^{s}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
c	Rlint_epi		34	1	1	`|||`	lint NS_lint_balls::TH_lint_balls		21	24	25	
c	Rlint_epi		34	2	1	`|.|`			21	26	26	
t	Rrhs_expr		4	3	9	Rlint_epi	*Rpotential_codeblk*$^{\epsilon}$ *Rlint_epi*		6	22	27	

⇒*Rlint_epi* State: 22 state type: $^{s/r}$

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
c	Rpotential_directives		24	1	1	ϵ			22	0	22	2
c	Rpotential_directives		24	2	1	`{`			22	88	93	
t	Rrhs_expr		4	3	10	Rpotential_codeblk	*Rlint_epi*		6	23	27	
c	Rpotential_codeblk		23	1	1	Rpotential_directives			22	94	94	

⇒*Rpotential_codeblk* State: 23 state type: ^{s}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
c	Rlint_epi		34	1	1	`|||`	lint NS_lint_balls::TH_lint_balls		23	24	25	
c	Rlint_epi		34	2	1	`|.|`			23	26	26	
t	Rrhs_expr		4	3	11	Rlint_epi			6	27	27	

⇒`|||` *arbitration−code*: ϵ State: 24 state type: ^{s}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rlint_epi		34	1	2	lint			23	25	25	

⇒*lint* State: 25 state type: ^{r}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rlint_epi		34	1	3				23	0	25	4

⇒`|.|` State: 26 state type: ^{r}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rlint_epi		34	2	2			23	0	26	4

⇒*Rlint_epi*

State: 27 state type: ^{r}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rrhs_expr		4	3	12			6	0	27	3

⇒*Rcalled_thd_eos*

State: 28 state type: ^{s}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rlint_epi		34	1	1	||| lint NS_lint_balls::TH_lint_balls		28	24	25	
c	Rlint_epi		34	2	1	|.|		28	26	26	
t	Rrhs_expr		4	2	7	Rlint_epi *Rpotential_codeblk*$^{\epsilon}$ *Rlint_epi*		6	29	31	

⇒*Rlint_epi*

State: 29 state type: $^{s/r}$

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rpotential_directives		24	1	1	ϵ		29	0	29	2
c	Rpotential_directives		24	2	1	{		29	88	93	
t	Rrhs_expr		4	2	8	Rpotential_codeblk *Rlint_epi*		6	30	31	
c	Rpotential_codeblk		23	1	1	Rpotential_directives		29	94	94	

⇒*Rpotential_codeblk*

State: 30 state type: ^{s}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rlint_epi		34	1	1	||| lint NS_lint_balls::TH_lint_balls		30	24	25	
c	Rlint_epi		34	2	1	|.|		30	26	26	
t	Rrhs_expr		4	2	9	Rlint_epi		6	31	31	

⇒*Rlint_epi*

State: 31 state type: ^{r}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rrhs_expr		4	2	10			6	0	31	3

⇒*Rcalled_proc_op*

State: 32 state type: ^{s}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rlint_epi		34	1	1	||| lint NS_lint_balls::TH_lint_balls		32	24	25	
c	Rlint_epi		34	2	1	|.|		32	26	26	
t	Rrhs_expr		4	4	2	Rlint_epi *Rbal_called_proc_expr*		6	33	46	
t	Rrhs_expr		4	5	2	Rlint_epi *Rbal_called_proc_expr*		6	33	42	

⇒*Rlint_epi*

State: 33 state type: ^{s}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rrtned_T		7	3	1	|?|		33	60	60	
c	Rrtned_T		7	1	1	||| T-in-stbl NS_rtn_component::TH_rtn_component		33	61	64	
c	Rrtned_T		7	2	1	||| |?| NULL		33	61	62	
c	Rrtned_T		7	4	1	||| ||| NULL		33	61	63	
c	Rbal_called_proc_expr		14	1	1	Rrtned_T *Rlint_epi*		33	95	101	
t	Rrhs_expr		4	4	3	Rbal_called_proc_expr *Rlint_epi*		6	34	46	
t	Rrhs_expr		4	5	3	Rbal_called_proc_expr *Rlint_epi*		6	34	42	

⇒*Rbal_called_proc_expr*

State: 34 state type: ^{s}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rlint_epi		34	1	1	||| lint NS_lint_balls::TH_lint_balls		34	24	25	
c	Rlint_epi		34	2	1	|.|		34	26	26	
t	Rrhs_expr		4	4	4	Rlint_epi |.|		6	35	46	

t	Rrhs_expr	4	5	4	Rlint_epi `|.|`	6	35	42	

⇒*Rlint_epi* State: 35 state type: *s*

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rrhs_expr		4	4	5	`|.|`		6	36	46	
t	Rrhs_expr		4	5	5	`|.|`		6	36	42	

⇒`|.|` State: 36 state type: *s*

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rcalled_proc_eos		22	2	1	`|?|`		36	102	102	
c	Rchained_dispatcher		19	1	1	`|||` rule-in-stbl NS_rhs_component::TH_rhs_component		36	103	85	
c	Rchained_dispatcher		19	3	1	`|||` `|?|` NULL		36	103	83	
c	Rcalled_proc_eos		22	1	1	`|||` eosubrule NS_rhs_bnd::TH_rhs_bnd		36	103	104	
c	Rchained_dispatcher		19	2	1	`|||` T-in-stbl NULL		36	103	84	
t	Rrhs_expr		4	5	6	Rchained_dispatcher *Rlint_epi*		6	37	42	
t	Rrhs_expr		4	4	6	Rcalled_proc_eos *Rlint_epi*		6	43	46	

⇒*Rchained_dispatcher* State: 37 state type: *s*

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rlint_epi		34	1	1	`|||` lint NS_lint_balls::TH_lint_balls		37	24	25	
c	Rlint_epi		34	2	1	`|.|`		37	26	26	
t	Rrhs_expr		4	5	7	Rlint_epi *Rcalled_proc_eos*		6	38	42	

⇒*Rlint_epi* State: 38 state type: *s*

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rcalled_proc_eos		22	2	1	`|?|`		38	102	102	
c	Rcalled_proc_eos		22	1	1	`|||` eosubrule NS_rhs_bnd::TH_rhs_bnd		38	105	104	
t	Rrhs_expr		4	5	8	Rcalled_proc_eos *Rlint_epi*		6	39	42	

⇒*Rcalled_proc_eos* State: 39 state type: *s*

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rlint_epi		34	1	1	`|||` lint NS_lint_balls::TH_lint_balls		39	24	25	
c	Rlint_epi		34	2	1	`|.|`		39	26	26	
t	Rrhs_expr		4	5	9	Rlint_epi *Rpotential_codeblk*$^{\epsilon}$ *Rlint_epi*		6	40	42	

⇒*Rlint_epi* State: 40 state type: *s/r*

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rpotential_directives		24	1	1	ϵ		40	0	40	2
c	Rpotential_directives		24	2	1	`{`		40	88	93	
t	Rrhs_expr		4	5	10	Rpotential_codeblk *Rlint_epi*		6	41	42	
c	Rpotential_codeblk		23	1	1	Rpotential_directives		40	94	94	

⇒*Rpotential_codeblk* State: 41 state type: *s*

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rlint_epi		34	1	1	`|||` lint NS_lint_balls::TH_lint_balls		41	24	25	
c	Rlint_epi		34	2	1	`|.|`		41	26	26	
t	Rrhs_expr		4	5	11	Rlint_epi		6	42	42	

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

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rrhs_expr		4	5	12			6	0	42	3

⇒*Rcalled_proc_eos* State: 43 state type: *s*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rlint_epi		34	1	1	||| lint NS_lint_balls::TH_lint_balls		43	24	25	
c	Rlint_epi		34	2	1	|.|		43	26	26	
t	Rrhs_expr		4	4	7	Rlint_epi *Rpotential_codeblk*$^{\epsilon}$ *Rlint_epi*		6	44	46	

⇒*Rlint_epi* State: 44 state type: *s/r*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rpotential_directives		24	1	1	ϵ		44	0	44	2
c	Rpotential_directives		24	2	1	{		44	88	93	
t	Rrhs_expr		4	4	8	Rpotential_codeblk *Rlint_epi*		6	45	46	
c	Rpotential_codeblk		23	1	1	Rpotential_directives		44	94	94	

⇒*Rpotential_codeblk* State: 45 state type: *s*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rlint_epi		34	1	1	||| lint NS_lint_balls::TH_lint_balls		45	24	25	
c	Rlint_epi		34	2	1	|.|		45	26	26	
t	Rrhs_expr		4	4	9	Rlint_epi		6	46	46	

⇒*Rlint_epi* State: 46 state type: *r*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rrhs_expr		4	4	10			6	0	46	3

⇒*RTorRcomponents_epi* State: 47 state type: *s*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rrhs_expr		4	1	2	|.|		6	48	51	

⇒\|.\| State: 48 state type: *s*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rmust_eos		20	2	1	|?|		48	106	106	
c	Rmust_eos		20	1	1	||| eosubrule NS_rhs_bnd::TH_rhs_bnd		48	107	109	
c	Rmust_eos		20	3	1	||| |?| NULL		48	107	108	
t	Rrhs_expr		4	1	3	Rmust_eos *Rpotential_codeblk*$^{\epsilon}$ *Rlint_epi*		6	49	51	

⇒*Rmust_eos* State: 49 state type: *s/r*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rpotential_directives		24	1	1	ϵ		49	0	49	2
c	Rpotential_directives		24	2	1	{		49	88	93	
t	Rrhs_expr		4	1	4	Rpotential_codeblk *Rlint_epi*		6	50	51	
c	Rpotential_codeblk		23	1	1	Rpotential_directives		49	94	94	

⇒*Rpotential_codeblk* State: 50 state type: *s*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rlint_epi		34	1	1	||| lint NS_lint_balls::TH_lint_balls		50	24	25	
c	Rlint_epi		34	2	1	|.|		50	26	26	
t	Rrhs_expr		4	1	5	Rlint_epi		6	51	51	

⇒*Rlint_epi* State: 51 state type: *r*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rrhs_expr		4	1	6			6	0	51	3

⇒*RTorRcomponents* State: 52 state type: $^{s/r}$

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	RTorRcomponents_epi		16	2	2			6	0	52	1
c	RTorRcomponent		18	1	1	||| rule-in-stbl NS_rhs_component::TH_rhs_component		52	110	12	
c	RTorRcomponent		18	2	1	||| T-in-stbl NULL		52	110	11	
c	RTorRcomponent		18	3	1	||| |?| NULL		52	110	8	
t	RTorRcomponents		17	2	2	RTorRcomponent *Rlint_epi*		6	53	54	

⇒*RTorRcomponent* State: 53 state type: ^{s}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rlint_epi		34	1	1	||| lint NS_lint_balls::TH_lint_balls		53	24	25	
c	Rlint_epi		34	2	1	|.|		53	26	26	
t	RTorRcomponents		17	2	3	Rlint_epi		6	54	54	

⇒*Rlint_epi* State: 54 state type: ^{r}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	RTorRcomponents		17	2	4			6	0	54	2

⇒*RTorRcomponent* State: 55 state type: ^{s}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rlint_epi		34	1	1	||| lint NS_lint_balls::TH_lint_balls		55	24	25	
c	Rlint_epi		34	2	1	|.|		55	26	26	
t	RTorRcomponents		17	1	2	Rlint_epi		6	56	56	

⇒*Rlint_epi* State: 56 state type: ^{r}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	RTorRcomponents		17	1	3			6	0	56	2

⇒*Rlint_epi* State: 57 state type: $^{s/r}$

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rcweb_k_epi		36	4	1	ϵ		57	0	57	2
c	Rcweb_k_epi		36	1	1	||| cweb-comment NS_cweb_or_c_k::TH_cweb_or_c_k		57	111	113	
c	Rcweb_k_epi		36	2	1	||| comment-overrun NULL		57	111	114	
c	Rcweb_k_epi		36	3	1	||| comment NULL		57	111	112	
t	Rcommon_comments		3	1	2	Rcweb_k_epi *Rlint_epi*		5	58	59	

⇒*Rcweb_k_epi* State: 58 state type: ^{s}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rlint_epi		34	1	1	||| lint NS_lint_balls::TH_lint_balls		58	24	25	
c	Rlint_epi		34	2	1	|.|		58	26	26	
t	Rcommon_comments		3	1	3	Rlint_epi		5	59	59	

⇒*Rlint_epi* State: 59 state type: ^{r}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rcommon_comments		3	1	4			5	0	59	2

⇒\|?\| State: 60 state type: ^{r}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rrtned_T		7	3	2			15	0	60	2

⇒\|\|\| *arbitration−code*: ϵ State: 61 state type: ^{s}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rrtned_T		7	2	2	|?|		15	62	62	
t	Rrtned_T		7	4	2	|||		15	63	63	
t	Rrtned_T		7	1	2	T-in-stbl		15	64	64	

⇒\|?\| State: 62 state type: ^{r}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rrtned_T		7	2	3			15	0	62	2

⇒\|\|\| *arbitration−code*: ϵ State: 63 state type: ^{r}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rrtned_T		7	4	3			15	0	63	2

⇒$^{T-in-stbl}$ State: 64 state type: ^{r}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rrtned_T		7	1	3			15	0	64	2

⇒$^{Rrtned\_T}$ State: 65 state type: ^{s}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rlint_epi		34	1	1	||| lint NS_lint_balls::TH_lint_balls		65	24	25	
c	Rlint_epi		34	2	1	|.|		65	26	26	
t	Rbal_thread_expr		6	1	2	Rlint_epi *Rcalled_thread*		15	66	72	

⇒$^{Rlint\_epi}$ State: 66 state type: ^{s}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rns		10	3	1	|?|		66	67	67	
c	Rns		10	1	1	||| identifier NULL		66	68	70	
c	RNULL		9	1	1	||| # NULL NS_identifier::TH_identifier		66	68	71	
c	Rns		10	2	1	||| |?| NULL		66	68	69	
t	Rbal_thread_expr		6	1	3	Rcalled_thread		15	72	72	
c	Rcalled_thread		8	2	1	RNULL		66	73	73	
c	Rcalled_thread		8	1	1	Rns *Rdbl_colon*		66	74	80	

⇒\|?\| State: 67 state type: ^{r}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rns		10	3	2			66	0	67	5

⇒\|\|\| *arbitration−code*: ϵ State: 68 state type: ^{s}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rns		10	2	2	|?|		66	69	69	
t	Rns		10	1	2	identifier		66	70	70	
t	RNULL		9	1	2	# NULL		66	71	71	

⇒\|?\| State: 69 state type: ^{r}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rns		10	2	3			66	0	69	5

⇒^{identifier} State: 70 state type: ^{r}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rns		10	1	3			66	0	70	5

⇒$^{\#NULL}$ State: 71 state type: ^{r}

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	RNULL			9	1	3				66	0	71	2

⇒*Rcalled_thread* State: 72 state type: *r*

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rbal_thread_expr			6	1	4				15	0	72	2

⇒*RNULL* State: 73 state type: *r*

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rcalled_thread			8	2	2				66	0	73	2

⇒*Rns* State: 74 state type: *s*

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
c	Rdbl_colon			11	3	1	|?|			74	115	115	
c	Rdbl_colon			11	2	1	||| |?| NULL			74	116	117	
c	Rdbl_colon			11	1	1	||| :: NS_dbl_colon::TH_dbl_colon			74	116	118	
t	Rcalled_thread			8	1	2	Rdbl_colon *Rth_name*			66	75	80	

⇒*Rdbl_colon* State: 75 state type: *s*

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
c	Rth_name			12	3	1	|?|			75	76	76	
c	Rth_name			12	1	1	||| identifier NS_identifier::TH_identifier			75	77	79	
c	Rth_name			12	2	1	||| |?| NULL			75	77	78	
t	Rcalled_thread			8	1	3	Rth_name			66	80	80	

⇒\|?\| State: 76 state type: *r*

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rth_name			12	3	2				75	0	76	2

⇒\|\|\| *arbitration−code*: ϵ State: 77 state type: *s*

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rth_name			12	2	2	|?|			75	78	78	
t	Rth_name			12	1	2	identifier			75	79	79	

⇒\|?\| State: 78 state type: *r*

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rth_name			12	2	3				75	0	78	2

⇒*identifier* State: 79 state type: *r*

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rth_name			12	1	3				75	0	79	2

⇒*Rth_name* State: 80 state type: *r*

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rcalled_thread			8	1	4				66	0	80	2

⇒\|?\| State: 81 state type: *r*

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rcalled_thd_eos			21	2	2				18	0	81	2

⇒\|\|\| *arbitration−code*: ϵ State: 82 state type: *s*

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

t	Rchained_dispatcher	19	3	2	`|?|`	18	83	83	
t	Rchained_dispatcher	19	2	2	T-in-stbl	18	84	84	
t	Rchained_dispatcher	19	1	2	rule-in-stbl	18	85	85	
t	Rcalled_thd_eos	21	1	2	eosubrule	18	86	86	

⇒`|?|` State: 83 state type: ^{r}

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rchained_dispatcher		19	3	3			18	0	83	2

⇒$^{T-in-stbl}$ State: 84 state type: ^{r}

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rchained_dispatcher		19	2	3			18	0	84	2

⇒$^{rule-in-stbl}$ State: 85 state type: ^{r}

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rchained_dispatcher		19	1	3			18	0	85	2

⇒^{eosubrule} State: 86 state type: ^{r}

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rcalled_thd_eos		21	1	3			18	0	86	2

⇒`|||` $arbitration-code$: ϵ State: 87 state type: ^{s}

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rcalled_thd_eos		21	1	2	eosubrule		20	86	86	

⇒{ State: 88 state type: ^{s}

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rlint_epi		34	1	1	`|||` lint NS_lint_balls::TH_lint_balls		88	24	25	
c	Rlint_epi		34	2	1	`|.|`		88	26	26	
t	Rpotential_directives		24	2	2	Rlint_epi *Rmust_directive_phrase*		22	89	93	

⇒$^{Rlint\_epi}$ State: 89 state type: $^{s/r}$

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rdirective_cweb_k_epi		35	1	1	ϵ		89	0	89	2
c	Rdirective_cweb_k_epi		35	2	1	`|||` cweb-comment NS_cweb_or_c_k::TH_cweb_or_c_k		89	119	121	
c	Rdirective_cweb_k_epi		35	3	1	`|||` comment-overrun NULL		89	119	122	
c	Rdirective_cweb_k_epi		35	4	1	`|||` comment NULL		89	119	120	
t	Rpotential_directives		24	2	3	Rmust_directive_phrase *Rmaybe_more_directives*$^{\epsilon}$ *Rclose_brace*		22	90	93	
c	Rmust_directive_phrase		25	1	1	Rdirective_cweb_k_epi *Rlint_epi*		89	123	127	

⇒$^{Rmust\_directive\_phrase}$ State: 90 state type: $^{s/r^2}$

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rdirective_cweb_k_epi		35	1	1	ϵ		90	0	90	2
c	Rmaybe_more_directives		28	1	1	ϵ		90	0	90	6
c	Rdirective		32	1	1	`|||` `#` op NS_identifier::TH_identifier		90	128	130	
c	Rdirective_cweb_k_epi		35	2	1	`|||` cweb-comment NS_cweb_or_c_k::TH_cweb_or_c_k		90	128	121	
c	Rdirective		32	2	1	`|||` `|?|` NULL		90	128	129	
c	Rdirective_cweb_k_epi		35	3	1	`|||` comment-overrun NULL		90	128	122	
c	Rdirective_cweb_k_epi		35	4	1	`|||` comment NULL		90	128	120	
t	Rpotential_directives		24	2	4	Rmaybe_more_directives *Rclose_brace*		22	91	93	
c	Rmaybe_more_directives		28	2	1	Rmaybe_directive_phrases		90	131	131	
c	Rmaybe_directive_phrases		29	2	1	Rmaybe_directive_phrases *Rmaybe_directive_phrase*		90	131	132	

c	Rmaybe_directive_phrases	29	1	1	Rmaybe_directive_phrase	90	141	141
c	Rmaybe_directive_phrase	30	2	1	Rdirective *Rsyntax_code*	90	133	135
c	Rmaybe_directive_phrase	30	1	1	Rdirective_cweb_k_epi *Rlint_epi*	90	136	140

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

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rclose_brace		27	1	1	|?|		91	142	142	
c	Rclose_brace		27	2	1	}		91	143	143	
t	Rpotential_directives		24	2	5	Rclose_brace *Rlint_epi*		22	92	93	

⇒*Rclose_brace* State: 92 state type: *s*

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rlint_epi		34	1	1	||| lint NS_lint_balls::TH_lint_balls		92	24	25	
c	Rlint_epi		34	2	1	|.|		92	26	26	
t	Rpotential_directives		24	2	6	Rlint_epi		22	93	93	

⇒*Rlint_epi* State: 93 state type: *r*

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rpotential_directives		24	2	7			22	0	93	2

⇒*Rpotential_directives* State: 94 state type: *r*

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rpotential_codeblk		23	1	2			22	0	94	2

⇒*Rrtned_T* State: 95 state type: *s*

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rlint_epi		34	1	1	||| lint NS_lint_balls::TH_lint_balls		95	24	25	
c	Rlint_epi		34	2	1	|.|		95	26	26	
t	Rbal_called_proc_expr		14	1	2	Rlint_epi *Rcalled_proc_name*		33	96	101	

⇒*Rlint_epi* State: 96 state type: *s*

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rns		10	3	1	|?|		96	67	67	
c	Rns		10	1	1	||| identifier NULL		96	68	70	
c	RNULL		9	1	1	||| # NULL NS_identifier::TH_identifier		96	68	71	
c	Rns		10	2	1	||| |?| NULL		96	68	69	
c	Rcalled_proc_name		15	2	1	RNULL		96	97	97	
c	Rcalled_proc_name		15	1	1	Rns *Rdbl_colon*		96	98	100	
t	Rbal_called_proc_expr		14	1	3	Rcalled_proc_name		33	101	101	

⇒*RNULL* State: 97 state type: *r*

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rcalled_proc_name		15	2	2			96	0	97	2

⇒*Rns* State: 98 state type: *s*

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rdbl_colon		11	3	1	|?|		98	115	115	
c	Rdbl_colon		11	2	1	||| |?| NULL		98	116	117	
c	Rdbl_colon		11	1	1	||| :: NS_dbl_colon::TH_dbl_colon		98	116	118	
t	Rcalled_proc_name		15	1	2	Rdbl_colon *Rth_name*		96	99	100	

⇒*Rdbl_colon* State: 99 state type: *s*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rth_name		12	3	1	|?|		99	76	76	
c	Rth_name		12	1	1	||| identifier NS_identifier::TH_identifier		99	77	79	
c	Rth_name		12	2	1	||| |?| NULL		99	77	78	
t	Rcalled_proc_name		15	1	3	Rth_name		96	100	100	

⇒*Rth_name*

State: 100 state type: *r*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rcalled_proc_name		15	1	4			96	0	100	2

⇒*Rcalled_proc_name*

State: 101 state type: *r*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rbal_called_proc_expr		14	1	4			33	0	101	2

⇒|?|

State: 102 state type: *r*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rcalled_proc_eos		22	2	2			36	0	102	2

⇒||| *arbitration−code*: ϵ

State: 103 state type: *s*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rchained_dispatcher		19	3	2	|?|		36	83	83	
t	Rchained_dispatcher		19	2	2	T-in-stbl		36	84	84	
t	Rchained_dispatcher		19	1	2	rule-in-stbl		36	85	85	
t	Rcalled_proc_eos		22	1	2	eosubrule		36	104	104	

⇒*eosubrule*

State: 104 state type: *r*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rcalled_proc_eos		22	1	3			36	0	104	2

⇒||| *arbitration−code*: ϵ

State: 105 state type: *s*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rcalled_proc_eos		22	1	2	eosubrule		38	104	104	

⇒|?|

State: 106 state type: *r*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rmust_eos		20	2	2			48	0	106	7

⇒||| *arbitration−code*: ϵ

State: 107 state type: *s*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rmust_eos		20	3	2	|?|		48	108	108	
t	Rmust_eos		20	1	2	eosubrule		48	109	109	

⇒|?|

State: 108 state type: *r*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rmust_eos		20	3	3			48	0	108	7

⇒*eosubrule*

State: 109 state type: *r*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rmust_eos		20	1	3			48	0	109	7

⇒||| *arbitration−code*: ϵ

State: 110 state type: *s*

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

t	RTorRcomponent	18	3	2	`|?|`	52	8	8	
t	RTorRcomponent	18	2	2	T-in-stbl	52	11	11	
t	RTorRcomponent	18	1	2	rule-in-stbl	52	12	12	

⇒`|||` *arbitration−code*: ϵ State: 111 state type: ^{s}

	← rule	→ R#	sr#	Po	← subrule element	→ Brn	Gto	Red	LA
t	Rcweb_k_epi	36	3	2	comment	57	112	112	
t	Rcweb_k_epi	36	1	2	cweb-comment	57	113	113	
t	Rcweb_k_epi	36	2	2	comment-overrun	57	114	114	

⇒*comment* State: 112 state type: ^{r}

	← rule	→ R#	sr#	Po	← subrule element	→ Brn	Gto	Red	LA
t	Rcweb_k_epi	36	3	3		57	0	112	2

⇒*cweb−comment* State: 113 state type: ^{r}

	← rule	→ R#	sr#	Po	← subrule element	→ Brn	Gto	Red	LA
t	Rcweb_k_epi	36	1	3		57	0	113	2

⇒*comment−overrun* State: 114 state type: ^{r}

	← rule	→ R#	sr#	Po	← subrule element	→ Brn	Gto	Red	LA
t	Rcweb_k_epi	36	2	3		57	0	114	2

⇒`|?|` State: 115 state type: ^{r}

	← rule	→ R#	sr#	Po	← subrule element	→ Brn	Gto	Red	LA
t	Rdbl_colon	11	3	2		74	0	115	5

⇒`|||` *arbitration−code*: ϵ State: 116 state type: ^{s}

	← rule	→ R#	sr#	Po	← subrule element	→ Brn	Gto	Red	LA
t	Rdbl_colon	11	2	2	`|?|`	74	117	117	
t	Rdbl_colon	11	1	2	::	74	118	118	

⇒`|?|` State: 117 state type: ^{r}

	← rule	→ R#	sr#	Po	← subrule element	→ Brn	Gto	Red	LA
t	Rdbl_colon	11	2	3		74	0	117	5

⇒:: State: 118 state type: ^{r}

	← rule	→ R#	sr#	Po	← subrule element	→ Brn	Gto	Red	LA
t	Rdbl_colon	11	1	3		74	0	118	5

⇒`|||` *arbitration−code*: ϵ State: 119 state type: ^{s}

	← rule	→ R#	sr#	Po	← subrule element	→ Brn	Gto	Red	LA
t	Rdirective_cweb_k_epi	35	4	2	comment	89	120	120	
t	Rdirective_cweb_k_epi	35	2	2	cweb-comment	89	121	121	
t	Rdirective_cweb_k_epi	35	3	2	comment-overrun	89	122	122	

⇒*comment* State: 120 state type: ^{r}

	← rule	→ R#	sr#	Po	← subrule element	→ Brn	Gto	Red	LA
t	Rdirective_cweb_k_epi	35	4	3		89	0	120	2

⇒*cweb−comment* State: 121 state type: ^{r}

	← rule	→ R#	sr#	Po	← subrule element	→ Brn	Gto	Red	LA
t	Rdirective_cweb_k_epi	35	2	3		89	0	121	2

⇒*comment−overrun* State: 122 state type: ^{r}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rdirective_cweb_k_epi		35	3	3			89	0	122	2

⇒*Rdirective_cweb_k_epi* State: 123 state type: ^{s}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rlint_epi		34	1	1	||| lint NS_lint_balls::TH_lint_balls		123	24	25	
c	Rlint_epi		34	2	1	|.|		123	26	26	
t	Rmust_directive_phrase		25	1	2	Rlint_epi *Rmust_directive*		89	124	127	

⇒*Rlint_epi* State: 124 state type: ^{s}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rmust_directive		26	3	1	|?|		124	144	144	
c	Rmust_directive		26	2	1	||| |?| NULL		124	145	146	
c	Rmust_directive		26	1	1	||| # op NS_identifier::TH_identifier		124	145	147	
t	Rmust_directive_phrase		25	1	3	Rmust_directive *Rsyntax_code*		89	125	127	

⇒*Rmust_directive* State: 125 state type: ^{s}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rsyntax_code		33	1	1	||| syntax-code NS_o2_sdc::TH_o2_sdc		125	148	150	
c	Rsyntax_code		33	2	1	||| |?| NULL		125	148	149	
t	Rmust_directive_phrase		25	1	4	Rsyntax_code *Rlint_epi*		89	126	127	

⇒*Rsyntax_code* State: 126 state type: ^{s}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rlint_epi		34	1	1	||| lint NS_lint_balls::TH_lint_balls		126	24	25	
c	Rlint_epi		34	2	1	|.|		126	26	26	
t	Rmust_directive_phrase		25	1	5	Rlint_epi		89	127	127	

⇒*Rlint_epi* State: 127 state type: ^{r}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rmust_directive_phrase		25	1	6			89	0	127	8

⇒\|\|\| *arbitration−code*: ϵ State: 128 state type: ^{s}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rdirective		32	2	2	|?|		90	129	129	
t	Rdirective_cweb_k_epi		35	4	2	comment		90	120	120	
t	Rdirective		32	1	2	# op		90	130	130	
t	Rdirective_cweb_k_epi		35	2	2	cweb-comment		90	121	121	
t	Rdirective_cweb_k_epi		35	3	2	comment-overrun		90	122	122	

⇒\|?\| State: 129 state type: ^{r}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rdirective		32	2	3			90	0	129	9

⇒*#op* State: 130 state type: ^{r}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rdirective		32	1	3			90	0	130	9

⇒*Rmaybe_directive_phrases* State: 131 state type: $^{s/r^2}$

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

c	Rdirective_cweb_k_epi	35	1	1	ϵ	131	0	131	2
t	Rmaybe_more_directives	28	2	2		90	0	131	6
c	Rdirective	32	1	1	||| # op NS_identifier::TH_identifier	131	128	130	
c	Rdirective_cweb_k_epi	35	2	1	||| cweb-comment NS_cweb_or_c_k::TH_cweb_or_c_k	131	128	121	
c	Rdirective	32	2	1	||| |?| NULL	131	128	129	
c	Rdirective_cweb_k_epi	35	3	1	||| comment-overrun NULL	131	128	122	
c	Rdirective_cweb_k_epi	35	4	1	||| comment NULL	131	128	120	
t	Rmaybe_directive_phrases	29	2	2	Rmaybe_directive_phrase	90	132	132	
c	Rmaybe_directive_phrase	30	2	1	Rdirective *Rsyntax_code*	131	133	135	
c	Rmaybe_directive_phrase	30	1	1	Rdirective_cweb_k_epi *Rlint_epi*	131	136	140	

⇒*Rmaybe_directive_phrase* State: 132 state type: *r*

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rmaybe_directive_phrases		29	2	3			90	0	132	8

⇒*Rdirective* State: 133 state type: *s*

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rsyntax_code		33	1	1	||| syntax-code NS_o2_sdc::TH_o2_sdc		133	148	150	
c	Rsyntax_code		33	2	1	||| |?| NULL		133	148	149	
t	Rmaybe_directive_phrase		30	2	2	Rsyntax_code *Rlint_epi*		131	134	135	

⇒*Rsyntax_code* State: 134 state type: *s*

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rlint_epi		34	1	1	||| lint NS_lint_balls::TH_lint_balls		134	24	25	
c	Rlint_epi		34	2	1	|.|		134	26	26	
t	Rmaybe_directive_phrase		30	2	3	Rlint_epi		131	135	135	

⇒*Rlint_epi* State: 135 state type: *r*

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rmaybe_directive_phrase		30	2	4			131	0	135	8

⇒*Rdirective_cweb_k_epi* State: 136 state type: *s*

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rlint_epi		34	1	1	||| lint NS_lint_balls::TH_lint_balls		136	24	25	
c	Rlint_epi		34	2	1	|.|		136	26	26	
t	Rmaybe_directive_phrase		30	1	2	Rlint_epi *Rdirective*		131	137	140	

⇒*Rlint_epi* State: 137 state type: *s*

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rdirective		32	1	1	||| # op NS_identifier::TH_identifier		137	151	130	
c	Rdirective		32	2	1	||| |?| NULL		137	151	129	
t	Rmaybe_directive_phrase		30	1	3	Rdirective *Rsyntax_code*		131	138	140	

⇒*Rdirective* State: 138 state type: *s*

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rsyntax_code		33	1	1	||| syntax-code NS_o2_sdc::TH_o2_sdc		138	148	150	
c	Rsyntax_code		33	2	1	||| |?| NULL		138	148	149	
t	Rmaybe_directive_phrase		30	1	4	Rsyntax_code *Rlint_epi*		131	139	140	

⇒*Rsyntax_code* State: 139 state type: *s*

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rlint_epi		34	1	1	||| lint NS_lint_balls::TH_lint_balls		139	24	25	

c	Rlint_epi	34	2	1	`|.|`		139	26	26	
t	Rmaybe_directive_phrase	30	1	5	Rlint_epi		131	140	140	

⇒*Rlint_epi* State: 140 state type: *r*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rmaybe_directive_phrase		30	1	6			131	0	140	8

⇒*Rmaybe_directive_phrase* State: 141 state type: *r*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rmaybe_directive_phrases		29	1	2			90	0	141	8

⇒`|?|` State: 142 state type: *r*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rclose_brace		27	1	2			91	0	142	2

⇒`}` State: 143 state type: *r*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rclose_brace		27	2	2			91	0	143	2

⇒`|?|` State: 144 state type: *r*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rmust_directive		26	3	2			124	0	144	9

⇒`|||` *arbitration−code*: *ϵ* State: 145 state type: *s*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rmust_directive		26	2	2	`|?|`		124	146	146	
t	Rmust_directive		26	1	2	`#` op		124	147	147	

⇒`|?|` State: 146 state type: *r*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rmust_directive		26	2	3			124	0	146	9

⇒*#op* State: 147 state type: *r*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rmust_directive		26	1	3			124	0	147	9

⇒`|||` *arbitration−code*: *ϵ* State: 148 state type: *s*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rsyntax_code		33	2	2	`|?|`		125	149	149	
t	Rsyntax_code		33	1	2	syntax-code		125	150	150	

⇒`|?|` State: 149 state type: *r*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rsyntax_code		33	2	3			125	0	149	2

⇒*syntax−code* State: 150 state type: *r*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rsyntax_code		33	1	3			125	0	150	2

⇒`|||` *arbitration−code*: *ϵ* State: 151 state type: *s*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rdirective		32	2	2	`|?|`		137	129	129	

t Rdirective 32 1 2 # op 137 130 130

108. Index.

subrule_def Grammar

Date: January 2, 2015 at 15:39

File: subrule_def.lex Ns: NS_subrule_def

Version: 1.0 Debug: true

Grammar Comments: Type: Thread

Parse a subrule: into the valley of sin...

562 element(s) in Lookahead Expression below

eolr − ||| − |.| − |t|