1. Copyright.

Parse a grammar's rule definition.

Example of a rule definition to parse:

```
/*
file: ruledef.txt
Why: example of text to parse by rule_def_phrase grammar.
*/
Rrule_def_phrase AD AB(
lhs
,parallel-control-monitor{
  arbitrator-code
  ***
  }
){
  -> |.|
  ->  ||| "cweb-marker" NS_cweb_marker::TH_cweb_marker {
        rhs-op
        Crule_def_phrase* fsm = (Crule_def_phrase*)parser()->fsm_tbl();
        fsm->rule_def_->add_cweb_marker(sf->p2__->ast());
        ***
        }
  ->  ||| |+| NULL {
      rhs-op
                RSVP(sf->p2__);
                parser()->set_stop_parse(true);
      ***
      }
}
```

2. Fsm Crule_def_phrase class.

3. Crule_def_phrase constructor directive.

⟨ Crule_def_phrase constructor directive 3 ⟩ ≡
  *rule_def_* = 0;

4. Crule_def_phrase op directive.

⟨ Crule_def_phrase op directive 4 ⟩ ≡
  *rule_def_* = 0;

5. Crule_def_phrase user-declaration directive.

⟨ Crule_def_phrase user-declaration directive 5 ⟩ ≡
public: *rule_def* ∗ *rule_def_*;

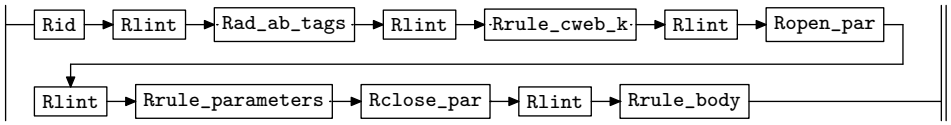
6. Crule_def_phrase user-prefix-declaration directive.

⟨ Crule_def_phrase user-prefix-declaration directive 6 ⟩ ≡

```
#include "lint_balls.h"
#include "cweb_or_c_k.h"
#include "identifier.h"
#include "c_string.h"
#include "t_def_delabort_tags.h"
#include "subrules_phrase.h"
#include "rule_lhs_phrase.h"
#include "yacco2_stbl.h"
#include "parallel_monitor_ph.h"
```

7. *Rrule_def_phrase* rule.

Rrule_def_phrase

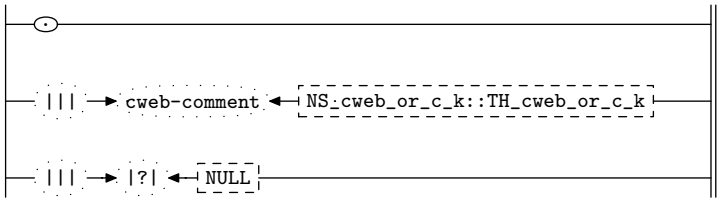


⟨ Rrule_def_phrase subrule 1 op directive 7 ⟩ ≡

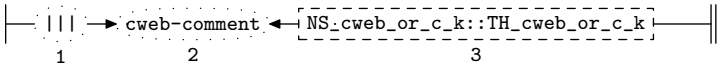
```
Crule_def_phrase * fsm = ( Crule_def_phrase * ) rule_info__.parser__->fsm_tbl__;
fsm->rule_def_->bld_its_tree( );
RSVP(fsm->rule_def_);
fsm->rule_def_ = 0;
```

8. *Rrule_cweb_k* rule.

Rrule_cweb_k



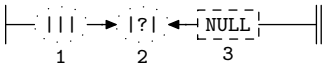
9. *Rrule_cweb_k*'s subrule 2.



⟨ Rrule_cweb_k subrule 2 op directive 9 ⟩ ≡

```
Crule_def_phrase * fsm = ( Crule_def_phrase * ) 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__);
fsm->rule_def_->add_cweb_marker(cweb_t_);
AST::set_content(*cweb_t_, *cw);
AST::join_pts(*cweb_t_, *cwebk_t_);
```

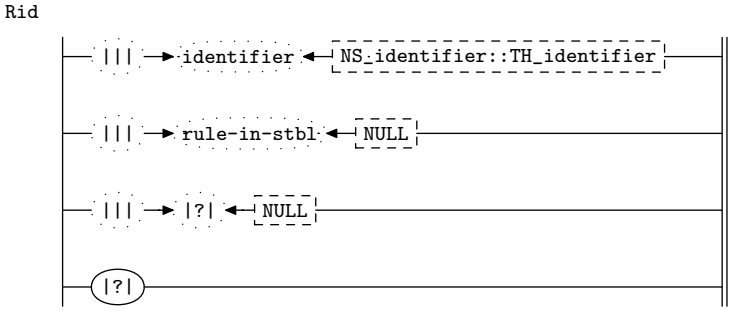
10. *Rrule_cweb_k*'**s subrule 3.**



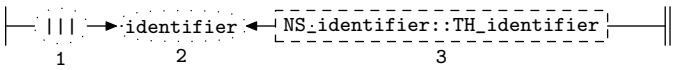
⟨ Rrule_cweb_k subrule 3 op directive 10 ⟩ ≡

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

11. *Rid* **rule.**



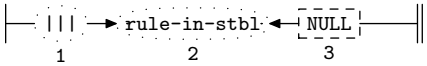
12. *Rid*'**s subrule 1.**



⟨ Rid subrule 1 op directive 12 ⟩ ≡

```
Crule_def_phrase * fsm = ( Crule_def_phrase * ) rule_info__.parser__→fsm_tbl__;
const char *skey = sf→p2__→identifier( )→c_str( );
fsm→rule_def_ = new rule_def(skey);
rule_in_stbl * ristbl = new rule_in_stbl(*fsm→rule_def_);
fsm→rule_def_→set_rc(*sf→p2__, __FILE__, __LINE__);
sf→p2__→set_auto_delete(true);
T_sym_tbl_report_card report_card;
using namespace yacco2_stbl;
add_sym_to_stbl(report_card, *skey, *ristbl, table_entry::defed, table_entry::rule);
if (report_card.status_ ≠ T_sym_tbl_report_card::okay) {
  report_card.err_entry_→set_rc(*sf→p2__, __FILE__, __LINE__);
  RSVP(report_card.err_entry_);
  rule_info__.parser__→set_stop_parse(true);
  return;
}
ristbl→stbl_idx(report_card.pos_);
report_card.tbl_entry_→defined_ = true;
```

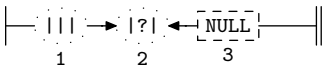
13. ***Rid*'s subrule 2.**



⟨ Rid subrule 2 op directive 13 ⟩ ≡

```
Crule_def_phrase * fsm = ( Crule_def_phrase * ) rule_info__.parser__->fsm_tbl__;
using namespace yacco2_stbl;
T_sym_tbl_report_card report_card;
find_sym_in_stbl(report_card, *sf->p2__->r_def( )->rule_name( )->c_str( ));
if (report_card.action_ == T_sym_tbl_report_card::not_fnd) {
  CAbs_lr1_sym * sym = new Err_rule_not_in_stbl;
  sym->set_rc(*rule_info__.parser__->start_token__, __FILE__, __LINE__);
  RSVP(sym);
  rule_info__.parser__->set_stop_parse(true);
  return;
}
if (report_card.tbl_entry_->defined_ == true) {
  CAbs_lr1_sym * sym = new Err_rule_already_defined;
  sym->set_rc(*rule_info__.parser__->start_token__, __FILE__, __LINE__);
  RSVP(sym);
  rule_info__.parser__->set_stop_parse(true);
  return;
}
fsm->rule_def_ = sf->p2__->r_def( );
report_card.tbl_entry_->defined_ = true;
```

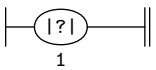
14. ***Rid*'s subrule 3.**



⟨ Rid subrule 3 op directive 14 ⟩ ≡

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

15. ***Rid*'s subrule 4.**

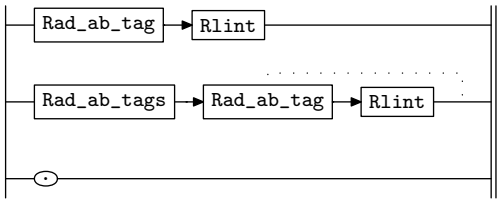


⟨ Rid subrule 4 op directive 15 ⟩ ≡

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

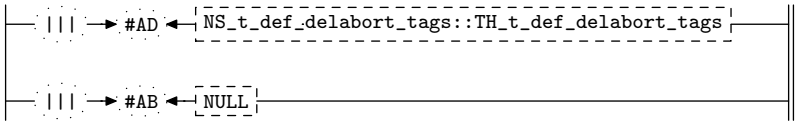
16. *Rad_ab_tags* **rule.**

Rad_ab_tags

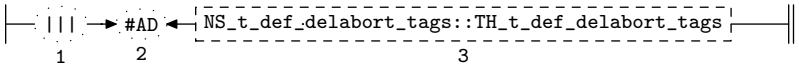


17. *Rad_ab_tag* **rule.**

Rad_ab_tag



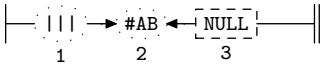
18. *Rad_ab_tag***'s subrule 1.**



⟨ Rad_ab_tag subrule 1 op directive 18 ⟩ ≡

```
Crule_def_phrase * fsm = ( Crule_def_phrase * ) rule_info__.parser__->fsm_tbl__;
if (fsm->rule_def_->autodelete( ) ≡ true) {
    CAbs_lr1_sym * sym = new Err_already_defined_AD;
    sym->set_rc(*sf->p2__, __FILE__, __LINE__);
    RSVP(sym);
    rule_info__.parser__->set_stop_parse(true);
    return;
}
fsm->rule_def_->autodelete(true);
```

19. *Rad_ab_tag***'s subrule 2.**

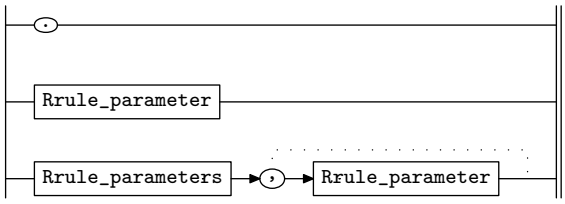


⟨ Rad_ab_tag subrule 2 op directive 19 ⟩ ≡

```
Crule_def_phrase * fsm = ( Crule_def_phrase * ) rule_info__.parser__->fsm_tbl__;
if (fsm->rule_def_->autoabort( ) ≡ true) {
    CAbs_lr1_sym * sym = new Err_already_defined_AB;
    sym->set_rc(*sf->p2__, __FILE__, __LINE__);
    RSVP(sym);
    rule_info__.parser__->set_stop_parse(true);
    return;
}
fsm->rule_def_->autoabort(true);
```

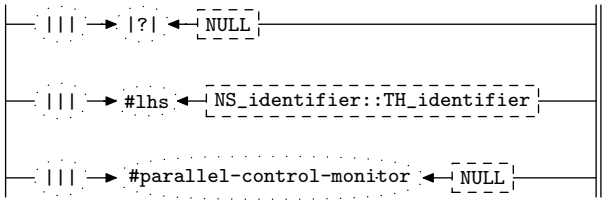
20. *Rrule_parameters* **rule.**

Rrule_parameters

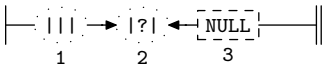


21. *Rrule_parameter* **rule.**

Rrule_parameter



22. *Rrule_parameter*'s **subrule 1.**

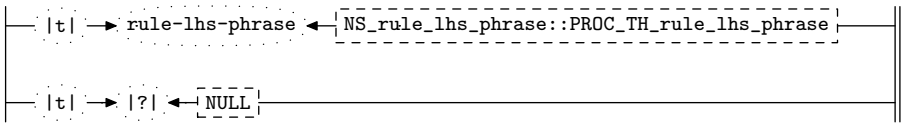


⟨ Rrule_parameter subrule 1 op directive 22 ⟩ ≡

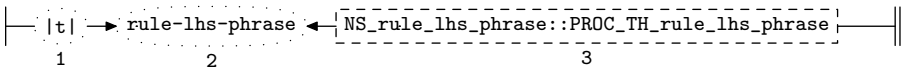
```
CAbs_lr1_sym * sym = new Err_not_lhs_pcnrl_mntr;
sym→set_rc(*sf→p2__, __FILE__, __LINE__);
RSVP(sym);
rule_info__.parser__→set_stop_parse(true);
```

23. *Rlhs_phrase* **rule.**

Rlhs_phrase



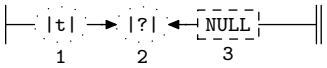
24. *Rlhs_phrase*'s **subrule 1.**



⟨ Rlhs_phrase subrule 1 op directive 24 ⟩ ≡

```
Crule_def_phrase * fsm = ( Crule_def_phrase * ) rule_info__.parser__→fsm_tbl__;
fsm→rule_def_→rule_lhs(sf→p2__);
```

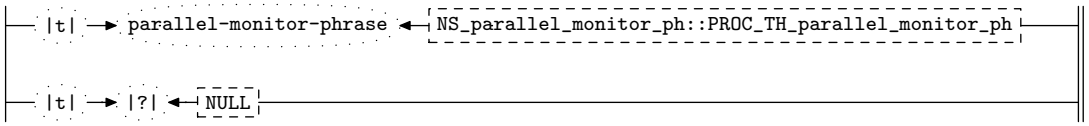
25. *Rlhs_phrase*'s subrule 2.



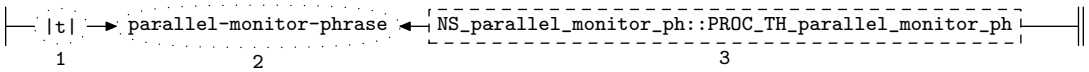
⟨ Rlhs_phrase subrule 2 op directive 25 ⟩ ≡
```
RSVP(sf→p2__);
rule_info__.parser__→set_stop_parse(true);
```

26. *Rparallel_monitor_phrase* **rule.**

Rparallel_monitor_phrase

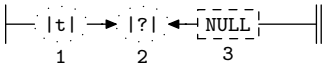


27. *Rparallel_monitor_phrase*'s subrule 1.



⟨ Rparallel_monitor_phrase subrule 1 op directive 27 ⟩ ≡
```
Crule_def_phrase * fsm = ( Crule_def_phrase * ) rule_info__.parser__→fsm_tbl__;
fsm→rule_def_→parallel_mntr(sf→p2__);
```

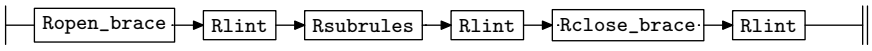
28. *Rparallel_monitor_phrase*'s subrule 2.



⟨ Rparallel_monitor_phrase subrule 2 op directive 28 ⟩ ≡
```
RSVP(sf→p2__);
rule_info__.parser__→set_stop_parse(true);
```

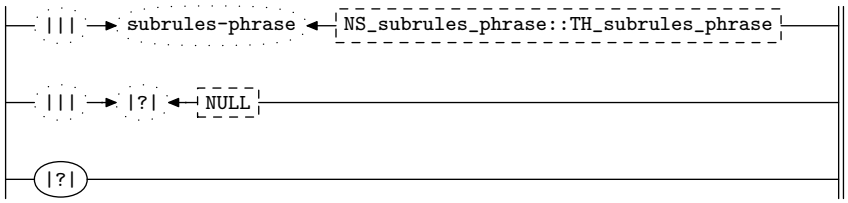
29. *Rrule_body* **rule.**

Rrule_body

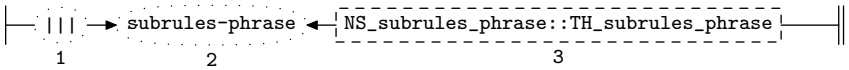


30. *Rsubrules* **rule.**

Rsubrules



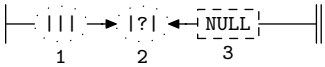
31. *Rsubrules*'s subrule 1.



⟨ Rsubrules subrule 1 op directive 31 ⟩ ≡

```
Crule_def_phrase * fsm = ( Crule_def_phrase * ) rule_info__.parser__→fsm_tbl__;
fsm→rule_def_→subrules(sf→p2__);      /* distribute rule def to its subrules */
std::vector < T_subrule_def *> ::iterator i = sf→p2__→subrules()→begin();
std::vector < T_subrule_def *> ::iterator ie = sf→p2__→subrules()→end();
for ( ; i ≠ ie; ++i) {
  T_subrule_def * srd = *i;
  srd→its_rule_def(fsm→rule_def_);
}
```

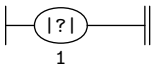
32. *Rsubrules*'s subrule 2.



⟨ Rsubrules subrule 2 op directive 32 ⟩ ≡

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

33. *Rsubrules*'s subrule 3.

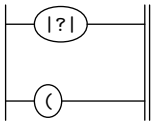


⟨ Rsubrules subrule 3 op directive 33 ⟩ ≡

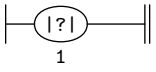
```
CAbs_lr1_sym * sym = new Err_no_sub_rule_present;
sym→set_rc(*sf→p1__, __FILE__, __LINE__);
RSVP(sym);
rule_info__.parser__→set_stop_parse(true);
```

34. *Ropen_par* **rule.**

Ropen_par



35. *Ropen_par*'s subrule 1.

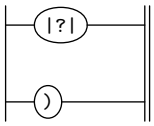


⟨ Ropen_par subrule 1 op directive 35 ⟩ ≡

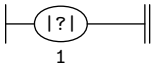
```
CAbs_lr1_sym * sym = new Err_no_open_parenthesis;
sym→set_rc(*sf→p1__, __FILE__, __LINE__);
RSVP(sym);
rule_info__.parser__→set_stop_parse(true);
```

36. *Rclose_par* **rule.**

Rclose_par



37. *Rclose_par*'**s subrule 1.**

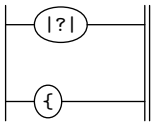


⟨ Rclose_par subrule 1 op directive 37 ⟩ ≡

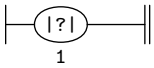
```
CAbs_lr1_sym * sym = new Err_no_close_parenthesis;
sym→set_rc(*sf→p1__, __FILE__, __LINE__);
RSVP(sym);
rule_info__.parser__→set_stop_parse(true);
```

38. *Ropen_brace* **rule.**

Ropen_brace



39. *Ropen_brace*'**s subrule 1.**

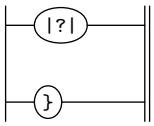


⟨ Ropen_brace subrule 1 op directive 39 ⟩ ≡

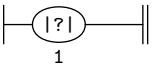
```
CAbs_lr1_sym * sym = new Err_no_open_brace;
sym→set_rc(*sf→p1__, __FILE__, __LINE__);
RSVP(sym);
rule_info__.parser__→set_stop_parse(true);
```

40. *Rclose_brace* **rule.**

Rclose_brace



41. *Rclose_brace*'s subrule 1.

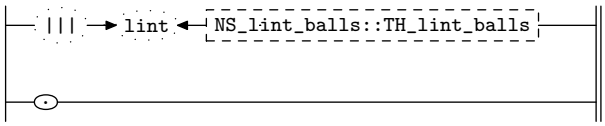


⟨ Rclose_brace subrule 1 op directive 41 ⟩ ≡

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

42. *Rlint* **rule.**

Rlint



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

```
/*
 File: rule_def_phrase.fsc
 Date and Time: Fri Jan  2 15:33:53 2015
*/
transitive   y
grammar-name "rule_def_phrase"
name-space   "NS_rule_def_phrase"
thread-name  "TH_rule_def_phrase"
monolithic   n
file-name    "rule_def_phrase.fsc"
no-of-T      569
list-of-native-first-set-terminals 1
   LR1_questionable_shift_operator
end-list-of-native-first-set-terminals
list-of-transitive-threads 1
  NS_identifier::TH_identifier
end-list-of-transitive-threads
list-of-used-threads 5
  NS_cweb_or_c_k::TH_cweb_or_c_k
  NS_identifier::TH_identifier
  NS_lint_balls::TH_lint_balls
  NS_subrules_phrase::TH_subrules_phrase
  NS_t_def_delabort_tags::TH_t_def_delabort_tags
end-list-of-used-threads
fsm-comments
"Parse a grammar's rule  definition."
```

44. Lr1 State Network.

⇒ State: 1 state type: ^{s}

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
c	Rid			3	4	1	|?|			1	2	2	
c	Rid			3	1	1	|||	identifier NS_identifier::TH_identifier		1	3	5	
c	Rid			3	2	1	|||	rule-in-stbl NULL		1	3	6	
c	Rid			3	3	1	||| |?|	NULL		1	3	4	
c	Rrule_def_phrase			1	1	1	Rid	$\underline{Rlint^{\epsilon}}$ $\underline{Rad\_ab\_tags^{\epsilon}}$ ...		1	7	38	

⇒\|?\| State: 2 state type: ^{r}

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

⇒\|\|\| *arbitration−code*: ϵ State: 3 state type: ^{s}

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rid			3	3	2	|?|			1	4	4	
t	Rid			3	1	2	identifier			1	5	5	
t	Rid			3	2	2	rule-in-stbl			1	6	6	

⇒\|?\| State: 4 state type: ^{r}

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

⇒*identifier* State: 5 state type: ^{r}

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

⇒*rule−in−stbl* State: 6 state type: ^{r}

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

⇒*Rid* 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	1
c	Rlint			16	1	1	|||	lint NS_lint_balls::TH_lint_balls		7	46	13	
t	Rrule_def_phrase			1	1	2	Rlint	$\underline{Rad\_ab\_tags^{\epsilon}}$ $\underline{Rlint^{\epsilon}}$ ...		1	8	38	

⇒*Rlint* State: 8 state type: $^{s/r}$

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
c	Rad_ab_tags			4	3	1	ϵ			8	0	8	1
c	Rad_ab_tag			5	2	1	||| #	AB NULL		8	48	12	
c	Rad_ab_tag			5	1	1	||| #	AD NS_t_def_delabort_tags::TH_t_def_delabort_tags		8	48	11	
c	Rad_ab_tags			4	2	1	Rad_ab_tags	$\underline{Rad\_ab\_tag}$		8	9	50	
t	Rrule_def_phrase			1	1	3	Rad_ab_tags	$\underline{Rlint^{\epsilon}}$ $\underline{Rrule\_cweb\_k^{\epsilon}}$ ...		1	9	38	
c	Rad_ab_tags			4	1	1	Rad_ab_tag	$\underline{Rlint^{\epsilon}}$		8	51	52	

⇒*Rad_ab_tags* 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	2
c	Rad_ab_tag			5	2	1	||| #	AB NULL		9	10	12	
c	Rad_ab_tag			5	1	1	||| #	AD NS_t_def_delabort_tags::TH_t_def_delabort_tags		9	10	11	

c	Rlint	16	1	1	||| lint NS_lint_balls::TH_lint_balls	9	10	13
t	Rad_ab_tags	4	2	2	Rad_ab_tag *Rlint*$^{\epsilon}$	8	49	50
t	Rrule_def_phrase	1	1	4	Rlint *Rrule_cweb_k*$^{\epsilon}$ *Rlint*$^{\epsilon}$ ...	1	14	38

⇒\|\|\| *arbitration−code*: ϵ State: 10 state type: ^{s}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rad_ab_tag		5	1	2	# AD		9	11	11	
t	Rad_ab_tag		5	2	2	# AB		9	12	12	
t	Rlint		16	1	2	lint		9	13	13	

⇒*#AD* State: 11 state type: ^{r}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rad_ab_tag		5	1	3			9	0	11	1

⇒*#AB* State: 12 state type: ^{r}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rad_ab_tag		5	2	3			9	0	12	1

⇒*lint* State: 13 state type: ^{r}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rlint		16	1	3			9	0	13	2

⇒*Rlint* State: 14 state type: $^{s/r}$

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rrule_cweb_k		2	1	1	ϵ		14	0	14	1
c	Rrule_cweb_k		2	2	1	||| cweb-comment NS_cweb_or_c_k::TH_cweb_or_c_k		14	53	55	
c	Rrule_cweb_k		2	3	1	||| |?| NULL		14	53	54	
t	Rrule_def_phrase		1	1	5	Rrule_cweb_k *Rlint*$^{\epsilon}$ *Ropen_par*		1	15	38	

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

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rlint		16	2	1	ϵ		15	0	15	3
c	Rlint		16	1	1	||| lint NS_lint_balls::TH_lint_balls		15	46	13	
t	Rrule_def_phrase		1	1	6	Rlint *Ropen_par*		1	16	38	

⇒*Rlint* State: 16 state type: ^{s}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Ropen_par		12	1	1	|?|		16	56	56	
c	Ropen_par		12	2	1	(		16	57	57	
t	Rrule_def_phrase		1	1	7	Ropen_par *Rlint*$^{\epsilon}$ *Rrule_parameters*$^{\epsilon}$ ...		1	17	38	

⇒*Ropen_par* State: 17 state type: $^{s/r}$

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rlint		16	2	1	ϵ		17	0	17	4
c	Rlint		16	1	1	||| lint NS_lint_balls::TH_lint_balls		17	46	13	
t	Rrule_def_phrase		1	1	8	Rlint *Rrule_parameters*$^{\epsilon}$ *Rclose_par*		1	18	38	

⇒*Rlint* State: 18 state type: $^{s/r}$

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rrule_parameters		6	1	1	ϵ		18	0	18	5
c	Rrule_parameter		7	1	1	||| |?| NULL		18	21	22	
c	Rrule_parameter		7	2	1	||| # lhs NS_identifier::TH_identifier		18	21	32	

c	Rrule_parameter	7	3	1	||| # parallel-control-monitor NULL	18	21	27
t	Rrule_def_phrase	1	1	9	Rrule_parameters *Rclose_par*	1	19	38
c	Rrule_parameters	6	3	1	Rrule_parameters ,	18	19	33
c	Rrule_parameters	6	2	1	Rrule_parameter	18	58	58

⇒*Rrule_parameters*

State: 19 state type: ^{s}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
c	Rclose_par		13	1	1	|?|			19	59	59	
c	Rclose_par		13	2	1	)			19	60	60	
t	Rrule_parameters		6	3	2	,			18	20	33	
t	Rrule_def_phrase		1	1	10	Rclose_par *Rlint$^{\epsilon}$* *Rrule_body*			1	34	38	

⇒,

State: 20 state type: ^{s}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
c	Rrule_parameter		7	1	1	||| |?| NULL			20	21	22	
c	Rrule_parameter		7	2	1	||| # lhs NS_identifier::TH_identifier			20	21	32	
c	Rrule_parameter		7	3	1	||| # parallel-control-monitor NULL			20	21	27	
t	Rrule_parameters		6	3	3	Rrule_parameter			18	33	33	

⇒\|\|\| *arbitration−code*: ϵ

State: 21 state type: ^{s}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rrule_parameter		7	1	2	|?|			20	22	22	
t	Rrule_parameter		7	3	2	# parallel-control-monitor			20	23	27	
t	Rrule_parameter		7	2	2	# lhs			20	28	32	

⇒\|?\|

State: 22 state type: ^{r}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rrule_parameter		7	1	3				20	0	22	5

⇒*#parallel−control−monitor*

State: 23 state type: ^{s}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red
c	Rparallel_monitor_phrase		9	1	1	|t| parallel-monitor-phrase NS_parallel_monitor_ph::PROC_TH_parallel_monitor_ph			23	24	26
c	Rparallel_monitor_phrase		9	2	1	|t| |?| NULL			23	24	25
t	Rrule_parameter		7	3	4	Rparallel_monitor_phrase			20	27	27

⇒\|t\|

State: 24 state type: ^{s}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rparallel_monitor_phrase		9	2	2	|?|			23	25	25	
t	Rparallel_monitor_phrase		9	1	2	parallel-monitor-phrase			23	26	26	

⇒\|?\|

State: 25 state type: ^{r}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rparallel_monitor_phrase		9	2	3				23	0	25	5

⇒*parallel−monitor−phrase*

State: 26 state type: ^{r}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rparallel_monitor_phrase		9	1	3				23	0	26	5

⇒*Rparallel_monitor_phrase*

State: 27 state type: ^{r}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rrule_parameter		7	3	5				20	0	27	5

⇒*#lhs* State: 28 state type: *s*

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rlhs_phrase		8	1	1	`|t|` rule-lhs-phrase NS_rule_lhs_phrase::PROC_TH_rule_lhs_phrase		28	29	31	
c	Rlhs_phrase		8	2	1	`|t|` `|?|` NULL		28	29	30	
t	Rrule_parameter		7	2	4	Rlhs_phrase		20	32	32	

⇒`|t|` State: 29 state type: *s*

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rlhs_phrase		8	2	2	`|?|`		28	30	30	
t	Rlhs_phrase		8	1	2	rule-lhs-phrase		28	31	31	

⇒`|?|` State: 30 state type: *r*

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rlhs_phrase		8	2	3			28	0	30	5

⇒*rule−lhs−phrase* State: 31 state type: *r*

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rlhs_phrase		8	1	3			28	0	31	5

⇒*Rlhs_phrase* State: 32 state type: *r*

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rrule_parameter		7	2	5			20	0	32	5

⇒*Rrule_parameter* State: 33 state type: *r*

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rrule_parameters		6	3	4			18	0	33	5

⇒*Rclose_par* State: 34 state type: *s/r*

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rlint		16	2	1	ϵ		34	0	34	6
c	Rlint		16	1	1	`|||` lint NS_lint_balls::TH_lint_balls		34	46	13	
t	Rrule_def_phrase		1	1	11	Rlint *Rrule_body*		1	35	38	

⇒*Rlint* State: 35 state type: *s*

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Ropen_brace		14	1	1	`|?|`		35	36	36	
c	Ropen_brace		14	2	1	`{`		35	37	37	
t	Rrule_def_phrase		1	1	12	Rrule_body		1	38	38	
c	Rrule_body		10	1	1	Ropen_brace *Rlint*$^{\epsilon}$ *Rsubrules*		35	39	47	

⇒`|?|` State: 36 state type: *r*

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Ropen_brace		14	1	2			35	0	36	7

⇒`{` State: 37 state type: *r*

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Ropen_brace		14	2	2			35	0	37	7

⇒*Rrule_body* State: 38 state type: *r*

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

⇒*Ropen_brace* State: 39 state type: *s/r*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rlint		16	2	1	ϵ		39	0	39	7
c	Rlint		16	1	1	||| lint NS_lint_balls::TH_lint_balls		39	46	13	
t	Rrule_body		10	1	2	Rlint *Rsubrules*		35	40	47	

⇒*Rlint* State: 40 state type: *s*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rsubrules		11	3	1	|?|		40	61	61	
c	Rsubrules		11	1	1	||| subrules-phrase NS_subrules_phrase::TH_subrules_phrase		40	62	64	
c	Rsubrules		11	2	1	||| |?| NULL		40	62	63	
t	Rrule_body		10	1	3	Rsubrules *Rlint*$^{\epsilon}$ *Rclose_brace*		35	41	47	

⇒*Rsubrules* State: 41 state type: *s/r*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rlint		16	2	1	ϵ		41	0	41	8
c	Rlint		16	1	1	||| lint NS_lint_balls::TH_lint_balls		41	46	13	
t	Rrule_body		10	1	4	Rlint *Rclose_brace*		35	42	47	

⇒*Rlint* State: 42 state type: *s*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rclose_brace		15	1	1	|?|		42	43	43	
c	Rclose_brace		15	2	1	}		42	44	44	
t	Rrule_body		10	1	5	Rclose_brace *Rlint*$^{\epsilon}$		35	45	47	

⇒\|?\| State: 43 state type: *r*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rclose_brace		15	1	2			42	0	43	2

⇒} State: 44 state type: *r*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rclose_brace		15	2	2			42	0	44	2

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

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rlint		16	2	1	ϵ		45	0	45	2
c	Rlint		16	1	1	||| lint NS_lint_balls::TH_lint_balls		45	46	13	
t	Rrule_body		10	1	6	Rlint		35	47	47	

⇒\|\|\| *arbitration−code*: ϵ State: 46 state type: *s*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rlint		16	1	2	lint		45	13	13	

⇒*Rlint* State: 47 state type: *r*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rrule_body		10	1	7			35	0	47	2

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

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rad_ab_tag		5	1	2	# AD		8	11	11	
t	Rad_ab_tag		5	2	2	# AB		8	12	12	

⇒*Rad_ab_tag* 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	1
c	Rlint		16	1	1	|||	lint NS_lint_balls::TH_lint_balls		49	46	13	
t	Rad_ab_tags		4	2	3	Rlint			8	50	50	

⇒*Rlint* State: 50 state type: *r*

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rad_ab_tags		4	2	4				8	0	50	1

⇒*Rad_ab_tag* State: 51 state type: *s/r*

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
c	Rlint		16	2	1	ϵ			51	0	51	1
c	Rlint		16	1	1	|||	lint NS_lint_balls::TH_lint_balls		51	46	13	
t	Rad_ab_tags		4	1	2	Rlint			8	52	52	

⇒*Rlint* State: 52 state type: *r*

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rad_ab_tags		4	1	3				8	0	52	1

⇒\|\|\| *arbitration−code*: ϵ State: 53 state type: *s*

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rrule_cweb_k		2	3	2	|?|			14	54	54	
t	Rrule_cweb_k		2	2	2	cweb-comment			14	55	55	

⇒\|?\| State: 54 state type: *r*

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rrule_cweb_k		2	3	3				14	0	54	1

⇒*cweb−comment* State: 55 state type: *r*

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rrule_cweb_k		2	2	3				14	0	55	1

⇒\|?\| State: 56 state type: *r*

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Ropen_par		12	1	2				16	0	56	4

⇒(State: 57 state type: *r*

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Ropen_par		12	2	2				16	0	57	4

⇒*Rrule_parameter* State: 58 state type: *r*

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rrule_parameters		6	2	2				18	0	58	5

⇒\|?\| State: 59 state type: *r*

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rclose_par		13	1	2				19	0	59	9

⇒) State: 60 state type: *r*

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

t	Rclose_par	13	2	2			19	0	60	9

⇒|?|

State: 61 state type: ^{r}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rsubrules		11	3	2				40	0	61	10

⇒||| *arbitration−code*: ϵ

State: 62 state type: ^{s}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rsubrules		11	2	2	|?|			40	63	63	
t	Rsubrules		11	1	2	subrules-phrase			40	64	64	

⇒|?|

State: 63 state type: ^{r}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rsubrules		11	2	3				40	0	63	10

⇒*subrules−phrase*

State: 64 state type: ^{r}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rsubrules		11	1	3				40	0	64	10

45. Index.

⟨ Crule_def_phrase constructor directive 3 ⟩
⟨ Crule_def_phrase op directive 4 ⟩
⟨ Crule_def_phrase user-declaration directive 5 ⟩
⟨ Crule_def_phrase user-prefix-declaration directive 6 ⟩
⟨ Rad_ab_tag subrule 1 op directive 18 ⟩
⟨ Rad_ab_tag subrule 2 op directive 19 ⟩
⟨ Rclose_brace subrule 1 op directive 41 ⟩
⟨ Rclose_par subrule 1 op directive 37 ⟩
⟨ Rid subrule 1 op directive 12 ⟩
⟨ Rid subrule 2 op directive 13 ⟩
⟨ Rid subrule 3 op directive 14 ⟩
⟨ Rid subrule 4 op directive 15 ⟩
⟨ Rlhs_phrase subrule 1 op directive 24 ⟩
⟨ Rlhs_phrase subrule 2 op directive 25 ⟩
⟨ Ropen_brace subrule 1 op directive 39 ⟩
⟨ Ropen_par subrule 1 op directive 35 ⟩
⟨ Rparallel_monitor_phrase subrule 1 op directive 27 ⟩
⟨ Rparallel_monitor_phrase subrule 2 op directive 28 ⟩
⟨ Rrule_cweb_k subrule 2 op directive 9 ⟩
⟨ Rrule_cweb_k subrule 3 op directive 10 ⟩
⟨ Rrule_def_phrase subrule 1 op directive 7 ⟩
⟨ Rrule_parameter subrule 1 op directive 22 ⟩
⟨ Rsubrules subrule 1 op directive 31 ⟩
⟨ Rsubrules subrule 2 op directive 32 ⟩
⟨ Rsubrules subrule 3 op directive 33 ⟩

rule_def_phrase Grammar

Date: January 2, 2015 at 15:38

File: rule_def_phrase.lex Ns: NS_rule_def_phrase

Version: 1.0 Debug: false

Grammar Comments: Type: Thread

Parse a grammar's rule definition.

1 element(s) in Lookahead Expression below

eolr