

**1. Copyright.**

Copyright © Dave Bone 1998 - 2015

**2. *parallel\_monitor\_ph* thread.**

Parse arbitration construct. Typically there is no arbitration to be done but when 2 or more threads of subset / superset context are deployed deplomacy is required. This thread is a “chained procedure call”.

Example of a parallel-control-monitor construct to parse:

```

/*
file: ppmntr.txt
Why: text example parallel monitor construct.
*/

parallel-control-monitor {
  /@
  unquoted-string supersets identifier
  and also includes other T like $-$$>$ and cweb comments ie cweb-marker.
  Code only gets tried when \# parms in accept queue $>$ 1.
  Single entry is one of identifier or cweb.
  @/
  arbitrator-code
  using namespace NS_yacco2_T_enum;
  {
    parallel_accept_queue_iterator_type z;
    z = To_judge->accept_queue__.find(NS_yacco2_T_enum::T_Enum::T_T_cweb_marker_);
    if(z != ie){// cweb marker fnd
      i = z;
      accept_parse_parm = (*z).second;
      (*z).second = 0;
      goto arbitrated_parameter;
    }
    // negative: throw out the superset: unquoted-string
    // cuz identifier or one of its derivatives should be accepted
    i = To_judge->accept_queue__.find(NS_yacco2_T_enum::T_Enum::T_T_unquoted_string_);
    delete (*i).second;
    To_judge->accept_queue__.erase(i);
    i = To_judge->accept_queue__.begin();
    accept_parse_parm = (*i).second;
  }
  ***
}

```

**3. Fsm Cparallel\_monitor\_ph class.****4. Cparallel\_monitor\_ph constructor directive.**

⟨Cparallel\_monitor\_ph constructor directive 4⟩ ≡  
*parallel\_monitor\_phrase\_* = 0;

**5. Cparallel\_monitor\_ph op directive.**

```

⟨Cparallel_monitor_ph op directive 5⟩ ≡
  if (parallel_monitor_phrase_ ≠ 0) {
    delete parallel_monitor_phrase_;
    parallel_monitor_phrase_ = 0;
  }
  parallel_monitor_phrase_ = new T_parallel_monitor_phrase;
  parallel_monitor_phrase_→set_rc(*parser_→start_token_, __FILE__, __LINE__);
  AST *t = new AST(*parallel_monitor_phrase_);
  parallel_monitor_phrase_→phrase_tree(t);

```

**6. Cparallel\_monitor\_ph user-declaration directive.**

```

⟨Cparallel_monitor_ph user-declaration directive 6⟩ ≡
public: void add_sdc_to_directive(yacco2 :: CAbs_lr1_sym *Dir, T_syntax_code *Sdc);
  T_parallel_monitor_phrase *parallel_monitor_phrase_;

```

**7. Cparallel\_monitor\_ph user-implementation directive.**

```

⟨Cparallel_monitor_ph user-implementation directive 7⟩ ≡
  void Cparallel_monitor_ph :: 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_arbitrator_code_: {
      T_arbitrator_code *k = ( T_arbitrator_code * ) 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);
      }
  } }

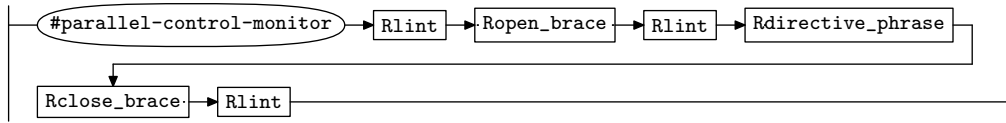
```

**8. Cparallel\_monitor\_ph user-prefix-declaration directive.**

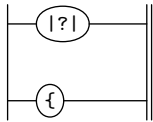
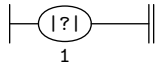
```

⟨Cparallel_monitor_ph user-prefix-declaration directive 8⟩ ≡
  using namespace NS_yacco2_terminals;
#include "lint_balls.h"
#include "eol.h"
#include "c_comments.h"
#include "cweb_or_c_k.h"
#include "identifier.h"
#include "c_string.h"
#include "o2_sdc.h"

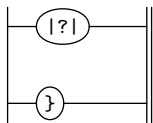
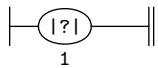
```

9. *Rparallel\_monitor\_phrase* rule.*Rparallel\_monitor\_phrase*

$\langle \text{Rparallel\_monitor\_phrase subrule 1 op directive 9} \rangle \equiv$   
 $C_{\text{parallel\_monitor\_ph}} * fsm = ( C_{\text{parallel\_monitor\_ph}} * ) rule\_info\_parser\_fsm\_tbl\_;$   
 $RSVP(fsm\_parallel\_monitor\_phrase\_);$   
 $fsm\_parallel\_monitor\_phrase\_ = 0;$

10. *Ropen\_brace* rule.*Ropen\_brace*11. *Ropen\_brace*'s subrule 1.

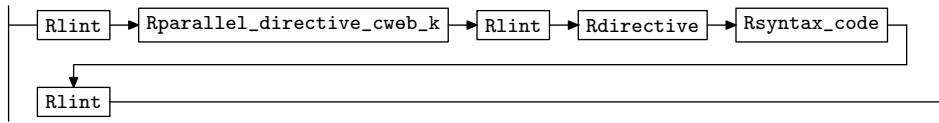
$\langle \text{Ropen\_brace subrule 1 op directive 11} \rangle \equiv$   
 $CAbs\_lr1\_sym * sym = \mathbf{new} Err\_no\_open\_brace;$   
 $sym\_set\_rc(*sf\_p1\_, \_FILE\_, \_LINE\_);$   
 $RSVP(sym);$   
 $rule\_info\_parser\_set\_stop\_parse(true);$

12. *Rclose\_brace* rule.*Rclose\_brace*13. *Rclose\_brace*'s subrule 1.

$\langle \text{Rclose\_brace subrule 1 op directive 13} \rangle \equiv$   
 $CAbs\_lr1\_sym * sym = \mathbf{new} Err\_no\_close\_brace;$   
 $sym\_set\_rc(*sf\_p1\_, \_FILE\_, \_LINE\_);$   
 $RSVP(sym);$   
 $rule\_info\_parser\_set\_stop\_parse(true);$

14. *Rdirective\_phrase* rule.

Rdirective\_phrase



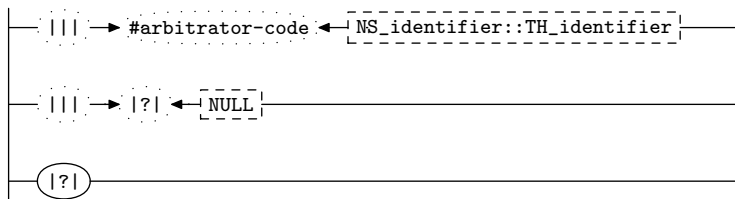
⟨ Rdirective\_phrase subrule 1 op directive 14 ⟩ ≡

```

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);
Cparallel_monitor_ph * fsm = ( Cparallel_monitor_ph * ) rule_info--.parser--fsm_tbl--;
fsm-add_sdc_to_directive(dir-directive_, sdc→syntax_code_);
  
```

15. *Rdirective* rule.

Rdirective



16. *Rdirective* op directive.

⟨ Rdirective op directive 16 ⟩ ≡

```

if (directive_ ≡ 0) return;
Cparallel_monitor_ph * fsm = ( Cparallel_monitor_ph * ) rule_info--.parser--fsm_tbl--;
CAbs_lr1_sym * result = fsm-parallel_monitor_phrase-→add_directive_to_mntr(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);
  
```

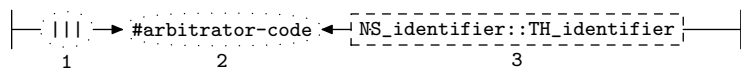
17. *Rdirective* user-declaration directive.

⟨ Rdirective user-declaration directive 17 ⟩ ≡

```

CAbs_lr1_sym * directive_;
  
```

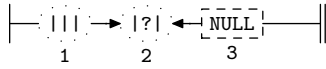
18. *Rdirective's* subrule 1.



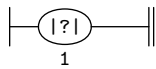
⟨ Rdirective subrule 1 op directive 18 ⟩ ≡

```

directive_ = sf-p2--;
  
```

**19. *Rdirective's subrule 2.***

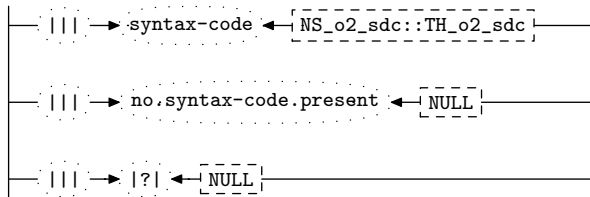
$\langle \text{Rdirective subrule 2 op directive 19} \rangle \equiv$   
*directive\_* = 0;  
*CAbs\_lr1\_sym* \* *sym* = **new** *Err\_not\_arbitration\_code\_kw*;  
*sym*→*set\_rc*(\**sf*→*p2*\_, \_\_FILE\_\_, \_\_LINE\_\_);  
*RSVP*(*sym*);  
*rule\_info*\_.*parser*\_→*set\_stop\_parse*(*true*);

**20. *Rdirective's subrule 3.***

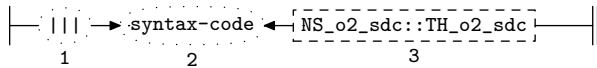
$\langle \text{Rdirective subrule 3 op directive 20} \rangle \equiv$   
*CAbs\_lr1\_sym* \* *sym* = **new** *Err\_no\_directive\_present*;  
*sym*→*set\_rc*(\**sf*→*p1*\_, \_\_FILE\_\_, \_\_LINE\_\_);  
*RSVP*(*sym*);  
*rule\_info*\_.*parser*\_→*set\_stop\_parse*(*true*);

**21. *Rsyntax\_code* rule.**

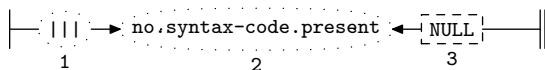
*Rsyntax\_code*

**22. *Rsyntax\_code* user-declaration directive.**

$\langle \text{Rsyntax_code user-declaration directive 22} \rangle \equiv$   
*T\_syntax\_code* \* *syntax\_code\_*;

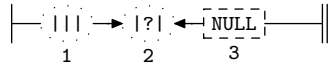
**23. *Rsyntax\_code's subrule 1.***

$\langle \text{Rsyntax_code subrule 1 op directive 23} \rangle \equiv$   
*syntax\_code\_* = *sf*→*p2*\_;

**24. *Rsyntax\_code's subrule 2.***

$\langle \text{Rsyntax_code subrule 2 op directive 24} \rangle \equiv$   
*syntax\_code\_* = 0;  
*sf*→*p2*\_→*set\_auto\_delete*(*true*);

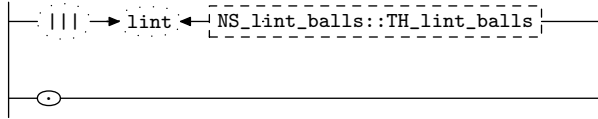
**25. Rsyntax\_code's subrule 3.**



⟨ Rsyntax\_code subrule 3 op directive 25 ⟩ ≡  
 syntax\_code\_ = 0;  
 RSVP(sf-p2\_);  
 rule\_info\_.parser\_--set\_stop\_parse(true);

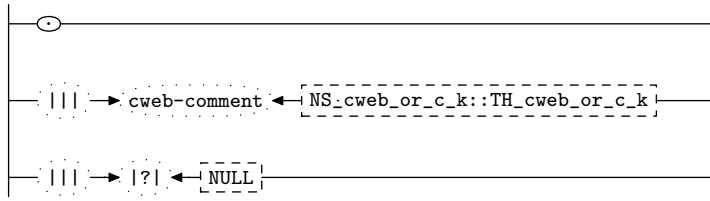
**26. Rlint rule.**

Rlint



**27. Rparallel\_directive\_cweb\_k rule.**

Rparallel\_directive\_cweb\_k



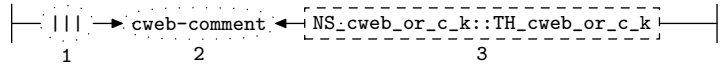
**28. Rparallel\_directive\_cweb\_k constructor directive.**

⟨ Rparallel\_directive\_cweb\_k constructor directive 28 ⟩ ≡  
 cweb\_t\_ = 0;

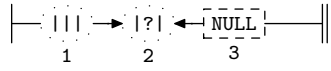
**29. Rparallel\_directive\_cweb\_k user-declaration directive.**

⟨ Rparallel\_directive\_cweb\_k user-declaration directive 29 ⟩ ≡  
 AST \* cweb\_t\_;

**30. Rparallel\_directive\_cweb\_k's subrule 2.**



⟨ Rparallel\_directive\_cweb\_k subrule 2 op directive 30 ⟩ ≡  
 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::set\_content(\*cweb\_t\_, \*cw);  
 AST::join\_pts(\*cweb\_t\_, \*cwebk\_t\_);

**31.** *Rparallel\_directive\_cweb\_k's subrule 3.*

$\langle Rparallel\_directive\_cweb\_k \text{ subrule } 3 \text{ op directive } 31 \rangle \equiv$   
 RSVP(*sf-p2\_*);  
*rule\_info\_.parser--set\_stop\_parse(true);*



**32. First Set Language for  $O_2^{linker}$ .**

```
/*
  File: parallel_monitor_ph.fsc
  Date and Time: Fri Jan  2 15:33:47 2015
*/
transitive      n
grammar-name    "parallel_monitor_ph"
name-space      "NS_parallel_monitor_ph"
thread-name     "TH_parallel_monitor_ph"
monolithic      n
file-name       "parallel_monitor_ph.fsc"
no-of-T         569
list-of-native-first-set-terminals 1
  T_parallel_control_monitor
end-list-of-native-first-set-terminals
list-of-transitive-threads 0
end-list-of-transitive-threads
list-of-used-threads 4
  NS_cweb_or_c_k::TH_cweb_or_c_k
  NS_identifier::TH_identifier
  NS_lint_balls::TH_lint_balls
  NS_o2_sdc::TH_o2_sdc
end-list-of-used-threads
fsm-comments
"Parse a rule's arbitration code: \n into the valley of someone's dementia..."
```

**33. Lr1 State Network.**

$\Rightarrow$					State: 1 state type: $s$				
$\leftarrow$	rule	$\rightarrow$	R#	sr#	Po	$\leftarrow$	subrule element	$\rightarrow$	Brn Gto Red LA
c	Rparallel_monitor_phrase		1	1	1		# parallel-control-monitor		1 2 12
$\Rightarrow$	#parallel-control-monitor						State: 2 state type: $s/r$		
$\leftarrow$	rule	$\rightarrow$	R#	sr#	Po	$\leftarrow$	subrule element	$\rightarrow$	Brn Gto Red LA
c	Rlint		7	2	1		$\epsilon$		2 0 2 1
c	Rlint		7	1	1		lint NS lint_balls::TH lint_balls		2 10 11
t	Rparallel_monitor_phrase		1	1	2		Rlint <u>Ropen_brace</u>		1 3 12
$\Rightarrow$	Rlint						State: 3 state type: $s$		
$\leftarrow$	rule	$\rightarrow$	R#	sr#	Po	$\leftarrow$	subrule element	$\rightarrow$	Brn Gto Red LA
c	Ropen_brace		2	1	1		?		3 13 13
c	Ropen_brace		2	2	1		{		3 14 14
t	Rparallel_monitor_phrase		1	1	3		Ropen_brace <u>Rlint<math>\epsilon</math> Rdirective_phrase</u>		1 4 12
$\Rightarrow$	Ropen_brace						State: 4 state type: $s/r$		
$\leftarrow$	rule	$\rightarrow$	R#	sr#	Po	$\leftarrow$	subrule element	$\rightarrow$	Brn Gto Red LA
c	Rlint		7	2	1		$\epsilon$		4 0 4 2
c	Rlint		7	1	1		lint NS lint_balls::TH lint_balls		4 10 11
t	Rparallel_monitor_phrase		1	1	4		Rlint <u>Rdirective_phrase</u>		1 5 12
$\Rightarrow$	Rlint						State: 5 state type: $s/r$		
$\leftarrow$	rule	$\rightarrow$	R#	sr#	Po	$\leftarrow$	subrule element	$\rightarrow$	Brn Gto Red LA
c	Rlint		7	2	1		$\epsilon$		5 0 5 2
c	Rlint		7	1	1		lint NS lint_balls::TH lint_balls		5 10 11
t	Rparallel_monitor_phrase		1	1	5		Rdirective_phrase <u>Rclose_brace</u>		1 6 12
c	Rdirective_phrase		4	1	1		Rlint <u>Rparallel_directive_cweb_k<math>\epsilon</math> Rlint<math>\epsilon</math> ...</u>		5 15 24
$\Rightarrow$	Rdirective_phrase						State: 6 state type: $s$		
$\leftarrow$	rule	$\rightarrow$	R#	sr#	Po	$\leftarrow$	subrule element	$\rightarrow$	Brn Gto Red LA
c	Rclose_brace		3	1	1		?		6 7 7
c	Rclose_brace		3	2	1		}		6 8 8
t	Rparallel_monitor_phrase		1	1	6		Rclose_brace <u>Rlint<math>\epsilon</math></u>		1 9 12
$\Rightarrow$	?						State: 7 state type: $r$		
$\leftarrow$	rule	$\rightarrow$	R#	sr#	Po	$\leftarrow$	subrule element	$\rightarrow$	Brn Gto Red LA
t	Rclose_brace		3	1	2				6 0 7 3
$\Rightarrow$	}						State: 8 state type: $r$		
$\leftarrow$	rule	$\rightarrow$	R#	sr#	Po	$\leftarrow$	subrule element	$\rightarrow$	Brn Gto Red LA
t	Rclose_brace		3	2	2				6 0 8 3
$\Rightarrow$	Rclose_brace						State: 9 state type: $s/r$		
$\leftarrow$	rule	$\rightarrow$	R#	sr#	Po	$\leftarrow$	subrule element	$\rightarrow$	Brn Gto Red LA
c	Rlint		7	2	1		$\epsilon$		9 0 9 3
c	Rlint		7	1	1		lint NS lint_balls::TH lint_balls		9 10 11
t	Rparallel_monitor_phrase		1	1	7		Rlint		1 12 12
$\Rightarrow$	arbitration-code: $\epsilon$						State: 10 state type: $s$		

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rlint		7	1	2	lint			9	11	11	
⇒ <i>lint</i>												
←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rlint		7	1	3				9	0	11	3
⇒ <i>Rlint</i>												
←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rparallel_monitor_phrase		1	1	8				1	0	12	3
⇒ <i> ? </i>												
←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Ropen_brace		2	1	2				3	0	13	2
⇒ <i>{</i>												
←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Ropen_brace		2	2	2				3	0	14	2
⇒ <i>Rlint</i>												
←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
c	Rparallel_directive_cweb_k		8	1	1	ε			15	0	15	2
c	Rparallel_directive_cweb_k		8	2	1	cweb-comment NS_cweb_or_c.k::TH_cweb_or_c.k			15	25	27	
c	Rparallel_directive_cweb_k		8	3	1	<i> ? </i> NULL			15	25	26	
t	Rdirective_phrase		4	1	2	Rparallel_directive_cweb_k <i>Rlint</i> <sup>ε</sup> <i>Rdirective</i>			5	16	24	
⇒ <i>Rparallel_directive_cweb_k</i>												
←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
c	Rlint		7	2	1	ε			16	0	16	2
c	Rlint		7	1	1	lint NS_lint_balls::TH_lint_balls			16	10	11	
t	Rdirective_phrase		4	1	3	Rlint <i>Rdirective</i>			5	17	24	
⇒ <i>Rlint</i>												
←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
c	Rdirective		5	3	1	<i> ? </i>			17	28	28	
c	Rdirective		5	1	1	# arbitrator-code NS_identifer::TH_identifier			17	29	31	
c	Rdirective		5	2	1	<i> ? </i> NULL			17	29	30	
t	Rdirective_phrase		4	1	4	Rdirective <i>Rsyntax_code</i>			5	18	24	
⇒ <i>Rdirective</i>												
←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
c	Rsyntax_code		6	1	1	syntax-code NS_o2_sdc::TH_o2_sdc			18	19	21	
c	Rsyntax_code		6	2	1	no syntax-code present NULL			18	19	22	
c	Rsyntax_code		6	3	1	<i> ? </i> NULL			18	19	20	
t	Rdirective_phrase		4	1	5	Rsyntax_code <i>Rlint</i> <sup>ε</sup>			5	23	24	
⇒ <i>    arbitration-code: ε</i>												
←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rsyntax_code		6	3	2	<i> ? </i>			18	20	20	
t	Rsyntax_code		6	1	2	syntax-code			18	21	21	
t	Rsyntax_code		6	2	2	no syntax-code present			18	22	22	

$\Rightarrow$  ?		State: 20 state type: $r$	
← rule	→ R# sr# Po ←	subrule element	→ Brn Gto Red LA
t Rsyntax_code	6 3 3		18 0 20 4
$\Rightarrow$ <i>syntax-code</i>		State: 21 state type: $r$	
← rule	→ R# sr# Po ←	subrule element	→ Brn Gto Red LA
t Rsyntax_code	6 1 3		18 0 21 4
$\Rightarrow$ <i>nosyntax-codepresent</i>		State: 22 state type: $r$	
← rule	→ R# sr# Po ←	subrule element	→ Brn Gto Red LA
t Rsyntax_code	6 2 3		18 0 22 4
$\Rightarrow$ <i>Rsyntax_code</i>		State: 23 state type: $s/r$	
← rule	→ R# sr# Po ←	subrule element	→ Brn Gto Red LA
c Rlint	7 2 1 $\epsilon$		23 0 23 5
c Rlint	7 1 1     lint NS_lint_balls::TH_lint_balls		23 10 11
t Rdirective_phrase	4 1 6 Rlint		5 24 24
$\Rightarrow$ <i>Rlint</i>		State: 24 state type: $r$	
← rule	→ R# sr# Po ←	subrule element	→ Brn Gto Red LA
t Rdirective_phrase	4 1 7		5 0 24 5
$\Rightarrow$     <i>arbitration-code: <math>\epsilon</math></i>		State: 25 state type: $s$	
← rule	→ R# sr# Po ←	subrule element	→ Brn Gto Red LA
t Rparallel_directive_cweb_k	8 3 2  ?		15 26 26
t Rparallel_directive_cweb_k	8 2 2 cweb-comment		15 27 27
$\Rightarrow$  ?		State: 26 state type: $r$	
← rule	→ R# sr# Po ←	subrule element	→ Brn Gto Red LA
t Rparallel_directive_cweb_k	8 3 3		15 0 26 2
$\Rightarrow$ <i>cweb-comment</i>		State: 27 state type: $r$	
← rule	→ R# sr# Po ←	subrule element	→ Brn Gto Red LA
t Rparallel_directive_cweb_k	8 2 3		15 0 27 2
$\Rightarrow$  ?		State: 28 state type: $r$	
← rule	→ R# sr# Po ←	subrule element	→ Brn Gto Red LA
t Rdirective	5 3 2		17 0 28 6
$\Rightarrow$     <i>arbitration-code: <math>\epsilon</math></i>		State: 29 state type: $s$	
← rule	→ R# sr# Po ←	subrule element	→ Brn Gto Red LA
t Rdirective	5 2 2  ?		17 30 30
t Rdirective	5 1 2 # arbitrator-code		17 31 31
$\Rightarrow$  ?		State: 30 state type: $r$	
← rule	→ R# sr# Po ←	subrule element	→ Brn Gto Red LA
t Rdirective	5 2 3		17 0 30 6
$\Rightarrow$ # <i>arbitrator-code</i>		State: 31 state type: $r$	
← rule	→ R# sr# Po ←	subrule element	→ Brn Gto Red LA
t Rdirective	5 1 3		17 0 31 6

**34. Index.**

$\epsilon$  : 26, 27.  
 # arbitrator-code: 15.  
 # parallel-control-monitor: 9.  
 |||: 15, 21, 26, 27.  
 |?|: 10, 12, 15, 21, 27.  
 \_\_FILE\_\_: 5, 7, 11, 13, 19, 20, 30.  
 \_\_LINE\_\_: 5, 7, 11, 13, 19, 20, 30.  
 add\_cweb\_marker: 14.  
 add\_directive\_to\_mntr: 16.  
 add\_sdc\_to\_directive: 6, 7, 14.  
 AST: 5, 14, 29, 30.  
 CAbs\_lr1\_sym: 6, 7, 11, 13, 16, 17, 19, 20.  
 Cparallel\_monitor\_ph: 7, 9, 14, 16.  
 cw: 30.  
 cweb-comment: 27.  
 cweb\_t: 14, 28, 29, 30.  
 cwebk\_t: 30.  
 cwebt: 14.  
 Dir: 6, 7.  
 dir: 14.  
 directive\_: 14, 16, 17, 18, 19.  
 eid: 7.  
 enumerated\_id\_: 7.  
 Err\_improper\_directive: 7.  
 Err\_no\_close\_brace: 13.  
 Err\_no\_directive\_present: 20.  
 Err\_no\_open\_brace: 11.  
 Err\_not\_arbitration\_code\_kw: 19.  
 fsm: 9, 14, 16.  
 fsm\_ttbl\_: 9, 14, 16.  
 INT: 7.  
 join\_pts: 30.  
 lint: 26.  
 no syntax-code present: 21.  
 NS\_cweb\_or\_c\_k::TH\_cweb\_or\_c\_k: 27.  
 NS\_identifier::TH\_identifier: 15.  
 NS\_lint\_balls::TH\_lint\_balls: 26.  
 NS\_o2\_sdc::TH\_o2\_sdc: 21.  
 NS\_yacco2\_T\_enum: 7.  
 NS\_yacco2\_terminals: 7, 8.  
 NULL: 15, 21, 27.  
 parallel\_monitor\_ph: 2.  
 parallel\_monitor\_phrase\_: 4, 5, 6, 9, 16.  
 parser\_: 5, 7, 9, 11, 13, 14, 16, 19, 20, 25, 31.  
 phrase\_tree: 5.  
 p1\_: 11, 13, 20.  
 p2\_: 14, 18, 19, 23, 24, 25, 30, 31.  
 p4\_: 14.  
 p5\_: 14.  
 Rclose\_brace: 9.  
 Rclose\_brace: 12, 13.  
 Rdirective: 14.  
 Rdirective: 14, 15, 18, 19, 20.  
 Rdirective\_phrase: 9.  
 Rdirective\_phrase: 14.  
 result: 16.  
 Rlint: 26.  
 Rlint: 9, 14.  
 Ropen\_brace: 9.  
 Ropen\_brace: 10, 11.  
 Rparallel\_directive\_cweb\_k: 14.  
 Rparallel\_directive\_cweb\_k: 27, 30, 31.  
 Rparallel\_monitor\_phrase: 9.  
 RSVP: 9, 11, 13, 16, 19, 20, 25, 31.  
 RSVP\_FSM: 7.  
 Rsyntax\_code: 14.  
 Rsyntax\_code: 14, 21, 23, 24, 25.  
 rule\_info\_: 9, 11, 13, 14, 16, 19, 20, 25, 31.  
 Sdc: 6, 7.  
 sdc: 14.  
 set\_auto\_delete: 16, 24.  
 set\_content: 30.  
 set\_rc: 5, 7, 11, 13, 19, 20, 30.  
 set\_stop\_parse: 7, 11, 13, 16, 19, 20, 25, 31.  
 sf: 11, 13, 14, 18, 19, 20, 23, 24, 25, 30, 31.  
 start\_token\_: 5.  
 sym: 7, 11, 13, 19, 20.  
 syntax-code: 21.  
 syntax\_code: 7.  
 syntax\_code\_: 14, 22, 23, 24, 25.  
 T\_arbitrator\_code: 7.  
 T\_cweb\_comment: 30.  
 T\_cweb\_marker: 30.  
 T\_Enum: 7.  
 T\_parallel\_monitor\_phrase: 5, 6.  
 T\_syntax\_code: 6, 7, 22.  
 T\_T\_arbitrator\_code\_: 7.  
 true: 7, 11, 13, 16, 19, 20, 24, 25, 31.  
 yacco2: 6, 7.

- ⟨ Cparallel\_monitor\_ph constructor directive 4 ⟩
- ⟨ Cparallel\_monitor\_ph op directive 5 ⟩
- ⟨ Cparallel\_monitor\_ph user-declaration directive 6 ⟩
- ⟨ Cparallel\_monitor\_ph user-implementation directive 7 ⟩
- ⟨ Cparallel\_monitor\_ph user-prefix-declaration directive 8 ⟩
- ⟨ Rclose\_brace subrule 1 op directive 13 ⟩
- ⟨ Rdirective op directive 16 ⟩
- ⟨ Rdirective subrule 1 op directive 18 ⟩
- ⟨ Rdirective subrule 2 op directive 19 ⟩
- ⟨ Rdirective subrule 3 op directive 20 ⟩
- ⟨ Rdirective user-declaration directive 17 ⟩
- ⟨ Rdirective\_phrase subrule 1 op directive 14 ⟩
- ⟨ Ropen\_brace subrule 1 op directive 11 ⟩
- ⟨ Rparallel\_directive\_cweb\_k constructor directive 28 ⟩
- ⟨ Rparallel\_directive\_cweb\_k subrule 2 op directive 30 ⟩
- ⟨ Rparallel\_directive\_cweb\_k subrule 3 op directive 31 ⟩
- ⟨ Rparallel\_directive\_cweb\_k user-declaration directive 29 ⟩
- ⟨ Rparallel\_monitor\_phrase subrule 1 op directive 9 ⟩
- ⟨ Rsyntax\_code subrule 1 op directive 23 ⟩
- ⟨ Rsyntax\_code subrule 2 op directive 24 ⟩
- ⟨ Rsyntax\_code subrule 3 op directive 25 ⟩
- ⟨ Rsyntax\_code user-declaration directive 22 ⟩

# parallel\_monitor\_ph Grammar

Date: January 2, 2015 at 15:37

File: parallel\_monitor\_ph.lex Ns: NS\_parallel\_monitor\_ph

Version: 1.0

Debug: false

Grammar Comments:

Type: Thread

Parse a rule's arbitration code: into the valley of someone's dementia...

1 element(s) in Lookahead Expression below

eolr

	Section	Page
<b>Copyright</b> .....	1	1
<i>parallel_monitor_ph</i> <b>thread</b> .....	2	2
Fsm Cparallel_monitor_ph class .....	3	2
Cparallel_monitor_ph constructor directive .....	4	2
Cparallel_monitor_ph op directive .....	5	3
Cparallel_monitor_ph user-declaration directive .....	6	3
Cparallel_monitor_ph user-implementation directive .....	7	3
Cparallel_monitor_ph user-prefix-declaration directive .....	8	3
<i>Rparallel_monitor_phrase</i> rule .....	9	4
<i>Ropen_brace</i> rule .....	10	4
<i>Ropen_brace</i> 's subrule 1 .....	11	4
<i>Rclose_brace</i> rule .....	12	4
<i>Rclose_brace</i> 's subrule 1 .....	13	4
<i>Rdirective_phrase</i> rule .....	14	5
<i>Rdirective</i> rule .....	15	5
<i>Rdirective</i> op directive .....	16	5
<i>Rdirective</i> user-declaration directive .....	17	5
<i>Rdirective</i> 's subrule 1 .....	18	5
<i>Rdirective</i> 's subrule 2 .....	19	6
<i>Rdirective</i> 's subrule 3 .....	20	6
<i>Rsyntax_code</i> rule .....	21	6
<i>Rsyntax_code</i> user-declaration directive .....	22	6
<i>Rsyntax_code</i> 's subrule 1 .....	23	6
<i>Rsyntax_code</i> 's subrule 2 .....	24	6
<i>Rsyntax_code</i> 's subrule 3 .....	25	7
<i>Rlint</i> rule .....	26	7
<i>Rparallel_directive_cweb_k</i> rule .....	27	7
<i>Rparallel_directive_cweb_k</i> constructor directive .....	28	7
<i>Rparallel_directive_cweb_k</i> user-declaration directive .....	29	7
<i>Rparallel_directive_cweb_k</i> 's subrule 2 .....	30	7
<i>Rparallel_directive_cweb_k</i> 's subrule 3 .....	31	8
<b>First Set Language for <math>O_2^{linker}</math></b> .....	32	9
<b>Lr1 State Network</b> .....	33	10

parallel\_monitor\_ph Grammar

TABLE OF CONTENTS 1

**Index** ..... 34 13