

1. Copyright.

Copyright © Dave Bone 1998 - 2015

2. pass3 stand alone Grammar.

Lexical / syntactic dispatcher for the grammar being parsed.

Why the name *pass3*? The short of it: when i was developing a long time ago there were 3 passes. Through attrition it became one. I was too lazy to rename it to *pass1*. So that's the blah blah blog of it all.

It calls `PROCESS_KEYWORD_FOR_SYNTAX_CODE` when keywords are discovered. This is a top/down approach to parsing as the "called procedure" then calls (dispatches on) the keyword's content to its parsing procedure that of course contains a monolithic bottom-up grammar to digest the grammar construct. Neat stuff to see how the 2 approaches to parsing can blend together: a monolithic grammar using threaded grammars to parse its stuff.

Let's look at this from above:

Here we have a standalone grammar calling a procedure that runs a standalone grammar that calls more threads. Well i think this is neat... no ice sir and hold the applause. thk u.

Changed: added keyword recognition by symbol table lookup. eliminated competing parallelism between identifier and *yacco2.keyword* that now no longer exists.

Note:

The `use_cnt_` deals with the problem of recursion using this grammar and to write only once the eog tokens to close off the producer container.

Please see the *Rprefile.inc.dispatcher* rule as an example of chained procedure call. The chaining is on the "@" which is also the start character of grammar *NS.prefile.inc::PROC.TH.prefile.inc* which is called as a procedure. This is an example of an optimization: thread activation is toooooo slow compared to a procedure call. Now what about inlining ugh?

An improvement: grammar file has no contents
21-Apr-2014

3. Fsm Cpass3 class.

4. Cpass3 op directive.

```
<Cpass3 op directive 4> ≡
++Cpass3::nested_use_cnt_;
```

5. Cpass3 user-declaration directive.

```
<Cpass3 user-declaration directive 5> ≡
public: static int nested_use_cnt_;
```

6. Cpass3 user-implementation directive.

```
<Cpass3 user-implementation directive 6> ≡
int Cpass3::nested_use_cnt_(0);
```

7. Cpass3 user-prefix-declaration directive.

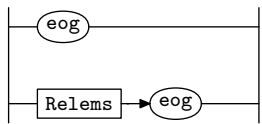
```

< Cpass3 user-prefix-declaration directive 7 > ≡
#include "ws.h"
#include "cweb_or_c_k.h"
#include "bad_char_set.h"
#include "eol.h"
#include "prefile_include.h"
#include "identifier.h"
#include "o2_externs.h"

```

8. Rpass3 rule.

Rpass3



9. Rpass3 op directive.

```

< Rpass3 op directive 9 > ≡
using namespace NS_yacco2_k_symbols;
if ( Cpass3 :: nested_use_cnt_ ≡ 1 ) {
    ADD_TOKEN_TO_PRODUCER_QUEUE(*yacco2 :: PTR_LR1_eog_);
    ADD_TOKEN_TO_PRODUCER_QUEUE(*yacco2 :: PTR_LR1_eog_);
}
-- Cpass3 :: nested_use_cnt_;

```

10. Rpass3's subrule 1.

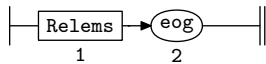


```

< Rpass3 subrule 1 op directive 10 > ≡
CAbs_lr1_sym * sym = new Err_empty_file;
sym->set_rc(*rule_info_.parser->current_token(), __FILE__, __LINE__);
ADD_TOKEN_TO_ERROR_QUEUE(*sym);
rule_info_.parser->set_abort_parse(true);

```

11. Rpass3's subrule 2.



```

< Rpass3 subrule 2 op directive 11 > ≡
if ( GRAMMAR_TREE ≡ 0 ) {
    CAbs_lr1_sym * sym = new Err_empty_file;
    sym->set_rc(*rule_info_.parser->current_token(), __FILE__, __LINE__);
    ADD_TOKEN_TO_ERROR_QUEUE(*sym);
    rule_info_.parser->set_abort_parse(true);
}

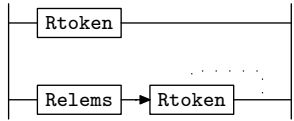
```

12. Relems rule.

13. Relems — left recursion diagram.

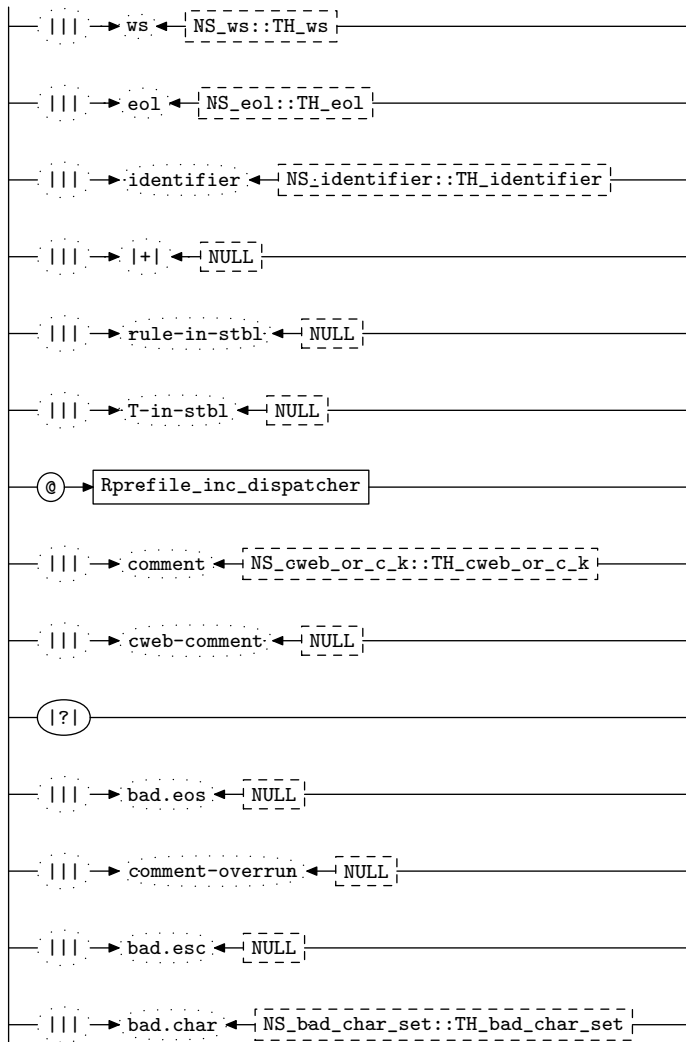
Note: the left recursion drawn as a Pascal railroad diagram.

Relems

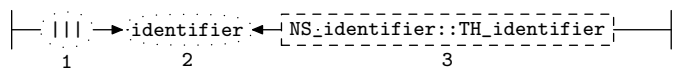


14. Rtoken rule.

Rtoken



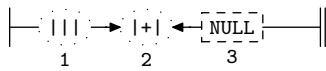
15. Rtoken's subrule 3.



16. Identifier slip thru.

As i use a top / down process to consume constructs, an Identifier some how is slipping thru due to either a premature ending of Rules top / down parse process or a before out-of-alignment token that should be within the Rules Vocabulary, or a misplaced misspelt T.

```
<Rtoken subrule 3 op directive 16> ≡
  sf-p2--set_auto_delete(true);
  CAbs_lr1_sym * sym = new Err_misplaced_or_misspelt_Rule_or_T;
  sym->set_rc(*sf-p2--, __FILE__, __LINE__);
  ADD_TOKEN_TO_ERROR_QUEUE(*sym);
  rule_info--.parser--set_abort_parse(true);
```

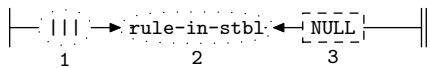
17. Rtoken's subrule 4.**18. Dispatch keyword to process its construct phrase.**

Again neat stuff with its co-operation of top/down and bottom-up parsing paradigms. Notice that i use the catch all “+” to show case it where as i could have referenced “keyword”. This is how the parser works:

- 1) check the state's table for specifically returned T from thread call
- 2) check for “catch all” returned presence from a thread call
- 3) try current token T to shift if thread call did not work or is not present in state
- 4) check for “catch all” presence in the state to shift

Note: there are 2 types of “catch all”: one for returned T from thread calls and the other for regular parsing.

```
<Rtoken subrule 4 op directive 18> ≡
  CAbs_lr1_sym * key = sf-p2--; /* extract specific keyword */
  yacco2 :: INT cont_pos = rule_info--.parser--current_token_pos--;
  CAbs_lr1_sym * cont_tok = rule_info--.parser--current_token();
  bool result = PROCESS_KEYWORD_FOR_SYNTAX_CODE(*rule_info--.parser--, key, &cont_tok, &cont_pos);
  if (result ≡ false) {
    rule_info--.parser--set_abort_parse(true);
    return;
  }
  ADD_TOKEN_TO_PRODUCER_QUEUE(*key); /* adv. to phrase's LA pos */
  rule_info--.parser--override_current_token(*cont_tok, cont_pos);
```

19. Rtoken's subrule 5.

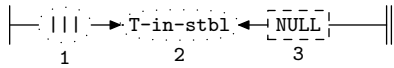
20. “rule-in-stbl” slip thru.

The top / down process lexes and consumes this token so this token never hits this grammar. Probable cause is a pre-mature stoppage of the “rules” construct. Why is “rule-in-stbl” returned? The token has been defined or referenced by one or more defining rules. The first occurrence in the rules construct enters it into the symbol table as defined or referenced. Depending on its next usage context, defined or referenced will update the its symbol table’s attributes.

```

⟨Rtoken subrule 5 op directive 20⟩ ≡
  sf-p2--set_auto_delete(true);
  CAbs_lr1_sym * sym = new Err_use_of_N_outside_Rules_construct;
  sym-set_rc(*sf-p2--, __FILE__, __LINE__);
  ADD_TOKEN_TO_ERROR_QUEUE(*sym);
  rule_info--.parser--set_abort_parse(true);

```

21. Rtoken’s subrule 6.**22. “T-in-stbl” slip thru.**

Could be an out-of-construct token that has been defined in the Terminal vocabulary. For example the token came before the Rules construct. Probably a typo mistake by the grammar writer.

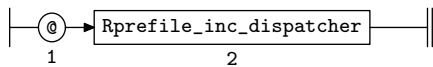
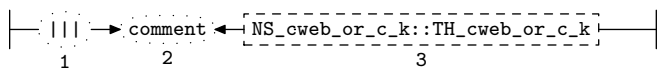
```

⟨Rtoken subrule 6 op directive 22⟩ ≡
  sf-p2--set_auto_delete(true);
  CAbs_lr1_sym * sym = new Err_use_of_T_outside_Rules_construct;
  sym-set_rc(*sf-p2--, __FILE__, __LINE__);
  ADD_TOKEN_TO_ERROR_QUEUE(*sym);
  rule_info--.parser--set_abort_parse(true);

```

23. Rtoken’s subrule 7.**24. Yac2o2’s pre-processor include directive.**

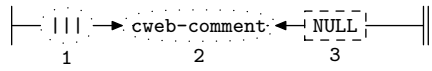
This demonstrates a nested environment where the grammar uses recursion by calling a function which contains the *pass3* grammar sequence. In this example, grammar *pass3* manually calls a thread via *start_manually_parallel_parsing* to get its file name to process. With the returned “file-inclusion” terminal, `PROCESS_INCLUDE_FILE` is called to parse the include file: a bom-de-bom-bom bump-and-grind sequence. The *use_cnt_* is a global variable that protects against the file include recursion of calling self until a stack overflow occurs.

**25. Rtoken’s subrule 8.**

```

⟨Rtoken subrule 8 op directive 25⟩ ≡
  T_comment * k = sf-p2--;
  k-set_auto_delete(true);

```

26. Rtoken's subrule 9.

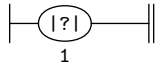
⟨ Rtoken subrule 9 op directive 26 ⟩ ≡

```

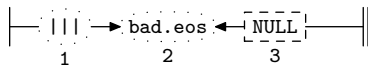
T_cweb_comment * k = sf-p2--;
AST * cwebk_t_ = new AST(*k);
AST * cweb_t_ = new AST();
T_cweb_marker * cw = new T_cweb_marker(cweb_t_);
cw->set_rc(*k, __FILE__, __LINE__);
AST::set_content(*cweb_t_, *cw);
AST::join_pts(*cweb_t_, *cwebk_t_);
BUILD_GRAMMAR_TREE(*cweb_t_);
CWEB_MARKER = 0;

```

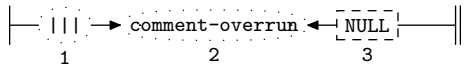
27. Rtoken's subrule 10.

28. Error subrules.

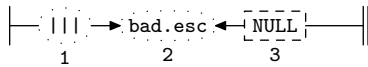
⟨Rtoken subrule 10 op directive 28⟩ ≡
CAbs_lr1_sym * *sym* = **new** *Err_not_kw_defining_grammar_construct*;
sym→*set_rc*(**sf*→*p1*__, __FILE__, __LINE__);
 ADD_TOKEN_TO_ERROR_QUEUE(**sym*);
rule_info→*parser*→*set_abort_parse*(*true*);

29. Rtoken's subrule 11.

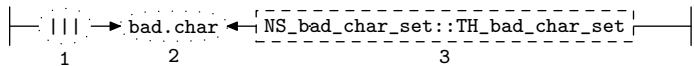
⟨Rtoken subrule 11 op directive 29⟩ ≡
 ADD_TOKEN_TO_ERROR_QUEUE(**sf*→*p2*__);
rule_info→*parser*→*set_abort_parse*(*true*);

30. Rtoken's subrule 12.

⟨Rtoken subrule 12 op directive 30⟩ ≡
 ADD_TOKEN_TO_ERROR_QUEUE(**sf*→*p2*__);
rule_info→*parser*→*set_abort_parse*(*true*);

31. Rtoken's subrule 13.

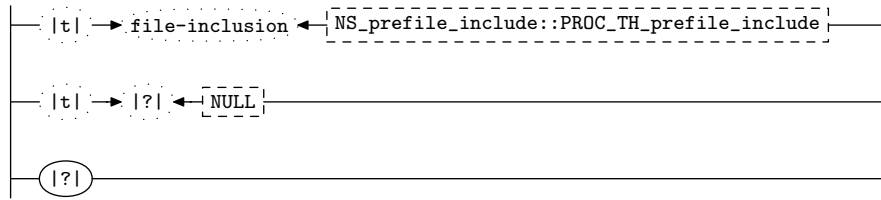
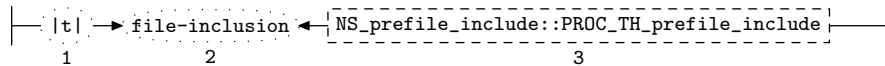
⟨Rtoken subrule 13 op directive 31⟩ ≡
 ADD_TOKEN_TO_ERROR_QUEUE(**sf*→*p2*__);
rule_info→*parser*→*set_abort_parse*(*true*);

32. Rtoken's subrule 14.

⟨Rtoken subrule 14 op directive 32⟩ ≡
Err_bad_char * *k* = *sf*→*p2*__;
 ADD_TOKEN_TO_ERROR_QUEUE(**k*);
rule_info→*parser*→*set_abort_parse*(*true*);

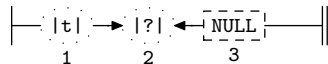
33. Rprefile_inc_dispatcher rule.

Rprefile_inc_dispatcher

**34. Rprefile_inc_dispatcher's subrule 1.**

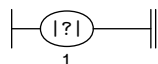
⟨Rprefile_inc_dispatcher subrule 1 op directive 34⟩ ≡

```
CAbs_lr1_sym * err = sf-p2--error_sym();
if (err ≠ 0) {
    rule_info--parser--set_abort_parse(true);
    ADD_TOKEN_TO_ERROR_QUEUE(*sf-p2--);
    ADD_TOKEN_TO_ERROR_QUEUE(*sf-p2--error_sym());
    return;
}
bool result = PROCESS_INCLUDE_FILE(*rule_info--parser--, *sf-p2--,
    *rule_info--parser--token_producer--);
if (result ≡ false) { /* exceeded nested file limit */
    rule_info--parser--set_abort_parse(true);
    return;
}
ADD_TOKEN_TO_RECYCLE_BIN(*sf-p2--); /* file name inside */
return;
```

35. Rprefile_inc_dispatcher's subrule 2.

⟨Rprefile_inc_dispatcher subrule 2 op directive 35⟩ ≡

```
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);
```

36. Rprefile_inc_dispatcher's subrule 3.

⟨Rprefile_inc_dispatcher subrule 3 op directive 36⟩ ≡

```
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);
```

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

```
/*
  File: pass3.fsc
  Date and Time: Fri Jan  2 15:33:48 2015
*/
transitive      y
grammar-name    "pass3"
name-space     "NS_pass3"
thread-name    "Cpass3"
monolithic     y
file-name      "pass3.fsc"
no-of-T        569
list-of-native-first-set-terminals 3
  LR1_questionable_shift_operator
  LR1_eog
  raw_at_sign
end-list-of-native-first-set-terminals
list-of-transitive-threads 5
  NS_ws::TH_ws
  NS_identifier::TH_identifier
  NS_eol::TH_eol
  NS_cweb_or_c_k::TH_cweb_or_c_k
  NS_bad_char_set::TH_bad_char_set
end-list-of-transitive-threads
list-of-used-threads 5
  NS_bad_char_set::TH_bad_char_set
  NS_cweb_or_c_k::TH_cweb_or_c_k
  NS_eol::TH_eol
  NS_identifier::TH_identifier
  NS_ws::TH_ws
end-list-of-used-threads
fsm-comments
"\02's lexer constructing tokens for syntax parser stage."
```

38. Lr1 State Network.

		State: 1 state type: ^s										
←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
c	Rtoken		3	10	1	←	?		1	2	2	
c	Rpass3		1	1	1	←	eog		1	3	3	
c	Rtoken		3	3	1	←	identifier NS_identifier::TH_identifier		1	4	9	
c	Rtoken		3	12	1	←	comment-overrun NULL		1	4	15	
c	Rtoken		3	2	1	←	eol NS_eol::TH_eol		1	4	6	
c	Rtoken		3	4	1	←	+ NULL		1	4	5	
c	Rtoken		3	5	1	←	rule-in-stbl NULL		1	4	11	
c	Rtoken		3	6	1	←	T-in-stbl NULL		1	4	10	
c	Rtoken		3	8	1	←	comment NS_cweb_or_c.k::TH_cweb_or_c.k		1	4	7	
c	Rtoken		3	9	1	←	cweb-comment NULL		1	4	12	
c	Rtoken		3	11	1	←	bad eos NULL		1	4	13	
c	Rtoken		3	13	1	←	bad esc NULL		1	4	14	
c	Rtoken		3	14	1	←	bad char NS_bad_char_set::TH_bad_char_set		1	4	16	
c	Rtoken		3	1	1	←	ws NS_ws::TH_ws		1	4	8	
c	Rtoken		3	7	1	←	@		1	17	22	
c	Relems		2	2	1	←	Relems <u>Rtoken</u>		1	23	25	
c	Rpass3		1	2	1	←	Relems <u>eog</u>		1	23	24	
c	Relems		2	1	1	←	Rtoken		1	26	26	

		State: 2 state type: ^r										
←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rtoken		3	10	2	←			1	0	2	1

		State: 3 state type: ^r										
←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rpass3		1	1	2	←			1	0	3	2

		State: 4 state type: ^s										
←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rtoken		3	4	2	←	+		1	5	5	
t	Rtoken		3	2	2	←	eol		1	6	6	
t	Rtoken		3	8	2	←	comment		1	7	7	
t	Rtoken		3	1	2	←	ws		1	8	8	
t	Rtoken		3	3	2	←	identifier		1	9	9	
t	Rtoken		3	6	2	←	T-in-stbl		1	10	10	
t	Rtoken		3	5	2	←	rule-in-stbl		1	11	11	
t	Rtoken		3	9	2	←	cweb-comment		1	12	12	
t	Rtoken		3	11	2	←	bad eos		1	13	13	
t	Rtoken		3	13	2	←	bad esc		1	14	14	
t	Rtoken		3	12	2	←	comment-overrun		1	15	15	
t	Rtoken		3	14	2	←	bad char		1	16	16	

		State: 5 state type: ^r										
←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rtoken		3	4	3	←			1	0	5	1

		State: 6 state type: ^r										
←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
						←						

t Rtoken		3	2	3				1	0	6	1
\Rightarrow <i>comment</i>						State: 7 state type: <i>r</i>					
← rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t Rtoken		3	8	3				1	0	7	1
\Rightarrow <i>ws</i>						State: 8 state type: <i>r</i>					
← rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t Rtoken		3	1	3				1	0	8	1
\Rightarrow <i>identifier</i>						State: 9 state type: <i>r</i>					
← rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t Rtoken		3	3	3				1	0	9	1
\Rightarrow <i>T-in-stbl</i>						State: 10 state type: <i>r</i>					
← rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t Rtoken		3	6	3				1	0	10	1
\Rightarrow <i>rule-in-stbl</i>						State: 11 state type: <i>r</i>					
← rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t Rtoken		3	5	3				1	0	11	1
\Rightarrow <i>cweb-comment</i>						State: 12 state type: <i>r</i>					
← rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t Rtoken		3	9	3				1	0	12	1
\Rightarrow <i>badeos</i>						State: 13 state type: <i>r</i>					
← rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t Rtoken		3	11	3				1	0	13	1
\Rightarrow <i>badesc</i>						State: 14 state type: <i>r</i>					
← rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t Rtoken		3	13	3				1	0	14	1
\Rightarrow <i>comment-overrun</i>						State: 15 state type: <i>r</i>					
← rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t Rtoken		3	12	3				1	0	15	1
\Rightarrow <i>badchar</i>						State: 16 state type: <i>r</i>					
← rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t Rtoken		3	14	3				1	0	16	1
\Rightarrow <i>@</i>						State: 17 state type: <i>s</i>					
← rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
c Rprofile_inc_dispatcher		4	3	1	?			17	18	18	
c Rprofile_inc_dispatcher		4	2	1	t ? NULL			17	19	20	
c Rprofile_inc_dispatcher		4	1	1	t file-inclusion NS_profile_include::PROC_TH_profile_include			17	19	21	
t Rtoken		3	7	2	Rprofile_inc_dispatcher			1	22	22	
\Rightarrow <i> ? </i>						State: 18 state type: <i>r</i>					
← rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t Rprofile_inc_dispatcher		4	3	2				17	0	18	1

\Rightarrow t					State: 19 state type: ^s			
← rule	→ R#	sr#	Po	←	subrule element	→	Brn	Gto Red LA
t Rprefile_inc_dispatcher	4	2	2	?			17	20 20
t Rprefile_inc_dispatcher	4	1	2	file-inclusion			17	21 21
\Rightarrow ?					State: 20 state type: ^r			
← rule	→ R#	sr#	Po	←	subrule element	→	Brn	Gto Red LA
t Rprefile_inc_dispatcher	4	2	3				17	0 20 1
\Rightarrow file-inclusion					State: 21 state type: ^r			
← rule	→ R#	sr#	Po	←	subrule element	→	Brn	Gto Red LA
t Rprefile_inc_dispatcher	4	1	3				17	0 21 1
\Rightarrow Rprefile_inc_dispatcher					State: 22 state type: ^r			
← rule	→ R#	sr#	Po	←	subrule element	→	Brn	Gto Red LA
t Rtoken	3	7	3				1	0 22 1
\Rightarrow Relems					State: 23 state type: ^s			
← rule	→ R#	sr#	Po	←	subrule element	→	Brn	Gto Red LA
c Rtoken	3	10	1	?			23	2 2
t Rpass3	1	2	2	eog			1	24 24
c Rtoken	3	3	1	identifier NS_identifier::TH_identifier			23	4 9
c Rtoken	3	12	1	comment-overflow NULL			23	4 15
c Rtoken	3	2	1	eol NS_eol::TH_eol			23	4 6
c Rtoken	3	4	1	+ NULL			23	4 5
c Rtoken	3	5	1	rule-in-stbl NULL			23	4 11
c Rtoken	3	6	1	T-in-stbl NULL			23	4 10
c Rtoken	3	8	1	comment NS_cweb_or_c_k::TH_cweb_or_c_k			23	4 7
c Rtoken	3	9	1	cweb-comment NULL			23	4 12
c Rtoken	3	11	1	bad eos NULL			23	4 13
c Rtoken	3	13	1	bad esc NULL			23	4 14
c Rtoken	3	14	1	bad char NS_bad_char_set::TH_bad_char_set			23	4 16
c Rtoken	3	1	1	ws NS_ws::TH_ws			23	4 8
c Rtoken	3	7	1	@			23	17 22
t Relems	2	2	2	Rtoken			1	25 25
\Rightarrow eog					State: 24 state type: ^r			
← rule	→ R#	sr#	Po	←	subrule element	→	Brn	Gto Red LA
t Rpass3	1	2	3				1	0 24 2
\Rightarrow Rtoken					State: 25 state type: ^r			
← rule	→ R#	sr#	Po	←	subrule element	→	Brn	Gto Red LA
t Relems	2	2	3				1	0 25 1
\Rightarrow Rtoken					State: 26 state type: ^r			
← rule	→ R#	sr#	Po	←	subrule element	→	Brn	Gto Red LA
t Relems	2	1	2				1	0 26 1

39. Index.

- |+|: 14.
- |||: 14.
- |?|: 14, 33.
- |t|: 33.
- __FILE__: 10, 11, 16, 20, 22, 26, 28, 35, 36.
- __LINE__: 10, 11, 16, 20, 22, 26, 28, 35, 36.
- ADD_TOKEN_TO_ERROR_QUEUE: 10, 11, 16, 20, 22, 28, 29, 30, 31, 32, 34.
- ADD_TOKEN_TO_PRODUCER_QUEUE: 9, 18.
- ADD_TOKEN_TO_RECYCLE_BIN: 34.
- AST: 26.
- bad char: 14.
- bad eos: 14.
- bad esc: 14.
- BUILD_GRAMMAR_TREE: 26.
- CAbs_lr1_sym*: 10, 11, 16, 18, 20, 22, 28, 34, 35, 36.
- comment: 14.
- comment-overrun: 14.
- cont_pos*: 18.
- cont_tok*: 18.
- Cpass3*: 4, 6, 9.
- current_token*: 10, 11, 18, 36.
- current_token_pos_*: 18.
- cw*: 26.
- cweb-comment: 14.
- CWEB_MARKER: 26.
- cweb_t*: 26.
- cwebk_t*: 26.
- eog: 8.
- eol: 14.
- err: 34.
- Err_bad_char*: 32.
- Err_bad_directive*: 35.
- Err_empty_file*: 10, 11.
- Err_misplaced_or_misspelt_Rule_or_T*: 16.
- Err_no_directive_present*: 36.
- Err_not_kw_defining_grammar_construct*: 28.
- Err_use_of_N_outside_Rules_construct*: 20.
- Err_use_of_T_outside_Rules_construct*: 22.
- error_sym*: 34.
- false*: 18, 34.
- file-inclusion: 33.
- GRAMMAR_TREE: 11.
- identifier: 14.
- INT: 18.
- join_pts*: 26.
- key*: 18.
- nested_use_cnt_*: 4, 5, 6, 9.
- NS_bad_char_set::TH_bad_char_set*: 14.
- NS_cweb_or_c_k::TH_cweb_or_c_k*: 14.
- NS_eol::TH_eol*: 14.
- NS_identifier::TH_identifier*: 14.
- NS_prefile_include::PROC_TH_prefile_include*: 33.
- NS_ws::TH_ws*: 14.
- NS_prefile_inc*: 2.
- NS_yacco2_k_symbols*: 9.
- NULL: 14, 33.
- override_current_token*: 18.
- parser_*: 10, 11, 16, 18, 20, 22, 28, 29, 30, 31, 32, 34, 35, 36.
- pass1*: 2.
- pass3*: 2, 24.
- PROC_TH_prefile_inc*: 2.
- PROCESS_INCLUDE_FILE: 24, 34.
- PROCESS_KEYWORD_FOR_SYNTAX_CODE: 2, 18.
- PTR_LR1_eog_*: 9.
- p1_*: 28.
- p2_*: 16, 18, 20, 22, 25, 26, 29, 30, 31, 32, 34, 35.
- Relems: 8, 13.
- Relems*: 12.
- result*: 18, 34.
- Rpass3*: 8, 10, 11.
- Rprefile_inc_dispatcher*: 14.
- Rprefile_inc_dispatcher*: 2, 33, 34, 35, 36.
- RSVP: 35, 36.
- Rtoken: 13.
- Rtoken*: 14, 15, 17, 19, 21, 23, 25, 26, 27, 29, 30, 31, 32.
- rule-in-stbl: 14.
- rule_info_*: 10, 11, 16, 18, 20, 22, 28, 29, 30, 31, 32, 34, 35, 36.
- set_abort_parse*: 10, 11, 16, 18, 20, 22, 28, 29, 30, 31, 32, 34.
- set_auto_delete*: 16, 20, 22, 25, 35.
- set_content*: 26.
- set_rc*: 10, 11, 16, 20, 22, 26, 28, 35, 36.
- set_stop_parse*: 35, 36.
- sf*: 16, 18, 20, 22, 25, 26, 28, 29, 30, 31, 32, 34, 35.
- start_manually_parallel_parsing*: 24.
- sym*: 10, 11, 16, 20, 22, 28, 35, 36.
- T-in-stbl: 14.
- T_comment*: 25.
- T_cweb_comment*: 26.
- T_cweb_marker*: 26.
- token_producer_*: 34.
- true*: 10, 11, 16, 18, 20, 22, 25, 28, 29, 30, 31, 32, 34, 35, 36.
- use_cnt_*: 24.
- ws: 14.
- yacco2*: 9, 18.
- yacco2_keyword*: 2.

⟨ Cpass3 op directive 4 ⟩
⟨ Cpass3 user-declaration directive 5 ⟩
⟨ Cpass3 user-implementation directive 6 ⟩
⟨ Cpass3 user-prefix-declaration directive 7 ⟩
⟨ Rpass3 op directive 9 ⟩
⟨ Rpass3 subrule 1 op directive 10 ⟩
⟨ Rpass3 subrule 2 op directive 11 ⟩
⟨ Rprefile_inc_dispatcher subrule 1 op directive 34 ⟩
⟨ Rprefile_inc_dispatcher subrule 2 op directive 35 ⟩
⟨ Rprefile_inc_dispatcher subrule 3 op directive 36 ⟩
⟨ Rtoken subrule 10 op directive 28 ⟩
⟨ Rtoken subrule 11 op directive 29 ⟩
⟨ Rtoken subrule 12 op directive 30 ⟩
⟨ Rtoken subrule 13 op directive 31 ⟩
⟨ Rtoken subrule 14 op directive 32 ⟩
⟨ Rtoken subrule 3 op directive 16 ⟩
⟨ Rtoken subrule 4 op directive 18 ⟩
⟨ Rtoken subrule 5 op directive 20 ⟩
⟨ Rtoken subrule 6 op directive 22 ⟩
⟨ Rtoken subrule 8 op directive 25 ⟩
⟨ Rtoken subrule 9 op directive 26 ⟩

pass3 Grammar

Date: January 2, 2015 at 15:38

File: pass3.lex

Ns: NS_pass3

Version: 1.0

Debug: false

Grammar Comments:

Type: Monolithic

*O*₂'s lexer constructing tokens for syntax parser stage.

pass3 stand alone Grammar	2	2
Fsm Cpass3 class	3	2
Cpass3 op directive	4	2
Cpass3 user-declaration directive	5	2
Cpass3 user-implementation directive	6	2
Cpass3 user-prefix-declaration directive	7	3
Rpass3 rule	8	3
Rpass3 op directive	9	3
Rpass3's subrule 1	10	3
Rpass3's subrule 2	11	3
Relems rule	12	3
Relems — left recursion diagram	13	4
Rtoken rule	14	4
Rtoken's subrule 3	15	4
Identifier slip thru	16	5
Rtoken's subrule 4	17	5
Dispatch keyword to process its construct phrase	18	5
Rtoken's subrule 5	19	5
“rule-in-stbl” slip thru	20	6
Rtoken's subrule 6	21	6
“T-in-stbl” slip thru	22	6
Rtoken's subrule 7	23	6
Yac ₂ o ₂ 's pre-processor include directive	24	6
Rtoken's subrule 8	25	6
Rtoken's subrule 9	26	7
Rtoken's subrule 10	27	7
Error subrules	28	8
Rtoken's subrule 11	29	8
Rtoken's subrule 12	30	8
Rtoken's subrule 13	31	8
Rtoken's subrule 14	32	8
Rprefile_inc_dispatcher rule	33	9
Rprefile_inc_dispatcher's subrule 1	34	9
Rprefile_inc_dispatcher's subrule 2	35	9
Rprefile_inc_dispatcher's subrule 3	36	9
First Set Language for O_2^{linker}	37	10
Lr1 State Network	38	11
Index	39	14