

1. Copyright.

Copyright © Dave Bone 1998 - 2015

2. *cweb_or_c_k* combined “c++ ” or my “cweb” comment Thread.

Why? To eliminate the non-determinism between these 2 threads due to their common prefix: “/”. Why again? Speed and stop the confusion across the threads address space when called by procedure instead of by thread. The thread gets called singly and hence the speed versus parallel threads competing where only one will win. Some interesting debugging issues were found when using Sun’s superb “mdb” and “dtrace” tools.

3. Cpp comments.

Handles both flavours of c++ comments.

- 1) // single line type comment
- 2) /* ... */ type comment

To speed things up, i’m using a finite-state approach to lower the pushdown deterministic rule push / pop cycles.

4. Cweb type comments.

Recognize *cweb* comments buried within a grammar. These comments start with “/@” followed by the comments and closed off by “@” and a forward slash — can’t declare it as this will prematurely end this comment. The enclosing comment prefix / suffix are not appended to the data as only its contents that are outputted in the various documents.

Note: Use of the |.| symbol to lower the Lookahead set members.

5. Fsm Cweb_or_c_k class.**6. Cweb_or_c_k constructor directive.**

```
< Cweb_or_c_k constructor directive 6 > ≡
  ddd_idx_ = 0;
  ddd_[ddd_idx_] = 0;
```

7. Cweb_or_c_k op directive.

```
< Cweb_or_c_k op directive 7 > ≡
  ddd_idx_ = 0;
  ddd_[ddd_idx_] = 0;
```

8. Cweb_or_c_k user-declaration directive.

```
< Cweb_or_c_k user-declaration directive 8 > ≡
public: char ddd_[1024 * 32];
int ddd_idx_;
void copy_str_into_buffer(std :: string * Str);
void copy_kstr_into_buffer(const char *Str);
```

9. Cweb_or_c_k user-implementation directive.

```
< Cweb_or_c_k user-implementation directive 9 > ≡
void Cweb_or_c_k :: copy_str_into_buffer(std :: string * Str)
{
  const char *y = Str->c_str();
  int x(0);
  for ( ; y[x] ≠ 0; ++x, ++ddd_idx_) ddd_[ddd_idx_] = y[x];
  ddd_[ddd_idx_] = 0;
}
```

10. *copy_kstr_into_buffer*.

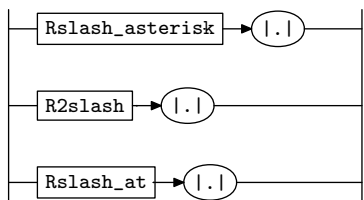
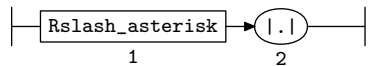
```

⟨ More code 10 ⟩ ≡
void Ccweb_or_c.k::copy_kstr_into_buffer(const char *Str)
{
    const char *y = Str;
    int x(0);
    for (; y[x] ≠ 0; ++x, ++ddd_idx_) ddd_[ddd_idx_] = y[x];
    ddd_[ddd_idx_] = 0;
}

```

11. *Rcweb_or_c.k* rule.

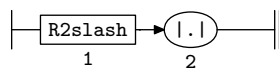
Rcweb_or_c.k

12. *Rcweb_or_c.k*'s subrule 1.

```

⟨ Rcweb_or_c.k subrule 1 op directive 12 ⟩ ≡
Ccweb_or_c.k * fsm = ( Ccweb_or_c.k * ) rule_info...parser...fsm_tbl_;
T_comment * com = new T_comment((const char *) &fsm_ddd_);
com->set_rc(*rule_info...parser...start_token_, __FILE__, __LINE__);
RSVP(com);

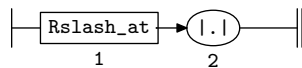
```

13. *Rcweb_or_c.k*'s subrule 2.

```

⟨ Rcweb_or_c.k subrule 2 op directive 13 ⟩ ≡
Ccweb_or_c.k * fsm = ( Ccweb_or_c.k * ) rule_info...parser...fsm_tbl_;
T_comment * com = new T_comment((const char *) &fsm_ddd_);
com->set_rc(*rule_info...parser...start_token_, __FILE__, __LINE__);
RSVP(com);

```

14. *Rcweb_or_c.k*'s subrule 3.

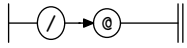
```

⟨ Rcweb_or_c.k subrule 3 op directive 14 ⟩ ≡
Ccweb_or_c.k * fsm = ( Ccweb_or_c.k * ) rule_info...parser...fsm_tbl_;
T_cweb_comment * com = new T_cweb_comment((const char *) &fsm_ddd_);
com->set_rc(*rule_info...parser...start_token_, __FILE__, __LINE__); /* file marker */
RSVP(com);

```

15. *Rslash_at* rule.

Rslash_at



⟨Rslash_at subrule 1 op directive 15⟩ ≡

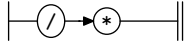
```

Ccweb_or_c.k * fsm = ( Ccweb_or_c.k * ) rule_info__parser__fsm_tbl__ ;
loop:
  switch (rule_info__parser__current_token()→enumerated_id__) {
  case T_Enum::T_raw_lf_: goto other;
  case T_Enum::T_raw_cr_: goto cr;
  case T_Enum::T_LR1_eog_: goto overrun;
  case T_Enum::T_raw_at_sign_: goto atsign;
  default: goto other;
  }
cr:
  { /* cr lf? */
  fsm→copy_kstr_into_buffer(rule_info__parser__current_token()→id__);
  rule_info__parser__get_next_token();
  if (rule_info__parser__current_token()→enumerated_id__ ≠ T_Enum::T_raw_lf_) goto loop;
  /* not cr lf */
  fsm→copy_kstr_into_buffer(rule_info__parser__current_token()→id__);
  rule_info__parser__get_next_token();
  goto loop;
  }
;
atsign:
  { /* end of k? */
  rule_info__parser__get_next_token();
  if (rule_info__parser__current_token()→enumerated_id__ ≠ T_Enum::T_raw_slash_) {
  fsm→copy_kstr_into_buffer("@");
  goto loop; /* false eok */
  }
  rule_info__parser__get_next_token();
  return;
  }
overrun:
  {
  CAbs_lr1_sym * sym = new Err_comment_overrun;
  sym→set_rc(*rule_info__parser__start_token__, __FILE__, __LINE__);
  RSVP(sym);
  rule_info__parser__set_stop_parse(true);
  return;
  }
other:
  {
  fsm→copy_kstr_into_buffer(rule_info__parser__current_token()→id__);
  rule_info__parser__get_next_token();
  goto loop;
  }

```

16. *Rslash_asterisk* rule.

Rslash_asterisk



(Rslash_asterisk subrule 1 op directive 16) ≡

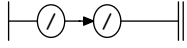
```

Ccweb_or_c.k * fsm = ( Ccweb_or_c.k * ) rule_info__parser__fsm_tbl__ ;
fsm->copy_kstr_into_buffer("/"); /* due to lex scanner */
fsm->copy_kstr_into_buffer("*");
loop:
switch (rule_info__parser__current_token()-enumerated_id__) {
case T_Enum::T_raw_lf_: goto other;
case T_Enum::T_raw_cr_: goto cr;
case T_Enum::T_LR1_eog_: goto overrun;
case T_Enum::T_raw_asteric_: goto aster;
default: goto other;
}
cr:
{ /* cr lf? */
fsm->copy_kstr_into_buffer(rule_info__parser__current_token()-id__);
rule_info__parser__get_next_token();
if (rule_info__parser__current_token()-enumerated_id__ ≠ T_Enum::T_raw_lf_) goto loop;
/* not cr lf */
fsm->copy_kstr_into_buffer(rule_info__parser__current_token()-id__);
rule_info__parser__get_next_token();
goto loop;
}
;
aster:
{ /* end of k? */
fsm->copy_kstr_into_buffer(rule_info__parser__current_token()-id__);
rule_info__parser__get_next_token();
if (rule_info__parser__current_token()-enumerated_id__ ≠ T_Enum::T_raw_slash_) goto loop;
/* false eok */
fsm->copy_kstr_into_buffer(rule_info__parser__current_token()-id__);
rule_info__parser__get_next_token();
return;
}
overrun:
{
CAbs_lr1_sym * sym = new Err_comment_overrun;
sym->set_rc(*rule_info__parser__start_token__, __FILE__, __LINE__);
RSVP(sym);
rule_info__parser__set_stop_parse(true);
return;
}
other:
{
fsm->copy_kstr_into_buffer(rule_info__parser__current_token()-id__);
rule_info__parser__get_next_token();
goto loop;
}

```

17. *R2slash* rule.

R2slash



⟨R2slash subrule 1 op directive 17⟩ ≡

```

Ccweb_or_c.k * fsm = ( Ccweb_or_c.k * ) rule_info__parser__fsm_tbl__ ;
fsm->copy_kstr_into_buffer("/");
fsm->copy_kstr_into_buffer("/");
for ( ; ; ) {
    switch (rule_info__parser__current_token()-enumerated_id__) {
        case T_Enum :: T_raw_lf_:
            {
                return;
            }
        case T_Enum :: T_raw_cr_:
            {
                return;
            }
        case T_Enum :: T_LR1_eog_: return;
    }
    fsm->copy_kstr_into_buffer(rule_info__parser__current_token()-id__);
    rule_info__parser__get_next_token();
}

```

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

```
/*
  File: cweb_or_c.k.fsc
  Date and Time: Fri Jan  2 15:33:31 2015
*/
transitive    n
grammar-name  "cweb_or_c.k"
name-space    "NS_cweb_or_c.k"
thread-name   "TH_cweb_or_c.k"
monolithic    n
file-name     "cweb_or_c.k.fsc"
no-of-T       569
list-of-native-first-set-terminals 1
  raw_slash
end-list-of-native-first-set-terminals
list-of-transitive-threads 0
end-list-of-transitive-threads
list-of-used-threads 0
end-list-of-used-threads
fsm-comments
"C++ or cweb type comments lexer."
```

19. Lr1 State Network.

⇒					State: 1 state type: ^s		
←	rule	→	R# sr# Po	←	subrule element	→	Brn Gto Red LA
c	Rslash_at		2 1 1 /				1 2 5
c	R2slash		4 1 1 /				1 2 4
c	Rslash_asterisk		3 1 1 /				1 2 3
c	Rcweb_or_c_k		1 3 1	Rslash_at .			1 6 7
c	Rcweb_or_c_k		1 1 1	Rslash_asterisk .			1 8 9
c	Rcweb_or_c_k		1 2 1	R2slash .			1 10 11
⇒/					State: 2 state type: ^s		
←	rule	→	R# sr# Po	←	subrule element	→	Brn Gto Red LA
t	Rslash_asterisk		3 1 2 *				1 3 3
t	R2slash		4 1 2 /				1 4 4
t	Rslash_at		2 1 2 @				1 5 5
⇒*					State: 3 state type: ^r		
←	rule	→	R# sr# Po	←	subrule element	→	Brn Gto Red LA
t	Rslash_asterisk		3 1 3				1 0 3 1
⇒/					State: 4 state type: ^r		
←	rule	→	R# sr# Po	←	subrule element	→	Brn Gto Red LA
t	R2slash		4 1 3				1 0 4 1
⇒@					State: 5 state type: ^r		
←	rule	→	R# sr# Po	←	subrule element	→	Brn Gto Red LA
t	Rslash_at		2 1 3				1 0 5 1
⇒ <i>Rslash_at</i>					State: 6 state type: ^s		
←	rule	→	R# sr# Po	←	subrule element	→	Brn Gto Red LA
t	Rcweb_or_c_k		1 3 2 .				1 7 7
⇒ .					State: 7 state type: ^r		
←	rule	→	R# sr# Po	←	subrule element	→	Brn Gto Red LA
t	Rcweb_or_c_k		1 3 3				1 0 7 2
⇒ <i>Rslash_asterisk</i>					State: 8 state type: ^s		
←	rule	→	R# sr# Po	←	subrule element	→	Brn Gto Red LA
t	Rcweb_or_c_k		1 1 2 .				1 9 9
⇒ .					State: 9 state type: ^r		
←	rule	→	R# sr# Po	←	subrule element	→	Brn Gto Red LA
t	Rcweb_or_c_k		1 1 3				1 0 9 2
⇒ <i>R2slash</i>					State: 10 state type: ^s		
←	rule	→	R# sr# Po	←	subrule element	→	Brn Gto Red LA
t	Rcweb_or_c_k		1 2 2 .				1 11 11
⇒ .					State: 11 state type: ^r		
←	rule	→	R# sr# Po	←	subrule element	→	Brn Gto Red LA
t	Rcweb_or_c_k		1 2 3				1 0 11 2

20. Index.

|. |: 11.
__FILE__: 12, 13, 14, 15, 16.
__LINE__: 12, 13, 14, 15, 16.
aster: 16.
atsign: 15.
c_str: 9.
CAbs_lr1_sym: 15, 16.
Cweb_or_c.k: 9, 10, 12, 13, 14, 15, 16, 17.
com: 12, 13, 14.
copy_kstr_into_buffer: 8, 10, 15, 16, 17.
copy_str_into_buffer: 8, 9.
cr: 15, 16.
current_token: 15, 16, 17.
cweb: 4.
cweb_or_c.k: 2.
ddd: 6, 7, 8, 9, 10, 12, 13, 14.
ddd_idx: 6, 7, 8, 9, 10.
enumerated_id: 15, 16, 17.
Err_comment_ouerrun: 15, 16.
fsm: 12, 13, 14, 15, 16, 17.
fsm_tbl: 12, 13, 14, 15, 16, 17.
get_next_token: 15, 16, 17.
id: 15, 16, 17.
loop: 15, 16.
other: 15, 16.
ouerrun: 15, 16.
parser: 12, 13, 14, 15, 16, 17.
Rcweb_or_c.k: 11, 12, 13, 14.
Rslash_asterisk: 11.
Rslash_at: 11.
Rslash_asterisk: 16.
Rslash_at: 15.
RSVP: 12, 13, 14, 15, 16.
rule_info: 12, 13, 14, 15, 16, 17.
R2slash: 17.
R2slash: 11.
set_rc: 12, 13, 14, 15, 16.
set_stop_parse: 15, 16.
start_token: 12, 13, 14, 15, 16.
std: 8, 9.
Str: 8, 9, 10.
string: 8, 9.
sym: 15, 16.
T_comment: 12, 13.
T_cweb_comment: 14.
T_Enum: 15, 16, 17.
T_LR1_eog: 15, 16, 17.
T_raw_asteric: 16.
T_raw_at_sign: 15.
T_raw_cr: 15, 16, 17.
T_raw_lf: 15, 16, 17.
T_raw_slash: 15, 16.
true: 15, 16.
x: 9, 10.
y: 9, 10.

< Cweb_or_c_k constructor directive [6](#) >
< Cweb_or_c_k op directive [7](#) >
< Cweb_or_c_k user-declaration directive [8](#) >
< Cweb_or_c_k user-implementation directive [9](#) >
< More code [10](#) >
< R2slash subrule 1 op directive [17](#) >
< Rweb_or_c_k subrule 1 op directive [12](#) >
< Rweb_or_c_k subrule 2 op directive [13](#) >
< Rweb_or_c_k subrule 3 op directive [14](#) >
< Rslash_asterisk subrule 1 op directive [16](#) >
< Rslash_at subrule 1 op directive [15](#) >

cweb_or_c_k Grammar

Date: January 2, 2015 at 15:34

File: cweb_or_c_k.lex

Ns: NS_cweb_or_c_k

Version: 1.0

Debug: false

Grammar Comments:

Type: Thread

C++ or cweb type comments lexer.

1 element(s) in Lookahead Expression below

eolr

	Section	Page
Copyright	1	1
<i>cweb_or_c_k</i> combined “c++ ” or my “cweb” comment Thread	2	2
C++ comments	3	2
Cweb type comments	4	2
Fsm Ccweb_or_c_k class	5	2
Ccweb_or_c_k constructor directive	6	2
Ccweb_or_c_k op directive	7	2
Ccweb_or_c_k user-declaration directive	8	2
Ccweb_or_c_k user-implementation directive	9	2
<i>copy_kstr_into_buffer</i>	10	3
<i>Rcweb_or_c_k</i> rule	11	3
<i>Rcweb_or_c_k</i> 's subrule 1	12	3
<i>Rcweb_or_c_k</i> 's subrule 2	13	3
<i>Rcweb_or_c_k</i> 's subrule 3	14	3
<i>Rslash_at</i> rule	15	4
<i>Rslash_asterisk</i> rule	16	5
<i>R2slash</i> rule	17	6
First Set Language for O_2^{linker}	18	7
Lr1 State Network	19	8
Index	20	9