

1. Copyright.

Copyright © Dave Bone 1998 - 2015

2. *o2_sdc* Thread.

A single crawler of characters scooping up a grammar directive's syntax directed code. Added to it is “cweb” comment support. This allows the grammar document to support better code formatting particularly when the “syntax directed code” is big. The caveat is the emitted cpp code when “-p” option is on contains the “cweave” formatting directives and bypassed when not to generate the grammar document. So be warned. Generate the cpp code without the “-p” option to avoid the c++ errors.

3. Fsm Co2_sdc class.**4. Co2_sdc constructor directive.**

```
<Co2_sdc constructor directive 4> ≡
  ddd_idx_ = 0;
  ddd_[ddd_idx_] = 0;
```

5. Co2_sdc op directive.

```
<Co2_sdc op directive 5> ≡
  ddd_idx_ = 0;
  ddd_[ddd_idx_] = 0;
  parser_→set_use_all_shift_on();
  CAbs_lr1_sym * sym = parser_→start_token_;
  switch (sym→enumerated_id_) {
  case T_Enum::T_LR1_eog.: break;
  default: return;
  }
  parser_→set_use_all_shift_off();
```

6. Co2_sdc user-declaration directive.

```
<Co2_sdc user-declaration directive 6> ≡
public: char ddd_[1024 * 32];
  int ddd_idx_;
  void copy_str_into_buffer(std::string * Str);
  void copy_kstr_into_buffer(const char *Str);
```

7. Co2_sdc user-implementation directive.

```
<Co2_sdc user-implementation directive 7> ≡
void Co2_sdc::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;
}
```

8. *copy_kstr_into_buffer*.

⟨ More code 8 ⟩ ≡

```

void Co2_sdc::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;
}

```

9. Co2_sdc user-prefix-declaration directive.

⟨ Co2_sdc user-prefix-declaration directive 9 ⟩ ≡

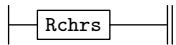
```

#include "cweb_or_c_k.h"
#include "c_string.h"
#include "c_literal.h"
#include "o2_code_end.h"
extern char PRT_SW;

```

10. *Ryacco2_syntax_code* rule.

Ryacco2_syntax_code

**11. *Ryacco2_syntax_code op* directive.**⟨ *Ryacco2_syntax_code op* directive 11 ⟩ ≡

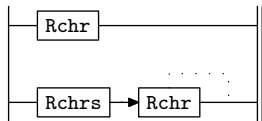
```

Co2_sdc * fsm = ( Co2_sdc * ) rule_info_.parser_--fsm_tbl_;
CAbs_lr1_sym * sym(0);
if ( fsm->ddd_idx_ ≡ 0 ) {
    sym = new Err_no_syntax_code_present;
    sym->set_rc(*rule_info_.parser_--start_token_, __FILE__, __LINE__);
    RSVP(sym);
    rule_info_.parser_--set_stop_parse(true);
    return;
}
sym = new T_syntax_code((const char *) &fsm->ddd_);
sym->set_rc(*rule_info_.parser_--start_token_, __FILE__, __LINE__);
RSVP(sym);

```

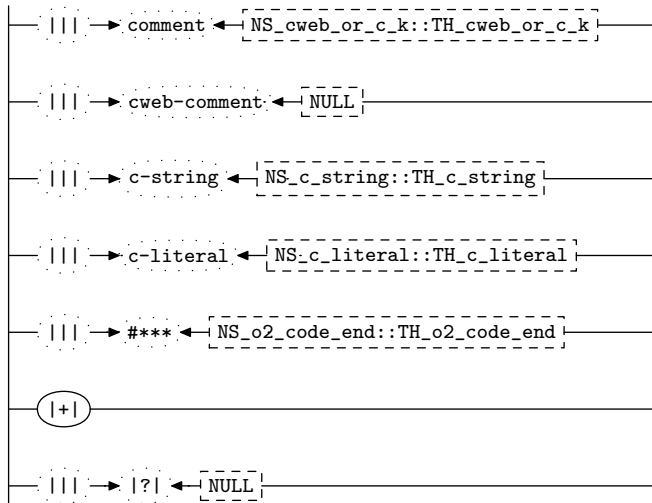
12. *Rchrs* rule.

Rchrs



13. Rchr rule.

Rchr



14. Rchr op directive.

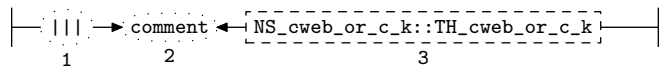
⟨Rchr op directive 14⟩ ≡

```

if (rule_info__parser__stop_parse__ ≡ true) return;
CAbs_lr1_sym * la_sym = rule_info__parser__current_token();
using namespace NS_yacco2_T_enum;
using namespace NS_yacco2_terminals; /* watch for overrun */
if (la_sym->enumerated_id__ ≡ T_Enum::T_LR1_eog_) {
    CAbs_lr1_sym * sym = new Err_no_end_of_code;
    sym->set_rc(*la_sym, __FILE__, __LINE__);
    RSVP(sym);
    rule_info__parser__set_stop_parse(true);
}

```

15. Rchr's subrule 1.



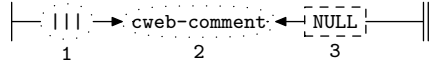
⟨Rchr subrule 1 op directive 15⟩ ≡

```

Co2_sdc * fsm = ( Co2_sdc * ) rule_info__parser__fsm_tbl__;
T_comment * k = sf-p2__;
k->set_auto_delete(true); /* delete it when it pops from stack */
fsm->copy_str_into_buffer(k->comment_data());

```

16. Rchr's subrule 2.

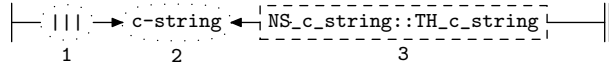


```

⟨Rchr subrule 2 op directive 16⟩ ≡
  Co2_sdc * fsm = ( Co2_sdc * ) rule_info...parser--fsm.tbl_;
  T_cweb_comment * k = sf-p2_;
  k-set_auto_delete(true); /* delete it when it pops from stack */
  if (PRT_SW ≡ 'n') return; /* bypass the cweve comments in the cpp code */
  fsm-copy_str_into_buffer(k-comment_data());
  string fmt("@<More_□code□>+=\n");
  fsm-copy_str_into_buffer(&fmt);

```

17. Rchr's subrule 3.

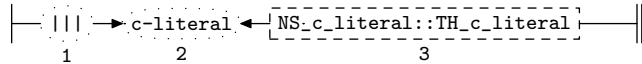


```

⟨Rchr subrule 3 op directive 17⟩ ≡
  Co2_sdc * fsm = ( Co2_sdc * ) rule_info...parser--fsm.tbl_;
  T_c_string * k = sf-p2_;
  k-set_auto_delete(true); /* delete it when it pops from stack */
  /* add back the bounding double quotes */
  fsm-ddd_[fsm-ddd_idx_] = '';
  ++fsm-ddd_idx_;
  fsm-copy_str_into_buffer(k-c_string());
  fsm-ddd_[fsm-ddd_idx_] = '';
  ++fsm-ddd_idx_;
  fsm-ddd_[fsm-ddd_idx_] = 0;

```

18. Rchr's subrule 4.

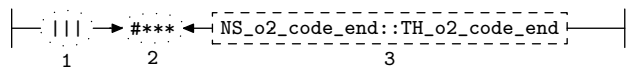


```

⟨Rchr subrule 4 op directive 18⟩ ≡
  Co2_sdc * fsm = ( Co2_sdc * ) rule_info...parser--fsm.tbl_;
  T_c_literal * k = sf-p2_;
  k-set_auto_delete(true); /* add back the bounding single quotes */
  fsm-ddd_[fsm-ddd_idx_] = '\'';
  ++fsm-ddd_idx_;
  fsm-copy_str_into_buffer(k-c_literal());
  fsm-ddd_[fsm-ddd_idx_] = '\'';
  ++fsm-ddd_idx_;
  fsm-ddd_[fsm-ddd_idx_] = 0;

```

19. Rchr's subrule 5.

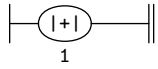


```

⟨Rchr subrule 5 op directive 19⟩ ≡
  sf-p2--set_auto_delete(true);
  rule_info...parser--set_use_all_shift_off();

```

20. Rchr's subrule 6.

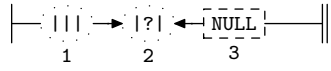


Add balance of characters to sdc code string: ex. comma. Special escape sequences are put back to their original declaration.

```

⟨Rchr subrule 6 op directive 20⟩ ≡
  Co2_sdc * fsm = ( Co2_sdc * ) rule_info...parser...fsm.tbl...;
  CAbs_lr1_sym * k = sf-p1...;
  switch (k-enumerated_id...) {
  case T_Enum:: T_LR1_eog_:
    {
      rule_info...parser...set_use_all_shift_off();
      break;
    }
  case T_Enum:: T_T_eol_:
    {
      fsm-copy_kstr_into_buffer("\n");
      break;
    }
  case T_Enum:: T_raw_sp_:
    {
      fsm-copy_kstr_into_buffer(" ");
      break;
    }
  case T_Enum:: T_raw_ff_:
    {
      fsm-copy_kstr_into_buffer("\f");
      break;
    }
  case T_Enum:: T_raw_vt_:
    {
      fsm-copy_kstr_into_buffer("\v");
      break;
    }
  case T_Enum:: T_raw_ht_:
    {
      fsm-copy_kstr_into_buffer("\t");
      break;
    }
  default:
    {
      fsm-ddd_[fsm-ddd_idx_] = k-id_[0];
      ++fsm-ddd_idx_;
      fsm-ddd_[fsm-ddd_idx_] = 0;
    }
  }
}

```

21. *Rchr's subrule 7.*

\langle Rchr subrule 7 op directive 21 $\rangle \equiv$

```
RSVP(sf-p2__);
rule_info__parser__set_stop_parse(true);
```

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

```
/*
  File: o2_sdc.fsc
  Date and Time: Fri Jan  2 15:33:46 2015
*/
transitive      y
grammar-name    "o2_sdc"
name-space     "NS_o2_sdc"
thread-name    "TH_o2_sdc"
monolithic     n
file-name      "o2_sdc.fsc"
no-of-T        569
list-of-native-first-set-terminals 1
  LR1_all_shift_operator
end-list-of-native-first-set-terminals
list-of-transitive-threads 4
  NS_cweb_or_c_k::TH_cweb_or_c_k
  NS_c_string::TH_c_string
  NS_c_literal::TH_c_literal
  NS_o2_code_end::TH_o2_code_end
end-list-of-transitive-threads
list-of-used-threads 4
  NS_c_literal::TH_c_literal
  NS_c_string::TH_c_string
  NS_cweb_or_c_k::TH_cweb_or_c_k
  NS_o2_code_end::TH_o2_code_end
end-list-of-used-threads
fsm-comments
"\02's syntax directed code extractor."
```


23. Lr1 State Network.

\Rightarrow					State: 1 state type: s				
\leftarrow	rule	\rightarrow	R#	sr#	Po	\leftarrow	subrule element	\rightarrow	Brn Gto Red LA
c	Rchr		3	1	1		comment NS_cweb_or_c_k::TH_cweb_or_c_k		1 2 4
c	Rchr		3	2	1		cweb-comment NULL		1 2 8
c	Rchr		3	3	1		c-string NS_c_string::TH_c_string		1 2 6
c	Rchr		3	4	1		c-literal NS_c_literal::TH_c_literal		1 2 5
c	Rchr		3	5	1		# *** NS_o2_code_end::TH_o2_code_end		1 2 7
c	Rchr		3	7	1		? NULL		1 2 3
c	Rchr		3	6	1	+			1 9 9
c	Ryacco2_syntax_code		1	1	1	Rchrs			1 10 10
c	Rchrs		2	2	1	Rchrs	<u>Rchr</u>		1 10 11
c	Rchrs		2	1	1	Rchr			1 12 12
\Rightarrow	<i>arbitration-code</i> : ϵ						State: 2 state type: s		
\leftarrow	rule	\rightarrow	R#	sr#	Po	\leftarrow	subrule element	\rightarrow	Brn Gto Red LA
t	Rchr		3	7	2	?			1 3 3
t	Rchr		3	1	2	comment			1 4 4
t	Rchr		3	4	2	c-literal			1 5 5
t	Rchr		3	3	2	c-string			1 6 6
t	Rchr		3	5	2	# ***			1 7 7
t	Rchr		3	2	2	cweb-comment			1 8 8
\Rightarrow	?						State: 3 state type: r		
\leftarrow	rule	\rightarrow	R#	sr#	Po	\leftarrow	subrule element	\rightarrow	Brn Gto Red LA
t	Rchr		3	7	3				1 0 3 1
\Rightarrow	<i>comment</i>						State: 4 state type: r		
\leftarrow	rule	\rightarrow	R#	sr#	Po	\leftarrow	subrule element	\rightarrow	Brn Gto Red LA
t	Rchr		3	1	3				1 0 4 1
\Rightarrow	<i>c-literal</i>						State: 5 state type: r		
\leftarrow	rule	\rightarrow	R#	sr#	Po	\leftarrow	subrule element	\rightarrow	Brn Gto Red LA
t	Rchr		3	4	3				1 0 5 1
\Rightarrow	<i>c-string</i>						State: 6 state type: r		
\leftarrow	rule	\rightarrow	R#	sr#	Po	\leftarrow	subrule element	\rightarrow	Brn Gto Red LA
t	Rchr		3	3	3				1 0 6 1
\Rightarrow	<i># ***</i>						State: 7 state type: r		
\leftarrow	rule	\rightarrow	R#	sr#	Po	\leftarrow	subrule element	\rightarrow	Brn Gto Red LA
t	Rchr		3	5	3				1 0 7 1
\Rightarrow	<i>cweb-comment</i>						State: 8 state type: r		
\leftarrow	rule	\rightarrow	R#	sr#	Po	\leftarrow	subrule element	\rightarrow	Brn Gto Red LA
t	Rchr		3	2	3				1 0 8 1
\Rightarrow	+						State: 9 state type: r		
\leftarrow	rule	\rightarrow	R#	sr#	Po	\leftarrow	subrule element	\rightarrow	Brn Gto Red LA
t	Rchr		3	6	2				1 0 9 1

$\Rightarrow R_{chrs}$		State: 10 state type: s/r										
\leftarrow	rule	\rightarrow	R#	sr#	Po	\leftarrow	subrule element	\rightarrow	Brn	Gto	Red	LA
t	Ryacco2_syntax_code		1	1	2				1	0	10	2
c	Rchr		3	1	1		comment NS_cweb_or_c_k::TH_cweb_or_c_k		10	2	4	
c	Rchr		3	2	1		cweb-comment NULL		10	2	8	
c	Rchr		3	3	1		c-string NS_c_string::TH_c_string		10	2	6	
c	Rchr		3	4	1		c-literal NS_c_literal::TH_c_literal		10	2	5	
c	Rchr		3	5	1		# *** NS_o2_code_end::TH_o2_code_end		10	2	7	
c	Rchr		3	7	1		? NULL		10	2	3	
c	Rchr		3	6	1	+			10	9	9	
t	Rchrs		2	2	2	Rchr			1	11	11	
$\Rightarrow R_{chr}$		State: 11 state type: r										
\leftarrow	rule	\rightarrow	R#	sr#	Po	\leftarrow	subrule element	\rightarrow	Brn	Gto	Red	LA
t	Rchrs		2	2	3				1	0	11	1
$\Rightarrow R_{chr}$		State: 12 state type: r										
\leftarrow	rule	\rightarrow	R#	sr#	Po	\leftarrow	subrule element	\rightarrow	Brn	Gto	Red	LA
t	Rchrs		2	1	2				1	0	12	1

24. Index.

|+|: 13.
 # ***: 13.
 |||: 13.
 |?|: 13.
 __FILE__: 11, 14.
 __LINE__: 11, 14.
 c-literal: 13.
 c-string: 13.
c.literal: 18.
c.str: 7.
c.string: 17.
CAbs_lr1_sym: 5, 11, 14, 20.
 comment: 13.
comment_data: 15, 16.
copy_kstr_into_buffer: 6, 8, 20.
copy_str_into_buffer: 6, 7, 15, 16, 17, 18.
Co2_sdc: 7, 8, 11, 15, 16, 17, 18, 20.
current_token: 14.
 cweb-comment: 13.
ddd: 4, 5, 6, 7, 8, 11, 17, 18, 20.
ddd_idx: 4, 5, 6, 7, 8, 11, 17, 18, 20.
enumerated_id: 5, 14, 20.
Err_no_end_of_code: 14.
Err_no_syntax_code_present: 11.
fmt: 16.
fsm: 11, 15, 16, 17, 18, 20.
fsm_tbl: 11, 15, 16, 17, 18, 20.
id: 20.
la_sym: 14.
NS_c_literal::TH_c_literal: 13.
NS_c_string::TH_c_string: 13.
NS_cweb_or_c_k::TH_cweb_or_c_k: 13.
NS_o2_code_end::TH_o2_code_end: 13.
NS_yacco2_T_enum: 14.
NS_yacco2_terminals: 14.
 NULL: 13.
o2_sdc: 2.
parser: 5, 11, 14, 15, 16, 17, 18, 19, 20, 21.
 PRT_SW: 9, 16.
p1: 20.
p2: 15, 16, 17, 18, 19, 21.
 Rchr: 12.
Rchr: 13, 15, 16, 17, 18, 19, 20, 21.
Rchrs: 12.
 Rchrs: 10, 12.
 RSVP: 11, 14, 21.
rule_info: 11, 14, 15, 16, 17, 18, 19, 20, 21.
Ryacco2_syntax_code: 10.
set_auto_delete: 15, 16, 17, 18, 19.
set_rc: 11, 14.
set_stop_parse: 11, 14, 21.
set_use_all_shift_off: 5, 19, 20.
set_use_all_shift_on: 5.
sf: 15, 16, 17, 18, 19, 20, 21.
start_token: 5, 11.
std: 6, 7.
stop_parse: 14.
Str: 6, 7, 8.
string: 6, 7, 16.
sym: 5, 11, 14.
T_c_literal: 18.
T_c_string: 17.
T_comment: 15.
T_cweb_comment: 16.
T_Enum: 5, 14, 20.
T_LR1_eog: 5, 14, 20.
T_raw_ff: 20.
T_raw_ht: 20.
T_raw_sp: 20.
T_raw_vt: 20.
T_syntax_code: 11.
T_T_eol: 20.
true: 11, 14, 15, 16, 17, 18, 19, 21.
x: 7, 8.
y: 7, 8.

⟨ Co2_sdc constructor directive 4 ⟩
⟨ Co2_sdc op directive 5 ⟩
⟨ Co2_sdc user-declaration directive 6 ⟩
⟨ Co2_sdc user-implementation directive 7 ⟩
⟨ Co2_sdc user-prefix-declaration directive 9 ⟩
⟨ More code 8 ⟩
⟨ Rchr op directive 14 ⟩
⟨ Rchr subrule 1 op directive 15 ⟩
⟨ Rchr subrule 2 op directive 16 ⟩
⟨ Rchr subrule 3 op directive 17 ⟩
⟨ Rchr subrule 4 op directive 18 ⟩
⟨ Rchr subrule 5 op directive 19 ⟩
⟨ Rchr subrule 6 op directive 20 ⟩
⟨ Rchr subrule 7 op directive 21 ⟩
⟨ Ryacco2_syntax_code op directive 11 ⟩

o2_sdc Grammar

Date: January 2, 2015 at 15:37

File: o2_sdc.lex

Ns: NS_o2_sdc

Version: 1.0

Debug: false

Grammar Comments:

Type: Thread

O_2 's syntax directed code extractor.

562 element(s) in Lookahead Expression below

eolr - | + | - |||

	Section	Page
Copyright	1	1
o2_sdc Thread	2	2
Fsm Co2_sdc class	3	2
Co2_sdc constructor directive	4	2
Co2_sdc op directive	5	2
Co2_sdc user-declaration directive	6	2
Co2_sdc user-implementation directive	7	2
<i>copy_kstr_into_buffer</i>	8	3
Co2_sdc user-prefix-declaration directive	9	3
<i>Ryacco2_syntax_code</i> rule	10	3
Ryacco2_syntax_code op directive	11	3
<i>Rchrs</i> rule	12	3
<i>Rchr</i> rule	13	4
Rchr op directive	14	4
<i>Rchr</i> 's subrule 1	15	4
<i>Rchr</i> 's subrule 2	16	5
<i>Rchr</i> 's subrule 3	17	5
<i>Rchr</i> 's subrule 4	18	5
<i>Rchr</i> 's subrule 5	19	5
<i>Rchr</i> 's subrule 6	20	6
<i>Rchr</i> 's subrule 7	21	7
First Set Language for O_2^{linker}	22	8
Lr1 State Network	23	9
Index	24	11