

1. Copyright.

Copyright © Dave Bone 1998 - 2015

2. *rhs_component* Thread.

A component of subrule.

Note: the forest versus trees on T and ||| which is a T but i wanted clarity within “subrule.lex” to break out a regular grammar expression from a thread expression

For clarity within the *subrule_def* grammar, eosubrule is recognized only by *rhs_bnd* grammar. Well now, \rightarrow is a T. So I trap it in *find_T_of_chr_str* and abort the parse. This stops the recognition on RToRcomponents of *subrule_def*.

I could have used the enumeration approach to test its presence but what is its value in the sea of Terminals. It’s not like the hardwired |||lr k terminal that is fixed in place and time.

3. Fsm Crhs_component class.**4. Crhs_component user-declaration directive.**

```
<Crhs_component user-declaration directive 4> ≡
public: void find_T_of_chr_str(const char *Str, yacco2 :: CAbs_lr1_sym * Tok);
```

5. Crhs_component user-implementation directive.

```
<Crhs_component user-implementation directive 5> ≡
void Crhs_component :: find_T_of_chr_str(const char *Str, yacco2 :: CAbs_lr1_sym * Tok){
    T_sym_tbl_report_card report_card;
    using namespace yacco2_stbl;
    find_sym_in_stbl(report_card, *Str);
    if (report_card.action_ ≡ T_sym_tbl_report_card :: not_fnd) {
        CAbs_lr1_sym * sym = new Err_subrule_use_undefined_T;
        sym->set_rc(*parser->start_token_, __FILE__, __LINE__);
        RSVP_FSM(sym);
        parser->set_stop_parse(true);
        return;
    }
    if (report_card.tbl_entry->type_ ≠ table_entry :: terminal) {
        CAbs_lr1_sym * sym = new Err_not_T_or_R_in_subrule_expr;
        sym->set_rc(*parser->start_token_, __FILE__, __LINE__);
        RSVP_FSM(sym);
        parser->set_stop_parse(true);
        return;
    }
    report_card.tbl_entry->used_ = true;
    CAbs_lr1_sym * sym = report_card.tbl_entry->symbol_; T_in_stbl * T = ( T_in_stbl * ) sym;
    refered_T * rT = new refered_T(*T);
    rT->set_rc(*Tok, __FILE__, __LINE__);
    T->add_T_into_xref(*rT);
    RSVP_FSM(T);
    parser->set_stop_parse(true); }
```

6. Crhs_component user-prefix-declaration directive.

```

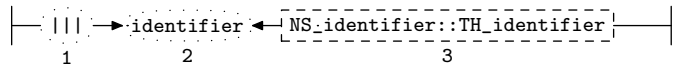
⟨Crhs_component user-prefix-declaration directive 6⟩ ≡
#include "identifier.h"
#include "c_string.h"
#include "cweb_or_c_k.h"
#include "unq_str.h"
#include "yacco2_stbl.h"

```

7. Rrhs_component rule.

Rrhs_component

	→	identifier	←	NS_identifier::TH_identifier
	→	c-string	←	NS_c_string::TH_c_string
	→	unquoted-string	←	NS_unq_str::TH_unq_str
	→	cweb-comment	←	NS_cweb_or_c_k::TH_cweb_or_c_k
	→	+	←	NULL
	→	rule-in-stbl	←	NULL
	→	T-in-stbl	←	NULL
	→	kw-in-stbl	←	NULL

8. *Rrhs_component*'s subrule 1.

Used rule and not defined yet, so create it.

\langle Rrhs_component subrule 1 op directive 8 $\rangle \equiv$

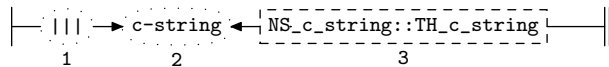
```

const char *skey = sf→p2→identifier()→c_str();
rule_def *rdef = new rule_def(skey);
rdef→set_rc(*sf→p2→, __FILE__, __LINE__);
rule_in_stbl *ristbl = new rule_in_stbl(*rdef);
ristbl→set_rc(*sf→p2→, __FILE__, __LINE__);
referred_rule *rr = new referred_rule(*ristbl);
rr→set_rc(*sf→p2→, __FILE__, __LINE__);
ristbl→add_R_into_xref(*rr);
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::used, 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_);
RSVP(ristbl);
rule_info→.parser→set_stop_parse(true);

```

9. *Rrhs_component*'s subrule 2.

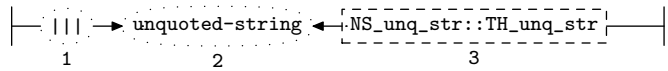
Considered a T type.

\langle Rrhs_component subrule 2 op directive 9 $\rangle \equiv$

```

Crhs_component *fsm = ( Crhs_component * ) rule_info→.parser→fsm_tbl→;
fsm→fnd_T_of_chr_str(sf→p2→c_string()→c_str(), sf→p2→);
sf→p2→set_auto_delete(true);

```

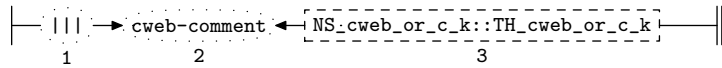
10. *Rrhs_component's subrule 3.*

Considered a T type. Why the check on subrule vector: \rightarrow ? This is a context problem whereby my grammars use it and what happens when a language being defined also has it? So i abort on the unprotected \rightarrow so that *rhs_bnd* grammar covers it as one of my grammar's terminals. The quoted version is passed thru as is. Fractal logic on self contexts: The inner self and the outer contextual grounds.

```

⟨Rrhs_component subrule 3 op directive 10⟩ ≡ /* only accept when no identifier */
  Crhs_component * fsm = ( Crhs_component * ) rule_info_.parser_--fsm_tbl_;
  sf-p2_--set_auto_delete(true);
  string eos_chk(">");
  if ( eos_chk ≡ sf-p2_--unquoted_string()-c_str() ) {
    rule_info_.parser_--set_abort_parse(true);
    return;
  }
  fsm-fnd_T_of_chr_str(sf-p2_--unquoted_string()-c_str(), sf-p2_--);

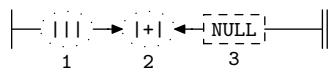
```

11. *Rrhs_component's subrule 4.*

```

⟨Rrhs_component subrule 4 op directive 11⟩ ≡
  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_);
  RSVP(cw);
  rule_info_.parser_--set_stop_parse(true);

```

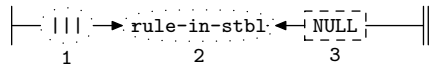
12. *Rrhs_component's subrule 5.*

Forward the error.

```

⟨Rrhs_component subrule 5 op directive 12⟩ ≡
  RSVP(sf-p2_--);
  rule_info_.parser_--set_stop_parse(true);

```

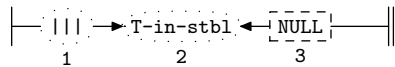
13. *Rrhs_component*'s subrule 6.

⟨Rrhs_component subrule 6 op directive 13⟩ ≡

```

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;
}
report_card.tbl_entry→used_ = true;
referred_rule * rr = new referred_rule(*sf→p2→);
rr→set_rc(*sf→p2→, __FILE__, __LINE__);
sf→p2→add_R_into_xref(*rr);
RSVP(sf→p2→);
rule_info→parser→set_stop_parse(true);

```

14. *Rrhs_component*'s subrule 7.

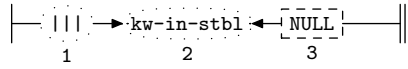
Used T, so indicate its xreference.

⟨Rrhs_component subrule 7 op directive 14⟩ ≡

```

using namespace yacco2_stbl;
T_sym_tbl_report_card report_card;
find_sym_in_stbl(report_card, *sf→p2→t_def()→t_name()→c_str());
if (report_card.action_ ≡ T_sym_tbl_report_card::not_fnd) {
    CAbs_lr1_sym * sym = new Err_T_not_in_stbl;
    sym→set_rc(*rule_info→parser→start_token_, __FILE__, __LINE__);
    RSVP(sym);
    rule_info→parser→set_stop_parse(true);
    return;
}
report_card.tbl_entry→used_ = true;
referred_T * rt = new referred_T(*sf→p2→);
rt→set_rc(*sf→p2→, __FILE__, __LINE__);
sf→p2→add_T_into_xref(*rt);
RSVP(sf→p2→);
rule_info→parser→set_stop_parse(true);

```

15. *Rrhs_component's subrule 8.*

\langle Rrhs_component subrule 8 op directive 15 $\rangle \equiv$

```
CAbs_lr1_sym * sym = new Err_not_T_or_R_in_subrule_expr;
sym->set_rc(*rule_info__parser__->start_token__, __FILE__, __LINE__);
RSVP(sym);
rule_info__parser__->set_stop_parse(true);
```

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

```
/*
  File: rhs_component.fsc
  Date and Time: Fri Jan  2 15:33:52 2015
*/
transitive      y
grammar-name    "rhs_component"
name-space     "NS_rhs_component"
thread-name     "TH_rhs_component"
monolithic     n
file-name      "rhs_component.fsc"
no-of-T        569
list-of-native-first-set-terminals 0
end-list-of-native-first-set-terminals
list-of-transitive-threads 4
  NS_unq_str::TH_unq_str
  NS_identifier::TH_identifier
  NS_c_string::TH_c_string
  NS_cweb_or_c_k::TH_cweb_or_c_k
end-list-of-transitive-threads
list-of-used-threads 4
  NS_c_string::TH_c_string
  NS_cweb_or_c_k::TH_cweb_or_c_k
  NS_identifier::TH_identifier
  NS_unq_str::TH_unq_str
end-list-of-used-threads
fsm-comments
"Subule's individual component recognizer except eosubrule."
```


17. Lr1 State Network.

⇒					State: 1 state type: <i>s</i>				
←	rule	→	R#	sr#	Po	←	subrule element	→	Brn Gto Red LA
c	Rrhs_component		1	3	1		unquoted-string NS.unq_str::TH.unq_str		1 2 5
c	Rrhs_component		1	6	1		rule-in-stbl NULL		1 2 8
c	Rrhs_component		1	8	1		kw-in-stbl NULL		1 2 9
c	Rrhs_component		1	1	1		identifier NS.identifier::TH.identifier		1 2 6
c	Rrhs_component		1	2	1		c-string NS.c_string::TH.c_string		1 2 4
c	Rrhs_component		1	4	1		cweb-comment NS.cweb_or_c.k::TH.cweb_or_c.k		1 2 10
c	Rrhs_component		1	5	1		+ NULL		1 2 3
c	Rrhs_component		1	7	1		T-in-stbl NULL		1 2 7
⇒		arbitration-code: AR_Rrhs_component					State: 2 state type: <i>s</i>		
←	rule	→	R#	sr#	Po	←	subrule element	→	Brn Gto Red LA
t	Rrhs_component		1	5	2	+			1 3 3
t	Rrhs_component		1	2	2	c-string			1 4 4
t	Rrhs_component		1	3	2	unquoted-string			1 5 5
t	Rrhs_component		1	1	2	identifier			1 6 6
t	Rrhs_component		1	7	2	T-in-stbl			1 7 7
t	Rrhs_component		1	6	2	rule-in-stbl			1 8 8
t	Rrhs_component		1	8	2	kw-in-stbl			1 9 9
t	Rrhs_component		1	4	2	cweb-comment			1 10 10
⇒	+						State: 3 state type: <i>r</i>		
←	rule	→	R#	sr#	Po	←	subrule element	→	Brn Gto Red LA
t	Rrhs_component		1	5	3				1 0 3 1
⇒	c-string						State: 4 state type: <i>r</i>		
←	rule	→	R#	sr#	Po	←	subrule element	→	Brn Gto Red LA
t	Rrhs_component		1	2	3				1 0 4 1
⇒	unquoted-string						State: 5 state type: <i>r</i>		
←	rule	→	R#	sr#	Po	←	subrule element	→	Brn Gto Red LA
t	Rrhs_component		1	3	3				1 0 5 1
⇒	identifier						State: 6 state type: <i>r</i>		
←	rule	→	R#	sr#	Po	←	subrule element	→	Brn Gto Red LA
t	Rrhs_component		1	1	3				1 0 6 1
⇒	T-in-stbl						State: 7 state type: <i>r</i>		
←	rule	→	R#	sr#	Po	←	subrule element	→	Brn Gto Red LA
t	Rrhs_component		1	7	3				1 0 7 1
⇒	rule-in-stbl						State: 8 state type: <i>r</i>		
←	rule	→	R#	sr#	Po	←	subrule element	→	Brn Gto Red LA
t	Rrhs_component		1	6	3				1 0 8 1
⇒	kw-in-stbl						State: 9 state type: <i>r</i>		
←	rule	→	R#	sr#	Po	←	subrule element	→	Brn Gto Red LA
t	Rrhs_component		1	8	3				1 0 9 1

\Rightarrow *cweb-comment*

←	rule	→	R#	sr#	Po	←
t	Rrhs_component		1	4	3	

State: 10 state type: *r*

	subrule	element
--	----------------	----------------

→	Brn	Gto	Red	LA
	1	0	10	1

18. Index.

|+|: 7.
 |||: 7.
 __FILE__: 5, 8, 11, 13, 14, 15.
 __LINE__: 5, 8, 11, 13, 14, 15.
 action_: 5, 13, 14.
 add_R_into_xref: 8, 13.
 add_sym_to_stbl: 8.
 add_T_into_xref: 5, 14.
 AR_Rrhs_component: 17.
 AST: 11.
 c-string: 7.
 c_str: 8, 9, 10, 13, 14.
 c_string: 9.
 CAbs_lr1_sym: 4, 5, 13, 14, 15.
 Crhs_component: 5, 9, 10.
 cw: 11.
 cweb-comment: 7.
 cweb_t: 11.
 cwebk_t: 11.
 eos_chk: 10.
 err_entry_: 8.
 Err_not_T_or_R_in_subrule_expr: 5, 15.
 Err_rule_not_in_stbl: 13.
 Err_subrule_use_undefined_T: 5.
 Err_T_not_in_stbl: 14.
 find_sym_in_stbl: 5, 13, 14.
 fnd_T_of_chr_str: 2, 4, 5, 9, 10.
 fsm: 9, 10.
 fsm_ttbl_: 9, 10.
 identifier: 7.
 identifier: 8.
 join_pts: 11.
 kw-in-stbl: 7.
 not_fnd: 5, 13, 14.
 NS_c_string::TH_c_string: 7.
 NS_cweb_or_c_k::TH_cweb_or_c_k: 7.
 NS_identifier::TH_identifier: 7.
 NS_unq_str::TH_unq_str: 7.
 NULL: 7.
 okay: 8.
 parser_: 5, 8, 9, 10, 11, 12, 13, 14, 15.
 pos_: 8.
 p2_: 8, 9, 10, 11, 12, 13, 14.
 r_def: 13.
 rdef: 8.
 refered_rule: 8, 13.
 refered_T: 5, 14.
 report_card: 5, 8, 13, 14.
 rhs_bnd: 2, 10.
 rhs_component: 2.
 ristbl: 8.
 rr: 8, 13.
 Rrhs_component: 7, 8, 9, 10, 11, 12, 13, 14, 15.
 RSVP: 8, 11, 12, 13, 14, 15.
 RSVP_FSM: 5.
 rt: 14.
 rT: 5.
 rule: 8.
 rule-in-stbl: 7.
 rule_def: 8.
 rule_in_stbl: 8.
 rule_info_: 8, 9, 10, 11, 12, 13, 14, 15.
 rule_name: 13.
 set_abort_parse: 10.
 set_auto_delete: 8, 9, 10.
 set_content: 11.
 set_rc: 5, 8, 11, 13, 14, 15.
 set_stop_parse: 5, 8, 11, 12, 13, 14, 15.
 sf: 8, 9, 10, 11, 12, 13, 14.
 skey: 8.
 start_token_: 5, 13, 14, 15.
 status_: 8.
 stbl_idx: 8.
 Str: 4, 5.
 string: 10.
 subrule_def: 2.
 sym: 5, 13, 14, 15.
 symbol_: 5.
 T-in-stbl: 7.
 T_cweb_comment: 11.
 T_cweb_marker: 11.
 t_def: 14.
 T_in_stbl: 5.
 t_name: 14.
 T_sym_ttbl_report_card: 5, 8, 13, 14.
 table_entry: 5, 8.
 tbl_entry_: 5, 13, 14.
 terminal: 5.
 Tok: 4, 5.
 true: 5, 8, 9, 10, 11, 12, 13, 14, 15.
 type_: 5.
 unquoted-string: 7.
 unquoted_string: 10.
 used: 8.
 used_: 5, 13, 14.
 yacco2: 4, 5.
 yacco2_stbl: 5, 8, 13, 14.

⟨ Crhs_component user-declaration directive 4 ⟩
⟨ Crhs_component user-implementation directive 5 ⟩
⟨ Crhs_component user-prefix-declaration directive 6 ⟩
⟨ Rrhs_component subrule 1 op directive 8 ⟩
⟨ Rrhs_component subrule 2 op directive 9 ⟩
⟨ Rrhs_component subrule 3 op directive 10 ⟩
⟨ Rrhs_component subrule 4 op directive 11 ⟩
⟨ Rrhs_component subrule 5 op directive 12 ⟩
⟨ Rrhs_component subrule 6 op directive 13 ⟩
⟨ Rrhs_component subrule 7 op directive 14 ⟩
⟨ Rrhs_component subrule 8 op directive 15 ⟩

rhs_component Grammar

Date: January 2, 2015 at 15:38

File: rhs_component.lex Ns: NS_rhs_component

Version: 1.0 Debug: true

Grammar Comments: Type: Thread

Subule's individual component recognizer except eosubrle.

1 element(s) in Lookahead Expression below

eolr

	Section	Page
Copyright	1	1
<i>rhs_component</i> Thread	2	2
Fsm Crhs_component class	3	2
Crhs_component user-declaration directive	4	2
Crhs_component user-implementation directive	5	2
Crhs_component user-prefix-declaration directive	6	3
<i>Rrhs_component</i> rule	7	3
<i>Rrhs_component</i> 's subrule 1	8	4
<i>Rrhs_component</i> 's subrule 2	9	4
<i>Rrhs_component</i> 's subrule 3	10	5
<i>Rrhs_component</i> 's subrule 4	11	5
<i>Rrhs_component</i> 's subrule 5	12	5
<i>Rrhs_component</i> 's subrule 6	13	6
<i>Rrhs_component</i> 's subrule 7	14	6
<i>Rrhs_component</i> 's subrule 8	15	7
First Set Language for O_2^{linker}	16	8
Lr1 State Network	17	9
Index	18	11