1. Copyright.

2. *linker_pass3* **grammar.**
Quasi lexical / syntatic Phase for Yacco2's first set linker.
Sample of a grammar's First set control file:

```
/*
 File: subrule_def.fsc
 Date and Time: Fri Dec  7 16:28:47 2007
*/
transitive   y
grammar-name "subrule_def"
name-space   "NS_subrule_def"
thread-name  "TH_subrule_def"
monolithic   n
file-name    "subrule_def.fsc"
no-of-T      567
list-of-native-first-set-terminals 0
end-list-of-native-first-set-terminals
list-of-transitive-threads 1
  NS_subrule_vector::TH_subrule_vector
end-list-of-transitive-threads
list-of-used-threads 10
  NS_cweb_marker::TH_cweb_marker
  NS_dbl_colon::TH_dbl_colon
  NS_identifier::TH_identifier
  NS_lint_balls::TH_lint_balls
  NS_o2_sdc::TH_o2_sdc
  NS_parallel_oper::TH_parallel_oper
  NS_rhs_bnd::TH_rhs_bnd
  NS_rhs_component::TH_rhs_component
  NS_rtn_component::TH_rtn_component
  NS_subrule_vector::TH_subrule_vector
end-list-of-used-threads
fsm-comments
"Parse a subrule: into the valley of sin..."
```

3. Fsm Clinker_pass3 class.

4. Clinker_pass3 user-declaration directive.

⟨ Clinker_pass3 user-declaration directive 4 ⟩ ≡
public: **bool** *chk_filename*(**const char** ∗*Filename*, *CAbs_lr1_sym* ∗ *Tok*, *CAbs_lr1_sym* ∗ *Err*);
bool *chk_ofilename*(**const char** ∗*Filename*, *CAbs_lr1_sym* ∗ *Tok*);
yacco2 :: *CAbs_lr1_sym* ∗ *chk_for_overrun*( );
std :: *string t_alphabet_filename_*;
std :: *string emitfile_filename_*;
T_syntax_code ∗ *preamble_srce_*;
std :: *vector* < *std* :: *string* > *grammars_fsc_files_*;
std :: *vector* < *std* :: *string* > *directories_*;

5. Clinker_pass3 user-implementation directive.

⟨ Clinker_pass3 user-implementation directive 5 ⟩ ≡

```
bool Clinker_pass3::chk_ofilename(const char *Filename, CAbs_lr1_sym * Tok)
{
  std::ofstream fle_chk;
  fle_chk.open(Filename, ios::out);
  if (¬fle_chk.good()) {
    CAbs_lr1_sym * sym = new Err_bad_filename(Filename);
    sym→set_rc(*Tok, __FILE__, __LINE__);
    parser__→add_token_to_error_queue(*sym);
    parser__→set_stop_parse(true);
    return false;
  }
  else {
    fle_chk.close();
  }
  return true;
}
bool Clinker_pass3::chk_filename(const char *Filename, CAbs_lr1_sym * Tok, CAbs_lr1_sym * Err)
{
  std::ifstream fle_chk;
  fle_chk.open(Filename);
  if (¬fle_chk.good()) {
    Err→set_rc(*Tok, __FILE__, __LINE__);
    parser__→add_token_to_error_queue(*Err);
    parser__→set_stop_parse(true);
    return false;
  }
  else {
    delete Err;
    fle_chk.close();
  }
  return true;
}
yacco2::CAbs_lr1_sym * Clinker_pass3::chk_for_overrun()
{
  switch (parser__→current_token()→enumerated_id__) {
  case T_Enum::T_LR1_eog_: break;
  default: return 0;
  }
  yacco2::CAbs_lr1_sym * sym = new Err_end_preamble_kw_not_present;
  sym→set_rc(*parser__→start_token__, __FILE__, __LINE__);
  parser__→set_use_all_shift_off();
  return sym;
}
```

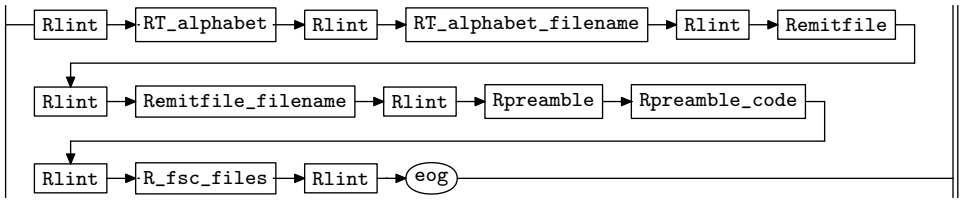
6. Clinker_pass3 user-prefix-declaration directive.

⟨ Clinker_pass3 user-prefix-declaration directive 6 ⟩ ≡

```
#include "linker_id.h"
#include "linker_preamble_code.h"
#include "lint_balls.h"
#include "c_string.h"
#include "angled_string.h"
#include "c_literal.h"
#include "bad_char_set.h"
  using namespace NS_yacco2_terminals;
```

7. *Rlinker_pass3* **rule.**

Rlinker_pass3



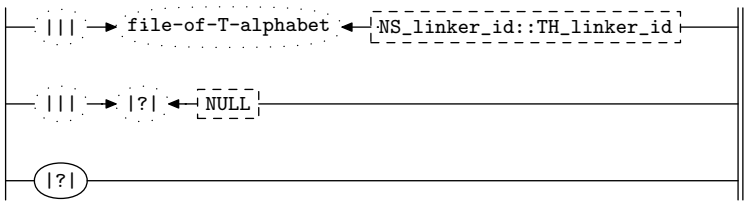
8. Rlinker_pass3 op directive.

⟨ Rlinker_pass3 op directive 8 ⟩ ≡

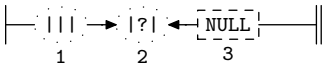
```
  using namespace NS_yacco2_k_symbols;
  ADD_TOKEN_TO_PRODUCER_QUEUE(*yacco2::PTR_LR1_eog__);
  ADD_TOKEN_TO_PRODUCER_QUEUE(*yacco2::PTR_LR1_eog__);
```

9. *RT_alphabet* **rule.**

RT_alphabet



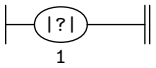
10. *RT_alphabet*'**s subrule 2.**



⟨ RT_alphabet subrule 2 op directive 10 ⟩ ≡

```
  CAbs_lr1_sym * sym = new Err_file_of_T_alphabet_kw_not_present;
  sym->set_rc(*sf->p2__, __FILE__, __LINE__);
  ADD_TOKEN_TO_ERROR_QUEUE(*sym);
  rule_info__.parser__->set_stop_parse(true);
  sf->p2__->set_auto_delete(true);
```

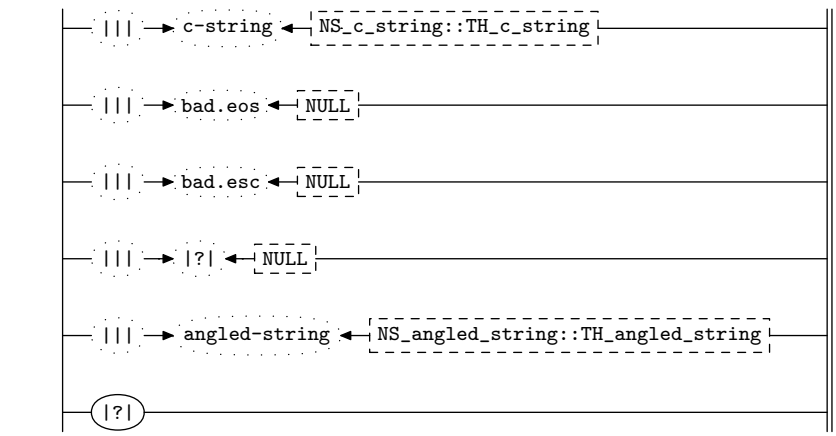
11. *RT_alphabet*'**s subrule 3.**



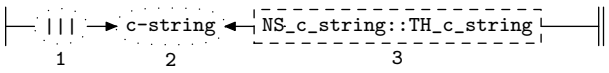
⟨ RT_alphabet subrule 3 op directive 11 ⟩ ≡

```
CAbs_lr1_sym * sym = new Err_file_of_T_alphabet_kw_not_present;
sym->set_rc(*rule_info__.parser__->current_token( ), __FILE__, __LINE__);
ADD_TOKEN_TO_ERROR_QUEUE(*sym);
rule_info__.parser__->set_stop_parse(true);
```

12. *RT_alphabet_filename* **rule.**



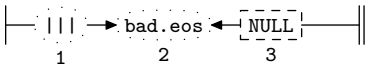
13. *RT_alphabet_filename*'**s subrule 1.**



⟨ RT_alphabet_filename subrule 1 op directive 13 ⟩ ≡

```
Clinker_pass3 * fsm = ( Clinker_pass3 * ) rule_info__.parser__->fsm_tbl__;
const char *fn = sf->p2__->c_string( )->c_str( );
if (fsm->chk_filename(fn, sf->p2__, new Err_T_alphabet_file_does_not_exist) ≡ false) return;
fsm->t_alphabet_filename_ += fn;
sf->p2__->set_auto_delete(true);
```

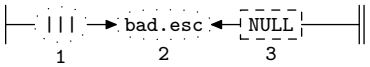
14. *RT_alphabet_filename*'**s subrule 2.**



⟨ RT_alphabet_filename subrule 2 op directive 14 ⟩ ≡

```
ADD_TOKEN_TO_ERROR_QUEUE(*sf->p2__);
rule_info__.parser__->set_stop_parse(true);
```

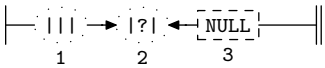
15. *RT_alphabet_filename*'s subrule 3.



⟨ RT_alphabet_filename subrule 3 op directive 15 ⟩ ≡

```
ADD_TOKEN_TO_ERROR_QUEUE(*sf→p2__);
rule_info__.parser__→set_stop_parse(true);
```

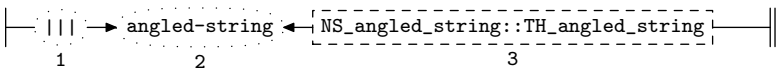
16. *RT_alphabet_filename*'s subrule 4.



⟨ RT_alphabet_filename subrule 4 op directive 16 ⟩ ≡

```
CAbs_lr1_sym * sym = new Err_no_filename_present;
sym→set_rc(*sf→p2__, __FILE__, __LINE__);
ADD_TOKEN_TO_ERROR_QUEUE(*sym);
rule_info__.parser__→set_stop_parse(true);
sf→p2__→set_auto_delete(true);
```

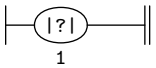
17. *RT_alphabet_filename*'s subrule 5.



⟨ RT_alphabet_filename subrule 5 op directive 17 ⟩ ≡

```
Clinker_pass3 * fsm = ( Clinker_pass3 * ) rule_info__.parser__→fsm_tbl__;
const char *fn = sf→p2__→angled_string()→c_str();
if (fsm→chk_filename(fn, sf→p2__, new Err_T_alphabet_file_does_not_exist) ≡ false) return;
fsm→t_alphabet_filename_ += fn;
sf→p2__→set_auto_delete(true);
```

18. *RT_alphabet_filename*'s subrule 6.

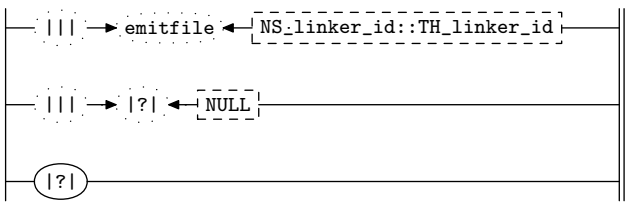


⟨ RT_alphabet_filename subrule 6 op directive 18 ⟩ ≡

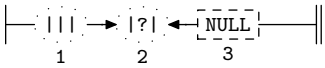
```
CAbs_lr1_sym * sym = new Err_T_alphabet_file_not_present;
sym→set_rc(*rule_info__.parser__→current_token(), __FILE__, __LINE__);
ADD_TOKEN_TO_ERROR_QUEUE(*sym);
rule_info__.parser__→set_stop_parse(true);
```

19. *Remitfile* **rule.**

Remitfile



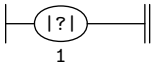
20. *Remitfile*'s subrule 2.



⟨ Remitfile subrule 2 op directive 20 ⟩ ≡

```
CAbs_lr1_sym * sym = new Err_emitfile_kw_not_present;
sym->set_rc(*sf->p2__, __FILE__, __LINE__);
ADD_TOKEN_TO_ERROR_QUEUE(*sym);
rule_info__.parser__->set_stop_parse(true);
sf->p2__->set_auto_delete(true);
```

21. *Remitfile*'s subrule 3.

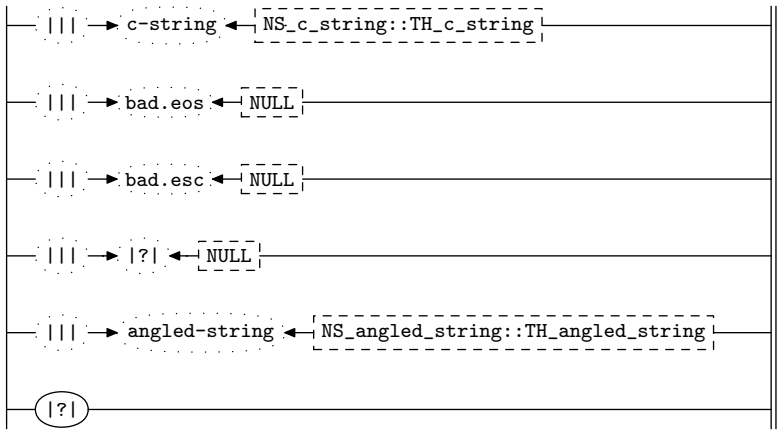


⟨ Remitfile subrule 3 op directive 21 ⟩ ≡

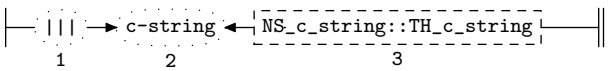
```
CAbs_lr1_sym * sym = new Err_emitfile_kw_not_present;
sym->set_rc(*rule_info__.parser__->current_token( ), __FILE__, __LINE__);
ADD_TOKEN_TO_ERROR_QUEUE(*sym);
rule_info__.parser__->set_stop_parse(true);
```

22. *Remitfile_filename* **rule.**

Remitfile_filename



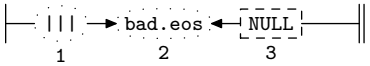
23. *Remitfile_filename*'s subrule 1.



⟨ Remitfile_filename subrule 1 op directive 23 ⟩ ≡

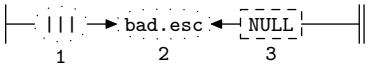
```
Clinker_pass3 * fsm = ( Clinker_pass3 * ) rule_info__.parser__->fsm_tbl__;
const char *fn = sf->p2__->c_string( )->c_str( );
if (fsm->chk_ofilename(fn, sf->p2__) ≡ false) return;
fsm->emitfile_filename_ += fn;
sf->p2__->set_auto_delete(true);
```

24. *Remitfile_filename*'**s subrule 2.**



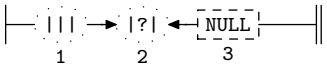
⟨ Remitfile_filename subrule 2 op directive 24 ⟩ ≡
```
ADD_TOKEN_TO_ERROR_QUEUE(*sf→p2__);
rule_info__.parser__→set_stop_parse(true);
```

25. *Remitfile_filename*'**s subrule 3.**



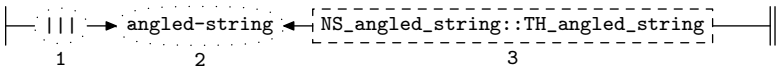
⟨ Remitfile_filename subrule 3 op directive 25 ⟩ ≡
```
ADD_TOKEN_TO_ERROR_QUEUE(*sf→p2__);
rule_info__.parser__→set_stop_parse(true);
```

26. *Remitfile_filename*'**s subrule 4.**



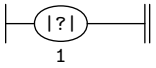
⟨ Remitfile_filename subrule 4 op directive 26 ⟩ ≡
```
CAbs_lr1_sym * sym = new Err_emitfile_file_not_present;
sym→set_rc(*sf→p2__, __FILE__, __LINE__);
ADD_TOKEN_TO_ERROR_QUEUE(*sym);
rule_info__.parser__→set_stop_parse(true);
sf→p2__→set_auto_delete(true);
```

27. *Remitfile_filename*'**s subrule 5.**



⟨ Remitfile_filename subrule 5 op directive 27 ⟩ ≡
```
Clinker_pass3 * fsm = ( Clinker_pass3 * ) rule_info__.parser__→fsm_tbl__;
const char *fn = sf→p2__→angled_string( )→c_str( );
if (fsm→chk_ofilename(fn, sf→p2__) ≡ false) return;
fsm→emitfile_filename_ += fn;
sf→p2__→set_auto_delete(true);
```

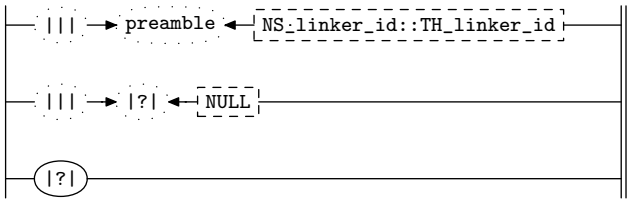
28. *Remitfile_filename*'**s subrule 6.**



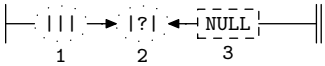
⟨ Remitfile_filename subrule 6 op directive 28 ⟩ ≡
```
CAbs_lr1_sym * sym = new Err_emitfile_file_not_present;
sym→set_rc(*rule_info__.parser__→current_token( ), __FILE__, __LINE__);
ADD_TOKEN_TO_ERROR_QUEUE(*sym);
rule_info__.parser__→set_stop_parse(true);
```

29. *Rpreamble* **rule.**

Rpreamble



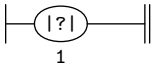
30. *Rpreamble*'**s subrule 2.**



⟨ Rpreamble subrule 2 op directive 30 ⟩ ≡

```
CAbs_lr1_sym * sym = new Err_preamble_kw_not_present;
sym->set_rc(*sf->p2__, __FILE__, __LINE__);
ADD_TOKEN_TO_ERROR_QUEUE(*sym);
rule_info__.parser__->set_stop_parse(true);
sf->p2__->set_auto_delete(true);
```

31. *Rpreamble*'**s subrule 3.**

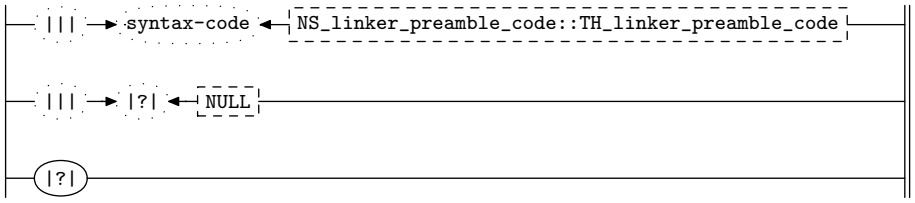


⟨ Rpreamble subrule 3 op directive 31 ⟩ ≡

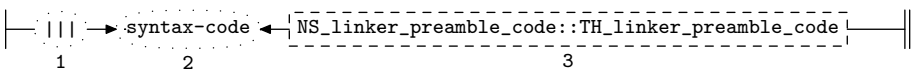
```
CAbs_lr1_sym * sym = new Err_preamble_kw_not_present;
sym->set_rc(*rule_info__.parser__->current_token( ), __FILE__, __LINE__);
ADD_TOKEN_TO_ERROR_QUEUE(*sym);
rule_info__.parser__->set_stop_parse(true);
```

32. *Rpreamble_code* **rule.**

Rpreamble_code



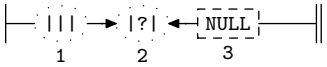
33. *Rpreamble_code*'**s subrule 1.**



⟨ Rpreamble_code subrule 1 op directive 33 ⟩ ≡

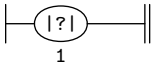
```
Clinker_pass3 * fsm = ( Clinker_pass3 * ) rule_info__.parser__->fsm_tbl__;
fsm->preamble_srce_ = sf->p2__;
```

34. *Rpreamble_code*'s subrule 2.



⟨ Rpreamble_code subrule 2 op directive 34 ⟩ ≡
```
ADD_TOKEN_TO_ERROR_QUEUE(*sf→p2__);
rule_info__.parser__→set_stop_parse(true);
```

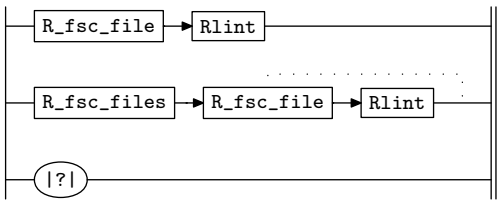
35. *Rpreamble_code*'s subrule 3.



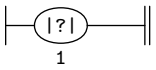
⟨ Rpreamble_code subrule 3 op directive 35 ⟩ ≡
```
CAbs_lr1_sym * sym = new Err_preamble_srce_code_not_present;
sym→set_rc(*rule_info__.parser__→current_token( ), __FILE__, __LINE__);
ADD_TOKEN_TO_ERROR_QUEUE(*sym);
rule_info__.parser__→set_stop_parse(true);
```

36. *R_fsc_files* rule.

R_fsc_files



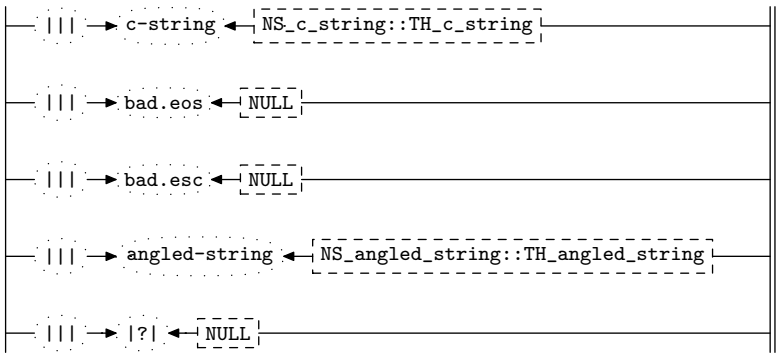
37. *R_fsc_files*'s subrule 3.



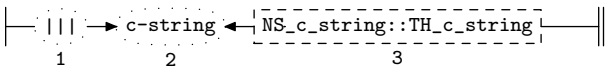
⟨ R_fsc_files subrule 3 op directive 37 ⟩ ≡
```
CAbs_lr1_sym * sym = new Err_fsc_cntl_file_not_present;
sym→set_rc(*rule_info__.parser__→current_token( ), __FILE__, __LINE__);
ADD_TOKEN_TO_ERROR_QUEUE(*sym);
rule_info__.parser__→set_stop_parse(true);
```

38. *R_fsc_file* rule.

R_fsc_file



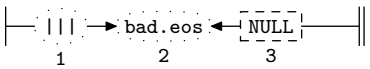
39. *R_fsc_file*'**s subrule 1.**



⟨ R_fsc_file subrule 1 op directive 39 ⟩ ≡

```
Clinker_pass3 * fsm = ( Clinker_pass3 * ) rule_info__.parser__->fsm_tbl__;
const char *fn = sf->p2__->c_string( )->c_str( );
if (fsm->chk_filename(fn, sf->p2__, new Err_fsc_file_does_not_exist) == false) return;
fsm->grammars_fsc_files_.push_back(string(fn));
sf->p2__->set_auto_delete(true);
```

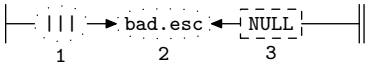
40. *R_fsc_file*'**s subrule 2.**



⟨ R_fsc_file subrule 2 op directive 40 ⟩ ≡

```
ADD_TOKEN_TO_ERROR_QUEUE(*sf->p2__);
rule_info__.parser__->set_stop_parse(true);
```

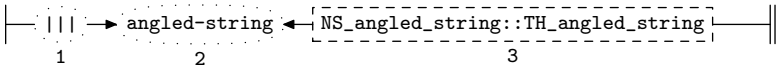
41. *R_fsc_file*'**s subrule 3.**



⟨ R_fsc_file subrule 3 op directive 41 ⟩ ≡

```
ADD_TOKEN_TO_ERROR_QUEUE(*sf->p2__);
rule_info__.parser__->set_stop_parse(true);
```

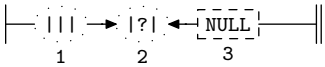
42. *R_fsc_file*'**s subrule 4.**



⟨ R_fsc_file subrule 4 op directive 42 ⟩ ≡

```
Clinker_pass3 * fsm = ( Clinker_pass3 * ) rule_info__.parser__->fsm_tbl__;
const char *fn = sf->p2__->angled_string( )->c_str( );
if (fsm->chk_filename(fn, sf->p2__, new Err_fsc_file_does_not_exist) == false) return;
fsm->grammars_fsc_files_.push_back(string(fn));
sf->p2__->set_auto_delete(true);
```

43. *R_fsc_file*'**s subrule 5.**

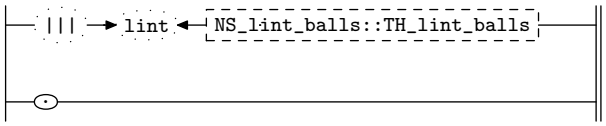


⟨ R_fsc_file subrule 5 op directive 43 ⟩ ≡

```
ADD_TOKEN_TO_ERROR_QUEUE(*sf->p2__);
rule_info__.parser__->set_stop_parse(true);
```

44. *Rlint* **rule.**

Rlint



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

```
/*
 File: linker_pass3.fsc
 Date and Time: Fri Jan  2 15:33:41 2015
*/
transitive   y
grammar-name "linker_pass3"
name-space   "NS_linker_pass3"
thread-name  "Clinker_pass3"
monolithic   y
file-name    "linker_pass3.fsc"
no-of-T      569
list-of-native-first-set-terminals 1
   LR1_questionable_shift_operator
end-list-of-native-first-set-terminals
list-of-transitive-threads 2
  NS_lint_balls::TH_lint_balls
  NS_linker_id::TH_linker_id
end-list-of-transitive-threads
list-of-used-threads 5
  NS_angled_string::TH_angled_string
  NS_c_string::TH_c_string
  NS_linker_id::TH_linker_id
  NS_linker_preamble_code::TH_linker_preamble_code
  NS_lint_balls::TH_lint_balls
end-list-of-used-threads
fsm-comments
"Lexer: constructing tokens for \\olinker parsing stage."
```

46. Lr1 State Network.

⇒ State: 1 state type: $^{s/r}$

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
c	Rlint			10	2	1	ϵ			1	0	1	1
c	Rlint			10	1	1	||| lint	NS_lint_balls::TH_lint_balls		1	2	3	
c	Rlinker_pass3			1	1	1	Rlint	*RT_alphabet*		1	4	24	

⇒\|\|\| *arbitration−code*: ϵ State: 2 state type: ^{s}

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rlint			10	1	2	lint			1	3	3	

⇒^{lint} State: 3 state type: ^{r}

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rlint			10	1	3				1	0	3	1

⇒^{Rlint} State: 4 state type: ^{s}

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
c	RT_alphabet			2	3	1	|?|			4	25	25	
c	RT_alphabet			2	1	1	|||	file-of-T-alphabet NS_linker_id::TH_linker_id		4	26	28	
c	RT_alphabet			2	2	1	||| |?|	NULL		4	26	27	
t	Rlinker_pass3			1	1	2	RT_alphabet	*Rlint*$^{\epsilon}$ *RT_alphabet_filename*		1	5	24	

⇒$^{RT\_alphabet}$ State: 5 state type: $^{s/r}$

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
c	Rlint			10	2	1	ϵ			5	0	5	2
c	Rlint			10	1	1	||| lint	NS_lint_balls::TH_lint_balls		5	2	3	
t	Rlinker_pass3			1	1	3	Rlint	*RT_alphabet_filename*		1	6	24	

⇒^{Rlint} State: 6 state type: ^{s}

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
c	RT_alphabet_filename			3	6	1	|?|			6	29	29	
c	RT_alphabet_filename			3	1	1	|||	c-string NS_c_string::TH_c_string		6	30	33	
c	RT_alphabet_filename			3	2	1	|||	bad eos NULL		6	30	34	
c	RT_alphabet_filename			3	3	1	|||	bad esc NULL		6	30	35	
c	RT_alphabet_filename			3	4	1	||| |?|	NULL		6	30	31	
c	RT_alphabet_filename			3	5	1	|||	angled-string NS_angled_string::TH_angled_string		6	30	32	
t	Rlinker_pass3			1	1	4	RT_alphabet_filename	*Rlint*$^{\epsilon}$ *Remitfile*		1	7	24	

⇒$^{RT\_alphabet\_filename}$ State: 7 state type: $^{s/r}$

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
c	Rlint			10	2	1	ϵ			7	0	7	2
c	Rlint			10	1	1	||| lint	NS_lint_balls::TH_lint_balls		7	2	3	
t	Rlinker_pass3			1	1	5	Rlint	*Remitfile*		1	8	24	

⇒^{Rlint} State: 8 state type: ^{s}

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
c	Remitfile			4	3	1	|?|			8	36	36	
c	Remitfile			4	1	1	|||	emitfile NS_linker_id::TH_linker_id		8	37	39	
c	Remitfile			4	2	1	||| |?|	NULL		8	37	38	
t	Rlinker_pass3			1	1	6	Remitfile	*Rlint*$^{\epsilon}$ *Remitfile_filename*		1	9	24	

$\Rightarrow^{Remitfile}$ State: 9 state type: $^{s/r}$

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rlint		10	2	1	ϵ		9	0	9	2
c	Rlint		10	1	1	||| lint NS_lint_balls::TH_lint_balls		9	2	3	
t	Rlinker_pass3		1	1	7	Rlint *Remitfile_filename*		1	10	24	

$\Rightarrow^{Rlint}$ State: 10 state type: ^{s}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Remitfile_filename		5	6	1	|?|		10	40	40	
c	Remitfile_filename		5	2	1	||| bad eos NULL		10	41	45	
c	Remitfile_filename		5	1	1	||| c-string NS_c_string::TH_c_string		10	41	44	
c	Remitfile_filename		5	3	1	||| bad esc NULL		10	41	46	
c	Remitfile_filename		5	4	1	||| |?| NULL		10	41	42	
c	Remitfile_filename		5	5	1	||| angled-string NS_angled_string::TH_angled_string		10	41	43	
t	Rlinker_pass3		1	1	8	Remitfile_filename *Rlint*$^{\epsilon}$ *Rpreamble*		1	11	24	

$\Rightarrow^{Remitfile\_filename}$ State: 11 state type: $^{s/r}$

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rlint		10	2	1	ϵ		11	0	11	2
c	Rlint		10	1	1	||| lint NS_lint_balls::TH_lint_balls		11	2	3	
t	Rlinker_pass3		1	1	9	Rlint *Rpreamble*		1	12	24	

$\Rightarrow^{Rlint}$ State: 12 state type: ^{s}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rpreamble		6	3	1	|?|		12	47	47	
c	Rpreamble		6	1	1	||| preamble NS_linker_id::TH_linker_id		12	48	50	
c	Rpreamble		6	2	1	||| |?| NULL		12	48	49	
t	Rlinker_pass3		1	1	10	Rpreamble *Rpreamble_code*		1	13	24	

$\Rightarrow^{Rpreamble}$ State: 13 state type: ^{s}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rpreamble_code		7	3	1	|?|		13	51	51	
c	Rpreamble_code		7	1	1	||| syntax-code NS_linker_preamble_code::TH_linker_preamble_code		13	52	54	
c	Rpreamble_code		7	2	1	||| |?| NULL		13	52	53	
t	Rlinker_pass3		1	1	11	Rpreamble_code *Rlint*$^{\epsilon}$ *R_fsc_files*		1	14	24	

$\Rightarrow^{Rpreamble\_code}$ State: 14 state type: $^{s/r}$

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rlint		10	2	1	ϵ		14	0	14	2
c	Rlint		10	1	1	||| lint NS_lint_balls::TH_lint_balls		14	2	3	
t	Rlinker_pass3		1	1	12	Rlint *R_fsc_files*		1	15	24	

$\Rightarrow^{Rlint}$ State: 15 state type: ^{s}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	R_fsc_files		8	3	1	|?|		15	55	55	
c	R_fsc_file		9	2	1	||| bad eos NULL		15	56	21	
c	R_fsc_file		9	4	1	||| angled-string NS_angled_string::TH_angled_string		15	56	19	
c	R_fsc_file		9	1	1	||| c-string NS_c_string::TH_c_string		15	56	20	
c	R_fsc_file		9	3	1	||| bad esc NULL		15	56	22	
c	R_fsc_file		9	5	1	||| |?| NULL		15	56	18	
t	Rlinker_pass3		1	1	13	R_fsc_files *Rlint*$^{\epsilon}$ *eog*		1	16	24	

c	R_fsc_files	8	2	1	R_fsc_files $\underline{R\_fsc\_file}$	15	16	58	
c	R_fsc_files	8	1	1	R_fsc_file $\underline{Rlint^{\epsilon}}$	15	59	60	

$\Rightarrow^{R\_fsc\_files}$ State: 16 state type: $^{s/r}$

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rlint		10	2	1	ϵ		16	0	16	3
c	R_fsc_file		9	2	1	||| bad eos NULL		16	17	21	
c	R_fsc_file		9	4	1	||| angled-string NS_angled_string::TH_angled_string		16	17	19	
c	Rlint		10	1	1	||| lint NS_lint_balls::TH_lint_balls		16	17	3	
c	R_fsc_file		9	1	1	||| c-string NS_c_string::TH_c_string		16	17	20	
c	R_fsc_file		9	3	1	||| bad esc NULL		16	17	22	
c	R_fsc_file		9	5	1	||| |?| NULL		16	17	18	
t	R_fsc_files		8	2	2	R_fsc_file $\underline{Rlint^{\epsilon}}$		15	57	58	
t	Rlinker_pass3		1	1	14	Rlint $\underline{eog}$		1	23	24	

$\Rightarrow$\|\|\| *arbitration−code*: ϵ State: 17 state type: ^{s}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	R_fsc_file		9	5	2	|?|		16	18	18	
t	R_fsc_file		9	4	2	angled-string		16	19	19	
t	R_fsc_file		9	1	2	c-string		16	20	20	
t	Rlint		10	1	2	lint		16	3	3	
t	R_fsc_file		9	2	2	bad eos		16	21	21	
t	R_fsc_file		9	3	2	bad esc		16	22	22	

$\Rightarrow$\|?\| State: 18 state type: ^{r}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	R_fsc_file		9	5	3			16	0	18	4

$\Rightarrow^{angled-string}$ State: 19 state type: ^{r}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	R_fsc_file		9	4	3			16	0	19	4

$\Rightarrow^{c-string}$ State: 20 state type: ^{r}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	R_fsc_file		9	1	3			16	0	20	4

$\Rightarrow^{badeos}$ State: 21 state type: ^{r}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	R_fsc_file		9	2	3			16	0	21	4

$\Rightarrow^{badesc}$ State: 22 state type: ^{r}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	R_fsc_file		9	3	3			16	0	22	4

$\Rightarrow^{Rlint}$ State: 23 state type: ^{s}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rlinker_pass3		1	1	15	eog		1	24	24	

$\Rightarrow^{eog}$ State: 24 state type: ^{r}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rlinker_pass3		1	1	16			1	0	24	5

⇒`|?|` State: 25 state type: ^{r}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	RT_alphabet		2	3	2				4	0	25	2

⇒`|||` *arbitration−code*: ϵ State: 26 state type: ^{s}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	RT_alphabet		2	2	2	`|?|`			4	27	27	
t	RT_alphabet		2	1	2	file-of-T-alphabet			4	28	28	

⇒`|?|` State: 27 state type: ^{r}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	RT_alphabet		2	2	3				4	0	27	2

⇒*file−of−T−alphabet* State: 28 state type: ^{r}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	RT_alphabet		2	1	3				4	0	28	2

⇒`|?|` State: 29 state type: ^{r}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	RT_alphabet_filename		3	6	2				6	0	29	2

⇒`|||` *arbitration−code*: ϵ State: 30 state type: ^{s}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	RT_alphabet_filename		3	4	2	`|?|`			6	31	31	
t	RT_alphabet_filename		3	5	2	angled-string			6	32	32	
t	RT_alphabet_filename		3	1	2	c-string			6	33	33	
t	RT_alphabet_filename		3	2	2	bad eos			6	34	34	
t	RT_alphabet_filename		3	3	2	bad esc			6	35	35	

⇒`|?|` State: 31 state type: ^{r}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	RT_alphabet_filename		3	4	3				6	0	31	2

⇒*angled−string* State: 32 state type: ^{r}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	RT_alphabet_filename		3	5	3				6	0	32	2

⇒*c−string* State: 33 state type: ^{r}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	RT_alphabet_filename		3	1	3				6	0	33	2

⇒*badeos* State: 34 state type: ^{r}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	RT_alphabet_filename		3	2	3				6	0	34	2

⇒*badesc* State: 35 state type: ^{r}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	RT_alphabet_filename		3	3	3				6	0	35	2

⇒`|?|` State: 36 state type: ^{r}

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA

t	Remitfile	4	3	2			8	0	36	2

⇒||| *arbitration−code*: ϵ State: 37 state type: *s*

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Remitfile		4	2	2	|?|			8	38	38	
t	Remitfile		4	1	2	emitfile			8	39	39	

⇒|?| State: 38 state type: *r*

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Remitfile		4	2	3				8	0	38	2

⇒*emitfile* State: 39 state type: *r*

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Remitfile		4	1	3				8	0	39	2

⇒|?| State: 40 state type: *r*

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Remitfile_filename		5	6	2				10	0	40	2

⇒||| *arbitration−code*: ϵ State: 41 state type: *s*

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Remitfile_filename		5	4	2	|?|			10	42	42	
t	Remitfile_filename		5	5	2	angled-string			10	43	43	
t	Remitfile_filename		5	1	2	c-string			10	44	44	
t	Remitfile_filename		5	2	2	bad eos			10	45	45	
t	Remitfile_filename		5	3	2	bad esc			10	46	46	

⇒|?| State: 42 state type: *r*

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Remitfile_filename		5	4	3				10	0	42	2

⇒*angled−string* State: 43 state type: *r*

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Remitfile_filename		5	5	3				10	0	43	2

⇒*c−string* State: 44 state type: *r*

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Remitfile_filename		5	1	3				10	0	44	2

⇒*badeos* State: 45 state type: *r*

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Remitfile_filename		5	2	3				10	0	45	2

⇒*badesc* State: 46 state type: *r*

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Remitfile_filename		5	3	3				10	0	46	2

⇒|?| State: 47 state type: *r*

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rpreamble		6	3	2				12	0	47	2

⇒||| *arbitration−code*: ϵ State: 48 state type: *s*

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rpreamble			6	2	2	|?|			12	49	49	
t	Rpreamble			6	1	2	preamble			12	50	50	

⇒\|?\| State: 49 state type: ^{r}

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rpreamble			6	2	3				12	0	49	2

⇒$preamble$ State: 50 state type: ^{r}

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rpreamble			6	1	3				12	0	50	2

⇒\|?\| State: 51 state type: ^{r}

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rpreamble_code			7	3	2				13	0	51	2

⇒\|\|\| $arbitration-code$: ϵ State: 52 state type: ^{s}

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rpreamble_code			7	2	2	|?|			13	53	53	
t	Rpreamble_code			7	1	2	syntax-code			13	54	54	

⇒\|?\| State: 53 state type: ^{r}

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rpreamble_code			7	2	3				13	0	53	2

⇒$syntax-code$ State: 54 state type: ^{r}

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rpreamble_code			7	1	3				13	0	54	2

⇒\|?\| State: 55 state type: ^{r}

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	R_fsc_files			8	3	2				15	0	55	4

⇒\|\|\| $arbitration-code$: ϵ State: 56 state type: ^{s}

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	R_fsc_file			9	5	2	|?|			15	18	18	
t	R_fsc_file			9	4	2	angled-string			15	19	19	
t	R_fsc_file			9	1	2	c-string			15	20	20	
t	R_fsc_file			9	2	2	bad eos			15	21	21	
t	R_fsc_file			9	3	2	bad esc			15	22	22	

⇒R_fsc_file State: 57 state type: $^{s/r}$

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
c	Rlint			10	2	1	ϵ			57	0	57	4
c	Rlint			10	1	1	||| lint NS_lint_balls::TH_lint_balls			57	2	3	
t	R_fsc_files			8	2	3	Rlint			15	58	58	

⇒$Rlint$ State: 58 state type: ^{r}

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	R_fsc_files			8	2	4				15	0	58	4

⇒R_fsc_file State: 59 state type: $^{s/r}$

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rlint		10	2	1	ϵ		59	0	59	4
c	Rlint		10	1	1	||| lint NS_lint_balls::TH_lint_balls		59	2	3	
t	R_fsc_files		8	1	2	Rlint		15	60	60	

$\Rightarrow^{Rlint}$ State: 60 state type: ^{r}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	R_fsc_files		8	1	3			15	0	60	4

47. Index.

ϵ : 44.
|||: 9, 12, 19, 22, 29, 32, 38, 44.
|?|: 9, 12, 19, 22, 29, 32, 36, 38.
__FILE__: 5, 10, 11, 16, 18, 20, 21, 26, 28, 30, 31, 35, 37.
__LINE__: 5, 10, 11, 16, 18, 20, 21, 26, 28, 30, 31, 35, 37.
add_token_to_error_queue: 5.
ADD_TOKEN_TO_ERROR_QUEUE: 10, 11, 14, 15, 16, 18, 20, 21, 24, 25, 26, 28, 30, 31, 34, 35, 37, 40, 41, 43.
ADD_TOKEN_TO_PRODUCER_QUEUE: 8.
angled-string: 12, 22, 38.
angled_string: 17, 27, 42.
bad eos: 12, 22, 38.
bad esc: 12, 22, 38.
c-string: 12, 22, 38.
c_str: 13, 17, 23, 27, 39, 42.
c_string: 13, 23, 39.
CAbs_lr1_sym: 4, 5, 10, 11, 16, 18, 20, 21, 26, 28, 30, 31, 35, 37.
chk_filename: 4, 5, 13, 17, 39, 42.
chk_for_overrun: 4, 5.
chk_ofilename: 4, 5, 23, 27.
Clinker_pass3: 5, 13, 17, 23, 27, 33, 39, 42.
close: 5.
current_token: 5, 11, 18, 21, 28, 31, 35, 37.
*directories_*: 4.
emitfile: 19.
*emitfile_filename_*: 4, 23, 27.
enumerated_id__: 5.
eog: 7.
Err: 4, 5.
Err_bad_filename: 5.
Err_emitfile_file_not_present: 26, 28.
Err_emitfile_kw_not_present: 20, 21.
Err_end_preamble_kw_not_present: 5.
Err_file_of_T_alphabet_kw_not_present: 10, 11.
Err_fsc_cntl_file_not_present: 37.
Err_fsc_file_does_not_exist: 39, 42.
Err_no_filename_present: 16.
Err_preamble_kw_not_present: 30, 31.
Err_preamble_srce_code_not_present: 35.
Err_T_alphabet_file_does_not_exist: 13, 17.
Err_T_alphabet_file_not_present: 18.
false: 5, 13, 17, 23, 27, 39, 42.
file-of-T-alphabet: 9.
Filename: 4, 5.
fle_chk: 5.
fn: 13, 17, 23, 27, 39, 42.
fsm: 13, 17, 23, 27, 33, 39, 42.
fsm_tbl__: 13, 17, 23, 27, 33, 39, 42.
good: 5.
*grammars_fsc_files_*: 4, 39, 42.
ifstream: 5.
ios: 5.
linker_pass3: 2.
lint: 44.
NS_angled_string::TH_angled_string: 12, 22, 38.
NS_c_string::TH_c_string: 12, 22, 38.
NS_linker_id::TH_linker_id: 9, 19, 29.
NS_linker_preamble_code::TH_linker_preamble_code: 32.
NS_lint_balls::TH_lint_balls: 44.
NS_yacco2_k_symbols: 8.
NS_yacco2_terminals: 6.
NULL: 9, 12, 19, 22, 29, 32, 38.
ofstream: 5.
open: 5.
out: 5.
parser__: 5, 10, 11, 13, 14, 15, 16, 17, 18, 20, 21, 23, 24, 25, 26, 27, 28, 30, 31, 33, 34, 35, 37, 39, 40, 41, 42, 43.
preamble: 29.
*preamble_srce_*: 4, 33.
PTR_LR1_eog__: 8.
push_back: 39, 42.
p2__: 10, 13, 14, 15, 16, 17, 20, 23, 24, 25, 26, 27, 30, 33, 34, 39, 40, 41, 42, 43.
R_fsc_file: 36.
R_fsc_files: 7, 36.
R_fsc_file: 38, 39, 40, 41, 42, 43.
R_fsc_files: 36, 37.
Remitfile: 19, 20, 21.
Remitfile: 7.
Remitfile_filename: 7.
Remitfile_filename: 22, 23, 24, 25, 26, 27, 28.
Rlinker_pass3: 7.
Rlint: 44.
Rlint: 7, 36.
Rpreamble: 29, 30, 31.
Rpreamble: 7.
Rpreamble_code: 7.
Rpreamble_code: 32, 33, 34, 35.
RT_alphabet: 7.
RT_alphabet_filename: 7.
RT_alphabet: 9, 10, 11.
RT_alphabet_filename: 12, 13, 14, 15, 16, 17, 18.
rule_info__: 10, 11, 13, 14, 15, 16, 17, 18, 20, 21, 23, 24, 25, 26, 27, 28, 30, 31, 33, 34, 35, 37, 39, 40, 41, 42, 43.

set_auto_delete: 10, 13, 16, 17, 20, 23, 26, 27, 30, 39, 42.
set_rc: 5, 10, 11, 16, 18, 20, 21, 26, 28, 30, 31, 35, 37.
set_stop_parse: 5, 10, 11, 14, 15, 16, 18, 20, 21, 24, 25, 26, 28, 30, 31, 34, 35, 37, 40, 41, 43.
set_use_all_shift_off: 5.
sf: 10, 13, 14, 15, 16, 17, 20, 23, 24, 25, 26, 27, 30, 33, 34, 39, 40, 41, 42, 43.
start_token__: 5.
std: 4, 5.
string: 4, 39, 42.
sym: 5, 10, 11, 16, 18, 20, 21, 26, 28, 30, 31, 35, 37.
`syntax-code`: 32.
*t_alphabet_filename_*: 4, 13, 17.
T_Enum: 5.
*T_LR1_eog_*: 5.
T_syntax_code: 4.
Tok: 4, 5.
true: 5, 10, 11, 13, 14, 15, 16, 17, 18, 20, 21, 23, 24, 25, 26, 27, 28, 30, 31, 34, 35, 37, 39, 40, 41, 42, 43.
vector: 4.
yacco2: 4, 5, 8.

⟨ Clinker_pass3 user-declaration directive 4 ⟩
⟨ Clinker_pass3 user-implementation directive 5 ⟩
⟨ Clinker_pass3 user-prefix-declaration directive 6 ⟩
⟨ RT_alphabet subrule 2 op directive 10 ⟩
⟨ RT_alphabet subrule 3 op directive 11 ⟩
⟨ RT_alphabet_filename subrule 1 op directive 13 ⟩
⟨ RT_alphabet_filename subrule 2 op directive 14 ⟩
⟨ RT_alphabet_filename subrule 3 op directive 15 ⟩
⟨ RT_alphabet_filename subrule 4 op directive 16 ⟩
⟨ RT_alphabet_filename subrule 5 op directive 17 ⟩
⟨ RT_alphabet_filename subrule 6 op directive 18 ⟩
⟨ R_fsc_file subrule 1 op directive 39 ⟩
⟨ R_fsc_file subrule 2 op directive 40 ⟩
⟨ R_fsc_file subrule 3 op directive 41 ⟩
⟨ R_fsc_file subrule 4 op directive 42 ⟩
⟨ R_fsc_file subrule 5 op directive 43 ⟩
⟨ R_fsc_files subrule 3 op directive 37 ⟩
⟨ Remitfile subrule 2 op directive 20 ⟩
⟨ Remitfile subrule 3 op directive 21 ⟩
⟨ Remitfile_filename subrule 1 op directive 23 ⟩
⟨ Remitfile_filename subrule 2 op directive 24 ⟩
⟨ Remitfile_filename subrule 3 op directive 25 ⟩
⟨ Remitfile_filename subrule 4 op directive 26 ⟩
⟨ Remitfile_filename subrule 5 op directive 27 ⟩
⟨ Remitfile_filename subrule 6 op directive 28 ⟩
⟨ Rlinker_pass3 op directive 8 ⟩
⟨ Rpreamble subrule 2 op directive 30 ⟩
⟨ Rpreamble subrule 3 op directive 31 ⟩
⟨ Rpreamble_code subrule 1 op directive 33 ⟩
⟨ Rpreamble_code subrule 2 op directive 34 ⟩
⟨ Rpreamble_code subrule 3 op directive 35 ⟩

linker_pass3 Grammar

Date: January 2, 2015 at 15:36

File: linker_pass3.lex

Ns: NS_linker_pass3

Version: 1.0 Debug: false

Grammar Comments: Type: Monolithic

Lexer: constructing tokens for O_2^{linker} parsing stage.

Section Page

Copyright 1 1

linker_pass3 **grammar** 2 2
Fsm Clinker_pass3 class 3 2
Clinker_pass3 user-declaration directive 4 2
Clinker_pass3 user-implementation directive 5 3
Clinker_pass3 user-prefix-declaration directive 6 4
Rlinker_pass3 rule 7 4
Rlinker_pass3 op directive 8 4
RT_alphabet rule 9 4
RT_alphabet's subrule 2 . . . . . 10 4
RT_alphabet's subrule 3 . . . . . 11 5
RT_alphabet_filename rule 12 5
RT_alphabet_filename's subrule 1 . . . . . 13 5
RT_alphabet_filename's subrule 2 . . . . . 14 5
RT_alphabet_filename's subrule 3 . . . . . 15 6
RT_alphabet_filename's subrule 4 . . . . . 16 6
RT_alphabet_filename's subrule 5 . . . . . 17 6
RT_alphabet_filename's subrule 6 . . . . . 18 6
Remitfile rule 19 6
Remitfile's subrule 2 . . . . . 20 7
Remitfile's subrule 3 . . . . . 21 7
Remitfile_filename rule 22 7
Remitfile_filename's subrule 1 . . . . . 23 7
Remitfile_filename's subrule 2 . . . . . 24 8
Remitfile_filename's subrule 3 . . . . . 25 8
Remitfile_filename's subrule 4 . . . . . 26 8
Remitfile_filename's subrule 5 . . . . . 27 8
Remitfile_filename's subrule 6 . . . . . 28 8
Rpreamble rule 29 9
Rpreamble's subrule 2 . . . . . 30 9
Rpreamble's subrule 3 . . . . . 31 9
Rpreamble_code rule 32 9
Rpreamble_code's subrule 1 . . . . . 33 9
Rpreamble_code's subrule 2 . . . . . 34 10
Rpreamble_code's subrule 3 . . . . . 35 10
R_fsc_files rule 36 10
R_fsc_files's subrule 3 . . . . . 37 10
R_fsc_file rule 38 10
R_fsc_file's subrule 1 . . . . . 39 11
R_fsc_file's subrule 2 . . . . . 40 11
R_fsc_file's subrule 3 . . . . . 41 11
R_fsc_file's subrule 4 . . . . . 42 11
R_fsc_file's subrule 5 . . . . . 43 11
Rlint rule 44 12
First Set Language for O_2^{linker} 45 13
Lr1 State Network 46 14
Index 47 21