1. Copyright.

2. *fsc_file* **stand alone Grammar.**
Parses the "first set control" files outputed from each grammar. This is the fodder of O_2's linker to generate the first sets per thread. It puts the grammar's attributes into yacco2's symbol table and into a global dictionary of threads for consolidation processing. See O_2linker documentation on the real lowdown.

Linker's control language.

```
file-of-T-alphabet "c:/yacco2/compiler/grammars/yacco2_T_enumeration.fsc"
emitfile "/yacco2/compiler/grammars/yacco2_fsc.cpp"
preamble
#include <yacco2.h>
#include <yacco2_T_enumeration.h>
#include <yacco2_err_symbols.h>
#include <yacco2_k_symbols.h>
#include <yacco2_terminals.h>
#include <yacco2_characters.h>
using namespace NS_yacco2_T_enum;
using namespace NS_yacco2_err_symbols;
using namespace NS_yacco2_k_symbols;
using namespace NS_yacco2_terminals;
using namespace NS_yacco2_characters;
end-preamble
"/yacco2/compiler/grammars/error_symbols_phrase.fsc"
"/yacco2/compiler/grammars/error_symbols_phrase_th.fsc"
"/yacco2/compiler/grammars/angled_string.fsc"
"/yacco2/compiler/grammars/bad_char_set.fsc"
"/yacco2/compiler/grammars/c_comments.fsc"
                    ...
```

3. Fsm Cfsc_file class.

4. Cfsc_file user-declaration directive.

⟨ Cfsc_file user-declaration directive 4 ⟩ ≡
public: *std* :: *string fully_qualified_th_name_*;
char *transitive_*;
char *monolithic_*;
NS_yacco2_terminals :: *T_c_string* ∗ *grammar_file_name_*;
NS_yacco2_terminals :: *T_c_string* ∗ *name_space_name_*;
NS_yacco2_terminals :: *T_c_string* ∗ *thread_name_*;
NS_yacco2_terminals :: *T_c_string* ∗ *file_name_*;
NS_yacco2_terminals :: *T_c_string* ∗ *fsm_comments_*;
NS_yacco2_terminals :: *T_int_no* ∗ *no_of_T_*; *std* :: *vector* < **int** > *list_of_Ts_*;
NS_yacco2_terminals :: *T_int_no* ∗ *no_native_T_*;
std :: *vector* < *NS_yacco2_terminals* :: *thread_attributes* ∗> *list_of_transitive_threads_*;
std :: *vector* < *NS_yacco2_terminals* :: *thread_attributes* ∗> *list_of_used_threads_*;
NS_yacco2_terminals :: *T_int_no* ∗ *no_threads_*;
NS_yacco2_terminals :: *T_int_no* ∗ *used_no_threads_*;
std :: *vector* < *std* :: *string* > ∗*used_threads_*;

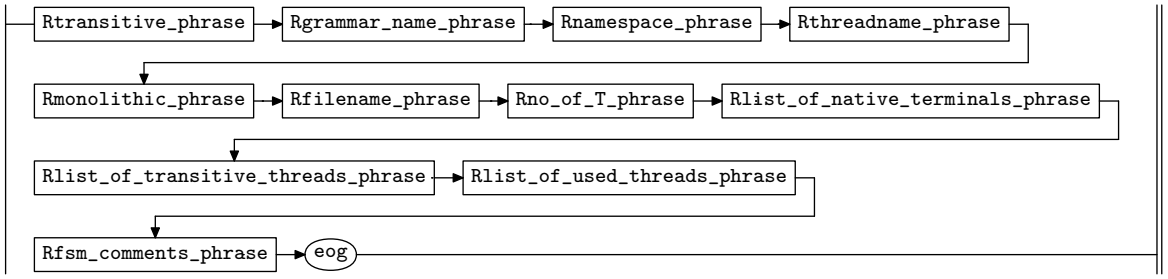
5. Cfsc_file user-prefix-declaration directive.

⟨ Cfsc_file user-prefix-declaration directive 5 ⟩ ≡

```
#include "yacco2_stbl.h"
  extern std::vector < NS_yacco2_terminals::table_entry *> GRAMMAR_DICTIONARY;
  extern std::vector < NS_yacco2_terminals::table_entry *> T_DICTIONARY;
  extern std::map < std::string, std::vector < std::string >> USED_THREADS_LIST;
```

6. *Rfsc_file* rule.

Rfsc_file



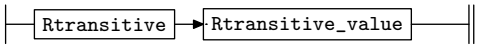
7. Rfsc_file op directive.

```
⟨ Rfsc_file op directive 7 ⟩ ≡
  Cfsc_file * fsm = ( Cfsc_file * ) rule_info__.parser__→fsm_tbl__;
  using namespace NS_yacco2_k_symbols;
  int no_T = fsm→no_native_T_→no( );
  int sze_T = fsm→list_of_Ts_.size( );
  if (sze_T ≠ no_T) {
    CAbs_lr1_sym * sym = new Err_no_of_native_Ts_in_list_not_equal(sze_T);
    sym→set_rc(*fsm→no_native_T_, __FILE__, __LINE__);
    ADD_TOKEN_TO_ERROR_QUEUE(*sym);
    return;
  }
  int no_th = fsm→no_threads_→no( );
  int sze_th = fsm→list_of_transitive_threads_.size( );
  if (sze_th ≠ no_th) {
    CAbs_lr1_sym * sym = new Err_no_of_threads_in_list_not_equal(sze_th);
    sym→set_rc(*fsm→no_threads_, __FILE__, __LINE__);
    ADD_TOKEN_TO_ERROR_QUEUE(*sym);
    return;
  }
  if (fsm→monolithic_ ≡ 'n') {      /* pass epsilon pass thru: thread grammar only */
    if ((no_T ≡ 0) ∧ (no_th ≡ 0)) {
      CAbs_lr1_sym * sym = new Err_epsilon_pass_thru;
      sym→set_rc(*fsm→no_native_T_, __FILE__, __LINE__);
      ADD_TOKEN_TO_ERROR_QUEUE(*sym);
      rule_info__.parser__→set_stop_parse(true);
      return;
    }
    if ((no_T ≡ 1) ∧ (no_th ≡ 0)) {
      int transience = fsm→list_of_Ts_[0];
      if (transience ≡ 7) {      /* |t| operator */
        CAbs_lr1_sym * sym = new Err_epsilon_pass_thru_no_Ts;
        sym→set_rc(*fsm→no_native_T_, __FILE__, __LINE__);
        ADD_TOKEN_TO_ERROR_QUEUE(*sym);
        rule_info__.parser__→set_stop_parse(true);
        return;
      }
    }
  }
  using namespace yacco2_stbl;
  T_sym_tbl_report_card report_card;
  find_sym_in_stbl(report_card,
      *fsm→fully_qualified_th_name_.c_str( )); if (report_card.action_ ≡ T_sym_tbl_report_card::fnd) {
    /* crted by another fsc file */
  if (report_card.tbl_entry_→defined_ ≡ true) {      /* probably dup fsc file */
    CAbs_lr1_sym * sym = new Err_already_defined_in_fsc_file;
    sym→set_rc(*fsm→thread_name_, __FILE__, __LINE__);
    ADD_TOKEN_TO_ERROR_QUEUE(*sym);
    rule_info__.parser__→set_stop_parse(true);
    return;
```

```
}
report_card.tbl_entry_→defined_ = true; th_in_stbl * th_in_tbl = ( th_in_stbl * ) report_card.tbl_entry_→symbol_;
thread_attributes * th_entry = th_in_tbl→thread_in_stbl( );
th_entry→transitive_ = fsm→transitive_;
th_entry→grammar_file_name_ = fsm→grammar_file_name_;
th_entry→name_space_name_ = fsm→name_space_name_;
th_entry→thread_name_ = fsm→thread_name_;
th_entry→monolithic_ = fsm→monolithic_;
th_entry→file_name_ = fsm→file_name_;
th_entry→fsm_comments_ = fsm→fsm_comments_;
if (¬fsm→list_of_Ts_.empty( ))
  copy(fsm→list_of_Ts_.begin( ), fsm→list_of_Ts_.end( ), back_inserter(th_entry→list_of_Ts_));
if (¬fsm→list_of_transitive_threads_.empty( )) copy(fsm→list_of_transitive_threads_.begin( ),
      fsm→list_of_transitive_threads_.end( ), back_inserter(th_entry→list_of_transitive_threads_));
return; } thread_attributes * th_id = new thread_attributes(fsm→fully_qualified_th_name_.c_str( ),
    fsm→transitive_, fsm→grammar_file_name_, fsm→name_space_name_, fsm→thread_name_,
    fsm→monolithic_, fsm→file_name_, fsm→list_of_Ts_, fsm→list_of_transitive_threads_, fsm→fsm_comments_);
th_in_stbl * t = new th_in_stbl(th_id);
add_sym_to_stbl(report_card, *fsm→fully_qualified_th_name_.c_str( ), *t, table_entry::defed,
    table_entry::thread);
if (report_card.status_ ≠ T_sym_tbl_report_card::okay) {
  t→stbl_idx(report_card.pos_);
  report_card.err_entry_→set_rc(*th_id, __FILE__, __LINE__);
  ADD_TOKEN_TO_ERROR_QUEUE(*report_card.err_entry_);
  rule_info__.parser__→set_stop_parse(true);
  return;
}
GRAMMAR_DICTIONARY.push_back(report_card.tbl_entry_);
th_id→th_enum_ = GRAMMAR_DICTIONARY.size( ) − 1;
ADD_TOKEN_TO_PRODUCER_QUEUE(*yacco2::PTR_LR1_eog__);
ADD_TOKEN_TO_PRODUCER_QUEUE(*yacco2::PTR_LR1_eog__);
```

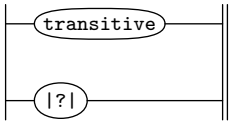
8. *Rtransitive_phrase* **rule.**

```
Rtransitive_phrase
```

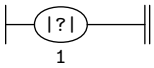


9. *Rtransitive* **rule.**

```
Rtransitive
```



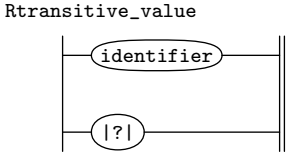
10. *Rtransitive*'**s subrule 2.**



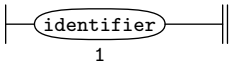
⟨ Rtransitive subrule 2 op directive 10 ⟩ ≡

```
CAbs_lr1_sym * sym = new Err_transitive_kw_not_present;
sym->set_rc(*sf->p1__, __FILE__, __LINE__);
ADD_TOKEN_TO_ERROR_QUEUE(*sym);
rule_info__.parser__->set_stop_parse(true);
```

11. *Rtransitive_value* **rule.**



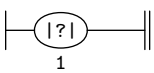
12. *Rtransitive_value*'**s subrule 1.**



⟨ Rtransitive_value subrule 1 op directive 12 ⟩ ≡

```
Cfsc_file * fsm = ( Cfsc_file * ) rule_info__.parser__->fsm_tbl__;
T_identifier * k = sf->p1__;
if (k->identifier( )->size( ) ≠ 1) {
  CAbs_lr1_sym * sym = new Err_transitive_value_bad;
  sym->set_rc(*sf->p1__, __FILE__, __LINE__);
  ADD_TOKEN_TO_ERROR_QUEUE(*sym);
  rule_info__.parser__->set_stop_parse(true);
  return;
}
char c = (*k->identifier( ))[0];
if (c ≡ 'n' ∨ c ≡ 'y') {
  fsm->transitive_ = c;
  return;
}
CAbs_lr1_sym * sym = new Err_transitive_value_bad;
sym->set_rc(*sf->p1__, __FILE__, __LINE__);
ADD_TOKEN_TO_ERROR_QUEUE(*sym);
rule_info__.parser__->set_stop_parse(true);
```

13. *Rtransitive_value*'**s subrule 2.**

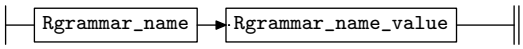


⟨ Rtransitive_value subrule 2 op directive 13 ⟩ ≡

```
CAbs_lr1_sym * sym = new Err_transitive_value_bad;
sym->set_rc(*sf->p1__, __FILE__, __LINE__);
ADD_TOKEN_TO_ERROR_QUEUE(*sym);
rule_info__.parser__->set_stop_parse(true);
```

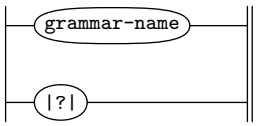
14. *Rgrammar_name_phrase* **rule.**

Rgrammar_name_phrase

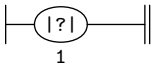


15. *Rgrammar_name* **rule.**

Rgrammar_name



16. *Rgrammar_name*'**s subrule 2.**

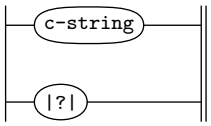


⟨ Rgrammar_name subrule 2 op directive 16 ⟩ ≡

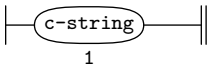
```
CAbs_lr1_sym * sym = new Err_grammar_name_kw_not_present;
sym->set_rc(*sf->p1__, __FILE__, __LINE__);
ADD_TOKEN_TO_ERROR_QUEUE(*sym);
rule_info__.parser__->set_stop_parse(true);
```

17. *Rgrammar_name_value* **rule.**

Rgrammar_name_value



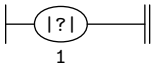
18. *Rgrammar_name_value*'**s subrule 1.**



⟨ Rgrammar_name_value subrule 1 op directive 18 ⟩ ≡

```
Cfsc_file * fsm = ( Cfsc_file * ) rule_info__.parser__->fsm_tbl__;
fsm->grammar_file_name_ = sf->p1__;
```

19. *Rgrammar_name_value*'**s subrule 2.**

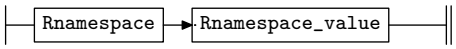


⟨ Rgrammar_name_value subrule 2 op directive 19 ⟩ ≡

```
CAbs_lr1_sym * sym = new Err_grammar_name_value_not_present;
sym->set_rc(*sf->p1__, __FILE__, __LINE__);
ADD_TOKEN_TO_ERROR_QUEUE(*sym);
rule_info__.parser__->set_stop_parse(true);
```

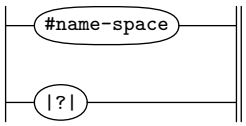
20. *Rnamespace_phrase* **rule.**

Rnamespace_phrase

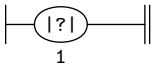


21. *Rnamespace* **rule.**

Rnamespace



22. *Rnamespace*'s **subrule 2.**

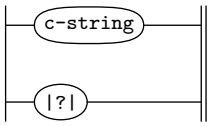


⟨ Rnamespace subrule 2 op directive 22 ⟩ ≡

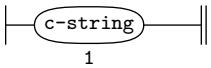
```
CAbs_lr1_sym * sym = new Err_namespace_kw_not_present;
sym->set_rc(*sf->p1__, __FILE__, __LINE__);
ADD_TOKEN_TO_ERROR_QUEUE(*sym);
rule_info__.parser__->set_stop_parse(true);
```

23. *Rnamespace_value* **rule.**

Rnamespace_value



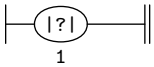
24. *Rnamespace_value*'s **subrule 1.**



⟨ Rnamespace_value subrule 1 op directive 24 ⟩ ≡

```
Cfsc_file * fsm = ( Cfsc_file * ) rule_info__.parser__->fsm_tbl__;
fsm->name_space_name_ = sf->p1__;
```

25. *Rnamespace_value*'s **subrule 2.**

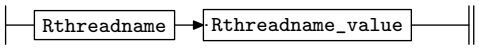


⟨ Rnamespace_value subrule 2 op directive 25 ⟩ ≡

```
CAbs_lr1_sym * sym = new Err_namespace_value_not_present;
sym->set_rc(*sf->p1__, __FILE__, __LINE__);
ADD_TOKEN_TO_ERROR_QUEUE(*sym);
rule_info__.parser__->set_stop_parse(true);
```

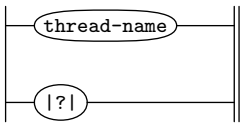
26. *Rthreadname_phrase* **rule.**

Rthreadname_phrase

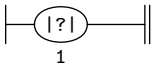


27. *Rthreadname* **rule.**

Rthreadname



28. *Rthreadname*'**s subrule 2.**

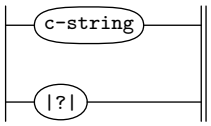


⟨ Rthreadname subrule 2 op directive 28 ⟩ ≡

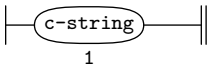
```
CAbs_lr1_sym * sym = new Err_threadname_kw_not_present;
sym->set_rc(*sf->p1__, __FILE__, __LINE__);
ADD_TOKEN_TO_ERROR_QUEUE(*sym);
rule_info__.parser__->set_stop_parse(true);
```

29. *Rthreadname_value* **rule.**

Rthreadname_value



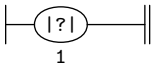
30. *Rthreadname_value*'**s subrule 1.**



⟨ Rthreadname_value subrule 1 op directive 30 ⟩ ≡

```
Cfsc_file * fsm = ( Cfsc_file * ) rule_info__.parser__->fsm_tbl__;
fsm->thread_name_ = sf->p1__;
fsm->fully_qualified_th_name_ += fsm->name_space_name_->c_string( )->c_str( );
fsm->fully_qualified_th_name_ += "::";
fsm->fully_qualified_th_name_ += fsm->thread_name_->c_string( )->c_str( );
T_identifier * th_id = new T_identifier(fsm->fully_qualified_th_name_.c_str( ));
th_id->set_rc(*fsm->thread_name_, __FILE__, __LINE__);
string gnm(sf->p1__->c_string( )->c_str( ));
USED_THREADS_LIST[gnm] = std::vector < string > ( );
std::map < std::string, std::vector < std::string >> ::iterator i = USED_THREADS_LIST.find(gnm);
fsm->used_threads_ = &i->second;
```

31. *Rthreadname_value*'**s subrule 2.**



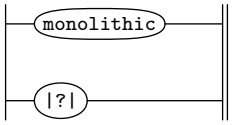
⟨ Rthreadname_value subrule 2 op directive 31 ⟩ ≡
```
CAbs_lr1_sym * sym = new Err_threadname_value_not_present;
sym->set_rc(*sf->p1__, __FILE__, __LINE__);
ADD_TOKEN_TO_ERROR_QUEUE(*sym);
rule_info__.parser__->set_stop_parse(true);
```

32. *Rmonolithic_phrase* **rule.**

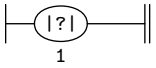
Rmonolithic_phrase

33. *Rmonolithic* **rule.**

Rmonolithic



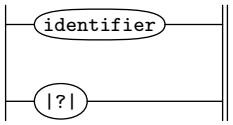
34. *Rmonolithic*'**s subrule 2.**



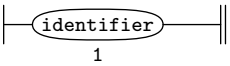
⟨ Rmonolithic subrule 2 op directive 34 ⟩ ≡
```
CAbs_lr1_sym * sym = new Err_monolithic_kw_not_present;
sym->set_rc(*sf->p1__, __FILE__, __LINE__);
ADD_TOKEN_TO_ERROR_QUEUE(*sym);
rule_info__.parser__->set_stop_parse(true);
```

35. *Rmonolithic_value* **rule.**

Rmonolithic_value



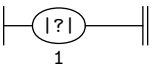
36. *Rmonolithic_value*'s subrule 1.



⟨ Rmonolithic_value subrule 1 op directive 36 ⟩ ≡

```
Cfsc_file * fsm = ( Cfsc_file * ) rule_info__.parser__→fsm_tbl__;
T_identifier * k = sf→p1__;
if (k→identifier()→size() ≠ 1) {
  CAbs_lr1_sym * sym = new Err_monolithic_value_bad;
  sym→set_rc(*sf→p1__, __FILE__, __LINE__);
  ADD_TOKEN_TO_ERROR_QUEUE(*sym);
  rule_info__.parser__→set_stop_parse(true);
  return;
}
char c = (*k→identifier())[0];
if (c ≡ 'n' ∨ c ≡ 'y') {
  fsm→monolithic_ = c;
  return;
}
CAbs_lr1_sym * sym = new Err_monolithic_value_bad;
sym→set_rc(*sf→p1__, __FILE__, __LINE__);
ADD_TOKEN_TO_ERROR_QUEUE(*sym);
rule_info__.parser__→set_stop_parse(true);
```

37. *Rmonolithic_value*'s subrule 2.

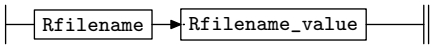


⟨ Rmonolithic_value subrule 2 op directive 37 ⟩ ≡

```
CAbs_lr1_sym * sym = new Err_monolithic_value_bad;
sym→set_rc(*sf→p1__, __FILE__, __LINE__);
ADD_TOKEN_TO_ERROR_QUEUE(*sym);
rule_info__.parser__→set_stop_parse(true);
```

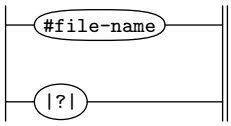
38. *Rfilename_phrase* **rule.**

Rfilename_phrase

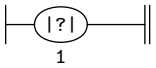


39. *Rfilename* **rule.**

Rfilename



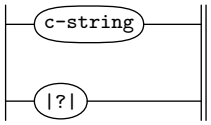
40. *Rfilename*'**s subrule 2.**



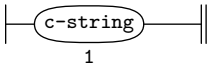
⟨ Rfilename subrule 2 op directive 40 ⟩ ≡
```
CAbs_lr1_sym * sym = new Err_filename_kw_not_present;
sym->set_rc(*sf->p1__, __FILE__, __LINE__);
ADD_TOKEN_TO_ERROR_QUEUE(*sym);
rule_info__.parser__->set_stop_parse(true);
```

41. *Rfilename_value* **rule.**

Rfilename_value

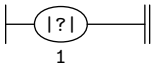


42. *Rfilename_value*'**s subrule 1.**



⟨ Rfilename_value subrule 1 op directive 42 ⟩ ≡
```
Cfsc_file * fsm = ( Cfsc_file * ) rule_info__.parser__->fsm_tbl__;
fsm->file_name_ = sf->p1__;
```

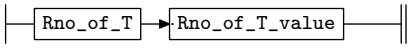
43. *Rfilename_value*'**s subrule 2.**



⟨ Rfilename_value subrule 2 op directive 43 ⟩ ≡
```
CAbs_lr1_sym * sym = new Err_filename_value_not_present;
sym->set_rc(*sf->p1__, __FILE__, __LINE__);
ADD_TOKEN_TO_ERROR_QUEUE(*sym);
rule_info__.parser__->set_stop_parse(true);
```

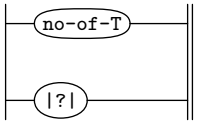
44. *Rno_of_T_phrase* **rule.**

Rno_of_T_phrase

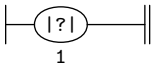


45. *Rno_of_T* **rule.**

Rno_of_T



46. *Rno_of_T*'**s subrule 2.**

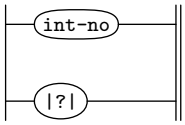


⟨ Rno_of_T subrule 2 op directive 46 ⟩ ≡

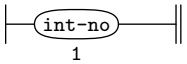
```
CAbs_lr1_sym * sym = new Err_no_of_T_kw_not_present;
sym→set_rc(*sf→p1__, __FILE__, __LINE__);
ADD_TOKEN_TO_ERROR_QUEUE(*sym);
rule_info__.parser__→set_stop_parse(true);
```

47. *Rno_of_T_value* **rule.**

Rno_of_T_value



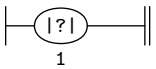
48. *Rno_of_T_value*'**s subrule 1.**



⟨ Rno_of_T_value subrule 1 op directive 48 ⟩ ≡

```
Cfsc_file * fsm = ( Cfsc_file * ) rule_info__.parser__→fsm_tbl__;
fsm→no_of_T_ = sf→p1__;      /* not eq make as a warning but continue parsing */
if (T_DICTIONARY.size( ) ≠ sf→p1__→no( )) {
  CAbs_lr1_sym * sym = new Err_bad_T_alphabet;
  sym→set_rc(*sf→p1__, __FILE__, __LINE__);
  ADD_TOKEN_TO_ERROR_QUEUE(*sym);
  rule_info__.parser__→set_stop_parse(true);
}
```

49. *Rno_of_T_value*'**s subrule 2.**

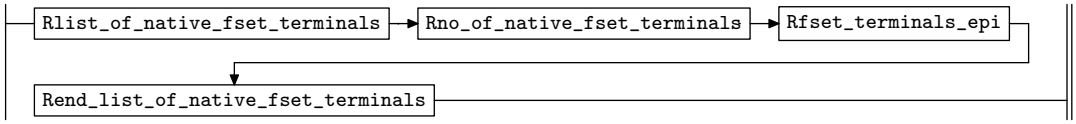


⟨ Rno_of_T_value subrule 2 op directive 49 ⟩ ≡

```
CAbs_lr1_sym * sym = new Err_no_of_T_value_not_present;
sym→set_rc(*sf→p1__, __FILE__, __LINE__);
ADD_TOKEN_TO_ERROR_QUEUE(*sym);
rule_info__.parser__→set_stop_parse(true);
```

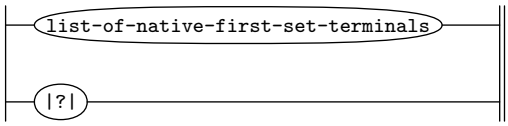
50. *Rlist_of_native_terminals_phrase* **rule.**

Rlist_of_native_terminals_phrase

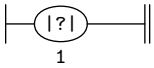


51. *Rlist_of_native_fset_terminals* **rule.**

Rlist_of_native_fset_terminals



52. *Rlist_of_native_fset_terminals*'s **subrule 2.**

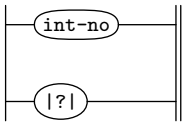


⟨ Rlist_of_native_fset_terminals subrule 2 op directive 52 ⟩ ≡

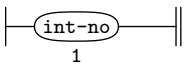
```
CAbs_lr1_sym * sym = new Err_list_of_terminals_kw_not_present;
sym->set_rc(*sf->p1__, __FILE__, __LINE__);
ADD_TOKEN_TO_ERROR_QUEUE(*sym);
rule_info__.parser__->set_stop_parse(true);
```

53. *Rno_of_native_fset_terminals* **rule.**

Rno_of_native_fset_terminals



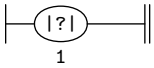
54. *Rno_of_native_fset_terminals*'s **subrule 1.**



⟨ Rno_of_native_fset_terminals subrule 1 op directive 54 ⟩ ≡

```
Cfsc_file * fsm = ( Cfsc_file * ) rule_info__.parser__->fsm_tbl__;
fsm->no_native_T_ = sf->p1__;
```

55. *Rno_of_native_fset_terminals*'s **subrule 2.**

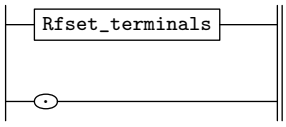


⟨ Rno_of_native_fset_terminals subrule 2 op directive 55 ⟩ ≡

```
CAbs_lr1_sym * sym = new Err_no_of_terminals_not_present;
sym->set_rc(*sf->p1__, __FILE__, __LINE__);
ADD_TOKEN_TO_ERROR_QUEUE(*sym);
rule_info__.parser__->set_stop_parse(true);
```

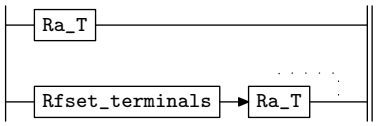
56. *Rfset_terminals_epi* **rule.**

Rfset_terminals_epi



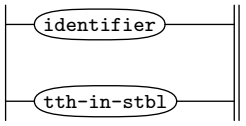
57. *Rfset_terminals* **rule.**

Rfset_terminals

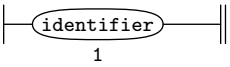


58. *Ra_T* **rule.**

Ra_T



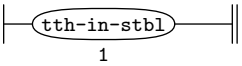
59. *Ra_T*'**s subrule 1.**



⟨ Ra_T subrule 1 op directive 59 ⟩ ≡

```
T_identifier * k = sf->p1__;
CAbs_lr1_sym * sym = new Err_bad_T_in_list;
sym->set_rc(*k, __FILE__, __LINE__);
ADD_TOKEN_TO_ERROR_QUEUE(*sym);
rule_info__.parser__->set_stop_parse(true);
```

60. *Ra_T*'**s subrule 2.**

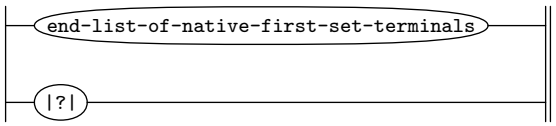


⟨ Ra_T subrule 2 op directive 60 ⟩ ≡

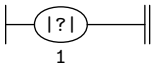
```
Cfsc_file * fsm = ( Cfsc_file * ) rule_info__.parser__->fsm_tbl__;
tth_in_stbl * k = sf->p1__;
T_attributes * T_att = k->t_in_stbl( );
fsm->list_of_Ts_.push_back( T_att->T_enum_);
```

61. *Rend_list_of_native_fset_terminals* **rule.**

Rend_list_of_native_fset_terminals



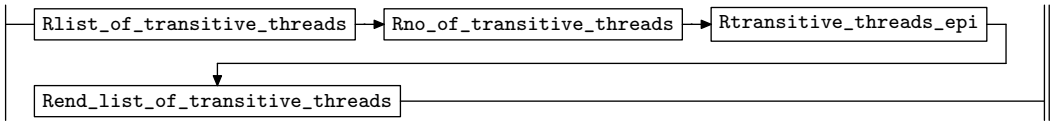
62. *Rend_list_of_native_fset_terminals*'s subrule 2.



⟨ Rend_list_of_native_fset_terminals subrule 2 op directive 62 ⟩ ≡
```
CAbs_lr1_sym * sym = new Err_end_list_native_T_kw_not_present;
sym→set_rc(*sf→p1__, __FILE__, __LINE__);
ADD_TOKEN_TO_ERROR_QUEUE(*sym);
rule_info__.parser__→set_stop_parse(true);
```

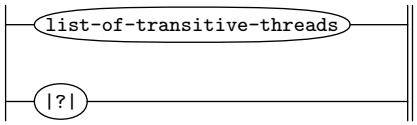
63. *Rlist_of_transitive_threads_phrase* **rule.**

Rlist_of_transitive_threads_phrase

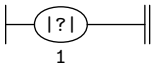


64. *Rlist_of_transitive_threads* **rule.**

Rlist_of_transitive_threads



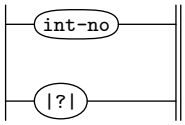
65. *Rlist_of_transitive_threads*'s subrule 2.



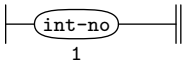
⟨ Rlist_of_transitive_threads subrule 2 op directive 65 ⟩ ≡
```
CAbs_lr1_sym * sym = new Err_no_list_of_trans_threads_kw;
sym→set_rc(*sf→p1__, __FILE__, __LINE__);
ADD_TOKEN_TO_ERROR_QUEUE(*sym);
rule_info__.parser__→set_stop_parse(true);
```

66. *Rno_of_transitive_threads* **rule.**

Rno_of_transitive_threads

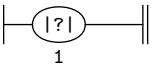


67. *Rno_of_transitive_threads*'s subrule 1.



⟨ Rno_of_transitive_threads subrule 1 op directive 67 ⟩ ≡
```
Cfsc_file * fsm = ( Cfsc_file * ) rule_info__.parser__→fsm_tbl__;
fsm→no_threads_ = sf→p1__;
```

68. *Rno_of_transitive_threads*'s subrule 2.

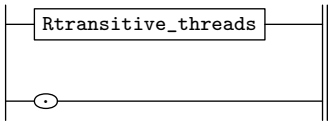


⟨ Rno_of_transitive_threads subrule 2 op directive 68 ⟩ ≡

```
CAbs_lr1_sym * sym = new Err_no_of_threads_not_present;
sym→set_rc(*sf→p1__, __FILE__, __LINE__);
ADD_TOKEN_TO_ERROR_QUEUE(*sym);
rule_info__.parser__→set_stop_parse(true);
```

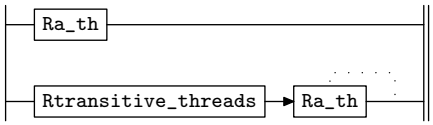
69. *Rtransitive_threads_epi* **rule.**

Rtransitive_threads_epi



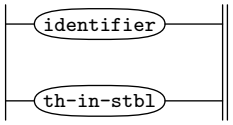
70. *Rtransitive_threads* **rule.**

Rtransitive_threads

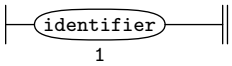


71. *Ra_th* **rule.**

Ra_th



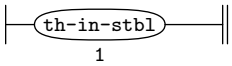
72. *Ra_th*'s subrule 1.



⟨ Ra_th subrule 1 op directive 72 ⟩ ≡

```
Cfsc_file * fsm = ( Cfsc_file * ) rule_info__.parser__->fsm_tbl__;
T_identifier * k = sf->p1__;
using namespace yacco2_stbl;
T_sym_tbl_report_card report_card;
thread_attributes * th = new thread_attributes(k->identifier( )->c_str( ));
th->set_rc(*k, __FILE__, __LINE__);
fsm->list_of_transitive_threads_.push_back(th);
th_in_stbl * t = new th_in_stbl(th);
add_sym_to_stbl(report_card, *k->identifier( )->c_str( ), *t, table_entry::used, table_entry::thread);
if (report_card.status_ ≠ T_sym_tbl_report_card::okay) {
  t->stbl_idx(report_card.pos_);
  report_card.err_entry_->set_rc(*k, __FILE__, __LINE__);
  ADD_TOKEN_TO_ERROR_QUEUE(*report_card.err_entry_);
  rule_info__.parser__->set_stop_parse(true);
  return;
}
GRAMMAR_DICTIONARY.push_back(report_card.tbl_entry_);
th->th_enum_ = GRAMMAR_DICTIONARY.size( ) − 1;
```

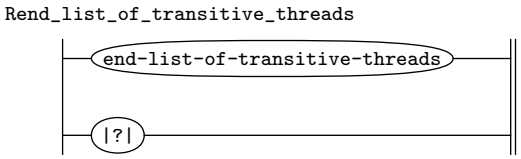
73. *Ra_th*'**s subrule 2.**



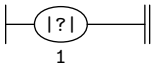
⟨ Ra_th subrule 2 op directive 73 ⟩ ≡

```
Cfsc_file * fsm = ( Cfsc_file * ) rule_info__.parser__→fsm_tbl__;
th_in_stbl * k = sf→p1__;
thread_attributes * th = k→thread_in_stbl( );
fsm→list_of_transitive_threads_.push_back(th);
using namespace yacco2_stbl;
T_sym_tbl_report_card report_card;
find_sym_in_stbl(report_card, *th→fully_qualified_th_name_.c_str( ));
if (report_card.status_ ≠ T_sym_tbl_report_card::okay) {
  report_card.err_entry_→set_rc(*k, __FILE__, __LINE__);
  ADD_TOKEN_TO_ERROR_QUEUE(*report_card.err_entry_);
  rule_info__.parser__→set_stop_parse(true);
  return;
}
if (report_card.action_ ≡ T_sym_tbl_report_card::fnd) {      /* crted by another fsc file */
  report_card.tbl_entry_→used_ = true;
  return;
}
if (report_card.action_ ≡ T_sym_tbl_report_card::not_fnd) {
  CAbs_lr1_sym * sym = new Err_no_sym_defs_present;
  sym→set_rc(*k, __FILE__, __LINE__);
  ADD_TOKEN_TO_ERROR_QUEUE(*sym);
  rule_info__.parser__→set_stop_parse(true);
  return;
}
```

74. *Rend_list_of_transitive_threads* **rule.**



75. *Rend_list_of_transitive_threads*'**s subrule 2.**

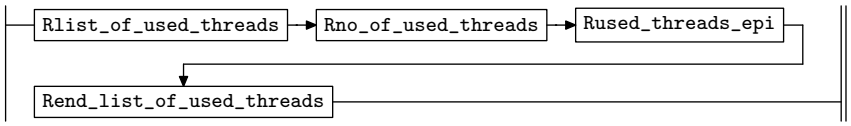


⟨ Rend_list_of_transitive_threads subrule 2 op directive 75 ⟩ ≡

```
CAbs_lr1_sym * sym = new Err_end_list_of_transitive_threads_kw_not_present;
sym→set_rc(*sf→p1__, __FILE__, __LINE__);
ADD_TOKEN_TO_ERROR_QUEUE(*sym);
rule_info__.parser__→set_stop_parse(true);
```

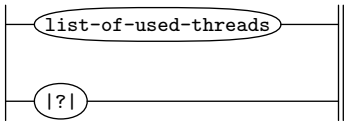
76. *Rlist_of_used_threads_phrase* **rule.**

Rlist_of_used_threads_phrase

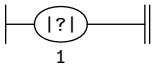


77. *Rlist_of_used_threads* **rule.**

Rlist_of_used_threads



78. *Rlist_of_used_threads*'s **subrule 2.**

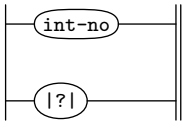


⟨ Rlist_of_used_threads subrule 2 op directive 78 ⟩ ≡

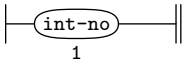
```
CAbs_lr1_sym * sym = new Err_no_list_of_trans_threads_kw;
sym->set_rc(*sf->p1__, __FILE__, __LINE__);
ADD_TOKEN_TO_ERROR_QUEUE(*sym);
rule_info__.parser__->set_stop_parse(true);
```

79. *Rno_of_used_threads* **rule.**

Rno_of_used_threads



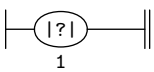
80. *Rno_of_used_threads*'s **subrule 1.**



⟨ Rno_of_used_threads subrule 1 op directive 80 ⟩ ≡

```
Cfsc_file * fsm = ( Cfsc_file * ) rule_info__.parser__->fsm_tbl__;
fsm->used_no_threads_ = sf->p1__;
```

81. *Rno_of_used_threads*'s **subrule 2.**

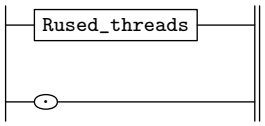


⟨ Rno_of_used_threads subrule 2 op directive 81 ⟩ ≡

```
CAbs_lr1_sym * sym = new Err_no_of_threads_not_present;
sym->set_rc(*sf->p1__, __FILE__, __LINE__);
ADD_TOKEN_TO_ERROR_QUEUE(*sym);
rule_info__.parser__->set_stop_parse(true);
```

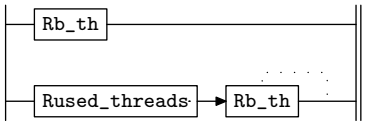
82. *Rused_threads_epi* **rule.**

Rused_threads_epi



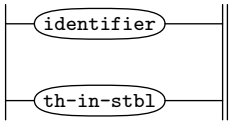
83. *Rused_threads* **rule.**

Rused_threads

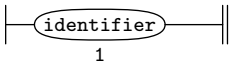


84. *Rb_th* **rule.**

Rb_th



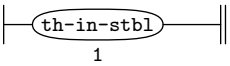
85. *Rb_th*'**s subrule 1.**



⟨ Rb_th subrule 1 op directive 85 ⟩ ≡

```
Cfsc_file * fsm = ( Cfsc_file * ) rule_info__.parser__->fsm_tbl__;
T_identifier * k = sf->p1__;
using namespace yacco2_stbl;
T_sym_tbl_report_card report_card;
thread_attributes * th = new thread_attributes(k->identifier()->c_str());
th->set_rc(*k, __FILE__, __LINE__);
fsm->list_of_used_threads_.push_back(th);
std::string::size_type x = th->fully_qualified_th_name_.find("::");
string tnm;
string & fqn = th->fully_qualified_th_name_;
for (int xx = x + 2; xx < fqn.size(); ++xx) {
  tnm += fqn[xx];
}
fsm->used_threads_->push_back(tnm);
```

86. *Rb_th*'**s subrule 2.**

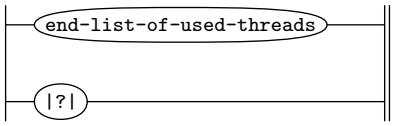


⟨ Rb_th subrule 2 op directive 86 ⟩ ≡

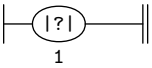
```
Cfsc_file * fsm = ( Cfsc_file * ) rule_info__.parser__->fsm_tbl__;
th_in_stbl * k = sf->p1__;
thread_attributes * th = k->thread_in_stbl( );
fsm->list_of_used_threads_.push_back(th);
std::string::size_type x = th->fully_qualified_th_name_.find("::");
string tnm;
string & fqn = th->fully_qualified_th_name_;
for (int xx = x + 2; xx < fqn.size( ); ++xx) {
  tnm += fqn[xx];
}
fsm->used_threads_->push_back(tnm);
```

87. *Rend_list_of_used_threads* **rule.**

Rend_list_of_used_threads



88. *Rend_list_of_used_threads*'**s subrule 2.**

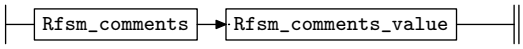


⟨ Rend_list_of_used_threads subrule 2 op directive 88 ⟩ ≡

```
CAbs_lr1_sym * sym = new Err_end_list_of_transitive_threads_kw_not_present;
sym->set_rc(*sf->p1__, __FILE__, __LINE__);
ADD_TOKEN_TO_ERROR_QUEUE(*sym);
rule_info__.parser__->set_stop_parse(true);
```

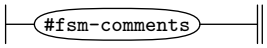
89. *Rfsm_comments_phrase* **rule.**

Rfsm_comments_phrase



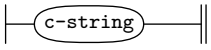
90. *Rfsm_comments* **rule.**

Rfsm_comments



91. *Rfsm_comments_value* **rule.**

Rfsm_comments_value



⟨ Rfsm_comments_value subrule 1 op directive 91 ⟩ ≡

```
Cfsc_file * fsm = ( Cfsc_file * ) rule_info__.parser__->fsm_tbl__;
fsm->fsm_comments_ = sf->p1__;
```

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

```
/*
 File: fsc_file.fsc
 Date and Time: Fri Jan  2 15:33:36 2015
*/
transitive   n
grammar-name "fsc_file"
name-space   "NS_fsc_file"
thread-name  "Cfsc_file"
monolithic   y
file-name    "fsc_file.fsc"
no-of-T      569
list-of-native-first-set-terminals 2
   T_transitive
   LR1_questionable_shift_operator
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
"\\Olinker's ``fsc'' control file parser."
```

93. Lr1 State Network.

⇒ State: 1 state type: ^{s}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rtransitive		3	2	1	|?|		1	2	2	
c	Rtransitive		3	1	1	transitive		1	3	3	
c	Rfsc_file		1	1	1	Rtransitive_phrase *Rgrammar_name_phrase*		1	4	15	
c	Rtransitive_phrase		2	1	1	Rtransitive *Rtransitive_value*		1	16	19	

⇒|?| State: 2 state type: ^{r}

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

⇒*transitive* State: 3 state type: ^{r}

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

⇒*Rtransitive_phrase* State: 4 state type: ^{s}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rgrammar_name		6	2	1	|?|		4	20	20	
c	Rgrammar_name		6	1	1	grammar-name		4	21	21	
t	Rfsc_file		1	1	2	Rgrammar_name_phrase *Rnamespace_phrase*		1	5	15	
c	Rgrammar_name_phrase		5	1	1	Rgrammar_name *Rgrammar_name_value*		4	22	25	

⇒*Rgrammar_name_phrase* State: 5 state type: ^{s}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rnamespace		9	2	1	|?|		5	26	26	
c	Rnamespace		9	1	1	# name-space		5	27	27	
t	Rfsc_file		1	1	3	Rnamespace_phrase *Rthreadname_phrase*		1	6	15	
c	Rnamespace_phrase		8	1	1	Rnamespace *Rnamespace_value*		5	28	31	

⇒*Rnamespace_phrase* State: 6 state type: ^{s}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rthreadname		12	2	1	|?|		6	32	32	
c	Rthreadname		12	1	1	thread-name		6	33	33	
t	Rfsc_file		1	1	4	Rthreadname_phrase *Rmonolithic_phrase*		1	7	15	
c	Rthreadname_phrase		11	1	1	Rthreadname *Rthreadname_value*		6	34	37	

⇒*Rthreadname_phrase* State: 7 state type: ^{s}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rmonolithic		15	2	1	|?|		7	38	38	
c	Rmonolithic		15	1	1	monolithic		7	39	39	
t	Rfsc_file		1	1	5	Rmonolithic_phrase *Rfilename_phrase*		1	8	15	
c	Rmonolithic_phrase		14	1	1	Rmonolithic *Rmonolithic_value*		7	40	43	

⇒*Rmonolithic_phrase* State: 8 state type: ^{s}

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rfilename		18	2	1	|?|		8	44	44	
c	Rfilename		18	1	1	# file-name		8	45	45	
t	Rfsc_file		1	1	6	Rfilename_phrase *Rno_of_T_phrase*		1	9	15	
c	Rfilename_phrase		17	1	1	Rfilename *Rfilename_value*		8	46	49	

⇒*Rfilename_phrase* State: 9 state type: *s*

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
c	Rno_of_T			21	2	1	`|?|`			9	50	50	
c	Rno_of_T			21	1	1	no-of-T			9	51	51	
t	Rfsc_file			1	1	7	Rno_of_T_phrase	*Rlist_of_native_terminals_phrase*		1	10	15	
c	Rno_of_T_phrase			20	1	1	Rno_of_T	*Rno_of_T_value*		9	52	55	

⇒*Rno_of_T_phrase* State: 10 state type: *s*

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
c	Rlist_of_native_fset_terminals			24	2	1	`|?|`			10	56	56	
c	Rlist_of_native_fset_terminals			24	1	1	list-of-native-first-set-terminals			10	57	57	
t	Rfsc_file			1	1	8	Rlist_of_native_terminals_phrase	*Rlist_of_transitive_threads_phrase*		1	11	15	
c	Rlist_of_native_terminals_phrase			23	1	1	Rlist_of_native_fset_terminals	*Rno_of_native_fset_terminals*		10	58	63	

⇒*Rlist_of_native_terminals_phrase* State: 11 state type: *s*

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
c	Rlist_of_transitive_threads			31	2	1	`|?|`			11	64	64	
c	Rlist_of_transitive_threads			31	1	1	list-of-transitive-threads			11	65	65	
t	Rfsc_file			1	1	9	Rlist_of_transitive_threads_phrase	*Rlist_of_used_threads_phrase*		1	12	15	
c	Rlist_of_transitive_threads_phrase			30	1	1	Rlist_of_transitive_threads	*Rno_of_transitive_threads*		11	66	71	

⇒*Rlist_of_transitive_threads_phrase* State: 12 state type: *s*

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
c	Rlist_of_used_threads			38	2	1	`|?|`			12	72	72	
c	Rlist_of_used_threads			38	1	1	list-of-used-threads			12	73	73	
t	Rfsc_file			1	1	10	Rlist_of_used_threads_phrase	*Rfsm_comments_phrase*		1	13	15	
c	Rlist_of_used_threads_phrase			37	1	1	Rlist_of_used_threads	*Rno_of_used_threads*		12	74	79	

⇒*Rlist_of_used_threads_phrase* State: 13 state type: *s*

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
c	Rfsm_comments			45	1	1	`#` fsm-comments			13	80	80	
t	Rfsc_file			1	1	11	Rfsm_comments_phrase	*eog*		1	14	15	
c	Rfsm_comments_phrase			44	1	1	Rfsm_comments	*Rfsm_comments_value*		13	81	83	

⇒*Rfsm_comments_phrase* State: 14 state type: *s*

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rfsc_file			1	1	12	eog			1	15	15	

⇒*eog* State: 15 state type: *r*

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rfsc_file			1	1	13				1	0	15	2

⇒*Rtransitive* State: 16 state type: *s*

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
c	Rtransitive_value			4	2	1	`|?|`			16	17	17	
c	Rtransitive_value			4	1	1	identifier			16	18	18	
t	Rtransitive_phrase			2	1	2	Rtransitive_value			1	19	19	

⇒`|?|` State: 17 state type: *r*

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rtransitive_value			4	2	2				16	0	17	3

⇒*identifier* State: 18 state type: *r*

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rtransitive_value		4	1	2				16	0	18	3

⇒*Rtransitive_value* State: 19 state type: *r*

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rtransitive_phrase		2	1	3				1	0	19	3

⇒|?| State: 20 state type: *r*

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rgrammar_name		6	2	2				4	0	20	4

⇒*grammar−name* State: 21 state type: *r*

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rgrammar_name		6	1	2				4	0	21	4

⇒*Rgrammar_name* State: 22 state type: *s*

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
c	Rgrammar_name_value		7	2	1	|?|			22	23	23	
c	Rgrammar_name_value		7	1	1	c-string			22	24	24	
t	Rgrammar_name_phrase		5	1	2	Rgrammar_name_value			4	25	25	

⇒|?| State: 23 state type: *r*

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rgrammar_name_value		7	2	2				22	0	23	5

⇒*c−string* State: 24 state type: *r*

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rgrammar_name_value		7	1	2				22	0	24	5

⇒*Rgrammar_name_value* State: 25 state type: *r*

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rgrammar_name_phrase		5	1	3				4	0	25	5

⇒|?| State: 26 state type: *r*

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rnamespace		9	2	2				5	0	26	4

⇒*#name−space* State: 27 state type: *r*

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rnamespace		9	1	2				5	0	27	4

⇒*Rnamespace* State: 28 state type: *s*

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
c	Rnamespace_value		10	2	1	|?|			28	29	29	
c	Rnamespace_value		10	1	1	c-string			28	30	30	
t	Rnamespace_phrase		8	1	2	Rnamespace_value			5	31	31	

⇒|?| State: 29 state type: *r*

←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rnamespace_value		10	2	2				28	0	29	6

⇒*c−string* State: 30 state type: *r*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rnamespace_value		10	1	2			28	0	30	6

⇒*Rnamespace_value* State: 31 state type: *r*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rnamespace_phrase		8	1	3			5	0	31	6

⇒|?| State: 32 state type: *r*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rthreadname		12	2	2			6	0	32	4

⇒*thread−name* State: 33 state type: *r*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rthreadname		12	1	2			6	0	33	4

⇒*Rthreadname* State: 34 state type: *s*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rthreadname_value		13	2	1	|?|		34	35	35	
c	Rthreadname_value		13	1	1	c-string		34	36	36	
t	Rthreadname_phrase		11	1	2	Rthreadname_value		6	37	37	

⇒|?| State: 35 state type: *r*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rthreadname_value		13	2	2			34	0	35	7

⇒*c−string* State: 36 state type: *r*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rthreadname_value		13	1	2			34	0	36	7

⇒*Rthreadname_value* State: 37 state type: *r*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rthreadname_phrase		11	1	3			6	0	37	7

⇒|?| State: 38 state type: *r*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rmonolithic		15	2	2			7	0	38	1

⇒*monolithic* State: 39 state type: *r*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rmonolithic		15	1	2			7	0	39	1

⇒*Rmonolithic* State: 40 state type: *s*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rmonolithic_value		16	2	1	|?|		40	41	41	
c	Rmonolithic_value		16	1	1	identifier		40	42	42	
t	Rmonolithic_phrase		14	1	2	Rmonolithic_value		7	43	43	

⇒|?| State: 41 state type: *r*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rmonolithic_value		16	2	2			40	0	41	8

⇒*identifier* State: 42 state type: *r*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rmonolithic_value		16	1	2			40	0	42	8

⇒*Rmonolithic_value* State: 43 state type: *r*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rmonolithic_phrase		14	1	3			7	0	43	8

⇒|?| State: 44 state type: *r*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rfilename		18	2	2			8	0	44	4

⇒*#file−name* State: 45 state type: *r*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rfilename		18	1	2			8	0	45	4

⇒*Rfilename* State: 46 state type: *s*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rfilename_value		19	2	1	|?|		46	47	47	
c	Rfilename_value		19	1	1	c-string		46	48	48	
t	Rfilename_phrase		17	1	2	Rfilename_value		8	49	49	

⇒|?| State: 47 state type: *r*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rfilename_value		19	2	2			46	0	47	9

⇒*c−string* State: 48 state type: *r*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rfilename_value		19	1	2			46	0	48	9

⇒*Rfilename_value* State: 49 state type: *r*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rfilename_phrase		17	1	3			8	0	49	9

⇒|?| State: 50 state type: *r*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rno_of_T		21	2	2			9	0	50	10

⇒*no−of−T* State: 51 state type: *r*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rno_of_T		21	1	2			9	0	51	10

⇒*Rno_of_T* State: 52 state type: *s*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rno_of_T_value		22	2	1	|?|		52	53	53	
c	Rno_of_T_value		22	1	1	int-no		52	54	54	
t	Rno_of_T_phrase		20	1	2	Rno_of_T_value		9	55	55	

⇒|?| State: 53 state type: *r*

	← rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rno_of_T_value		22	2	2			52	0	53	11

⇒*int−no*

State: 54 state type: *r*

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rno_of_T_value			22	1	2				52	0	54	11

⇒*Rno_of_T_value*

State: 55 state type: *r*

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rno_of_T_phrase			20	1	3				9	0	55	11

⇒|?|

State: 56 state type: *r*

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rlist_of_native_fset_terminals			24	2	2				10	0	56	10

⇒*list−of−native−first−set−terminals*

State: 57 state type: *r*

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rlist_of_native_fset_terminals			24	1	2				10	0	57	10

⇒*Rlist_of_native_fset_terminals*

State: 58 state type: *s*

	←	rule	→	R#	sr#	Po	←	subrule element	→
c	Rno_of_native_fset_terminals			25	2	1	|?|		
c	Rno_of_native_fset_terminals			25	1	1	int-no		
t	Rlist_of_native_terminals_phrase			23	1	2	Rno_of_native_fset_terminals	*Rfset_terminals_epi*$^{\epsilon}$	*Rend_list_of_native_fset_termina*

⇒*Rno_of_native_fset_terminals*

State: 59 state type: *s/r*

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
c	Rfset_terminals_epi			26	2	1	ϵ			59	0	59	12
c	Ra_T			28	1	1	identifier			59	86	86	
c	Ra_T			28	2	1	tth-in-stbl			59	87	87	
t	Rlist_of_native_terminals_phrase			23	1	3	Rfset_terminals_epi	*Rend_list_of_native_fset_terminals*		10	60	63	
c	Rfset_terminals_epi			26	1	1	Rfset_terminals			59	88	88	
c	Rfset_terminals			27	2	1	Rfset_terminals	*Ra_T*		59	88	89	
c	Rfset_terminals			27	1	1	Ra_T			59	90	90	

⇒*Rfset_terminals_epi*

State: 60 state type: *s*

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
c	Rend_list_of_native_fset_terminals			29	2	1	|?|			60	61	61	
c	Rend_list_of_native_fset_terminals			29	1	1	end-list-of-native-first-set-terminals			60	62	62	
t	Rlist_of_native_terminals_phrase			23	1	4	Rend_list_of_native_fset_terminals			10	63	63	

⇒|?|

State: 61 state type: *r*

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rend_list_of_native_fset_terminals			29	2	2				60	0	61	13

⇒*end−list−of−native−first−set−terminals*

State: 62 state type: *r*

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rend_list_of_native_fset_terminals			29	1	2				60	0	62	13

⇒*Rend_list_of_native_fset_terminals*

State: 63 state type: *r*

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rlist_of_native_terminals_phrase			23	1	5				10	0	63	13

⇒|?|

State: 64 state type: *r*

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rlist_of_transitive_threads			31	2	2				11	0	64	10

⇒*list−of−transitive−threads*

State: 65 state type: *r*

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rlist_of_transitive_threads			31	1	2				11	0	65	10

⇒*Rlist_of_transitive_threads*

State: 66 state type: *s*

	←	rule	→	R#	sr#	Po	←	subrule element	→
c	Rno_of_transitive_threads			32	2	1	|?|		
c	Rno_of_transitive_threads			32	1	1	int-no		
t	Rlist_of_transitive_threads_phrase			30	1	2	Rno_of_transitive_threads	*Rtransitive_threads_epi*$^{\epsilon}$	*Rend_list_of_transitive_threads*

⇒*Rno_of_transitive_threads*

State: 67 state type: *s/r*

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
c	Rtransitive_threads_epi			33	2	1	ϵ			67	0	67	14
c	Ra_th			35	1	1	identifier			67	93	93	
c	Ra_th			35	2	1	th-in-stbl			67	94	94	
t	Rlist_of_transitive_threads_phrase			30	1	3	Rtransitive_threads_epi	*Rend_list_of_transitive_threads*		11	68	71	
c	Rtransitive_threads_epi			33	1	1	Rtransitive_threads			67	95	95	
c	Rtransitive_threads			34	2	1	Rtransitive_threads	*Ra_th*		67	95	96	
c	Rtransitive_threads			34	1	1	Ra_th			67	97	97	

⇒*Rtransitive_threads_epi*

State: 68 state type: *s*

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
c	Rend_list_of_transitive_threads			36	2	1	|?|			68	69	69	
c	Rend_list_of_transitive_threads			36	1	1	end-list-of-transitive-threads			68	70	70	
t	Rlist_of_transitive_threads_phrase			30	1	4	Rend_list_of_transitive_threads			11	71	71	

⇒|?|

State: 69 state type: *r*

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rend_list_of_transitive_threads			36	2	2				68	0	69	15

⇒*end−list−of−transitive−threads*

State: 70 state type: *r*

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rend_list_of_transitive_threads			36	1	2				68	0	70	15

⇒*Rend_list_of_transitive_threads*

State: 71 state type: *r*

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rlist_of_transitive_threads_phrase			30	1	5				11	0	71	15

⇒|?|

State: 72 state type: *r*

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rlist_of_used_threads			38	2	2				12	0	72	10

⇒*list−of−used−threads*

State: 73 state type: *r*

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rlist_of_used_threads			38	1	2				12	0	73	10

⇒*Rlist_of_used_threads*

State: 74 state type: *s*

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
c	Rno_of_used_threads			39	2	1	|?|			74	98	98	

c	Rno_of_used_threads	39	1	1	int-no	74	99	99
t	Rlist_of_used_threads_phrase	37	1	2	Rno_of_used_threads *Rused_threads_epi*$^{\epsilon}$ *Rend_list_of_used_threads*	12	75	79

$\Rightarrow^{Rno\_of\_used\_threads}$ State: 75 state type: $^{s/r}$

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rused_threads_epi		40	2	1	ϵ		75	0	75	16
c	Rb_th		42	1	1	identifier		75	100	100	
c	Rb_th		42	2	1	th-in-stbl		75	101	101	
t	Rlist_of_used_threads_phrase		37	1	3	Rused_threads_epi *Rend_list_of_used_threads*		12	76	79	
c	Rused_threads_epi		40	1	1	Rused_threads		75	102	102	
c	Rused_threads		41	2	1	Rused_threads *Rb_th*		75	102	103	
c	Rused_threads		41	1	1	Rb_th		75	104	104	

$\Rightarrow^{Rused\_threads\_epi}$ State: 76 state type: ^{s}

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rend_list_of_used_threads		43	2	1	|?|		76	77	77	
c	Rend_list_of_used_threads		43	1	1	end-list-of-used-threads		76	78	78	
t	Rlist_of_used_threads_phrase		37	1	4	Rend_list_of_used_threads		12	79	79	

$\Rightarrow^{|?|}$ State: 77 state type: ^{r}

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rend_list_of_used_threads		43	2	2			76	0	77	17

$\Rightarrow^{end-list-of-used-threads}$ State: 78 state type: ^{r}

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rend_list_of_used_threads		43	1	2			76	0	78	17

$\Rightarrow^{Rend\_list\_of\_used\_threads}$ State: 79 state type: ^{r}

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rlist_of_used_threads_phrase		37	1	5			12	0	79	17

$\Rightarrow^{\#fsm-comments}$ State: 80 state type: ^{r}

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rfsm_comments		45	1	2			13	0	80	18

$\Rightarrow^{Rfsm\_comments}$ State: 81 state type: ^{s}

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
c	Rfsm_comments_value		46	1	1	c-string		81	82	82	
t	Rfsm_comments_phrase		44	1	2	Rfsm_comments_value		13	83	83	

$\Rightarrow^{c-string}$ State: 82 state type: ^{r}

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rfsm_comments_value		46	1	2			81	0	82	19

$\Rightarrow^{Rfsm\_comments\_value}$ State: 83 state type: ^{r}

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rfsm_comments_phrase		44	1	3			13	0	83	19

$\Rightarrow^{|?|}$ State: 84 state type: ^{r}

←	rule	→	R#	sr#	Po	← subrule element	→	Brn	Gto	Red	LA
t	Rno_of_native_fset_terminals		25	2	2			58	0	84	20

⇒*int−no*

	←	rule	→	R#	sr#	Po	←
t	Rno_of_native_fset_terminals			25	1	2	

⇒*identifier*

	←	rule	→	R#	sr#	Po	←
t	Ra_T			28	1	2	

⇒*tth−in−stbl*

	←	rule	→	R#	sr#	Po	←
t	Ra_T			28	2	2	

⇒*Rfset_terminals*

	←	rule	→	R#	sr#	Po	←
t	Rfset_terminals_epi			26	1	2	
c	Ra_T			28	1	1	identifier
c	Ra_T			28	2	1	tth-in-stbl
t	Rfset_terminals			27	2	2	Ra_T

⇒*Ra_T*

	←	rule	→	R#	sr#	Po	←
t	Rfset_terminals			27	2	3	

⇒*Ra_T*

	←	rule	→	R#	sr#	Po	←
t	Rfset_terminals			27	1	2	

⇒|?|

	←	rule	→	R#	sr#	Po	←
t	Rno_of_transitive_threads			32	2	2	

⇒*int−no*

	←	rule	→	R#	sr#	Po	←
t	Rno_of_transitive_threads			32	1	2	

⇒*identifier*

	←	rule	→	R#	sr#	Po	←
t	Ra_th			35	1	2	

⇒*th−in−stbl*

	←	rule	→	R#	sr#	Po	←
t	Ra_th			35	2	2	

⇒*Rtransitive_threads*

	←	rule	→	R#	sr#	Po	←
t	Rtransitive_threads_epi			33	1	2	
c	Ra_th			35	1	1	identifier
c	Ra_th			35	2	1	th-in-stbl
t	Rtransitive_threads			34	2	2	Ra_th

⇒*Ra_th*

State: 85 state type: *r*

subrule element	→	Brn	Gto	Red	LA
		58	0	85	20

State: 86 state type: *r*

subrule element	→	Brn	Gto	Red	LA
		59	0	86	20

State: 87 state type: *r*

subrule element	→	Brn	Gto	Red	LA
		59	0	87	20

State: 88 state type: *s/r*

subrule element	→	Brn	Gto	Red	LA
		59	0	88	12
		88	86	86	
		88	87	87	
		59	89	89	

State: 89 state type: *r*

subrule element	→	Brn	Gto	Red	LA
		59	0	89	20

State: 90 state type: *r*

subrule element	→	Brn	Gto	Red	LA
		59	0	90	20

State: 91 state type: *r*

subrule element	→	Brn	Gto	Red	LA
		66	0	91	21

State: 92 state type: *r*

subrule element	→	Brn	Gto	Red	LA
		66	0	92	21

State: 93 state type: *r*

subrule element	→	Brn	Gto	Red	LA
		67	0	93	21

State: 94 state type: *r*

subrule element	→	Brn	Gto	Red	LA
		67	0	94	21

State: 95 state type: *s/r*

subrule element	→	Brn	Gto	Red	LA
		67	0	95	14
		95	93	93	
		95	94	94	
		67	96	96	

State: 96 state type: *r*

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rtransitive_threads			34	2	3				67	0	96	21

⇒*Ra_th*

State: 97 state type: *r*

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rtransitive_threads			34	1	2				67	0	97	21

⇒|?|

State: 98 state type: *r*

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rno_of_used_threads			39	2	2				74	0	98	22

⇒*int−no*

State: 99 state type: *r*

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rno_of_used_threads			39	1	2				74	0	99	22

⇒*identifier*

State: 100 state type: *r*

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rb_th			42	1	2				75	0	100	22

⇒*th−in−stbl*

State: 101 state type: *r*

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rb_th			42	2	2				75	0	101	22

⇒*Rused_threads*

State: 102 state type: *s/r*

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rused_threads_epi			40	1	2				75	0	102	16
c	Rb_th			42	1	1	identifier			102	100	100	
c	Rb_th			42	2	1	th-in-stbl			102	101	101	
t	Rused_threads			41	2	2	Rb_th			75	103	103	

⇒*Rb_th*

State: 103 state type: *r*

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rused_threads			41	2	3				75	0	103	22

⇒*Rb_th*

State: 104 state type: *r*

	←	rule	→	R#	sr#	Po	←	subrule element	→	Brn	Gto	Red	LA
t	Rused_threads			41	1	2				75	0	104	22

94. Index.

fsc_file Grammar

Date: January 2, 2015 at 15:35

File: fsc_file.lex

Ns: NS_fsc_file

Version: 1.0 Debug: false

Grammar Comments: Type: Monolithic

O_2^{linker}'s "fsc" control file parser.