

1. Copyright.

Copyright © Dave Bone 1998 - 2015

2. *o2_lcl_opt* Thread.

The grammar *o2_lcl_opts* calls it per option detected. Options can be inputted in any order and all are optional. Their appearance in the command line means 'y'. Originally options were prefixed by a slash but i converted to the UNIX use of a minus sign. Here they are:

- t — generate Terminals definitions
- rc — generate Raw Chars definitions: deprecated
- err — generate Error terminals definitions
- lrk — generate yacc2's LRK terminal constants definitions: deprecated
- p — generate pdf + post script(ps) documents

Returned is the appropriate T indicating the option.

Note:

In the "parallel-la-boundary" construct this grammar demonstrates how one can fine tune the look-ahead boundary to cope with a false option prefix eg, -tt. error reported at -t and not the next option re: 2nd t. *eolr* is too general for this refinement.

Also shown is the use of meta-terminals to indicate or contain the parsed results. This is the cleanest way to communicate between threads. The agreement between the calling / called threads is the meta-terminal. Use of Global variables is efficient but ugly and unco-ordinated due to the random write access to the Global variable.

Modifications:

Date: 19 Oct. 2003

Reason:

Modify rule names to be shorter and more concise. Originally the encoding rule for the grammar rule name was R + "thread name prefix" + "name given to rule" where the plus sign indicates concatenation. This made it volumous in the traces.

Reason:

Add new switches to support:

/dbg turn on debugging of grammar to be compiled. this is a convenience to the author instead of using the editor to change the grammar's source code of fsm-debug "true" to "true"; once finished debugging, re-use the editor to restore the source code back to fsm-debug "true".

Date: 2 June 2005

Reason:

Remove /s:xxx switch. Compilers should be robust enough to deal with reasonably large files. Nod to Microsoft current compiler

3. Fsm Co2_lcl_opt class.**4. Co2_lcl_opt failed directive.**

Trap the failed option and return a bad command. This covers errors like the premature prefix -e where it should be -err. i could have been less specific to trap non first set options (-z) by defaulting to this facility but i'm teaching myself... As this thread is executed according to its first set "-", any failed attempt is a bad option. Please note the use of the *RSVP_FSM* macro. Its context is different than the normal Rule's use of *RSVP* macro.

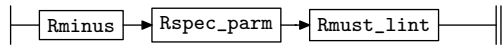
```

⟨ Co2_lcl_opt failed directive 4 ⟩ ≡
  CAbs_lr1_sym * sym = new Err_bad_cmd_line_opt;
  sym->set_rc(*parser->start_token--, __FILE__, __LINE__);
  RSVP_FSM(sym);
  return true;

```

5. Ro2_lcl.opt rule.

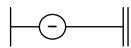
Ro2_lcl.opt



⟨ Ro2_lcl.opt subrule 1 op directive 5 ⟩ ≡
`sf-p2--rtn_parm--set_rc(*rule_info_.parser--start_token_, __FILE__, __LINE__);`
`RSVP(sf-p2--rtn_parm_);`

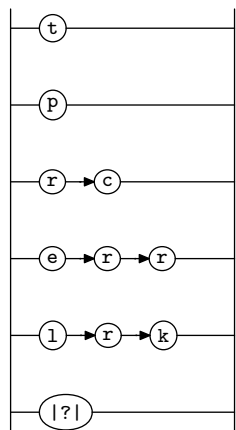
6. Rminus rule.

Rminus



7. Rspec_parm rule.

Rspec_parm

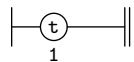


8. Rspec_parm user-declaration directive.

⟨ Rspec_parm user-declaration directive 8 ⟩ ≡
`public: CAbs_lr1_sym * rtn_parm_;`

9. Rspec_parm's subrule 1.

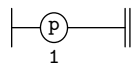
Generate Terminals vocabulary.



⟨ Rspec_parm subrule 1 op directive 9 ⟩ ≡
`rtn_parm_ = new T_option_t;`

10. Rspec_parm's subrule 2.

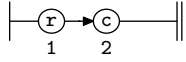
Generate Terminals vocabulary.



⟨ Rspec_parm subrule 2 op directive 10 ⟩ ≡
`rtn_parm_ = new T_option_p;`

11. *Rspec_parm*'s subrule 3.

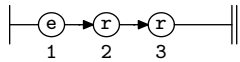
Generate Raw characters vocabulary. Watch out as this is only used at bootstrap time of O_2 . Users of O_2 should not use it unless they are experimenting.



\langle *Rspec_parm* subrule 3 op directive 11 $\rangle \equiv$
`rtn_parm_ = new Err_bad_cmd_line_opt;`
`rtn_parm_→set_rc(*rule_info_.parser_→start_token_, __FILE__, __LINE__);`

12. *Rspec_parm*'s subrule 4.

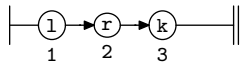
Generate Error vocabulary.



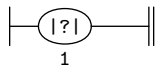
\langle *Rspec_parm* subrule 4 op directive 12 $\rangle \equiv$
`rtn_parm_ = new T_option_err;`

13. *Rspec_parm*'s subrule 5.

Generate LR K vocabulary. Watch out as this is only used at bootstrap time of O_2 . Users of O_2 should not use it unless they are experimenting. Currently not supported.



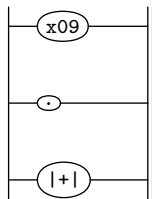
\langle *Rspec_parm* subrule 5 op directive 13 $\rangle \equiv$
`rtn_parm_ = new Err_bad_cmd_line_opt;`
`rtn_parm_→set_rc(*rule_info_.parser_→start_token_, __FILE__, __LINE__);`

14. *Rspec_parm*'s subrule 6.

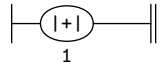
\langle *Rspec_parm* subrule 6 op directive 14 $\rangle \equiv$
`rtn_parm_ = new Err_bad_cmd_line_opt;`
`rtn_parm_→set_rc(*rule_info_.parser_→start_token_, __FILE__, __LINE__);`

15. *Rmust_lint* rule.

Rmust_lint



16. *Rmust_lint*'s subrule 3.



$\langle \text{Rmust_lint subrule 3 op directive 16} \rangle \equiv$
`parser()-set_abort_parse(true);`

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

```
/*
  File: o2_lcl_opt.fsc
  Date and Time: Fri Jan  2 15:33:45 2015
*/
transitive      n
grammar-name    "o2_lcl_opt"
name-space      "NS_o2_lcl_opt"
thread-name     "TH_o2_lcl_opt"
monolithic      n
file-name       "o2_lcl_opt.fsc"
no-of-T         569
list-of-native-first-set-terminals 1
  raw_minus
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
"\02's individual command line option recognizer."
```

18. Lr1 State Network.

\Rightarrow					State: 1 state type: s		
\leftarrow	rule	\rightarrow	R# sr# Po	\leftarrow	subrule element	\rightarrow	Brn Gto Red LA
c	Rminus		2 1 1 -				1 2 2
c	Ro2.lcl.opt		1 1 1	Rminus	<u>Rspec_parm</u>		1 3 8
\Rightarrow^-					State: 2 state type: r		
\leftarrow	rule	\rightarrow	R# sr# Po	\leftarrow	subrule element	\rightarrow	Brn Gto Red LA
t	Rminus		2 1 2				1 0 2 1
\Rightarrow	<u>Rminus</u>				State: 3 state type: s		
\leftarrow	rule	\rightarrow	R# sr# Po	\leftarrow	subrule element	\rightarrow	Brn Gto Red LA
c	Rspec_parm		3 6 1 ?				3 9 9
c	Rspec_parm		3 4 1 e				3 10 12
c	Rspec_parm		3 5 1 l				3 13 15
c	Rspec_parm		3 2 1 p				3 16 16
c	Rspec_parm		3 3 1 r				3 17 18
c	Rspec_parm		3 1 1 t				3 19 19
t	Ro2.lcl.opt		1 1 2	Rspec_parm	<u>Rmust_lint</u>		1 4 8
\Rightarrow	<u>Rspec_parm</u>				State: 4 state type: s		
\leftarrow	rule	\rightarrow	R# sr# Po	\leftarrow	subrule element	\rightarrow	Brn Gto Red LA
c	Rmust_lint		4 3 1 +				4 5 5
c	Rmust_lint		4 1 1 x09				4 6 6
c	Rmust_lint		4 2 1				4 7 7
t	Ro2.lcl.opt		1 1 3	Rmust_lint			1 8 8
\Rightarrow	<u> + </u>				State: 5 state type: r		
\leftarrow	rule	\rightarrow	R# sr# Po	\leftarrow	subrule element	\rightarrow	Brn Gto Red LA
t	Rmust_lint		4 3 2				4 0 5 2
\Rightarrow	<u>x09</u>				State: 6 state type: r		
\leftarrow	rule	\rightarrow	R# sr# Po	\leftarrow	subrule element	\rightarrow	Brn Gto Red LA
t	Rmust_lint		4 1 2				4 0 6 2
\Rightarrow					State: 7 state type: r		
\leftarrow	rule	\rightarrow	R# sr# Po	\leftarrow	subrule element	\rightarrow	Brn Gto Red LA
t	Rmust_lint		4 2 2				4 0 7 2
\Rightarrow	<u>Rmust_lint</u>				State: 8 state type: r		
\leftarrow	rule	\rightarrow	R# sr# Po	\leftarrow	subrule element	\rightarrow	Brn Gto Red LA
t	Ro2.lcl.opt		1 1 4				1 0 8 2
\Rightarrow	<u> ? </u>				State: 9 state type: r		
\leftarrow	rule	\rightarrow	R# sr# Po	\leftarrow	subrule element	\rightarrow	Brn Gto Red LA
t	Rspec_parm		3 6 2				3 0 9 3
\Rightarrow^e					State: 10 state type: s		
\leftarrow	rule	\rightarrow	R# sr# Po	\leftarrow	subrule element	\rightarrow	Brn Gto Red LA
t	Rspec_parm		3 4 2 r				3 11 12

\Rightarrow^r	\leftarrow rule	\rightarrow R# sr# Po \leftarrow	State: 11 state type: s	
t Rspec_parm		3 4 3 r	subrule element	\rightarrow Brn Gto Red LA
				3 12 12
\Rightarrow^r	\leftarrow rule	\rightarrow R# sr# Po \leftarrow	State: 12 state type: r	
t Rspec_parm		3 4 4	subrule element	\rightarrow Brn Gto Red LA
				3 0 12 3
\Rightarrow^l	\leftarrow rule	\rightarrow R# sr# Po \leftarrow	State: 13 state type: s	
t Rspec_parm		3 5 2 r	subrule element	\rightarrow Brn Gto Red LA
				3 14 15
\Rightarrow^r	\leftarrow rule	\rightarrow R# sr# Po \leftarrow	State: 14 state type: s	
t Rspec_parm		3 5 3 k	subrule element	\rightarrow Brn Gto Red LA
				3 15 15
\Rightarrow^k	\leftarrow rule	\rightarrow R# sr# Po \leftarrow	State: 15 state type: r	
t Rspec_parm		3 5 4	subrule element	\rightarrow Brn Gto Red LA
				3 0 15 3
\Rightarrow^p	\leftarrow rule	\rightarrow R# sr# Po \leftarrow	State: 16 state type: r	
t Rspec_parm		3 2 2	subrule element	\rightarrow Brn Gto Red LA
				3 0 16 3
\Rightarrow^r	\leftarrow rule	\rightarrow R# sr# Po \leftarrow	State: 17 state type: s	
t Rspec_parm		3 3 2 c	subrule element	\rightarrow Brn Gto Red LA
				3 18 18
\Rightarrow^c	\leftarrow rule	\rightarrow R# sr# Po \leftarrow	State: 18 state type: r	
t Rspec_parm		3 3 3	subrule element	\rightarrow Brn Gto Red LA
				3 0 18 3
\Rightarrow^t	\leftarrow rule	\rightarrow R# sr# Po \leftarrow	State: 19 state type: r	
t Rspec_parm		3 1 2	subrule element	\rightarrow Brn Gto Red LA
				3 0 19 3

19. Index.

|+|: 15.
|?|: 7.
__FILE__: 4, 5, 11, 13, 14.
__LINE__: 4, 5, 11, 13, 14.
CAbs_lr1_sym: 4, 8.
eobr: 2.
Err_bad_cmd_line_opt: 4, 11, 13, 14.
o2.lcl.opt: 2.
o2.lcl.opts: 2.
parser: 16.
parser_: 4, 5, 11, 13, 14.
p2_: 5.
Rminus: 5.
Rminus: 6.
Rmust_lint: 5.
Rmust_lint: 15, 16.
Ro2.lcl.opt: 5.
Rspec_parm: 5.
Rspec_parm: 7, 9, 10, 11, 12, 13, 14.
RSVP: 4, 5.
RSVP_FSM: 4.
rtn_parm_: 5, 8, 9, 10, 11, 12, 13, 14.
rule_info_: 5, 11, 13, 14.
set_abort_parse: 16.
set_rc: 4, 5, 11, 13, 14.
sf: 5.
start_token_: 4, 5, 11, 13, 14.
sym: 4.
T_option_err: 12.
T_option_p: 10.
T_option_t: 9.
true: 4, 16.

⟨ Co2.lcl.opt failed directive 4 ⟩
⟨ Rmust.lint subrule 3 op directive 16 ⟩
⟨ Ro2.lcl.opt subrule 1 op directive 5 ⟩
⟨ Rspec_parm subrule 1 op directive 9 ⟩
⟨ Rspec_parm subrule 2 op directive 10 ⟩
⟨ Rspec_parm subrule 3 op directive 11 ⟩
⟨ Rspec_parm subrule 4 op directive 12 ⟩
⟨ Rspec_parm subrule 5 op directive 13 ⟩
⟨ Rspec_parm subrule 6 op directive 14 ⟩
⟨ Rspec_parm user-declaration directive 8 ⟩

o2_lcl_opt Grammar

Date: January 2, 2015 at 15:37

File: o2_lcl_opt.lex

Ns: NS_o2_lcl_opt

Version: 1.1

Debug: false

Grammar Comments:

Type: Thread

O_2 's individual command line option recognizer.

1 element(s) in Lookahead Expression below

eolr

	Section	Page
Copyright	1	1
<i>o2_lcl_opt</i> Thread	2	2
Fsm Co2_lcl_opt class	3	2
Co2_lcl_opt failed directive	4	2
<i>Ro2_lcl_opt</i> rule	5	3
<i>Rminus</i> rule	6	3
<i>Rspec_parm</i> rule	7	3
Rspec_parm user-declaration directive	8	3
<i>Rspec_parm</i> 's subrule 1	9	3
<i>Rspec_parm</i> 's subrule 2	10	3
<i>Rspec_parm</i> 's subrule 3	11	4
<i>Rspec_parm</i> 's subrule 4	12	4
<i>Rspec_parm</i> 's subrule 5	13	4
<i>Rspec_parm</i> 's subrule 6	14	4
<i>Rmust_lint</i> rule	15	4
<i>Rmust_lint</i> 's subrule 3	16	5
First Set Language for O_2^{linker}	17	6
Lr1 State Network	18	7
Index	19	9