1. Eol Thread.
This thread's claim-to-fame is its end-of-line matching for Unix, Mac, or Windows variants. See *Rdelim* rule for the variant in end-of-line recognition. The other thing of interest is its use of the meta terminal Without it in the subrule, the traditional way is to subtract "x0a" from *eolr* representing "all termi in the Terminal alphabet within the lookahead expression of "parallel-thread-function" to prevent a s reduce conflict due to the "xod" common prefix subrules. This works but is very inefficient in the s the lookahead set generated caused by the number of terminals in the Terminal alphabet. The `|.|` app adds a shift in the subrule but only "eolr" in its reduce set whereas the traditional way has to binary s thru the lookahead set of approximately .5k terminals to see if the current token is a member. Under cu set implementation, this is expensive as the partition number is binary searched first followed by the ele within the 8 member set.

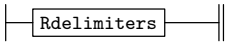
How does `|.|` work? Being a meta-terminal, it is not part of the token stream. It is one of the pa conditionals tested for by its presence within the finite automaton's current state. `|+|` is another such terminal example.

Use a global pointer to it as it is just an indicator. The new / delete cycle is too expensive.

2. Fsm *Ceol* class.

3. *Reol* rule.

`Reol`



Return the *eol* token back to the caller.

⟨ *Reol* subrule 1 op directive 3 ⟩ ≡
```
CAbs_lr1_sym * sym = NS_yacco2_terminals::PTR_eol__;
sym->set_rc(*rule_info__.parser__->start_token__, __FILE__, __LINE__);
RSVP(sym);
```