NAME

LRXplain − tries to explain LALR conflicts

SYNOPSIS

LRXplain [**−s** *N* **−v**[**v**[**v**]]] *G*.**output**

DESCRIPTION

LRXplain tries to assist the user in finding the underlying causes of LALR conflicts in *bison* grammars. *LRXplain* bases its analysis on the *G*.**output** file produced for a grammar *G* by a call of

bison --report=itemset,look-ahead *G*.**y.**

Although *bison* indicates which LALR states harbor conflicts, and even shows the contents of all states, finding the cause of a conflict is difficult, for two reasons. First, the cause of the conflict usually lies several steps away from the point of detection, the inadequate state. This is because an LALR parser tries to postpone the resolution of a conflict as long as possible, hoping in the end a look-ahead will solve the problem. When in a state *S* it becomes clear that that will not happen, *S* is declared inadequate. Finding the cause of the conflict from the inadequate state forces the user to search back through the transition table, a task better done by computer. The second is that it is often far from obvious how the parser can arrive at the inadequate state at all.

LRXplain starts by printing a short input sequence which brings the parser in the inadequate state *S*, and the look-ahead involved. A marker # in this sequence indicates the point in which the choice arises that could not be resolved in state *S*. This is followed by two sets of core items valid at #, one for each of the choices. This information is often sufficient to allow a conclusion like

> "After this input, with this look-ahead, the parser cannot decide whether the segment from the # until the look-ahead is the beginning of one rule or of another."

Under the **−v** option, the LALR parsing stacks are shown, in three segments: the common segment; the segment from # to look-ahead for choice 1; ditto for choice 2. The items in the states on these stack are shown, but only those that could be active in the complete reduction of the stack once the choice has been made. This is often only a fraction of all the items in a state, which simplifies understanding.

The **−vv** option causes *all* items in the stack segments to be printed; the active items are then marked by @s. (The **−vvv** option shows intermediate stages of how the activity information is carried back through the stacks; this produces more output than you should like to look at.)

(There are two secret options: **−T** (print the transition table) and **−M** (print a memory usage report).)

Often a grammar exhibits more that one inadequate state. By default *LRXplain* analyses and shows one conflict for each such state. A specific state can be singled out by supplying its number in a **−s** *N* option. In that case all different two-way conflicts in that state are analyzed and shown. Multiway conflicts are handled as if composed of two-way conflicts.

EXAMPLES

The grammar `test.y`

```
%%
S: 'x' T;
T: A | B;
A: 'y' C 'a' 'a';
B: 'y' D 'a' 'b';
C: 'c';
D: 'c';
```

processed by the calls

```
bison -d --report=look-ahead,itemset test.y
LRXplain  test.output
```

produces

```
There is 1 conflict in state 8
```

```
Input for reaching state 8:
'x' # 'y' 'c' ['a']
<<<< reduce 1
  A: 'y' . C 'a' 'a'
==== reduce 2
  B: 'y' . D 'a' 'b'
>>>>
```

This says that with an input string `xyc` the parser cannot see if `yc` is the start of an `A` or a `B`, even given a look-ahead `a`. A call of **LRXplain −v** reveals that this is because `c` can be reduced to both `C` and `D`:

```
<<<< reduce 1
Stack element 1: 'x' ->
  S: 'x' . T
Stack element 2: 'y' ->
  A: 'y' . C 'a' 'a'
Stack element 3: 'c' ->
  C: 'c' .  ['a']
==== reduce 2
Stack element 1: 'x' ->
  S: 'x' . T
Stack element 2: 'y' ->
  B: 'y' . D 'a' 'b'
Stack element 3: 'c' ->
  D: 'c' .  ['a']
>>>>
```

SEE ALSO

bison(1)

BUGS

Occasionally items are marked active that cannot be involved in processing the indicated input. This is because *bison* produces LALR(1) items and the *LRXplain* algorithm expects LR(1) items.

AUTHOR

Dick Grune, dick@dickgrune.com

COPYRIGHT