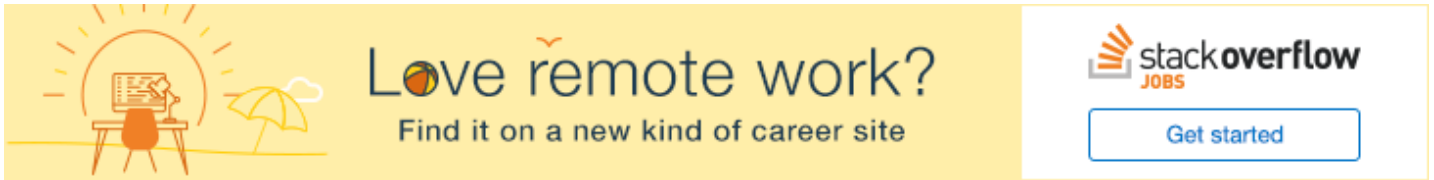


Save and restore parser state manually

[Ask Question](#)

0 I've setup a flex/bison system that can run and parse through yyparse and is able to be used inside a repl-type system. When a user enters a certain state (say: defining a method), and an systax error has been made, i'm looking for a way to deal with this in such a way, that bison can act like the user has never entered the line at all.

For instance:

```
1 1> class foo { <enter>
  2>   a = 1;  <enter>
  3>   asfasdfa <enter>
```

In this case, the parser should return back to the state it was prior to reading the last line. Right now, it will trigger a syntax error and rewinds the complete stack.

I know that bisons yyparse() function has got some local stack state values (yyvsp, yyssp and yyfsp) that is used for state tracking, but there seems no way to modify these outside the standard functionality like YYERROR etc.

My question is: is it even remotely possible to do these save/restore states in bison, and if not, what is the best way to deal with such errors.

[bison](#) [read-eval-print-loop](#)

[share](#) [improve this question](#)

[add a comment](#)

asked May 17 '13 at 15:23



[JayTaph](#)

2,041 1 15 28

2 Answers

[active](#)

[oldest](#)

[votes](#)

1 bison doesn't have a facility to save and restore parser states. You could probably create one by reverse-engineering the bison skeleton, but that would be a lot of work, and it would also be quite fragile since bison's runtime might change without notice. (To be fair, you're not forced to update bison, but you probably want the flexibility to be able to do so.)

However, parsing is pretty fast. In a REPL environment, you don't notice the parsing time. So there's nothing stopping you from simply reparsing the input from the beginning, up to the line before the error.

However, there's a catch: bison parsers don't necessarily detect a syntax error until the *next* token is read (although sometimes they do, depending on the precise nature of the error). So you cannot be sure that the last line read is the line actually containing the syntax error.

[share](#) [improve this answer](#)

answered May 18 '13 at 4:44



[rici](#)

123k 12 100 163

I'm not 100% sure if the right way to implement a repl is the way i suggested. For instance, i can only detect the states from the start of a userline until the error. But it might be possible that other rules on the line actually succeeded (for instance: "i=1; b=;") – [JayTaph](#) May 19 '13 at 7:53

[add a comment](#)



Tired of recruiter spam?

Want jobs tailored to your needs?


[Get started](#)

You should have a look at push parsers in Bison, they might provide you with the control you are looking for.

0

http://www.gnu.org/software/bison/manual/html_node/Push-Decl.html

[share](#) [improve this answer](#)

answered May 20 '13 at 12:46



[akim](#)

3,540

19

32

if there were a `yystate_copy()` interface, then it would provide exactly what OP is looking for. But there doesn't seem to be such a thing. Of course, copying the stack is tricky since it can contain arbitrary user objects, so I can understand why such an interface is not present. In the case of a REPL, parse checkpointing is over-engineering, but there may be some use for it, if you're looking for feature requests. – [rici](#) May 20 '13 at 23:22

[add a comment](#)