DPL

(P3)

Yeray Méndez Romero

[yeray.mendez@udc.es](mailto:yeray.mendez@udc.es)

Daniel Rivera López

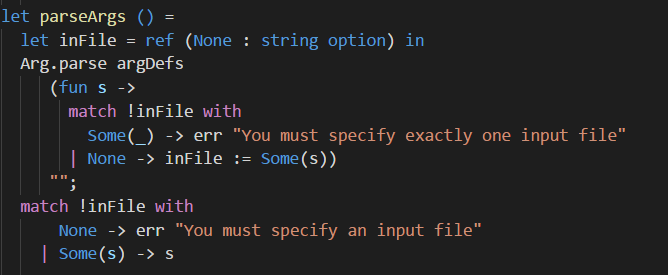
[d.rivera1@udc.es](mailto:d.rivera1@udc.es)

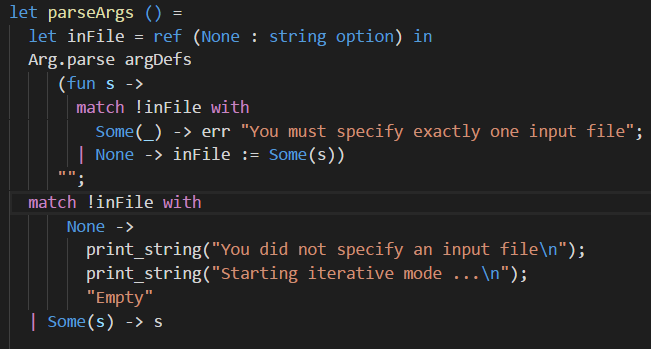
Changes to the original code:

In order to build the top level, we needed to make some changes to the original implementation:

1. Changes to ‘parseArgs’:

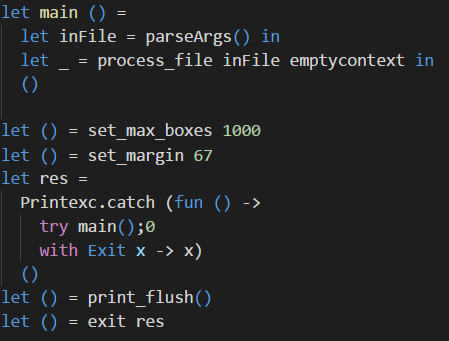
The error that happened when no input was provided has been changed to return a value ‘Empty’ that will be later used to start the top level.

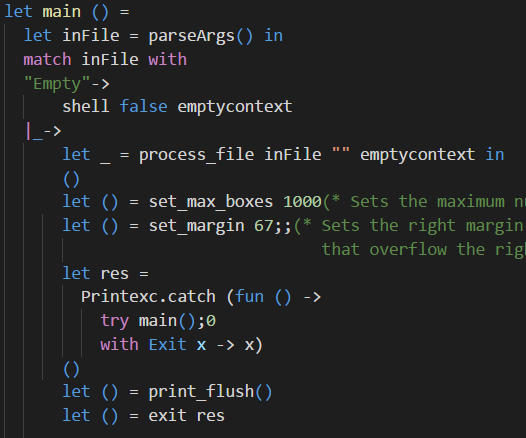




1. Changes to ‘main’:

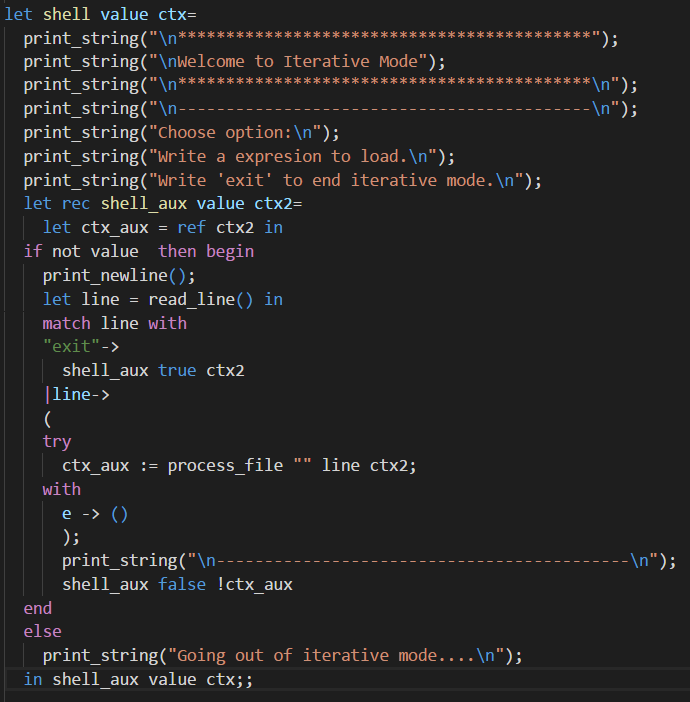
Now the ‘inFile’ in ‘main’ is matched to detect “Empty” and start the top level.





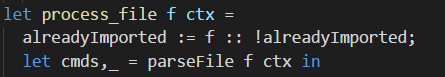
1. New function ‘shell’ added:

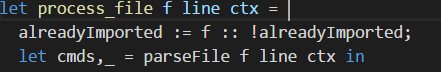
This new function starts the top level and process each line wrote by the user.



1. The function ‘process\_file’ now receives a new parameter:

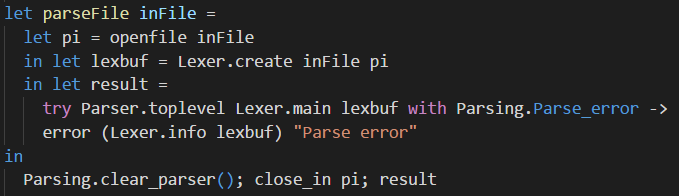
This new parameter ‘line’ represents each line that the user writes on the top level.

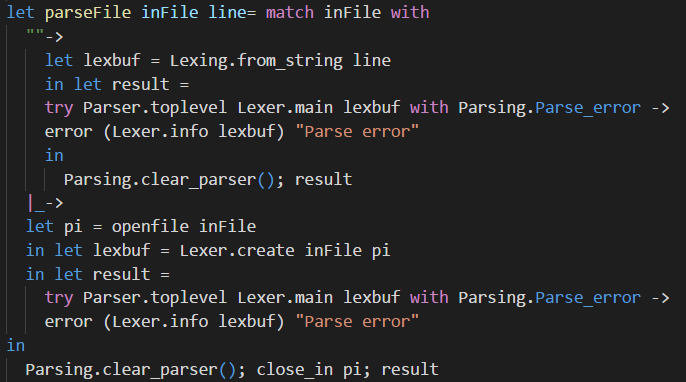




1. Changes to ‘parseFile’:

Now receives the new parameter ‘line’ and use it when the ‘inFile’ does not contain anything to build a ‘lexbuf’ with it and send it to the Parser.





The top level:

-User manual:

To start the top level just execute the program with no arguments.

To end the top level write ‘exit’ on the standard input and press Enter.

Inside the top level you can write lambda expressions on the standard input and press Enter to make the program process them.

-Performance:

When the shell is running, he process all lines that receive until he reads a “exit”, at this moment the user go out of iterative mode. The output of each processed line by the interpreter will be showing by prompt.