

Dynamic Syntax Checking

Ari Chae, Caitlin Riley,
Jazmin Gonzalez-Rivero

The Problem

- Errors don't provide enough information
 - Only return “parsing error” or “evaluation error”
- Error checking only occurs when the person hits enter

The Solution

- Change the parser and the evaluator to carry and present information if an error occurs
- Create a simple Windows Form IDE in C# that checks the user's program when they stop typing and returns error messages

The Structure

- Homework 4 and lecture 10 integration
- Error gathering
- Message production
- Integration with C#

Dead Demo

Input:

[3,2,4,]

Output:

Parsing error: error in list- expected expr

[3 , 2 , 4 , 'expr']

Dead Demo

Input:

{a=20; b=20}

Output:

Parsing error: cannot tokenize ; b=20}

Dead Demo

Input:

`filter (\x true) [1,2,3,4]`

Output:

Parsing error: error in function- expected right arrow
(/ x '->'true) [1 , 2 , 3 , 4] Function Call Failed

Dead Demo

Input:

let max a = if (a<b) then b else a in max 10 20

Output:

Evaluation error: There is no function called b, please check your spelling or your inputs

Dead Demo

Input:

let double x = x + in double 10

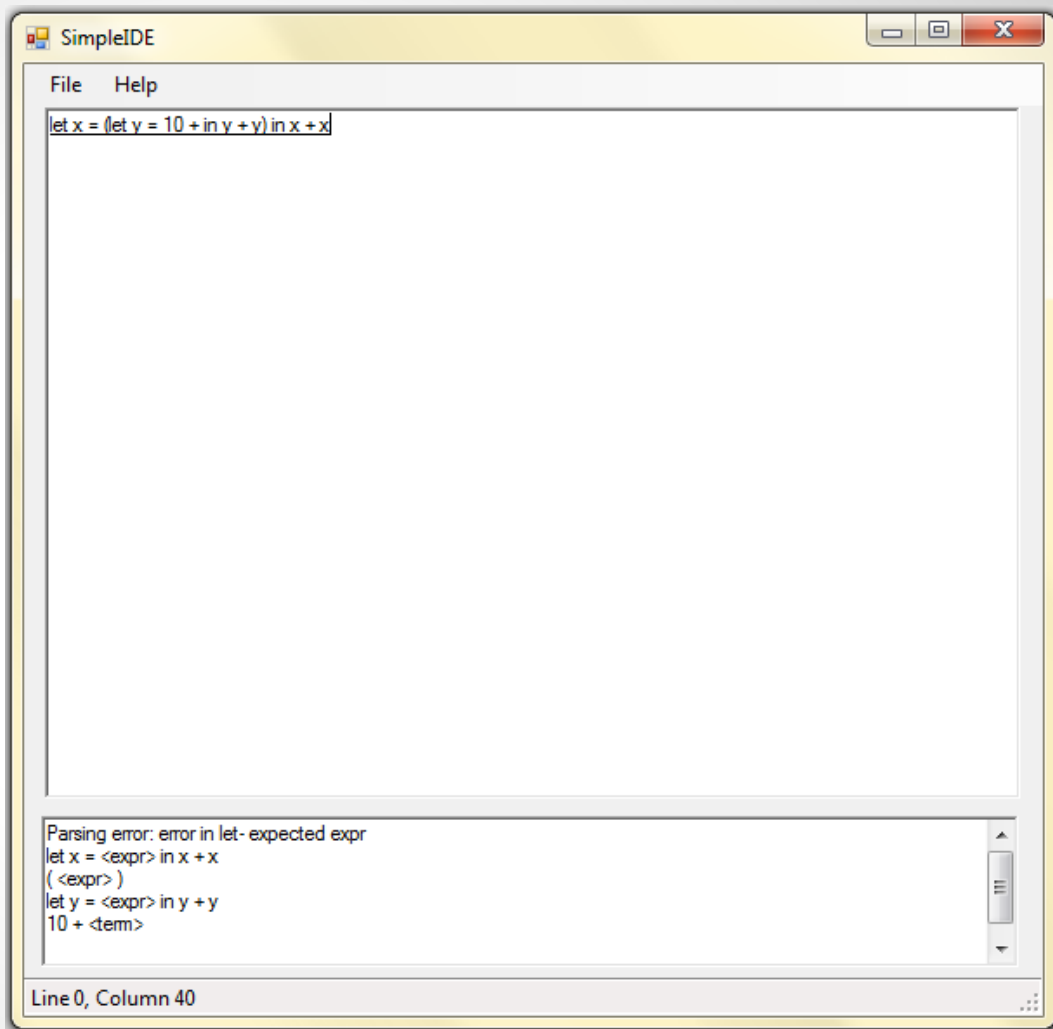
Output:

Parsing error: error in let- expected expr

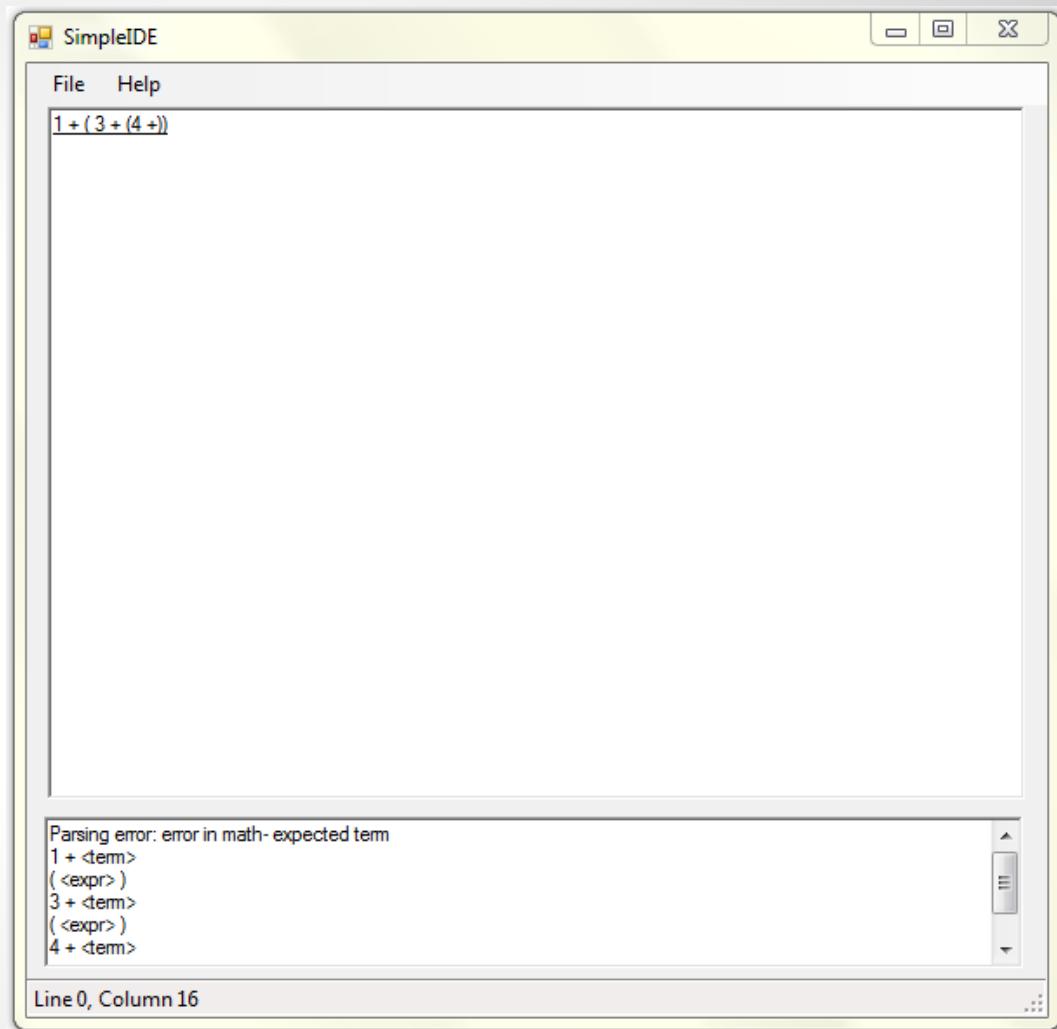
let double x = <expr> in double 10

x + <term>

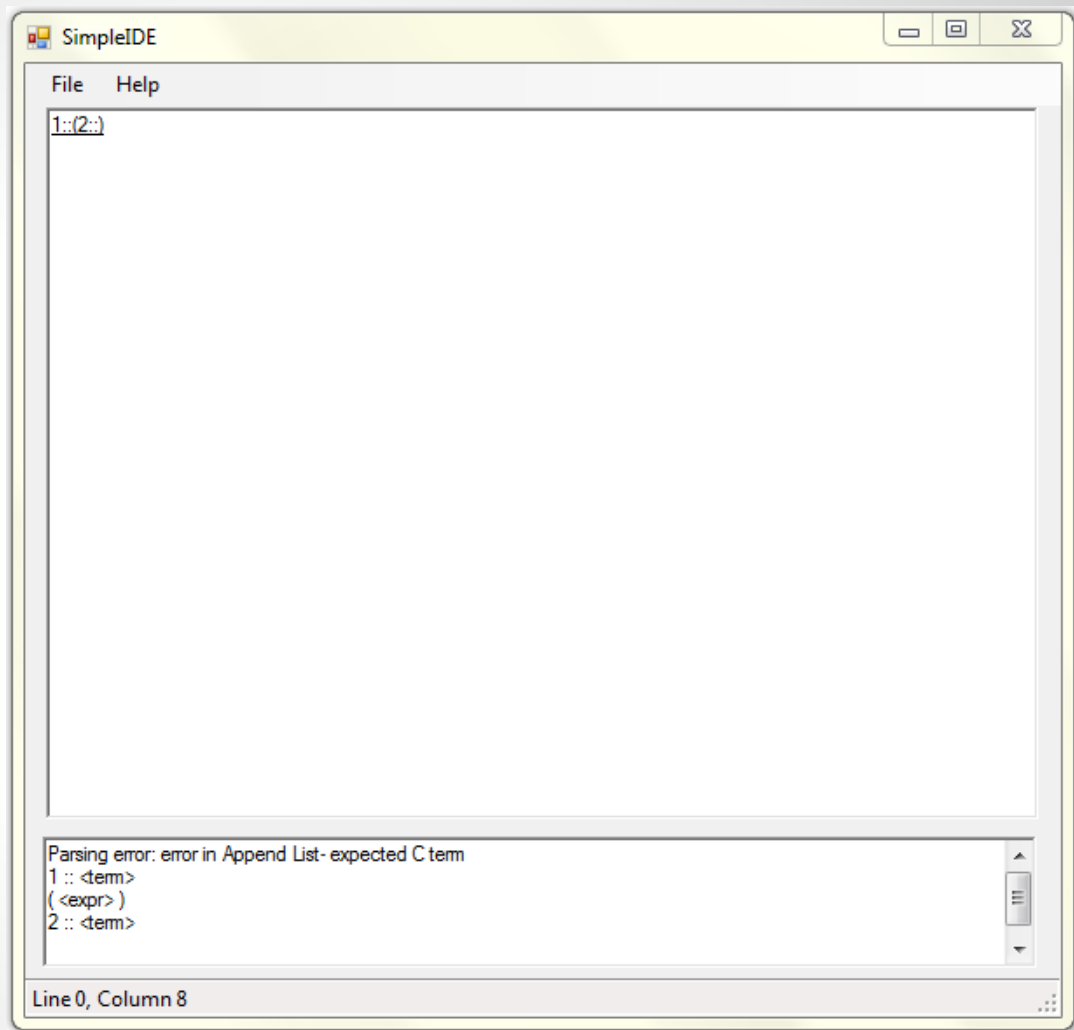
Dead Demo



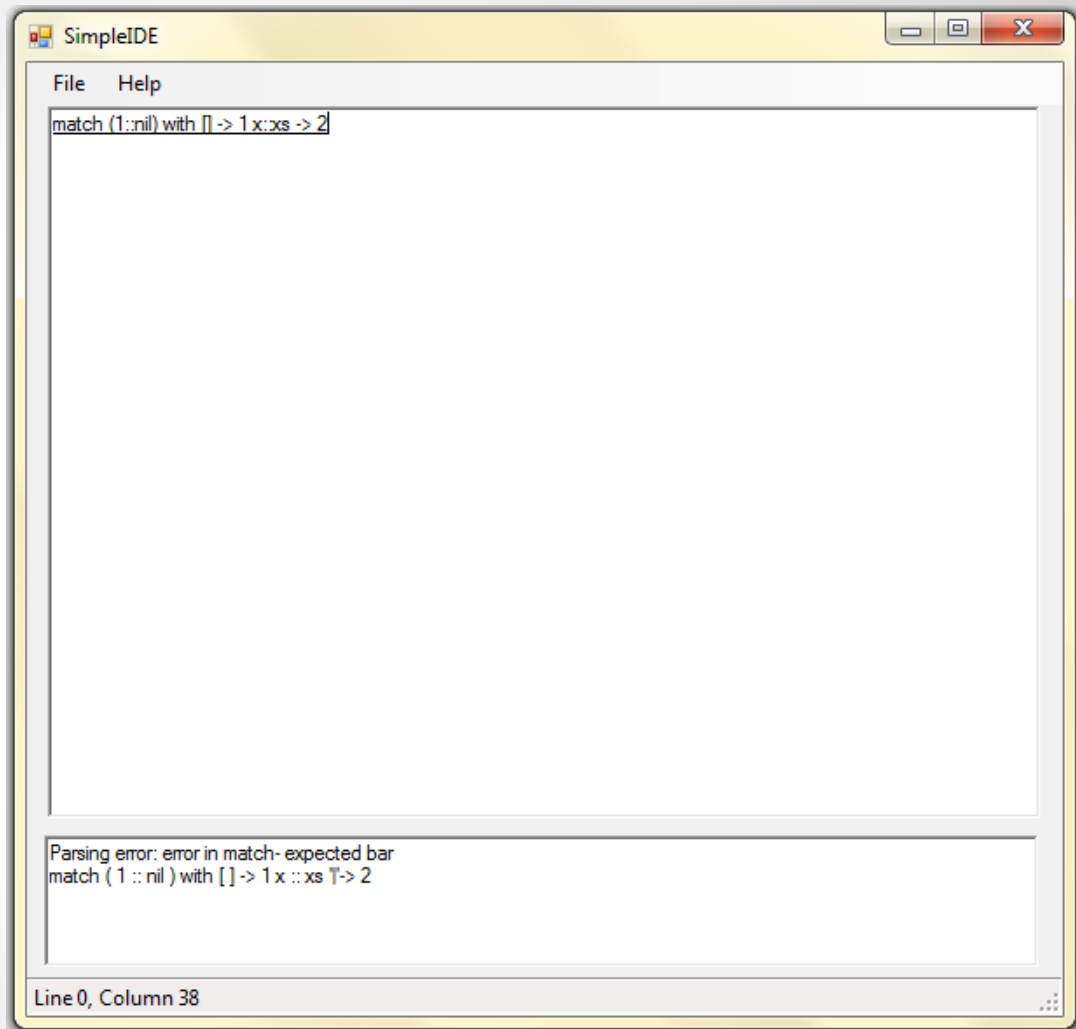
Dead Demo



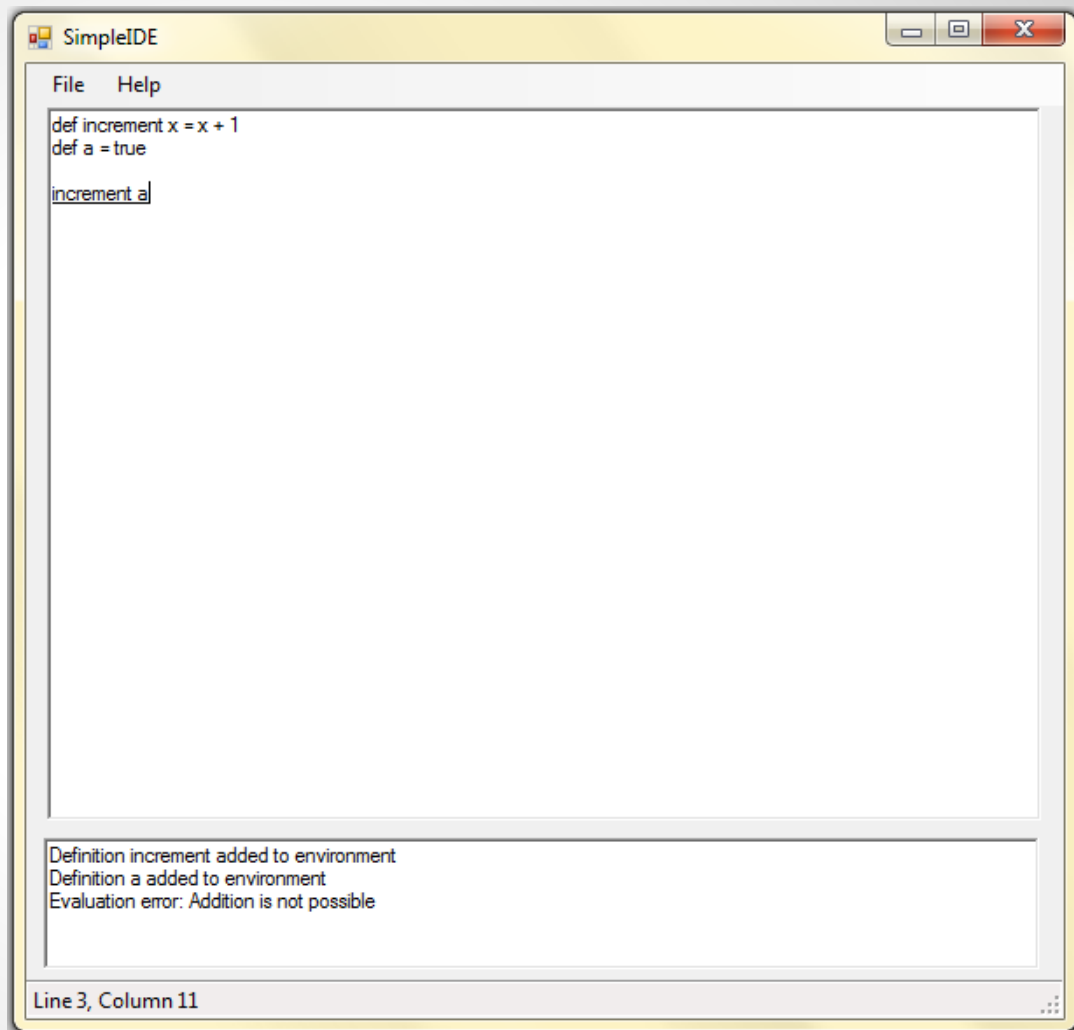
Dead Demo



Dead Demo



Dead Demo



Challenges

- Present errors only relevant to the actual expression entered by the user and not the other ones checked by the program

Example: `let x = 10 + in x + x`

Challenges

- Functions that are syntactically the same before forking

Example: Map - [e1 | s <- e2]

Filter - [e1 | s <- e2, e3]

Challenges

- Integrating SML with C#

Next Steps

- Syntax Completion
- Line Numbering

Questions?

