

CSCE 531 Spring 2020 Project 2a Type Checking and Interpretation

Project 2 builds on Project 1 (through 1c) and is a combination of assignments 2 and 3 in [R] (pp.195-196).

More complete descriptions are provided in Arne Ranta's website for [R], <http://www.grammaticalframework.org/ipl-book/assignments/assignment2/assignment2.html>. We will follow very closely the description by Dr. Ranta's colleague, Andreas Abel, at <http://www.cse.chalmers.se/edu/year/2018/course/DAT151/laborations/lab2/lab2.html>. Dr. Abel's 2018-2019 assignment includes both a Type Checker and an Interpreter; this makes sense, since the Interpreter is a "straightforward variant of the type checker" (section 5.6 [R]).

As noted in the last update for Project 1c (https://cse.sc.edu/~mgv/csce531sp20/pr/project1c_Update2.pdf), the grammar in section 2.10 [R] is (slightly) different from the ones on the book website. For this assignment, you must use the grammar of Dr. Abel's 2018-2019 assignment, which is a variation of CPP described in detail at <http://www.cse.chalmers.se/edu/year/2018/course/DAT151/laborations/lab2/lab2.html>, and found at the direct link <http://www.cse.chalmers.se/edu/year/2018/course/DAT151/laborations/lab2/ CPP.cf>.

For this assignment, you must use Haskell as a programming language. You may use one of Windows (as I do), Linux, and Mac OS. See tips in the project section of the course site, concerning the use of Powershell, the installation of make, etc.

You will submit two pdf files on the departmental dropbox two files. The details are below.

1. Run `bnfc -m CPP.cf`. Then, run `make`. Then, copy the CPP program below in file `test1.cc` and run the resulting parser on this file. The parse should succeed.

```
/* Andreas Abel, dont-bind-too-eagerly.cc, from
http://www.cse.chalmers.se/edu/year/2018/course/DAT151/laborations/lab
2/testsuite/
*/
int snd (int x, int y) {
    return y;
}

int main () {
    int x = 0;
    int r = snd(1,x);
    printInt(r); // Should print 0
    return r;
}
```

The first file you submit should contain the list of files generated by `bnfc -m CPP.cf`, the list of files generated by `make`, and the result of running the parser thus obtained on file `test1.cc`. Call this first file `pr2a1.pdf`

2. For the second part of the project, you will not need the Makefile, because a different one will be provided to you. Run `bncf CPP.cf`. Create a pdf file with a script or screenshot showing the resulting files in your directory or folder. Call the second file `pr2a.pdf`.