CS152 Homework: Type Systems

Due Tuesday, March 15, at 11:59 PM, *except for Ramsey & Kamin problem 1*, which is due Tuesday, March 15 at 5:00PM. Note that this time is *before* the rest of the assignment.

The purpose of this assignment is to help you learn about type systems.

Setup and Guidelines

In your cs152 directory create a subdirectory called `typesys`. Copy the `tuscheme.sml` file from `~cs152/software/bare/tuscheme/tuscheme.sml`. Also copy the `timpcore.sml` file from `~cs152/software/bare/timpcore/timpcore.sml`.

As in the ML homework, use function definition by pattern matching. In particular, do not use the functions `null`, `hd`, and `tl`.

The problems

1. [16pts] Do Exercise 1 on page 244 of Ramsey and Kamin: write type rules for lists. **Turn in this exercise on paper by 5:00PM Tuesday, March 15**.

Hint: Exercise 1 is more difficult than it first appears. I encourage you to scrutinize the lecture notes for similar cases, and to remember that you have to be able to type check every *expression* at compile time.

2. [16pts] Do Exercise 2 on page 244 of Ramsey and Kamin: finish the type checker for Typed Impcore by implementing the rules for arrays. Turn in this exercise in file `timpcore.sml`.
My solution to this problem is 21 lines of ML.

11. [14pts] Do Exercise 11 on page 245 of Ramsey and Kamin: write `exists?` and `all?` in Typed uScheme. Turn in this exercise in file `11.scm`.

13. [40pts] Do Exercise 13 on page 245 of Ramsey and Kamin: write a type checker for Typed uScheme. Turn in this exercise in file `tuscheme.sml`, which should also include your solution to Exercise 15 below. Don't worry about the quality of your error messages, but do remember that your code must compile *without errors or warnings*.
My solution to this problem is about 120 lines of ML. It is very similar to the type checker for Typed Impcore that appears in the book. It could have been a little shorter if I had given worse error messages.

16. [20pts] Do Exercise 16 on page 246 of Ramsey and Kamin: add sum types to Typed uScheme. As a hint, remember that you can call `eval` from a primitive function. Turn in the code for this exercise in file `tuscheme.sml`, which should also include your solution to Exercise 13 above. Please include the answers to parts (a)–(c) in your README file.
My solution to this problem is about 15 lines of ML.

What to submit

Submit the files `README`, `timpcore.sml`, `11.scm`, and `tuscheme.sml`. In comments at the top of your `README` file, please include your name, the names of any collaborators, and the number of hours you spent on the assignment.

If you want to submit up to 3 test cases to be applied to other people's type checkers, submit them in a file called `tests`, complete with comments that tell us how to run them.

Submit using `submit-typesys`.