CS 565 Out: November 6, 1996
Jens Palsberg Due: November 13, 1996

Homework 11:

Efficient Recursive Subtyping

1. Implement the algorithm from *Efficient Recursive Subtyping*. Use the grammar in the file `types.scm` on the CS 565 homepage. For a given subtype problem, your solution must output `#t` or `#f`. Here are two examples taken from *Efficient Recursive Subtyping*.

```
> (run "mu u.((u -> u) -> bot) <=
        mu v.((v -> bot) -> top)")

#t


> (run "mu v.(v -> bot) <=
        mu u.(u -> top)")

#f
```

Note: part of the problem is to figure out how to translate a type expression into an automaton. You can rely on that types do not contain black holes.

2. (Extra credit) Extend your solution to Problem 1 with a check such that an error is reported if a black hole occurs in one of the types.