COMP 321 Fall 2021 Homework 0:
Operational Semantics, λ-calculus

This is a short(er than usual) written homework for you to begin getting used to some of the concepts we will cover in the class.

To hand in, please upload a file `hw00.pdf` to your Google Drive handin folder.

1 Operational Semantics

For this problem, you may want to review the grammar, step rules, and progress proof for number programs in the Lecture 1 board.

1.1 Simple Traffic Light

A standard traffic light can be either red, yellow, or green (so we are modeling one side/direction of the light, not the whole thing). For this problem, we will think of the colors of the traffic light as "expressions" and give them an operational semantics.

Task 1 (3 points). Give a grammar for the colors.

Task 2 (5 points). Give operational semantics (stepping rules) that codify the usual transitions of a traffic light.

Task 3 (7 points). State and prove a *progress theorem* that says that you didn't forget any step rules.

1.2 Walk Light

Next, we will consider a traffic light with a pedestrian walk sign. The light should cycle as usual between colors, except *every other* time it turns red, it also illuminates a pedestrian walk sign (with the idea that the opposing light is also red at that the time, so pedestrians can cross diagonally). The walk sign should only be illuminated when the light is red.

Task 4 (15 points). Give a grammar and operational semantics for this kind of traffic light.

2 λ-calculus

In SML, we can represent the simple traffic light as a datatype

```
datatype Light = Red | Yellow | Green
```

which can give a case-analysis operation

```
case (e : Light) of
  Red => e1
| Yellow => e2
| Green => e3
```

As a concrete example of using case analysis, the following function scrambles the colors in an arbitrary way:

```
fun scramble(c : Light) =
  case c of
    Red => Green
  | Yellow => Yellow
  | Green => Red
```

Task 1 (20 points). Encode this datatype in the untyped $lambda$-calculus: Give definitions of `Red`, `Green`, `Yellow` and case analysis for `Light`. Show that the following operational semantics steps hold in the untyped λ-calculus:

```
(case Red    of Red => e1 | Yellow => e2 | Green => e3) |->* e1
(case Yellow of Red => e1 | Yellow => e2 | Green => e3) |->* e2
(case Green  of Red => e1 | Yellow => e2 | Green => e3) |->* e3
```

Hint: As a model, you should use the encoding of natural numbers discussed in Friday's class. You also might want to look at the encoding of Booleans, which follows a similar pattern but is simpler — see Exercise 21.3 in PFPL (and the solution on page 509 of the online preview).