CS51 FINAL PROJECT: IMPLEMENTING MINIML
FURTHER NOTES

STUART M. SHIEBER

Students may find some of the notations in the final project writeup unfamiliar, especially in Figure 1. In this document, I describe some of the notations for reference.

1. Equational definition of functions

Mathematics is full of functions, and of ways of defining them. A standard technique is to define functions using a set of equations. Each of the equations provides a part of the definition based on a particular subset of the possible argument values of the function. For instance, the factorial function, denoted by a postfix ! in standard mathematical notation, is defined by these two equations:

$$0! = 1$$

$$n! = n \cdot (n-1)! \quad \text{for } n > 0$$

Notice the following conventions:

- A 'for' or 'where' clause after an equation provides further constraint on the applicability of that equation. In the case at hand, the second equation applies only when the argument n is greater than 0.
- Mathematics often uses different conventions for denoting operations than any given programming language. Here, for instance, a center dot $\cdot$ is used for multiplication instead of the * more common in programming languages. In other cases, juxtaposition is used for multiplication, as in[1]

  $$\frac{d}{dx} x^3 = 3x^2$$

  where the juxtaposition of the 3 and the x^2 indicates that they are to be multiplied. The details of these notations are often unspecified in mathematical writing, reflecting the reality that mathematics is written to be read by *people* with sufficient common knowledge with the author to know the

Date: April 16, 2016.

[1]By the way, the notation $\frac{d}{dx}x^3$ is yet another example of a nonstandard notation for a function application. Here, the function being applied is the derivative function $\frac{d}{dx}$, its argument the expression x^3. The notational profligacy of mathematics – especially having many different notations for functions – hides a lot of commonality shared among mathematical processes.

background assumptions or to figure them out from context. We don't have such a privilege with computers, so notations are typically more carefully explicated in programming language documentation.

- The kind of thing that the argument must be is often left implicit in mathematical notation. In the factorial example, we didn't state explicitly that the argument of factorial must be a nonnegative integer, yet the definition is only appropriate for that case. Negative integers are not provided a well-founded definition for instance, nor are noninteger numbers. Again, the omission of these requirements is based on an assumption of shared context with the reader. So as not to have to make that assumption, computer programs that implement function definitions make use of type constraints (whether explicit or inferred) or invariant assertions or (as a last resort) documentation to capture these assumptions.
- The entire set of equations defines a single function, so that in converting definitions of this sort to code, they will end up in a single function definition. The individual equations correspond to different cases, which will likely be manifest by conditionals or case statements (such as OCaml `match` expressions).

2. Glossary of notations

In the definition of the FV function (which you will implement in the form of `free_vars`) we take advantage of some standard set notations, which we review here.

EMPTY SET

- The EMPTY SET, notated $\varnothing$, is the set containing no members.

EXTENSIONAL

- An EXTENSIONAL set definition (given by an explicit list of its members) is notated by listing the elements in braces separated by commas, as, for instance, $\{1, 2, 3, 4\}$. Obviously, this notation only works for finite sets, although infinite sets can be indicated with ellipses (as $\{1, 2, 3, \ldots\}$) in cases where the rule for filling in the remaining elements is sufficiently obvious to the reader.

INTENSIONAL

- An INTENSIONAL set definition (given by describing all members of the set rather than listing them) is notated by placing in braces a schematic element of the set, followed by a vertical bar, followed by a description of the range of any variables in the schema. For instance, the set of all even numbers might be $\{\, x \mid x \pmod 2 = 0 \,\}$, read "the set of all x such that x is evenly divisible by 2." Similarly, the set of all squares of prime numbers would be $\{\, x^2 \mid x \text{ is prime} \,\}$. (Note the combination of mathematical notation and natural language, a typical instance of "code switching" in mathematical writing.)

- The standard operations on sets are notated with infix operators:

  Union: $s \cup t$ is the UNION of sets s and t, that is, the set containing all the elements that are in either of the two sets; UNION

  Intersection: $s \cap t$ is the INTERSECTION, containing just the elements that are in both of the sets; INTERSECTION

  Difference: $s - t$ is the set DIFFERENCE, all elements in s except for those in t; and DIFFERENCE

  Membership: $x \in s$ specifies MEMBERSHIP, stating that x is a member of the set s. MEMBERSHIP

  By way of example, the following are all true statements, expressed in this notation:

$$
\begin{aligned}
\{1, 2, 3\} \cup \{3, 4\} &= \{1, 2, 3, 4\} \\
\{1, 2, 3\} \cap \{3, 4\} &= \{3\} \\
\{1, 2, 3\} - \{3, 4\} &= \{1, 2\} \\
3 &\in \{1, 2, 3\} \\
3 &\notin \{2, 4, 6\}
\end{aligned}
$$

Note the use of a slash through a symbol to indicate its NEGATION: $\notin$ for 'is not a member of'. NEGATION

There are different notions of identity used in mathematical notation. The $=$ symbol typically connotes two values being the same "semantically". The $\equiv$ symbol connotes a stronger notion of syntactic identity, so that $x \equiv y$ means that x and y are the same syntactic entity (variable say) rather than that they have the same value (in whatever context that might be appropriate). For instance, consider these equations found in the definition of substitution

$$
\begin{aligned}
x[x \mapsto P] &= P \\
y[x \mapsto P] &= y \qquad \text{where } x \not\equiv y
\end{aligned}
$$

(Recall that $P[Q \mapsto x]$ specifies the expression P with all free occurrences of x replaced by the expression Q (with care taken not to capture any free occurrences of x in Q). The notation $x \not\equiv y$ indicates that the variable y that constitutes the expression being substituted into is a different variable from the variable x that is being substituted for. (In this context, x and y don't have natural values that might or might not be identical in any case.)