Course Comments
- Some of the reading for this class was confusing or I feel like we didn't have enough of it. I would like a Scala-based assigned textbook to have to read every week, and I really liked the professor's PDF notes he posted. ... I think the professor and the TA did a really good job updating assignments and generally being available. Agreed!
- I came into this class hoping to gain a better understanding of how programming languages work, and I think this is what I got. Awesome!
- It was very nice to have professor and TA who greatly cared and put serious effort into their teaching. A lot of classes will say that's more important to understand the material than to worry about your grade, and this is one of the few classes I've taken where I have actually felt that this was true. Thanks!

Lab 6 Comments
- I found the hardest part to be debugging the parser. Once all the cases were implemented, it became hard to tell which one was causing a failure. Sometimes a failure in one case was only due to an incorrect implementation in another. ... I really enjoyed this lab. It was an excellent way to sum up everything we've learned this semester.
- The continuation passing test function was very subtle; it was easy to feel "almost there" without having the right structure at all.

Lab 6: 14.23 avg, 5.7 stdev, 14.25 median | Hard: 4.4 avg, 0.89 stdev, 4 median

T-Shirt Design Competition
e::=n le1+c2

Subtyping

VAR x: [1 to 10]
x:=1

INTEGER:=[1 to 10]
\[ \text{abstract } [1 \rightarrow 1] \quad \text{concave } 0.1 \leq \quad [1 \rightarrow 10] \]

\[ 1704 \leq 17040 \]

\text{Subtyping}
$\exists_1 < : \exists_2$  $\exists_1$ is a subtype of $\exists_2$

$[[1 \text{ to } 2]] < : [[1 \text{ to } 10]]$

$[[1 \text{ to } 10]] < : \text{INTEGER}$

Subtyping is a relation between two types

- If $\exists_1 < : \exists_2$, then an expression of type $\exists_1$ can be used in a context that expects a $\exists_2$

$\exists_1 \rightarrow \exists_2$, widening conversion / upcast

$\exists_2 \rightarrow \exists_1$, narrowing conversion / downcast

Reflexive

Transitive

Transitive

Transitive
1. Define subtyping for base types based on "subset interpretation"

\[ \text{short <: integer} \quad \text{and} \quad [a \to b] <: \text{integer} \]

2. Define subtyping on constructed types based on their components

\[ \begin{align*}
\text{Unsafe subtyping} \\
\text{int <: int} & \quad 3(1) \\
\end{align*} \]

\[ \begin{align*}
\Gamma \vdash e : z_i & \quad \Gamma \vdash e : z_i \\
\text{Subsumption} & \quad \Gamma \vdash e : z_i \\
\end{align*} \]

Immutable Records

\[ \begin{align*}
& \quad z_i : z_i' \\
\text{Depth subtyping} & \quad \begin{cases} 
\text{f}_1 : z_i, \ldots, \text{f}_n : z_i' \\
\text{g}_1 : z_1, \ldots, \text{g}_n : z_n' 
\end{cases}
\end{align*} \]