CS225 Spring 2018—Final Project Proposal

https://github.com/lgove/cs225

Madison Anderson git:@mander34

LeAnn Gove git:@lgove

Nikki Allen git:@nikkiallen

April 20, 2018

Project: System F with Bounded Quantification

We propose to implement a semantics and type checker for System F extended with bounded quantification.

Base Language We will work with simply typed lambda calculus with booleans, natural numbers, let-binding, and products as the base language.

Extended Language We will extend this language with the features of System F and bounded quantification. This consists of a new type:

1. A universal type variable, written $\forall X <: T.T$
2. An existential type variable, written $\exists X <: T, t$

a new value:

1. Type abstraction, written $\lambda X. <: T.t$

Applications Bounded quantification, in itself, is equivalent to combining Subtyping and Universal Quantification. This substantially increases both the expressive power of the system and its metatheoretic complexity. The calculus used in this language, written F <: ("F sub"), has played a large role in programming language research since it was developed in the 1980's, particularly in relation to the studies on the foundations of object-oriented programming.

Project Goals For this project, we plan to complete:

1. A small-step semantics for System F extended with bounded quantification
2. A type checker for System F extended with bounded quantification

Expected Challenges We expect the need to support to make the semantics as we have not explored this topic in depth in class and thus don't have clear notes and only the textbook as reference material. We also expect the implementation of the typechecker to be more challenging than what we have seen in class thus far.

Timeline and Milestones By the checkpoint we hope to have completed:

1. A prototype implementation of the small-step semantics
2. A suite of test-cases for the small-step semantics and well-typed relation
3. One medium-sized program encoded in the language which demonstrates a real-world application of the language

By the final project draft we hope to have completed:

1. The full implementation of small-step semantics and type checking
2. A fully comprehensive test suite, with all tests passing
3. The medium-sized program running through both the semantics and type checker implementation
4. A draft writeup that explains the on-paper formalism of our implementation
5. A draft of a presentation with 5 slides as the starting point for our in-class presentation

By the final project submissions we hope to have completed:

1. The final writeup and presentation
2. Any remaining implementation work that was missing in the final project draft