Assignment 7
Parametricity

15-814: Types and Programming Languages
Jan Hoffmann & C.B. Aberlé

Due Thursday, November 7, 2024
75 points

1 Mini Project Proposal

Important: Briefly discuss your project idea with the instructor before writing the project proposal.

The final assignment for this course is a mini project. The work on the project will start on November 16 and is due on December 7. *You can complete the final project individually or with a partner.* You have the following four options for the project.

Continuations In this project you will develop the semantics of continuations and extend an implementation of the K machine with continuations. This project has a mix of theory and implementation.

Denotational Semantics In this project you will develop a denotational semantics for PCF as an extension of the domain-theoretic semantics sketched in Homework 5, and show that it is equivalent to the structural dynamic semantics that we discussed in lecture. This project is theory only.

Formalizing the metatheory of System T in dependent type theory In this project you will formalize (a variant of) the proof from assignment three in the Agda programming language/proof assistant, using dependent type theory under the propositions-as-types interpretation to prove properties of programs. This project is implementation only, but will expose you to both the theory and practice of using dependent type theory as a proof system. *Warning:* You should only pick this project if you have experience with a proof assistant like Agda or if you are able to invest a lot of time.

Custom Project If you have an alternative idea for a project then you can submit a project proposal in this assignment. More information follows.

1.1 Project Proposal for Custom Project

Even if you later choose to do one of the predefined projects, we want you to explore topics in programming language research and develop a project proposal in this assignment. *You can only work on a custom project if your submitted project proposal has been accepted by the course staff.* If your proposal has been accepted, you can still choose to work on one of the three predefined projects.

A custom project can be a short paper or an implementation. If you choose an implementation, you still have to submit a pdf that describes the implementation and how to use it.

It is good to be ambitious but keep time management in mind. The total time you spend with the project should be similar to the time you have spent with 2.5 of the weekly homework assignments so far.

We highly suggest agreeing with the instructor on a topic before beginning to write your custom project proposal, especially if you wish to pursue a custom project over the predefined ones.

Task 1 (10 points) *Your proposal should be roughly* 1 page long *and include the following parts.*

1) *Title and project type (short paper or implementation)*

2) *Motivation (Why are you interested in this project? How does it relate to the topics discussed in class?)*

3) *Background (Summarize the existing technical results that are most relevant for your project.)*

4) *Project description (Outline the scope and goals of your project, e.g., a sketch of the short paper.)*

5) *Milestones and timeline (Define at least three milestones with dates that you need to complete for a successful project)*

The points for this task is based on completing all the required sections of the proposal and does not imply that your project proposal was accepted.

2 Parametricity

We will work with a version of System F extended with booleans and natural numbers. *You should refer to the lecture notes for the definition of logical equivalence.*

Task 2 (20 points) (i) Prove $f\ [\texttt{nat}]\ 27\ 42 \mapsto^* 27$ given

- $\cdot \vdash f : \forall\alpha.\, \alpha \to \alpha \to \alpha$
- $f\ [\texttt{bool}]\ \mathsf{true}\ \mathsf{false} \mapsto^* \mathsf{true}$

(ii) Prove there is some natural number n such that $f\ [\texttt{nat}]\ \mathsf{succ}\ \overline{0} \mapsto^* \overline{2n+1}$ given

- $\cdot \vdash f : \forall\alpha.\, (\alpha \to \alpha) \to \alpha \to \alpha$
- $f\ [\texttt{bool}]\ \mathsf{not}\ \mathsf{true} \mapsto^* \mathsf{false}$

Task 3 (25 points) For the following subtasks, assume that $\cdot \vdash e : \forall\alpha.\, \alpha \to \alpha \to \alpha$.

(i) Show that $e\,[\tau]\,v_1\,v_2 \mapsto^* v_1$ or $e\,[\tau]\,v_1\,v_2 \mapsto^* v_2$ for closed *values* v_1, v_2 of type τ.

(ii) Verify that the previous subtask's result is sufficient to show logical equivalence. Show that either $e \approx \Lambda\alpha.\, \lambda x : \alpha.\, \lambda y : \alpha.\, x \in [\![\forall\alpha.\, \alpha\to\alpha\to\alpha]\!]$ or $e \approx \Lambda\alpha.\, \lambda x : \alpha.\, \lambda y : \alpha.\, y \in [\![\forall\alpha.\, \alpha\to\alpha\to\alpha]\!]$. This is a more powerful generalization of the previous subtask's result, since we can instantiate it at arbitrary relations between possibly-different types and values.

Task 4 (20 points) We can also use parametricity to exhibit type isomorphisms. For this task, we consider System F extended with booleans. An isomorphism between types σ and τ consists of functions $f : \sigma \to \tau$ and $g : \tau \to \sigma$ that are mutually inverse up to logical equivalence. Using parametricity, prove that $\forall \alpha.\, \alpha \to \alpha \to \alpha$ is isomorphic to `bool`. Recall the static and dynamic semantics of `bool`:

$$\frac{}{\Gamma \vdash \texttt{tt}, \texttt{ff} : \texttt{bool}} \qquad \frac{\Gamma \vdash e : \texttt{bool} \qquad \Gamma \vdash e_1 : \tau \qquad \Gamma \vdash e_2 : \tau}{\Gamma \vdash \texttt{if}(e, e_1, e_2) : \tau}$$

$$\frac{}{\texttt{tt}, \texttt{ff}\ \mathsf{val}} \qquad \frac{e \mapsto e'}{\texttt{if}(e, e_1, e_2) \mapsto \texttt{if}(e', e_1, e_2)} \qquad \frac{}{\texttt{if}(\texttt{tt}, e_1, e_2) \mapsto e_1} \qquad \frac{}{\texttt{if}(\texttt{ff}, e_1, e_2) \mapsto e_2}$$

Logical equivalence for booleans is defined as follows:

$$v \sim v' \in [\texttt{bool}] \text{ iff } v = v'$$