Assignment 8
Parametricity

15-814: Types and Programming Languages
Frank Pfenning & David M Kahn

Due Thursday, November 4, 2021
70 points

This assignment is due by 11:59pm on the above date and it must be submitted electronically as a PDF file on Gradescope.

1 Parametricity

In all the tasks below, we assume that all functions are terminating and polymorphism is parametric.

Task 1 (L15, 15 points) In lecture, we provided some initial intuition-based examples of parametricity, but never proved them rigorously. In this task, you will provide rigorous proofs for those examples. Make sure your proofs use parametricity, and not, e.g., representation theorems from early in the semester that may or may not hold here.

(i) Prove $f\ [\mathit{nat}]\ 27\ 42 \mapsto^* 27$ given

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

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

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

Task 2 (L16.3, 20 points) Now we can explore one use of parametricity. Previously in the course, we went over *representation theorems*, which tell you that expressions of a given type may only have a couple of normal forms. The proofs of these theorems were usually not particularly elegant, and it was unclear how to generalize them or scale their approach.

Now we are in a setting of evaluation rather than normalization, and will prove similar theorems. However, parametricity will make the process much cleaner, and yield a more general result. In this task, we will exemplify how to do it by working with our our old Booleans, $\forall\alpha.\, \alpha \to \alpha \to \alpha$.

For the following subtasks, assume that $\cdot \vdash e : \forall\alpha.\, \alpha \to \alpha \to \alpha$.

(i) First we prove something analagous to our old representation theorems by showing that e can only behave like $\Lambda\alpha.\,\lambda x.\,\lambda y.\,x$ or $\Lambda\alpha.\,\lambda x.\,\lambda y.\,y$. Specifically, 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 $e \approx \Lambda\alpha.\,\lambda x.\,\lambda y.\,x \in [\![\forall\alpha.\,\alpha \to \alpha \to \alpha]\!]$ or $e \approx \Lambda\alpha.\,\lambda x.\,\lambda y.\,y \in [\![\forall\alpha.\,\alpha \to \alpha \to \alpha]\!]$. 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 3 (L16, 15 points) We can also use parametricity to help out with proving another kind of property: isomorphisms. Previously, we merely applied the compositions of our functions *forth* and *back* to test inputs to test (but not prove) that they were were extensionally equivalent to the identity function. And even that was difficult, since if one of the types related by the isomorphism was itself a function, we also had to resort to testing *it* on test inputs as well.

Now, with parametricity, we can skip all that and prove extensional equivalence directly. Using parametricity, prove that $\forall\alpha.\,\alpha \to \alpha \to \alpha \cong 2$. Feel free to use the results from the previous subtask.

Task 4 (L16, 20 points) Consider the family of types

$$\mathsf{tree}\ \tau = \mu\alpha.(\mathbf{leaf} : 1) + (\mathbf{node} : \alpha \times \tau \times \alpha)$$

(i) Define $v \sim v' \in [\mathsf{tree}\ \tau]$.

(ii) Define a function *tmap* such that $v \sim v' \in [\mathsf{tree}\ R]$ iff *tmap* $[\tau]\ [\tau']\ R\ v = v'$ for $R : \tau \to \tau'$.

(iii) Derive a result that parallels Wadler's "theorem for free" on a value $f : [\forall\alpha.\,\mathsf{tree}\ \alpha \to \mathsf{list}\ \alpha]$.

(iv) Prove that every element of type τ in the list l for $f\ [\tau]\ t \mapsto^* l$ must already be present in tree t.