Assignment 10
Linearity

15-814: Types and Programming Languages
Frank Pfenning

Due Friday, December 6, 2019

Linear Functions

Task 1 (L22.3, 20 points) Recall the definition of a purely positive type, updated to reflect the notation for linear types.

$$\tau^+ ::= 1 \mid \tau_1^+ \otimes \tau_2^+ \mid \oplus_{i \in I}(i : \tau_i^+) \mid \rho\alpha^+.\, \tau^+ \mid \alpha^+$$

Even in the purely linear language, it is possible to *copy* a value of purely linear type. Define a family of functions

$$\mathit{copy}_{\tau^+} : \tau^+ \multimap (\tau^+ \otimes \tau^+)$$

such that $\mathit{copy}_{\tau^+}\ v \mapsto^* \langle v, v\rangle$ for every $v : \tau^+$. You do not need to prove this property, just give the definitions of the *copy* functions. Your definitions may be mutually recursive.

Task 2 (L22.4, 20 points) A type isomorphism is *linear* if the functions *Forth* and *Back* are both linear. For each of the following pairs of types provide linear functions witnessing an isomorphism if they exist, or indicate no linear isomorphism exists. You may assume all functions terminate and use either extensional or logical equality as the basis for your judgment.

1. $\tau \multimap (\sigma \multimap \rho)$ and $\sigma \multimap (\tau \multimap \rho)$
2. $\tau \multimap (\sigma \multimap \rho)$ and $(\tau \otimes \sigma) \multimap \rho$
3. $\tau \multimap (\sigma \otimes \rho)$ and $(\tau \multimap \sigma) \otimes (\tau \multimap \rho)$
4. $(\tau \oplus \sigma) \multimap \rho$ and $(\tau \multimap \rho) \otimes (\sigma \multimap \rho)$
5. $(1 \oplus 1) \multimap \tau$ and $\tau \otimes \tau$

Linear Processes

Task 3 (L23.2, 20 points) Write a linear function *inc* on the binary representation of natural numbers.

1. Provide the code as a functional expression.

2. Following the conventions of this lecture, show the result of the translation into a process expression. You may use the optimization we presented here. Concretely, define *inc_proc* and *inc* so that the program representation as a configuration would be !cell *inc inc_proc*.

3. Show the initial and final configuration of computation for incrementing the number 1 represented as fold $(\texttt{B1} \cdot (\text{fold } (\texttt{E} \cdot \langle\,\rangle)))$.