Models of computation (MOD) 2015/16
Exam – June 8, 2016

[Ex. 1 (1st mid-term)] Let IMP^{undo} be the variant of IMP where the **while-do** construct is replaced by the construct **while** b **undo** c, whose operational semantics is defined by the rules

$$\frac{\langle b,\sigma\rangle \to \textbf{false}}{\langle \textbf{while } b \textbf{ undo } c,\sigma\rangle \to \sigma} \qquad \frac{\langle b,\sigma\rangle \to \textbf{true} \quad \langle c,\sigma''\rangle \to \sigma \quad \langle \textbf{while } b \textbf{ undo } c,\sigma''\rangle \to \sigma'}{\langle \textbf{while } b \textbf{ undo } c,\sigma\rangle \to \sigma'}$$

Exhibit a concrete counterexample (and explain it in detail) to disprove that command evaluation in IMP^{undo} is deterministic.

[Ex. 2 (1st mid-term)] Let IMP^{seq} the variant of IMP with no conditional statements, no cycles and where general assignments $x := a$ are replaced by updates of the form $x := x + 1$ or $x := x - 1$ (for any location x) whose operational semantics is defined by the rules:

$$\frac{}{\langle x := x+1,\sigma\rangle \to \sigma[^{\sigma(x)+1}/_x]} \qquad \frac{}{\langle x := x-1,\sigma\rangle \to \sigma[^{\sigma(x)-1}/_x]}$$

1. Prove (by rule induction, considering in detail all the rules of the language IMP^{seq}) that command evaluation in IMP^{seq} is *backward deterministic*, i.e., that for any c, σ, σ':

$$P(\langle c,\sigma\rangle \to \sigma') \stackrel{\text{def}}{=} \forall \sigma''.\ \langle c,\sigma''\rangle \to \sigma' \Rightarrow \sigma = \sigma''$$

2. Suppose we add conditional statements to IMP^{seq}. Exhibit a concrete counterexample to backward determinism.

[Ex. 3 (1st mid-term)] Let $\mathcal{D} = (D, \sqsubseteq)$ be a CPO and $\{d_i\}_{i\in\mathbb{N}}$ be a chain in D such that $\exists i, j \in \mathbb{N}.\ d_i \neq d_j$. Moreover, let $\mathcal{N} = (\mathbb{N}, \leq)$ the PO of natural numbers. Is it possible to define a monotone function $f : \mathcal{N} \to \mathcal{N}$ such that

$$\bigsqcup_{i\in\mathbb{N}} d_i \neq \bigsqcup_{i\in\mathbb{N}} d_{f(i)}?$$

[Ex. 4 (2nd mid-term)] Consider the HOFL terms

$$t \stackrel{\text{def}}{=} \lambda x.\ \lambda y.\ (x+1) \qquad t' \stackrel{\text{def}}{=} \lambda x.\ \lambda y.\ \textbf{if } y \textbf{ then } (x+1) \textbf{ else } (x+1).$$

1. Under which hypotheses are t and t' assigned the same type?
2. Are t and t' equivalent according to the (lazy) denotational semantics?
3. Let t_0, t_1 two closed HOFL terms of type int. Under which hypotheses does $((t\ t_0)\ t_1)$ converge operationally? And $((t'\ t_0)\ t_1)$?

[Ex. 5 (2nd mid-term)] Consider the HM-formulas

$$\phi_0 \stackrel{\text{def}}{=} \Box_\alpha((\Diamond_\beta true) \vee (\Box_\gamma false)^c) \qquad \phi_1 \stackrel{\text{def}}{=} \phi_0 \wedge \Diamond_\alpha true$$

1. Define a CCS process p such that $p \not\models \phi_0$.
2. Define a CCS process q such that $q \models \phi_0$ but $q \not\models \phi_1$.
3. Define a CCS process r such that $r \models \phi_0$ and $r \models \phi_1$.

[Ex. 6 (2nd mid-term)] Suppose two different printers Pr_1 and Pr_2 are on sale such that their lifecycles alternate between states s_1 (broken), s_2 (on repair) and s_3 (working), as modeled by the DTMCs in Figures 1 and 2. Which printer would you buy and why?

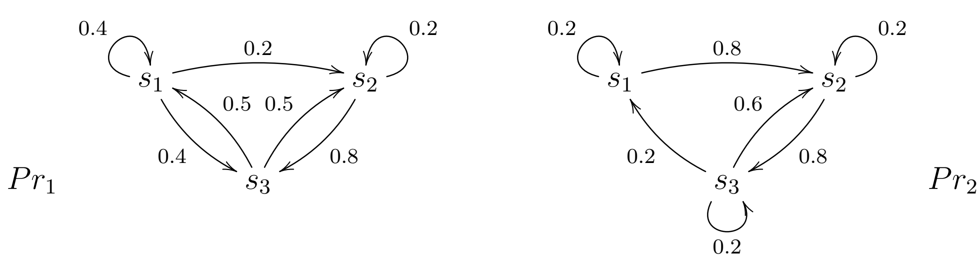

Figure 1: Two DTMCs

Pr_1	s_1	s_2	s_3
s_1	0.4	0.2	0.4
s_2	0	0.2	0.8
s_3	0.5	0.5	0

Pr_2	s_1	s_2	s_3
s_1	0.2	0.8	0
s_2	0	0.2	0.8
s_3	0.2	0.6	0.2

Figure 2: Their transition matrices