Logics of Programs	**Homework 3**
CS686 Spring 2001	**Solutions Posted Thursday, March 15**

1. HKT Exercise 5.3 p. 188.
2. HKT Exercise 5.4 p. 188.
3. The *iterative guarded command* construct **do** $\varphi_1 \to \alpha_1 \mid \cdots \mid \varphi_n \to \alpha_n$ **od** is like the **while** loop, except there are more clauses. Intuitively, a clause $\varphi_i \to \alpha_i$ is *enabled* in a state s if $s \vDash \varphi_i$. When entering the loop, if at least one clause is enabled in the current state, an enabled clause $\varphi_i \to \alpha_i$ is chosen nondeterministically and α_i is executed. After α_i halts, we test again for an enabled clause and repeat the process. This continues for as long as at least one clause is enabled. If ever we return to the beginning of the loop in a state in which no clause is enabled, we exit the loop. The formal semantics is defined by the translation to PDL given on p. 167 of HKT.
   (a) Give a Hoare-style partial correctness rule for this construct that generalizes the Hoare **while** rule (HKT Theorem 5.19(iii), p. 186).
   (b) Using PDL, prove the correctness of your rule of part (a).
4. A *trace* in a Kripke frame $\mathfrak{K} = (K, \mathfrak{m}_{\mathfrak{K}})$ is a sequence $\sigma = s_0 a_1 s_1 a_2 s_2 \cdots s_{n-1} a_n s_n$, where the s_i are states, the a_i are atomic programs, and $(s_{i-1}, s_i) \in \mathfrak{m}_{\mathfrak{K}}(a_i)$, $1 \le i \le n$. We define $\mathbf{first}(\sigma) = s_0$, $\mathbf{last}(\sigma) = s_n$, $\mathbf{length}(\sigma) = n$. Traces of length 0 are just single states. If $\mathbf{last}(\sigma) = \mathbf{first}(\tau)$, then the trace $\sigma\tau$ is the concatenation of the two traces; thus $\mathbf{first}(\sigma\tau) = \mathbf{first}(\sigma)$, $\mathbf{last}(\sigma\tau) = \mathbf{last}(\tau)$, and $\mathbf{length}(\sigma\tau) = \mathbf{length}(\sigma) + \mathbf{length}(\tau)$.

   A *trace model* for PDL is defined on the Kripke frame $\mathfrak{K}$ in a way similar to the definition of the relational model on $\mathfrak{K}$ as described in HKT Sec. 5.2 pp. 167ff., except that programs are interpreted as sets of traces and propositions as sets of states (traces of length 0). The propositional operators have their usual set-theoretic meanings, and

$$\begin{aligned}
\mathrm{tr}_{\mathfrak{K}}(p) &\stackrel{\mathrm{def}}{=} \mathfrak{m}_{\mathfrak{K}}(p),\ p \text{ an atomic proposition}\\
\mathrm{tr}_{\mathfrak{K}}(a) &\stackrel{\mathrm{def}}{=} \{uav \mid (u,v) \in \mathfrak{m}_{\mathfrak{K}}(a)\},\ a \text{ an atomic program}\\
\mathrm{tr}_{\mathfrak{K}}(\alpha\,;\,\beta) &\stackrel{\mathrm{def}}{=} \{\sigma\tau \mid \sigma \in \mathrm{tr}_{\mathfrak{K}}(\alpha),\ \tau \in \mathrm{tr}_{\mathfrak{K}}(\beta),\ \mathbf{last}(\sigma) = \mathbf{first}(\tau)\}\\
\mathrm{tr}_{\mathfrak{K}}(\alpha \cup \beta) &\stackrel{\mathrm{def}}{=} \mathrm{tr}_{\mathfrak{K}}(\alpha) \cup \mathrm{tr}_{\mathfrak{K}}(\beta)\\
\mathrm{tr}_{\mathfrak{K}}(\alpha^*) &\stackrel{\mathrm{def}}{=} \textstyle\bigcup_{n \ge 0} \mathrm{tr}_{\mathfrak{K}}(\alpha)^n\\
\mathrm{tr}_{\mathfrak{K}}([\alpha]\varphi) &\stackrel{\mathrm{def}}{=} \{s \mid \forall\sigma\ \mathbf{first}(\sigma) = s \wedge \sigma \in \mathrm{tr}_{\mathfrak{K}}(\alpha) \to \mathbf{last}(\sigma) \in \mathrm{tr}_{\mathfrak{K}}(\varphi)\}\\
\mathrm{tr}_{\mathfrak{K}}(\varphi?) &\stackrel{\mathrm{def}}{=} \mathrm{tr}_{\mathfrak{K}}(\varphi).
\end{aligned}$$

   There is a strong relationship between trace models and relational models. Formulate and prove a theorem characterizing this relationship.