INFR11248: Modelling Concurrent Systems

University of Edinburgh
2024-2025 SEM1

Homework 13

Gautam Yajaman

1. $P = a.b.c.0$

   $Q = a.b.a.0$

   If a process does not satisfy $\varphi = [a]([b]\langle c\rangle T \wedge \langle b\rangle[a]F)$, then it must satisfy $\neg\varphi$.

$$\begin{aligned}\neg\varphi &= \neg[a]([b]\langle c\rangle T \wedge \langle b\rangle[a]F)\\ &= \langle a\rangle\neg([b]\langle c\rangle T \wedge \langle b\rangle[a]F)\\ &= \langle a\rangle(\neg[b]\langle c\rangle T \vee \neg\langle b\rangle[a]F)\\ &= \langle a\rangle(\langle b\rangle\neg\langle c\rangle T \vee [b]\neg[a]F)\\ &= \langle a\rangle(\langle b\rangle[c]\neg T \vee [b]\langle a\rangle\neg F)\\ &= \langle a\rangle(\langle b\rangle[c]F \vee [b]\langle a\rangle T)\end{aligned}$$

2. To show that completed trace equivalence and simualation equivalence are not comparable, we must show that:

   (a) There exist a pair of processes that are simulation equivalent but are not completed trace equivalent

   (b) There exist a pair of processes that are completed trace equivalent but are not simulation equivalent

   In other words, we must show that the venn diagram of these sets are not concentric circles.

   Consider the processes $P = a.0 + a.b.0$ and $Q = a.b.0$. Process P and Q can simulate each other, i.e., $P \equiv_S Q$ (it is trivial to see that P and Q can always match each other's moves). However, $CT(P) = \{ab, a\}$ and $CT(Q) = \{ab\}$. Therefore $P \neq_{CT} Q$.

   Conversely, consider the processes $X = a.(b.0 + c.0)$ and $Y = a.b.0 + a.c.0$. Both these processes have the completed traces $\{ab, ac\}$, therefore $X =_{CT} Y$. However, consider the process Y attempting to simulate the process X. The first step process X makes will always be an a, and Y can match this with an a of its own. Note that, here, Y *must* commit to a particular branch of its process. As such, based on the branch it commits to, X always has a move it can do which Y cannot. For example, if Y commits to the $a.b.0$ branch, then after its first move, X can do a c which Y cannot match and hence $Y \not\sqsubseteq_S X$. Therefore we see that $X \not\equiv_S Y$.

   Hence we have shown that the two relations are incomparable.

3. (a) Let $[x]_\equiv \leq [y]_\equiv$. Then we must have that $x \sqsubseteq y$ by the definition of $\leq$. To show that $\leq$ is well defined, consider two arbitrary elements $x' \in [x]_\equiv$ and $y' \in [y]_\equiv$. We must show that $x' \sqsubseteq y'$.

   By the definition of equivalence classes, we have that $x' \equiv x$ and $y' \equiv y$. Since $\equiv$ is the kernel of $\sqsubseteq$, we have that $x' \equiv x \iff x \sqsubseteq x' \wedge x' \sqsubseteq x$.

   Hence we have:

$$x' \equiv x \Longrightarrow x' \sqsubseteq x$$
$$y' \equiv y \Longrightarrow y \sqsubseteq y'$$

   Therefore, we have that $x' \sqsubseteq x \sqsubseteq y \sqsubseteq y'$. By transitivity, $x' \sqsubseteq y'$. Hence the choice of representative from each equivalence class does not matter.

   (b) To show that $\leq$ is a partial order:

   i. Reflexivity: $x \sqsubseteq x$, since $\sqsubseteq$ is a preorder. By the definition of $\leq$, we have that $[x]_\equiv \leq [x]_\equiv$. Therefore $\leq$ is reflexive.

   ii. Transitivity: If $[x]_\equiv \leq [y]_\equiv$ and $[y]_\equiv \leq [z]_\equiv$, then we have that $x \sqsubseteq y$ and $y \sqsubseteq z$. By transitivity of $\sqsubseteq$, we have that $x \sqsubseteq z$ and therefore $[x]_\equiv \leq [z]_\equiv$. Therefore $\leq$ is transitive.

   iii. Antisymmetry: If $[x]_\equiv \leq [y]_\equiv$ and $[y]_\equiv \leq [x]_\equiv$, then we have that $x \sqsubseteq y$ and $y \sqsubseteq x$. This implies that $x \equiv y$, by the definition of the kernel. This means that x and y belong to the same equivalence class, which means that $[x]_\equiv = [y]_\equiv$. Therefore $\leq$ is antisymmetric.

   Therefore $\leq$ is a partial order.