A correction to the definition of hiding in tcc, Default tcc and Hybrid cc

Vineet Gupta Radha Jagadeesan Vijay Saraswat

August 7, 2002

Catuscia Palamidessi and Frank Valencia recently pointed out a problem with the definition of hiding in [SJG94]. They showed that the tcc program **new** X **in** [**if** $(X = 1)$ **else** $(Y = 2)$] has the observation $(\texttt{true}, \texttt{true})$, obtained by hiding X in the sequence $(X = 1, \texttt{true})$, which is a valid observation of **if** $(X = 1)$ **else** $(Y = 2)$. However this program is operationally equivalent to **next** $(y = 2)$, for which $(\texttt{true}, \texttt{true})$ is not an observation. Similar problems arise in Default tcc [SJG96] and Hybrid cc [GJS98].

The problems occur due to a missing condition in the definition of hiding. Essentially we need to make a closed world assumption in the program with respect to X-information. Thus we would like to remove from consideration any observations which at any instant contain X-information that was not generated by the program (since now no X-information can come from the environment).

Timed cc: The definition of hiding in [SJG94] was

$$\mathcal{P}[\![\textbf{new } X \textbf{ in } A]\!] \stackrel{d}{=} \{s \in \textbf{Obs} \mid \exists X.s = \exists X.t, \text{ for some } t \in \mathcal{P}[\![A]\!]\}$$

We change this to:

$$\begin{aligned} \mathcal{P}[\![A]\!]_X &\stackrel{d}{=} \{\epsilon\} \cup \{s \cdot a \in \mathcal{P}[\![A]\!] \mid s \in \mathcal{P}[\![A]\!]_X, \forall s \cdot b \in \mathcal{P}[\![A]\!][b \supseteq \exists_X a \rightarrow b \supseteq a]\}. \\ \mathcal{P}[\![\textbf{new } X \textbf{ in } A]\!] &\stackrel{d}{=} \{s \in \textbf{Obs} \mid \exists X.s = \exists X.t, \text{ for some } t \in \mathcal{P}[\![A]\!]_X\} \end{aligned}$$

The process $\mathcal{P}[\![A]\!]_X$ ensures that only those sequences which generate all their X-information are retained — if we have a fixed-point b that contains all the external information contained in a, then b should contain all the X-information conatined in a as well, since this information should have been generated by the program using only the external information. Otherwise a contains some extra information which could not possibly have been generated by the program, so it must be rejected.

For instance in the above example, the sequence $(X = 1, \texttt{true}) \in \mathcal{P}[\![\textbf{if } (X = 1) \textbf{ else } (Y = 2)]\!]$. However $(X = 1) \notin \mathcal{P}[\![\textbf{if } (X = 1) \textbf{ else } (Y = 2)]\!]_X$, because the sequence $(\texttt{true}) \in \mathcal{P}[\![\textbf{if } (X = 1) \textbf{ else } (Y = 2)]\!]$, and $\texttt{true} \supseteq \exists_X X = 1$, but $\texttt{true} \not\supseteq X = 1$. Thus $(X = 1, \texttt{true}) \notin \mathcal{P}[\![\textbf{if } (X = 1) \textbf{ else } (Y = 2)]\!]_X$, and so $\mathcal{P}[\![\textbf{new } X \textbf{ in if } (X = 1) \textbf{ else } (Y = 2)]\!] = \mathcal{P}[\![\textbf{next } Y = 2]\!]$, as desired.

Default tcc. In [SJG96] we had defined hiding for any Default cc process Z in three stages:

1. Define $Z_1 \stackrel{d}{=} \bigcup\{(Z_v, v) \subseteq Z \mid \forall u \in Z_v, u \supseteq \exists_X v \Rightarrow u = v\}$.
2. Define $Z_2 \stackrel{d}{=} \bigcup\{(Z_v, v) \subseteq Z_1 \mid \forall v' \in |D|, (v', v') \in Z_1, \exists_X v = \exists_X v' \Rightarrow \exists_X Z_v = \exists_X Z_{v'}\}$.
3. Now $\textbf{new}_X Z \stackrel{d}{=} \bigcup\{(S, v) \subseteq \textbf{DObs} \mid \exists v'[(v', v') \in Z_2, \exists_X v = \exists_X v', \exists_X S = \exists_X Z_{v'}]\}$.

So, we have: $\mathcal{P}[\![\textbf{new } X \textbf{ in } A]\!] = \textbf{new}_X \mathcal{P}[\![A]\!]$.

Here the first step ensured that all X-information is generated, while the second step ensures that the program remains determinate. We need to impose both of these conditions in the timed extension as well.

The definition of hiding for Default tcc in [SJG96] was:

$$\mathcal{D}[\![\textbf{new } X \textbf{ in } A]\!] \stackrel{d}{=} \left\{ s \in \textbf{TDccObs} \mid \exists s' \in \mathcal{D}[\![A]\!].\exists_X s = \exists_X s', \quad \forall n < |s| \left[s(n) \in \textbf{new}_X(\mathcal{D}[\![A]\!] \textbf{ after } (s')^n)(0) \right] \right\}$$

We need to change the condition, because this does not guarantee that the witnesses of the various elements of s come from the same observation sequence in $\mathcal{D}[\![A]\!]$. For example in the program **new** X **in** [**if** $X = 1$ **else next** $Y = 2$], the observation $((\texttt{true}, \texttt{true}), (\texttt{true}, \texttt{true}))$ is a valid observation according to this definition. The witnessing sequence s' is $((X = 1, X = 1), (\texttt{true}, \texttt{true}))$ because for $n = 0$, $(true, true)$ is in $\textbf{new}_X(\mathcal{D}[\![A]\!] \textbf{ after } (s')^n)(0)$ (where $A =$ **if** $X = 1$ **else next** $Y = 2$) due to the observation $(\texttt{true}, \texttt{true})$ in $\mathcal{D}[\![A]\!]$.

$$\mathcal{D}[\![\textbf{new } X \textbf{ in } A]\!] \stackrel{d}{=} \left\{ s \in \textbf{TDccObs} \mid \exists s' \in \mathcal{D}[\![A]\!].\exists_X s = \exists_X s', \quad \forall n < |s'| \left[s'(n) \in (\mathcal{D}[\![A]\!] \textbf{ after } (s')^n)(0)_2 \right] \right\}$$

This ensures that the same s' is used as a witness. In the above example, $(X = 1, X = 1)$ will not be in $\mathcal{D}[\![(A \textbf{ after } (s')^0)(0)]\!]_2$, eliminating this sequence from consideration.

Hybrid cc. The same fix needs to be added to the definition of hiding in [GJS98]. The previous definition was:

$$\begin{array}{l} \mathcal{H}[\![\textbf{new } X \textbf{ in } A]\!] \stackrel{d}{=} \\ \quad \{ f \in \textbf{HObs} \mid \exists g \in \mathcal{H}[\![A]\!].\exists_X f = \exists_X g, \\ \quad\quad (\forall t \in \texttt{dom}(f))\ [f(t) \in \textbf{new}_X(\mathcal{H}[\![A]\!] \textbf{after } g \restriction [0,t))(0)], \\ \quad\quad (\forall t \in \texttt{Int}(\texttt{dom}(f)) \cup \{0\})\ [f(t^+) \in \textbf{new}_X(\mathcal{H}[\![A]\!] \textbf{after } g \restriction [0,t])(0^+)] \} \end{array}$$

The corrected definition is:

$$\begin{array}{l} \mathcal{H}[\![\textbf{new } X \textbf{ in } A]\!] \stackrel{d}{=} \\ \quad \{ f \in \textbf{HObs} \mid \exists g \in \mathcal{H}[\![A]\!].\exists_X f = \exists_X g, \\ \quad\quad (\forall t \in \texttt{dom}(g))\ [g(t) \in (\mathcal{H}[\![A]\!] \textbf{after } g \restriction [0,t))(0)_2], \\ \quad\quad (\forall t \in \texttt{Int}(\texttt{dom}(g)) \cup \{0\})\ [g(t^+) \in (\mathcal{H}[\![A]\!] \textbf{after } g \restriction [0,t])(0^+)_2] \} \end{array}$$

References

[GJS98] Vineet Gupta, Radha Jagadeesan, and Vijay Saraswat. Computing with continuous change. *Science of Computer Programming*, 30(1-2):3–50, 1998.

[SJG94] V.A. Saraswat, R. Jagadeesan, and V. Gupta. Foundations of Timed Concurrent Constraint Programming. In Samson Abramsky, editor, *Proceedings of the Ninth Annual IEEE Symposium on Logic in Computer Science*, pages 71–80. IEEE Computer Press, July 1994.

[SJG96] V.A. Saraswat, R. Jagadeesan, and V. Gupta. Timed Default Concurrent Constraint Programming. *Journal of Symbolic Computation*, 22(5-6):475–520, November/December 1996. Extended abstract appeared in the *Proceedings of the 22nd ACM Symposium on Principles of Programming Languages*, San Francisco, January 1995.