Prof. Dr. Ir. Dr. h. c. Joost-Pieter Katoen
Christian Hensel, Matthias Volk

Introduction to Model Checking (Summer Term 2018)

— Solution 2 (due 7th May) —

General Remarks

- The exercises are to be solved in groups of *three* students.
- You may hand in your solutions for the exercises just before the exercise class starts at 12:15 or by dropping them into the "Introduction to Model Checking" box at our chair *before 12:00*. Do *not* hand in your solutions via L2P or via e-mail.

Exercise 1 (2 + 5 + 1 Points)

Whenever transition systems are compared via $=$ or $\neq$, this means (in)equality **up to renaming of states** (i.e. isomorphism).

(a) Show that the handshaking $\|_H$ operator **is not** associative, i.e. that in general

$$(\mathrm{TS}_1 \parallel_H TS_2) \parallel_{H'} \mathrm{TS}_3 \neq \mathrm{TS}_1 \parallel_H (\mathrm{TS}_2 \parallel_{H'} \mathrm{TS}_3)$$

(b) The handshaking operator $\|$ that forces transition systems to synchronize over *all* common actions **is** associative. Show that for arbitrary transition systems $\mathrm{TS}_i = (S_i, \mathrm{Act}_i, \to_i, S_0^i, \mathrm{AP}_i, L_i)$ for $i \in \{1, 2, 3\}$, it is

$$\underbrace{(\mathrm{TS}_1 \parallel \mathrm{TS}_2) \parallel \mathrm{TS}_3}_{L} = \underbrace{\mathrm{TS}_1 \parallel (\mathrm{TS}_2 \parallel \mathrm{TS}_3)}_{R}.$$

To this end, show that the bijective function $f_{\approx}\colon ((S_1 \times S_2) \times S_3) \to (S_1 \times (S_2 \times S_3))$ given by $f_{\approx}(\langle\langle s_1, s_2\rangle, s_3\rangle) = \langle s_1, \langle s_2, s_3\rangle\rangle$ preserves the transition relation in the sense that for all $\alpha \in \mathrm{Act}_1 \cup \mathrm{Act}_2 \cup \mathrm{Act}_3$ we have

$$\ell \xrightarrow{\alpha}_L \ell' \iff f_{\approx}(\ell) \xrightarrow{\alpha}_R f_{\approx}(\ell') \tag{2.1}$$

where $\ell, \ell' \in S_L$, S_L is the state space of transition system L and $\longrightarrow_L$, $\longrightarrow_R$ are the transition relations of L and R, respectively.

Hint: When considering an action α, you need only distinguish the cases

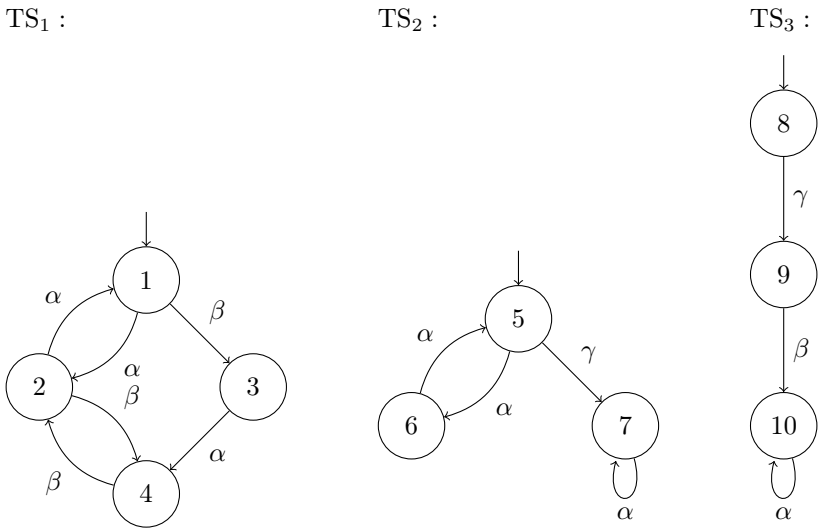
(i) $\alpha \in \mathrm{Act}_1 \backslash (Act_2 \cup Act_3)$

(ii) $\alpha \in (\mathrm{Act}_1 \cap \mathrm{Act}_2) \backslash Act_3$

(iii) $\alpha \in \mathrm{Act}_1 \cap Act_2 \cap Act_3$

as all other cases are symmetric. Also, for simplicity, it suffices to show the direction "$\Longrightarrow$" of condition (2.1). However, keep in mind that L and R are not necessarily action-deterministic (see exercise sheet 1).

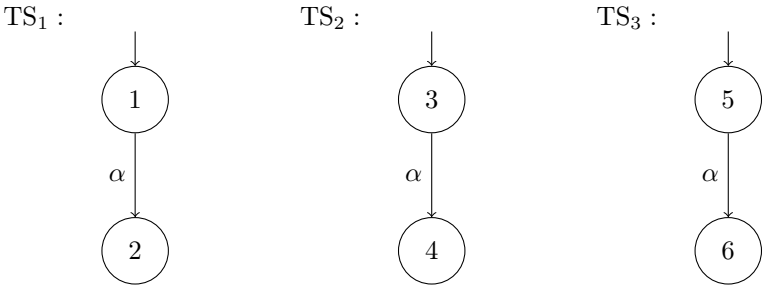
(c) Consider the following three transition systems:



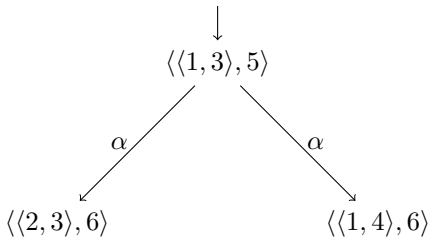
Build the composition $(\mathrm{TS}_1 \parallel \mathrm{TS}_2) \parallel \mathrm{TS}_3$.

Solution:

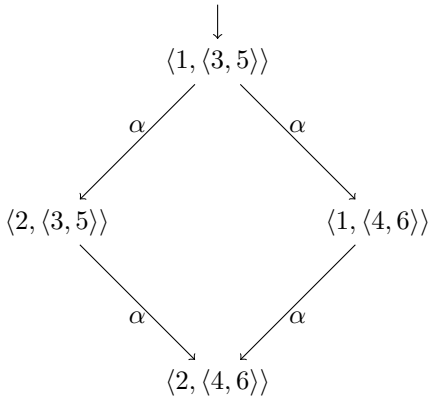
(a) Consider the following three transition systems:



The result of the composition $(\mathrm{TS}_1 \parallel_\emptyset \mathrm{TS}_2) \parallel_{\{\alpha\}} \mathrm{TS}_3$ is:



whereas the result of the composition $\mathrm{TS}_1 \parallel_\emptyset (\mathrm{TS}_2 \parallel_{\{\alpha\}} \mathrm{TS}_3)$ is:



(b) It remains to show the $\Longrightarrow$ of condition (2.1), i.e.

$$l \xrightarrow{\alpha}_L l' \implies f_{\approx}(l) \xrightarrow{\alpha}_R f_{\approx}(l')$$

Fix any action $\alpha \in Act_L$. We distinguish the following cases:

(i) $\alpha \in \text{Act}_1 \setminus (Act_2 \cup Act_3)$

(ii) $\alpha \in (\text{Act}_1 \cap \text{Act}_2) \setminus Act_3$

(iii) $\alpha \in \text{Act}_1 \cap Act_2 \cap Act_3$

The case where α does not belong to any action set is irrelevant and all other cases are symmetric. Fix any triple of states from TS_1, TS_2 and TS_3, say s_1, s_2, s_3. Let $l = \langle\langle s_1, s_2\rangle, s_3\rangle$ and $r = f_{\approx}(l) = \langle s_1, \langle s_2, s_3\rangle\rangle$

(i) There is no α-transition from s_2 and s_3. If there is no α-transition from s_1 then there is also no α-transition in the composed systems neither from $\langle\langle s_1, s_2\rangle, s_3\rangle$ nor $\langle s_1, \langle s_2, s_3\rangle\rangle$ and there is nothing to show.

Now assume there is (at least) one α-transition from s_1. Pick one α-successor, say s_1'. TS_1 and TS_2 interleave on α: $\langle s_1, s_2\rangle \xrightarrow{\alpha} \langle s_1', s_2\rangle$. This subsystem again interleaves on α with TS_3 so in L we get a transition $\langle\langle s_1, s_2\rangle, s_3\rangle \xrightarrow{\alpha} \langle\langle s_1', s_2\rangle, s_3\rangle$. On the other hand in $\text{TS}_2 \parallel \text{TS}_3$ there is no α-transition from $\langle s_2, s_3\rangle$. In R we get the transition $\langle s_1, \langle s_2, s_3\rangle\rangle \xrightarrow{\alpha} \langle s_1', \langle s_2, s_3\rangle\rangle$. We have shown that for the transition $l = \langle\langle s_1, s_2\rangle, s_3\rangle \xrightarrow{\alpha}_L \langle\langle s_1', s_2\rangle, s_3\rangle = l'$ there exists a transition $f_{\approx}(l) = \langle s_1, \langle s_2, s_3\rangle\rangle \xrightarrow{\alpha}_R \langle s_1', \langle s_2, s_3\rangle\rangle = f_{\approx}(l')$.

(ii) There is no α-transition from s_3. If there is no α-transition from s_1 or no α transition in s_2, again there is no α-transition in l and there is nothing to show.

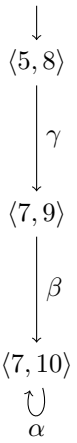
Now assume there is (at least) one α-transition from s_1 and one α transition in s_2. Pick α-successors, say s_1' and s_2'. In $\text{TS}_1 \parallel \text{TS}_2$ the systems synchronize on α and we have the transition $\langle s_1, s_2\rangle \xrightarrow{\alpha} \langle s_1', s_2'\rangle$. This subsystem interleaves with TS_3 on α and so in L we get a transition $\langle\langle s_1, s_2\rangle, s_3\rangle \xrightarrow{\alpha} \langle\langle s_1', s_2'\rangle, s_3\rangle$. On the other hand TS_2 and TS_3 interleave on α: $\langle s_2, s_3\rangle \xrightarrow{\alpha} \langle s_2', s_3\rangle$. This subsystem synchronizes with TS_1 on α and so in R we get the transition $\langle s_1, \langle s_2, s_3\rangle\rangle \xrightarrow{\alpha} \langle s_1', \langle s_2', s_3\rangle\rangle$. We have shown that for the transition $l = \langle\langle s_1, s_2\rangle, s_3\rangle \xrightarrow{\alpha}_L \langle\langle s_1', s_2'\rangle, s_3\rangle = l'$ there exists a transition $f_{\approx}(l) = \langle s_1, \langle s_2, s_3\rangle\rangle \xrightarrow{\alpha}_R \langle s_1', \langle s_2', s_3\rangle\rangle = f_{\approx}(l')$.

(iii) If there is no α transition in any one of the states s_1, s_2, s_3, then in both the composed systems L and R there is no α-transition from both $\langle\langle s_1, s_2\rangle, s_3\rangle$ and $\langle s_1, \langle s_2, s_3\rangle\rangle$ and there is nothing to show.

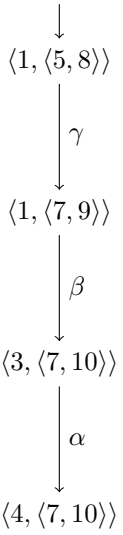
Assume there is (at least) one α-transition from s_1, one from s_2 and one from s_3. Pick α-successors, say s_1', s_2' and s_3'. In $\text{TS}_1 \parallel \text{TS}_2$ the systems synchronize on α and we have the transition $\langle s_1, s_2\rangle \xrightarrow{\alpha} \langle s_1', s_2'\rangle$. This subsystem synchronizes with TS_3 on α and so in L we get a transition $\langle\langle s_1, s_2\rangle, s_3\rangle \xrightarrow{\alpha}_L \langle\langle s_1', s_2'\rangle, s_3'\rangle$. On the other hand in TS_2 and TS_3 also synchronize on α: $\langle s_2, s_3\rangle \xrightarrow{\alpha} \langle s_2', s_3'\rangle$. This subsystem synchronizes with TS_1 on α and so in R we get the transition $\langle s_1, \langle s_2, s_3\rangle\rangle \xrightarrow{\alpha}_R \langle s_1', \langle s_2', s_3'\rangle\rangle$. We have shown that for the transition $l = \langle\langle s_1, s_2\rangle, s_3\rangle \xrightarrow{\alpha}_L \langle\langle s_1', s_2'\rangle, s_3'\rangle = l'$ there exists a transition $f_{\approx}(l) = \langle s_1, \langle s_2, s_3\rangle\rangle \xrightarrow{\alpha}_R \langle s_1', \langle s_2', s_3'\rangle\rangle = f_{\approx}(l')$.

(c) We use the associativity of the handshaking operator $\parallel$ and compute $\text{TS}_1 \parallel (\text{TS}_2 \parallel \text{TS}_3)$ instead of $(\text{TS}_1 \parallel \text{TS}_2) \parallel \text{TS}_3$.

$\mathrm{TS}_2 \parallel \mathrm{TS}_3$:



$\mathrm{TS}_1 \parallel (\mathrm{TS}_2 \parallel \mathrm{TS}_3)$:



Note, that using the original order of composition we would get an intermediate state space of 10 states.

Exercise 2 (2 + 5 Points)

(a) In the lecture, channel systems using FIFO (or queue) channels were introduced. We now consider LIFO (or stack) channels. Formally define the SOS rules for the communication such that in the induced transition system the channels have LIFO semantics.

(b) Consider the following decision problem.

Input: A LIFO channel system $[\mathcal{P}_1 \mid \ldots \mid \mathcal{P}_n]$ with program graphs

$$\mathcal{P}_i = (\mathrm{Loc}_i, \mathrm{Act}_i, \mathrm{Effect}_i, \hookrightarrow_i, \mathrm{Loc}_0^i, g_0^i)$$

over $(\mathrm{Var}, \mathrm{Chan})$ and a set $F \subseteq \mathrm{Loc}_1 \times \ldots \mathrm{Loc}_n \times \mathrm{Eval}(\mathrm{Var}) \times \mathrm{Eval}(\mathrm{Chan})$.
Question: Is some state of F reachable in $\mathrm{TS}([\mathcal{P}_1 \mid \ldots \mid \mathcal{P}_n])$?

Prove that this problem is undecidable. For this, reduce the halting problem for (nondeterministic) Turing machines started on an empty tape to the above problem.

Solution:

(a) As there is no difference between FIFO and LIFO semantics for channels c with $cap(c) = 0$ (handshaking), we formulate the SOS rules for asynchronous message passing as follows.

$$\frac{\ell_i \xhookrightarrow{g:c?x}_i \ell_i' \wedge \eta \models g \wedge \xi(c) = v_1 \ldots v_k \wedge k \geq 1}{\langle \ell_1, \ldots, \ell_i, \ldots, \ell_n, \eta, \xi \rangle \xrightarrow{\tau} \langle \ell_1, \ldots, \ell_i', \ldots, \ell_n, \eta', \xi' \rangle}$$

where $\eta' = \eta[x := v_k]$ and $\xi' = \xi[c := v_1 \ldots v_{k-1}]$.

$$\frac{\ell_i \xhookrightarrow{g:c!v'}_i \ell_i' \wedge \eta \models g \wedge \xi(c) = v_1 \ldots v_k \wedge k < cap(c) \geq 1}{\langle \ell_1, \ldots, \ell_i, \ldots, \ell_n, \eta, \xi \rangle \xrightarrow{\tau} \langle \ell_1, \ldots, \ell_i', \ldots, \ell_n, \eta, \xi' \rangle}$$

where $\xi' = \xi[c := v_1 \ldots v_k v']$. Note that the channel interactions may be guarded here, which is done exclusively for readability and is not needed for the result itself.

(b) Let $M = (Q, \Sigma, \Gamma, \delta, q_0, \square, F)$ be a nondeterministic Turing machine where

- $Q \neq \emptyset$ is a (finite) non-empty set of states,
- $\Sigma \neq \emptyset$ is the (finite) non-empty input alphabet,
- $\Gamma \neq \emptyset, \Sigma \subset \Gamma$ is the (finite) non-empty tape alphabet,
- $\delta \subseteq (Q \setminus F) \times \Gamma \times Q \times \Gamma \times \{L, N, R\}$ is the (finite) transition relation,
- q_0 is the starting state,
- $\square \in \Gamma \setminus \Sigma$ is the blank symbol, and
- $F \subseteq Q$ is a set of halting states.

From M, we construct a LIFO channel system $C = [\mathcal{P}]$ that consists of

- one program graph $\mathcal{P}$,
- two LIFO channels $\text{Chan}' = \{c_l, c_r\}$ with $cap(c_l) = cap(c_r) = \infty$, and
- a single variable $\text{Var} = \{x\}$ with $\text{Dom}(x) = \Gamma \cup \{\$_l, \$_r\}$.

The program graph $\mathcal{P} = (\text{Loc}, \text{Act}, \text{Effect}, \hookrightarrow, \text{Loc}_0, g_0)$ is given by

- $\text{Loc} = Q \cup \times\{L, N, R, B\} \cup \{init_0, init_1\}$,
- $\text{Act} = \{c_l!v, c_r!v \mid v \in \Gamma\} \cup \{c_l?x, c_r?x\} \cup \{x := \gamma \mid \gamma \in \Gamma\}$,
- $\text{Effect}(x := \gamma, \eta) = \eta[x := \gamma]$
- $\text{Loc}_0 = \langle start, N\rangle$,
- $g_0 = (x = \square)$

and

$$\langle init_0\rangle \overset{c_l!\$_l}{\hookrightarrow} \langle init_1\rangle \quad \langle init_1\rangle \overset{c_r!\$_r}{\hookrightarrow} \langle q_0, N\rangle$$

$$\frac{(q, \gamma, q', \gamma', N) \in \delta}{\langle q, N\rangle \overset{x=\gamma:x:=\gamma'}{\hookrightarrow} \langle q', N\rangle}$$

$$\frac{(q, \gamma, q', \gamma', L) \in \delta}{\langle q, N\rangle \overset{x=\gamma:c_r!\gamma'}{\hookrightarrow} \langle q', L\rangle \quad \wedge \quad \langle q', L\rangle \overset{c_l?x}{\hookrightarrow} \langle q', N\rangle}$$

$$\frac{(q, \gamma, q', \gamma', R) \in \delta}{\langle q, N\rangle \overset{x=\gamma:c_l!\gamma'}{\hookrightarrow} \langle q', R\rangle \quad \wedge \quad \langle q', R\rangle \overset{c_r?x}{\hookrightarrow} \langle q', N\rangle.}$$

$$\frac{}{\langle q, N\rangle \overset{x=\$_l:c_l!\$_l}{\hookrightarrow} \langle q, B\rangle} \quad \frac{}{\langle q, B\rangle \overset{x=\$_l:x:=\square}{\hookrightarrow} \langle q, N\rangle}$$

$$\frac{}{\langle q, N\rangle \overset{x=\$_r:c_r!\$_r}{\hookrightarrow} \langle q, B\rangle} \quad \frac{}{\langle q, B\rangle \overset{x=\$_r:x:=\square}{\hookrightarrow} \langle q, N\rangle}$$

The intuition is as follows. The content of the tape is stored in the two channels c_l and c_r. Initially, they are filled with special symbols $\$_l$ and $\$_r$ that do not appear in the transitions of the TM. c_l contains the content of the tape *left* of the current position and the topmost element of the (LIFO) channel is the one that is *one* to the left of the current position. Similarly, c_r stores the content of the tape *right* of the current position. The symbol that is stored on the tape at the current position is stored within the variable x. The symbols $\$_l$ and $\$_r$ mark the left and right "end" of the tape, respectively, where "end" means that everything left of $\$_l$ and everything right of $\$_r$ is the empty tape.

The locations of $\mathcal{P}$ essentially store the state of the TM M and whether or not an element needs to be read from the left or right (stored within the second component of the elements in Loc where N = read no element, L = read an element from the left, R = read an element from the right and B is used as a special flag that the content of the current tape cell is empty despite any value that x might hold). Initially, the left and right channel contents are initialized with $\$_l$ and $\$_r$, respectively. The transitions of $\mathcal{P}$ then make sure that the behavior of a TM transition is simulated. If M remains at the current position, all we need to do is update the content of the current tape cell, i.e. the variable x, and the state within M. If M moves to either the left or the right, we break the step into two parts. First, we write the updated (according to the TM transition) symbol at the current tape cell to the appropriate channel (c_l if the TM moves to the right and c_r if it moves to the left) and remember in the state not only the state transition of the TM but also that we need to read one more element from the *dual* channel to again have the current tape cell stored within the variable x. If the head is moved to any of the current "end of tape" markers, the marker is pushed one step further and the content of the variable x that stores the current tape cell content is set to $\square$.

Since the channel system C simulates the TM M "step by step", we have that M halts on the empty tape if and only if some state of $F' = \{\langle q, N\rangle \mid q \in F\}$ is reachable in $\mathrm{TS}(C)$. If the latter problem was decidable, the halting problem would also be, contradiction.

We remark that the result still holds if the channel interactions may not be guarded, if the channels are FIFO channels and even for program graphs with (finitely many) unbounded integer variables (i.e. without any channels).

Exercise 3* (1 + 3 + 1 Points)

We consider the problem of the *dining philosophers*. There are n philosophers sitting around a table and one fork is placed in-between any two philosophers sitting next to each other. The philosophers alternate between thinking and eating. Whenever a philosopher wants to eat, she first has to pick up her left and her right fork (in arbitrary order). After having finished eating, she puts both forks back on the table.

(a) Model the dining philosophers as a channel system $\mathcal{C}$.

Hint: Model each fork as a separate channel.

(b) Construct the transition system $\mathrm{TS}(\mathcal{C})$ for the channel system $\mathcal{C}$ defined in (a) for $n = 3$ philosophers. For simplification, you may assume that each philosopher first picks up her left fork and then her right fork. This order is also preserved when placing the forks on the table again. Furthermore, you may assume that each channel initially contains a fixed number of messages. Finally, you can exploit symmetries when constructing the transition system and merge symmetric states. If you do, briefly justify your approach.

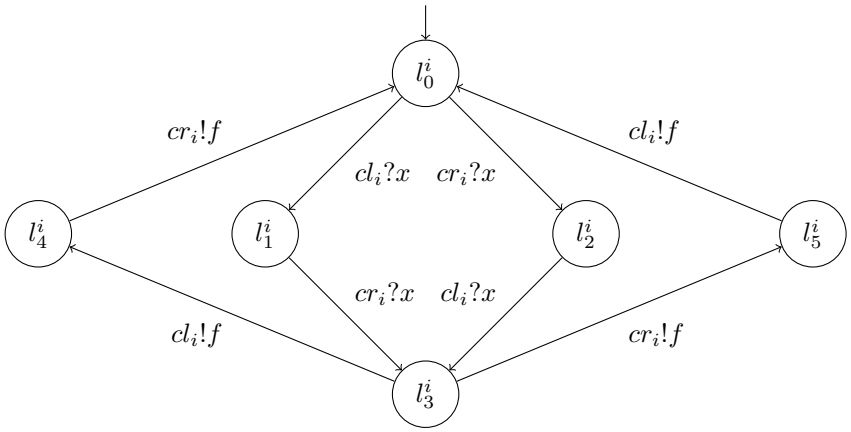
(c) Does the transition system of (b) contain a deadlock?

Solution:

(a) The dining philosophers are modeled as a channel system $[\mathcal{P}_1 \mid \cdots \mid \mathcal{P}_n \mid \mathcal{F}]$ over $(\mathrm{Var}, \mathrm{Chan})$ where $\mathrm{Var} = \{x\}$ and $\mathrm{Chan} = \{c_1, \ldots, c_n\}$ with capacity $\mathrm{cap}(c_i) = 1$. The program graphs for the philosophers $\mathcal{P}_i = (\mathrm{Loc}_i, \mathrm{Act}_i, \mathrm{Effect}_i, \hookrightarrow_i, \mathrm{Loc}_0^i, g_0^i)$ are given as:

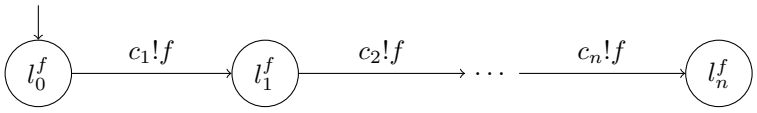
- $\mathrm{Loc}_i = \{l_0^i, \ldots, l_5^i\}$
- $\mathrm{Loc}_0^i = \{l_0^i\}$
- $g_0^i = \{true\}$
- Act_i, Effect_i and $\hookrightarrow_i$ as depicted below

In the following we use $cr_i = c_i$ to indicate the channel for the right fork of the i-th philosopher and $cl_i = \begin{cases} c_{i-1} & \text{if i>1} \\ c_n & \text{otherwise} \end{cases}$ to indicate the channel for the left fork.

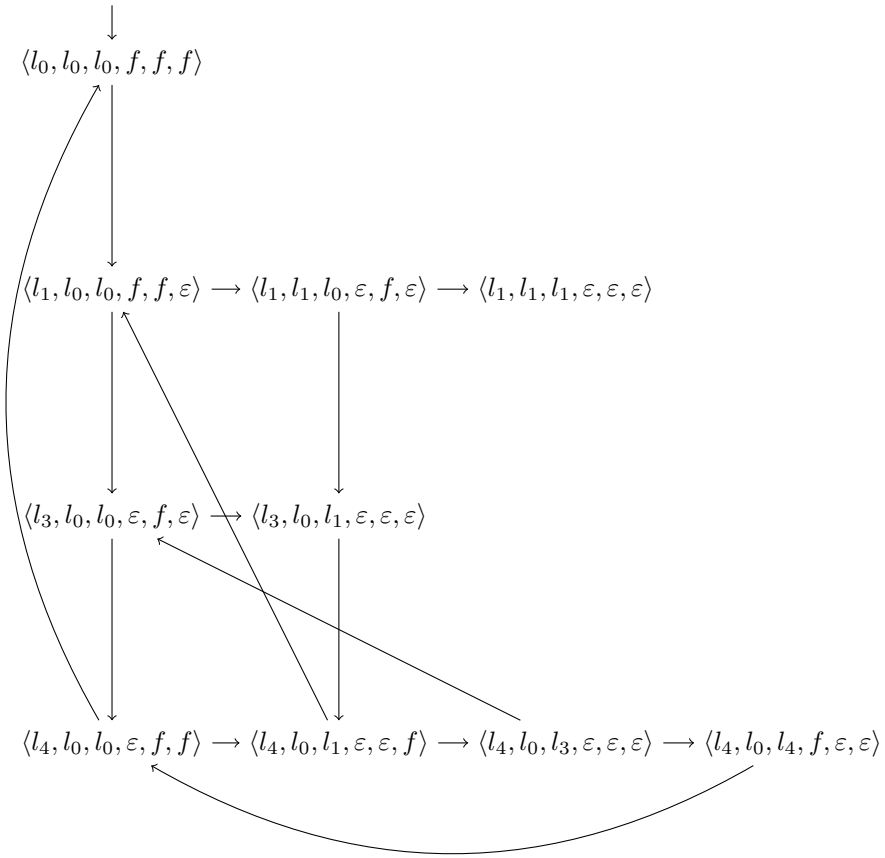


The program graph $\mathcal{F} = (\text{Loc}_f, \text{Act}_f, \text{Effect}_f, \hookrightarrow_f, \text{Loc}_0^f, g_0^f)$ initializing the channels of the forks is given as:

- $\text{Loc}_f = \left\{l_0^f, \ldots, l_n^f\right\}$
- $\text{Loc}_0^f = \left\{l_0^f\right\}$
- $g_0^f = \{true\}$
- Act_f, Effect_f and $\hookrightarrow_f$ as depicted below



(b) Using the simplification we only have to consider the left part of each program graph, i.e., the locations l_0, l_1, l_3, l_4. We also let each channel initially contain the message f. The resulting transition system $\text{TS}_\mathcal{C}$ is depicted below. Here $\langle l_0, l_0, l_0, f, f, f\rangle$ is the short form of $\langle l_0^1, l_0^2, l_0^3, c_1 = f, c_2 = f, c_3 = f\rangle$.



(c) The transition system contains a deadlock in $\langle l_1, l_1, l_1, \varepsilon, \varepsilon, \varepsilon \rangle$ which is reachable with the following path

$$\langle l_0, l_0, l_0, f, f, f \rangle \to \langle l_1, l_0, l_0, f, f, \varepsilon \rangle \to \langle l_1, l_1, l_0, \varepsilon, f, \varepsilon \rangle \to \langle l_1, l_1, l_1, \varepsilon, \varepsilon, \varepsilon \rangle$$