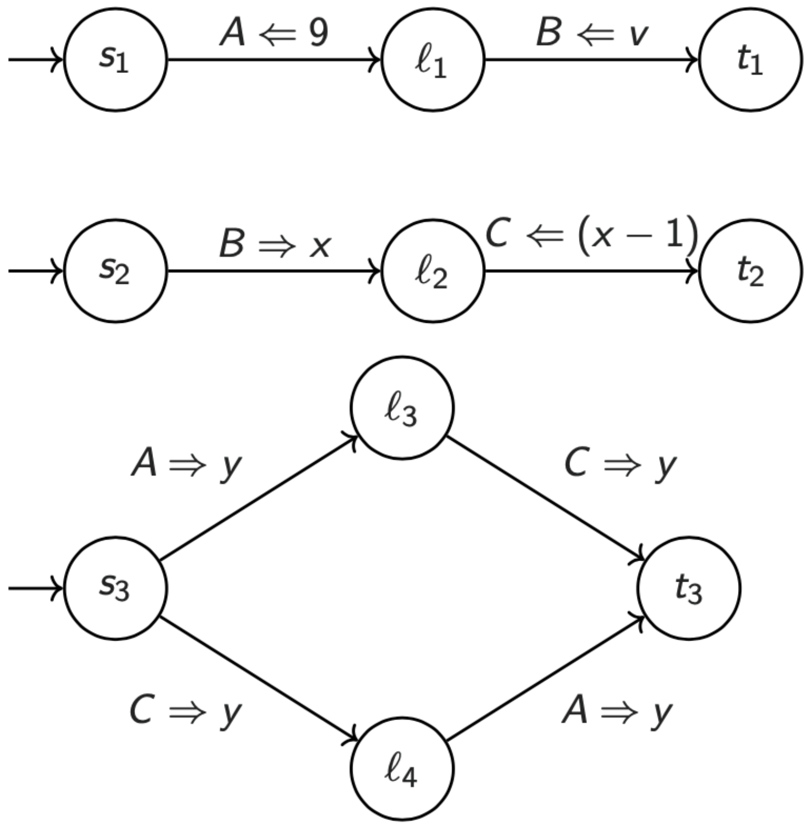
## Question 1 *Non-compositional Verification[6 marks]*

Here is a three process message passing system presented as a transition diagram of three processes $P1$, $P2$, and $P3$.

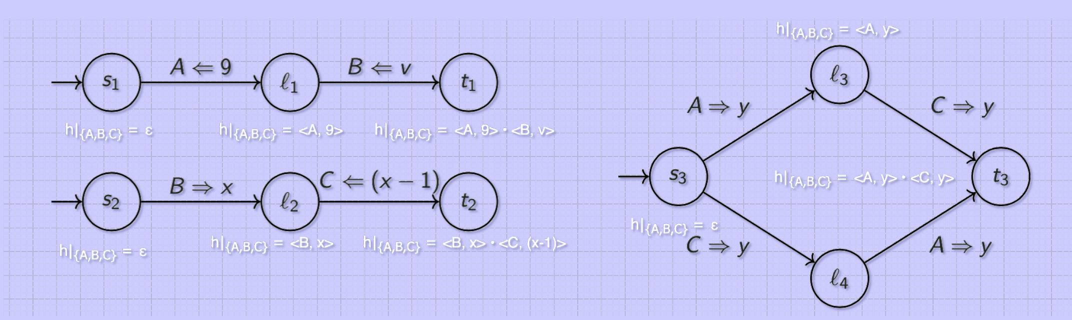



Prove using the Levin and Gries or AFR methods that the following Hoare triple holds:

$$\{True\}\ P1 \parallel P2 \parallel P3\ \{y = v - 1\}$$

You don't need to explicitly discharge your proof obligations; instead, it suffices to give your assertion networks, your extra auxiliary variable wrangling, and (if using AFR) your communication invariant.

Answer:



Answer:

For the three process passing system, there are three channel $(A, B, C)$ to pass the message.

- For channel $A$, $A \Leftarrow 9$ $(s_1 \to l_1)$, $A \Rightarrow y$ $(s_3 \to l_3)$, $A \Rightarrow y$ $(l_4 \to t_3)$
- For channel $B$, $B \Leftarrow v$ $(l_1 \to t_1)$, $B \Rightarrow x$ $(s_2 \to l_2)$
- For channel $C$, $C \Leftarrow (x-1)$ $(l_2 \to t_2)$, $C \Rightarrow y$ $(l_3 \to t_3)$, $C \Rightarrow y$ $(s_3 \to l_4)$

As the channel is not Asynchronous channel, a sender input must match a receiver output.
For channel $A$ and $C$, there is a input and two output. for channel; For channel $B$, there is one input and one output. The premise of finishing the message passing in channel $B$ is that complete the $A \Leftarrow 9$ in $P_1$. In $P_3$, it exists two output, however, $A \Rightarrow y$ $(l_4 \to t_3)$ has a premise of $C \Rightarrow y$, the input of channel $C$ is after the $B \Rightarrow x$ in $P_2$, therefore, this way will blocked. The other passing in C $(s_{333})$ is suitable of the topic.

Proof:

**Basic diagram rule gives us:**

$$\{h|_{\{A,B,C\}} = \varepsilon\}\ P_1\ \{h|_{\{A,B,C\}} = < A, 9 > \cdot < B, v >\} \quad (1)$$
$$\{h|_{\{A,B,C\}} = \varepsilon\}\ P_2\ \{h|_{\{A,B,C\}} = < B, x > \cdot < C, (x-1) >\} \quad (2)$$
$$\{h|_{\{A,B,C\}} = \varepsilon\}\ P_3\ \{h|_{\{A,B,C\}} = < A, y > \cdot < C, y >\} \quad (3)$$

**Apply the parallel composition rule.**

$\{h|_{\{A,B,C\}} = \varepsilon\ \wedge\ h|_{\{A,B,C\}} = \varepsilon\ \wedge\ h|_{\{A,B,C\}} = \varepsilon\}\ P_1 \parallel P_2 \parallel P_3\ \{h|_{\{A,B,C\}} =$
$< A, 9 > \cdot < B, v >\ \wedge h|_{\{A,B,C\}} = < B, x > \cdot < C, (x-1) >\ \wedge h|_{\{A,B,C\}}$
$= < A, y > \cdot < C, y >\}$ (4)

**According to the topic, $y$ is assigned to channel $A$'s output as 9, and then to channel $C$'s output as $(x-1)$.**
**Using the rule of consequence with (4) we get:**

$$\{h|_{\{A,B,C\}} = \varepsilon\}\ P_1 \parallel P_2 \parallel P_3\ \{x = v \wedge y = (x-1)\}\ (5)$$

**As $x = v, y = (x-1) \to y = (v-1)$:**

$$\{h|_{\{A,B,C\}} = \varepsilon\}\ P_1 \parallel P_2 \parallel P_3\ \{y = (v-1)\}\ (6)$$

**Using the rule of consequence:**

$$\{True \wedge h|_{\{A,B,C\}} = \varepsilon\}\ P_1 \parallel P_2 \parallel P_3\ \{y = (v-1)\}\ (7)$$

**Using the initialization rule:**

$$\{True\}\ P_1 \parallel P_2 \parallel P_3\ \{y = (v-1)\}\ (8)$$

Therefore, the result is $\{True\}\ P_1 \parallel P_2 \parallel P_3\ \{y = (v-1)\}$

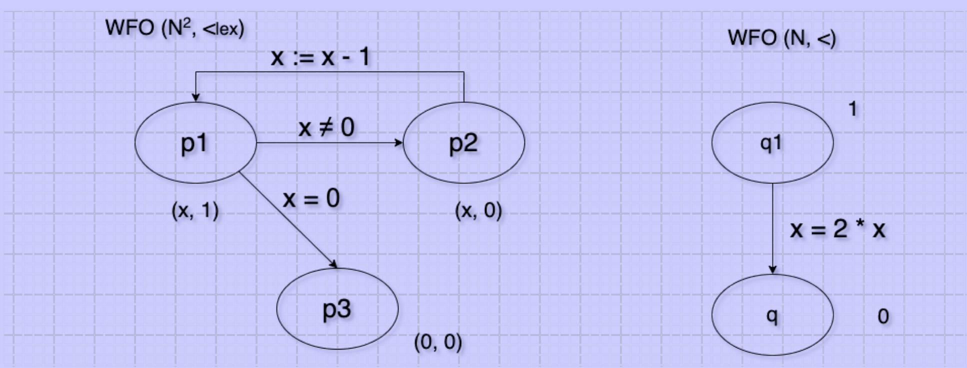
## Question 2 *Termination[6 marks]*

Consider the following program:

| **int** $x$ | |
|---|---|
| $\mathsf{p}_1$ **while** $x \neq 0$<br>$\mathsf{p}_2$ $x := x - 1$ | $\mathsf{q}_1$: $x := 2 * x$ |

**2.1** Use the local method to prove $x \geq 0$ -convergence for this program. You'll need exit locations for $p$ and $q$ (not shown in the above pseudocode). You don't need to explicitly discharge your proof obligations; specifying your assertion networks, your wellfounded set, and your ranking functions is sufficient.

Answer:




Assertion networks: $\{x \geq 0\}\ p \parallel q\ \{x = 0\}$
ranking:
p:
transition $p1 \xrightarrow{x \neq 0} p2$: $\models (x, 1) >_{lex} (x, 0) \wedge (0, 1) >_{lex} (0, 0)$

transition $p2 \xrightarrow{x := x-1} p1$: $\models x \neq 0 \Rightarrow x > x - 1 \geq 0$

transition $p1 \xrightarrow{x=0} p3$: $\models (0, 1) >_{lex} (0, 0)$

q:
transition $q1 \xrightarrow{x := x*2} q2$: $\models 1 > 0$

**2.2** Is this program $\top$ -convergent? Briefly motivate your answer.

Answer:

No, for this program, when $x < 0$, the $x$ will decrease forever and never reach the $x = 0$ to stop. When $x > 0$, the program, $p$ only minus 1, $q$ will double the value of $x$, it is difficulty to reach the purpose. Therefore, it is not $\top$ -convergent.

**2.3** Is this program $\perp$ -convergent? Briefly motivate your answer.

Answer:

Yes, although it is difficulty to reach the purpose, when the initial state is satisfied, the purpose can be achieved. Therefore, it iss $\perp$ -convergent.