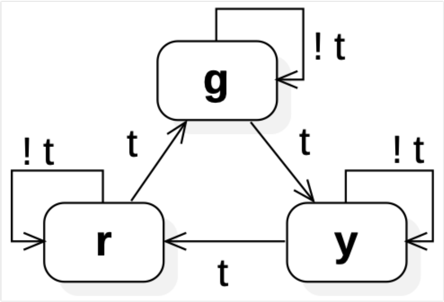
CCSL Specification Synthesis for A Traffic Light Control System

As a specific real-time embedded system, traffic light control systems are widely used in worldwide at road intersections to manage the traffic network flows. This example is about a traffic light system to regulate traffic flow at the cross intersection, which involves two traffic lights controlling the North-South lanes and East-West lanes, respectively. Here, each traffic light has three LED signals with green, yellow, and red colors, respectively. Note that at a time only one color is allowed to be shown by a traffic light, and the colors are shown following an order of green, yellow, and red in each iteration.



- $t\_g2y \triangleq t * g$
- $t\_y2r \triangleq t * y$
- $t\_r2g \triangleq t\_ew * r$
- $t\_y2y \triangleq n\_t * y$
- $t\_r2r \triangleq n\_t * r$
- $t\_g2g \triangleq n\_t * g$

- $glb\_g\_y \triangleq g + y$
- $glb\_tl \triangleq g + glb\_g\_y$
- $glb = glb\_tl$
- $glb\_t \triangleq t + n\_t$
- $glb\_t = glb$
- $t \# n\_t$

- $g \# r$
- $y \# r$
- $g \# y$

- $next\_t\_r \triangleq t\_y2r\$1\ on\ glb$
- $next\_t\_r \subseteq r$
- $next\_t\_g \triangleq t\_r2g\$1\ on\ glb$
- $next\_t\_g \subseteq g$
- $next\_t\_y \triangleq t\_g2y\$1\ on\ glb$
- $next\_t\_y \subseteq y$

- $next\_t\_r \triangleq t\_y2r\$1\ on\ glb$
- $next\_t\_r \subseteq r$
- $next\_t\_g \triangleq t\_r2g\$1\ on\ glb$
- $next\_t\_g \subseteq g$
- $next\_t\_y \triangleq t\_g2y\$1\ on\ glb$
- $next\_t\_y \subseteq y$

Figure 2: Traffic light statemachine and its CCSL template

At the early design stage, it is difficult for requirement engineers to directly figure out the CCSL specification of the whole traffic light control system for the cross intersection. Instead, they would like to construct the CCSL specification in a bottom-up manner. Firstly, they would like to resort to the CCSL template of a single traffic light. The left part of Figure 2 presents the statemachine for the traffic light and its corresponding CCSL specification. The statemachine consists of three states, i.e., “g”, “y”, and “r”, indicating the states for green, yellow, and red colors, respectively, and “t” is a transfer signal to control transitions between states. Once the traffic light receives a transfer signal , it will change its current state immediately.

Based on the common sense, the timing behaviors of traffic lights should obey the following properties:
Property 1: A traffic light shall never turn on two or more of its LEDs at the same time.
Property 2: A traffic light must change its current state immediately after receiving a transfer signal .
Property 3: Without receiving any transfer signal , a traffic light should not change its current state.
Property 4: The N-S traffic light and the E-W traffic light cannot be in the state of green light or yellow light simultaneously.
Property 5: If the N-S traffic light is red, the E-W traffic light cannot turn red simultaneously, and vice versa.

The right part of Figure 2 presents the CCSL constraints for a traffic light, which are generated from both the statemachine diagram shown in Figure 2(a) and the specified properties 1-3. These constraints can be used as the CCSL template of a traffic light to model its timing behaviors.

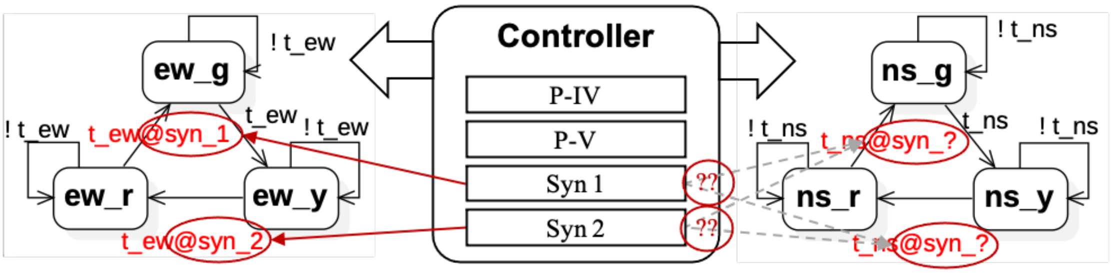


Figure 3: Design of the traffic light control system

Figure 3 shows the design of the traffic light system, which includes the E-W (East-West) traffic light, the N-S (North-South) traffic light, and a controller. Here, we use the controller to conduct the synchronization between the two traffic lights, where we use the clocks t_ew and t_ns to denote the transfer signals for the E-W traffic light and N-S traffic light, respectively. Based on the above safety properties 4-5, requirement engineers need to design a controller to regulate the intersection traffic via transfer signals. Since properties 4-5 can be directly encoded into CCSL constraints, requirement engineers only need to figure out the synchronization implementation for t_ew and t_ns defined in statemachines. The annotation @syn_i in Figure 3 indicates that the associated transition in the current statemachine needs to be synchronized with a transition in the other state machine, where is the synchronization index. The annotation @syn_? indicates that the counterpart transition for synchronization is unknown, where the synchronization index "?" needs to be figured out by the synthesizer. As shown in Figure 3, when a traffic light turns from red to green or from yellow to red, the traffic light of the other direction also needs to change accordingly at the same time. In Figure 3, the notations @syn_1 and @syn_2 in the left statemachine are used to enable the synchronization between two traffic lights. However, in the right statemachine such synchronization information is unknown. As shown in Figure 3, the CCSL constraints of the controller are generated from two properties (i.e., property 4 and property 5) and two incomplete synchronizations (i.e., Syn1 and Syn2), where the incomplete synchronizations are encoded into CCSL constraints as follows:

$$ew\_r * t\_ew = ??$$
$$ew\_y * t\_ew = ??$$

For both the incomplete CCSL model shown in Figure 3 and specified properties, we generate an incomplete CCSL specification with 11 atomic clocks and 61 CCSL constraints (31 expression constraints and 30 relation constraints), leaving the timing behaviors of interactions between traffic lights as unknown holes. Figure 4 shows the incomplete CCSL specification generated from the templates of traffic lights and the properties 1-5.

- t_ew_g2y ≜ t_ew ∗ ew_g
- t_ew_y2r ≜ t_ew ∗ ew_y
- t_ew_r2g ≜ t_ew ∗ ew_r
- t_ew_y2y ≜ n_t_ew ∗ ew_y
- t_ew_r2r ≜ n_t_ew ∗ ew_r
- t_ew_g2g ≜ n_t_ew ∗ ew_g
- glb_ew0 ≜ ew_g + ew_y
- glb_ew ≜ ew_g + glb_ew0
- glb_ew = glb
- glb_t_ew ≜ t_ew + n_t_ew
- glb_t_ew = glb
- t_ew#n_t_ew
- ew_g # ew_r
- ew_y # ew_r
- ew_g # ew_y
- next_t_ew_r ≜ t_ew_y2r$1 on glb
- next_t_ew_r ⊆ ew_r
- next_t_ew_g ≜ t_ew_r2g$1 on glb
- next_t_ew_g ⊆ ew_g
- next_t_ew_y ≜ t_ew_g2y$1 on glb
- next_t_ew_y ⊆ ew_y
- next_t_ew_r ≜ t_ew_y2r$1 on glb
- next_t_ew_r ⊆ ew_r
- next_t_ew_g ≜ t_ew_r2g$1 on glb
- next_t_ew_g ⊆ ew_g
- next_t_ew_y ≜ t_ew_g2y$1 on glb
- next_t_ew_y ⊆ ew_y

(a) CCSL constraints of E-W traffic light

- t_ns_g2y ≜ t_ns ∗ ns_g
- t_ns_y2r ≜ t_ns ∗ ns_y
- t_ns_r2g ≜ t_ns ∗ ns_r
- t_ns_y2y ≜ n_t_ns ∗ ns_y
- t_ns_r2r ≜ n_t_ns ∗ ns_r
- t_ns_g2g ≜ n_t_ns ∗ ns_g
- glb_ns0 ≜ ns_g + ns_y
- glb_ns ≜ ns_g + glb_ns0
- glb_ns = glb
- glb_t_ns ≜ t_ns + n_t_ns
- glb_t_ns = glb
- t_ns#n_t_ns
- ns_g # ns_r
- ns_y # ns_r
- ns_g # ns_y
- next_ew_r ≜ t_ns_r2r$1 on glb
- next_ew_r ⊆ ns_r
- next_ew_g ≜ t_ns_g2g$1 on glb
- next_ew_g ⊆ ns_g
- next_ew_y ≜ t_ns_y2y$1 on glb
- next_ew_y ⊆ ns_y
- next_ns_r ≜ t_ns_r2r$1 on glb
- next_ns_r ⊆ ns_r
- next_ns_g ≜ t_ns_g2g$1 on glb
- next_ns_g ⊆ ns_g
- next_ns_y ≜ t_ns_y2y$1 on glb
- next_ns_y ⊆ ns_y

(b) CCSL constraints of N-S traffic light

- ew_g_y ≜ ew_g + ew_y
- ns_g_y ≜ ns_g + ns_y
- ew_g_y#ns_g_y
- ew_r#ns_r
- t_ew_y2r =??
- t_ew_r2g =??
- glb ≜ glb_ew + glb_ns

(c) CCSL constraints of Controller

Figure 4: Incomplete CCSL specification generated from templates and properties

To fill the holes by using our RL-based CCSL synthesis approach, we set the number of RL training rounds to 500. To guide the synthesis, we generated 5 timing traces with a length of 50 each, which describe the behaviors of different traffic scenarios. Figure 5 shows the synthesis result, where the filled parts are in red, denoting that the transition from ns_r to ns_g is involved in Syn2 and the transition from ns_y to ns_r is involved in Syn1.

- $t\_ew\_g2y \triangleq t\_ew \ast ew\_g$
- $t\_ew\_y2r \triangleq t\_ew \ast ew\_y$
- $t\_ew\_r2g \triangleq t\_ew \ast ew\_r$
- $t\_ew\_y2y \triangleq n\_t\_ew \ast ew\_y$
- $t\_ew\_r2r \triangleq n\_t\_ew \ast ew\_r$
- $t\_ew\_g2g \triangleq n\_t\_ew \ast ew\_g$
- $glb\_ew0 \triangleq ew\_g + ew\_y$
- $glb\_ew \triangleq ew\_g + glb\_ew0$
- $glb\_ew = glb$
- $glb\_t\_ew \triangleq t\_ew + n\_t\_ew$
- $glb\_t\_ew = glb$
- $t\_ew\#n\_t\_ew$
- $ew\_g \# ew\_r$
- $ew\_y \# ew\_r$
- $ew\_g \# ew\_y$
- $next\_t\_ew\_r \triangleq t\_ew\_y2r\$1\ on\ glb$
- $next\_t\_ew\_r \subseteq ew\_r$
- $next\_t\_ew\_g \triangleq t\_ew\_r2g\$1\ on\ glb$
- $next\_t\_ew\_g \subseteq ew\_g$
- $next\_t\_ew\_y \triangleq t\_ew\_g2y\$1\ on\ glb$
- $next\_t\_ew\_y \subseteq ew\_y$
- $next\_t\_ew\_r \triangleq t\_ew\_y2r\$1\ on\ glb$
- $next\_t\_ew\_r \subseteq ew\_r$
- $next\_t\_ew\_g \triangleq t\_ew\_r2g\$1\ on\ glb$
- $next\_t\_ew\_g \subseteq ew\_g$
- $next\_t\_ew\_y \triangleq t\_ew\_g2y\$1\ on\ glb$
- $next\_t\_ew\_y \subseteq ew\_y$

(a) CCSL constraints of E-W traffic light

- $t\_ns\_g2y \triangleq t\_ns \ast ns\_g$
- $t\_ns\_y2r \triangleq t\_ns \ast ns\_y$
- $t\_ns\_r2g \triangleq t\_ns \ast ns\_r$
- $t\_ns\_y2y \triangleq n\_t\_ns \ast ns\_y$
- $t\_ns\_r2r \triangleq n\_t\_ns \ast ns\_r$
- $t\_ns\_g2g \triangleq n\_t\_ns \ast ns\_g$
- $glb\_ns0 \triangleq ns\_g + ns\_y$
- $glb\_ns \triangleq ns\_g + glb\_ns0$
- $glb\_ns = glb$
- $glb\_t\_ns \triangleq t\_ns + n\_t\_ns$
- $glb\_t\_ns = glb$
- $t\_ns\#n\_t\_ns$
- $ns\_g \# ns\_r$
- $ns\_y \# ns\_r$
- $ns\_g \# ns\_y$
- $next\_ew\_r \triangleq t\_ns\_r2r\$1\ on\ glb$
- $next\_ew\_r \subseteq ns\_r$
- $next\_ew\_g \triangleq t\_ns\_g2g\$1\ on\ glb$
- $next\_ew\_g \subseteq ns\_g$
- $next\_ew\_y \triangleq t\_ns\_y2y\$1\ on\ glb$
- $next\_ew\_y \subseteq ns\_y$
- $next\_ns\_r \triangleq t\_ns\_r2r\$1\ on\ glb$
- $next\_ns\_r \subseteq ns\_r$
- $next\_ns\_g \triangleq t\_ns\_g2g\$1\ on\ glb$
- $next\_ns\_g \subseteq ns\_g$
- $next\_ns\_y \triangleq t\_ns\_y2y\$1\ on\ glb$
- $next\_ns\_y \subseteq ns\_y$

(b) CCSL constraints of N-S traffic light

- $ew\_g\_y \triangleq ew\_g + ew\_y$
- $ns\_g\_y \triangleq ns\_g + ns\_y$
- $ew\_g\_y\#ns\_g\_y$
- $ew\_r\#ns\_r$
- $t\_ew\_y2r = \boldsymbol{t\_ns\_r2g}$
- $t\_ew\_r2g = \boldsymbol{t\_ns\_y2r}$
- $glb \triangleq glb\_ew + glb\_ns$

(c) CCSL constraints of Controller

Figure 5: Synthesized complete CCSL specification for the traffic light control system