Model-Checking General Temporal Formulas

Given an FDS $\mathcal{D}$ and a temporal formula φ, we wish to check that φ is valid over $\mathcal{D}$. This can be done according to the following recipe:

1. Construct the temporal tester $T_{\neg\varphi}$. This is an FDS whose observations are all the sequences satisfying $\neg\varphi$.
2. Form the synchronous composition $\mathcal{D}_C = \mathcal{D} \;|||\; T_{\neg\varphi}$. This is an FDS whose computations correspond to computations of $\mathcal{D}$ which satisfy $\neg\varphi$, i.e., violate φ.
3. Check that $\mathcal{D}_C$ is infeasible, i.e., have no computations.
4. Conclude that φ is $\mathcal{D}$-valid.
5. In case $\mathcal{D}_C$ is feasible, then any computation of $\mathcal{D}_C$ is a counter-example, i.e., a computation of $\mathcal{D}$ which violates φ.

The correctness of this prescription follows from

Claim 12. *The formula φ is $\mathcal{D}$-valid iff the FDS $\mathcal{D} \;|||\; T_{\neg\varphi}$ has no computations.*

It only remains to show how to check feasibility of an FDS and how to construct the tester T_ψ.

Checking the Feasibility of an FDS

The following algorithm checks whether the FDS $\mathcal{D}$ is feasible.

Algorithm CK-FEAS $(\mathcal{D})$ — Check whether system $\mathcal{D}$ is feasible

$feas$: **assertion**

1. $feas :=$ SET-FEASIBLE$(\mathcal{D})$ — — All states initiating a fair run
2. **return** $\Theta_{\mathcal{D}} \wedge feas$ — — All initial states initiating a computation

This algorithm returns a 0 result iff FDS $\mathcal{D}$ is infeasible. In case it returns a non-empty result, we can use it to extract and print a computation of $\mathcal{D}$ in a way similar to Algorithm SMC-RESP.

Temporal Testers

The missing element in the plan for reducing the verification problem $\mathcal{D} \models \psi$ to checking feasibility of the composed system $\mathcal{D} \;|||\; Tester_{\neg\psi}$, is a recipe for constructing the temporal tester $T_{\neg\psi}$.

Given a temporal formula φ, the tester T_φ is an FDS whose observations are all the sequences satisfying φ. We describe a construction of such a tester, called the tableau construction, for building such an FDS.

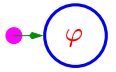
Transforming to positive form As a first step, we transform φ to a formula in a positive form, which means that negations are only applied to state-formulas. This transformation is achieved by repeated application of the following rewrite rules until the formula is in positive form:

$$
\begin{array}{lcl}
\neg\neg p & \longrightarrow & p \\
\neg(p \;\wedge\; q) & \longrightarrow & \neg p \;\vee\; \neg q \\
\neg(p \;\vee\; q) & \longrightarrow & \neg p \;\wedge\; \neg q \\
\neg\Box\, p & \longrightarrow & \Diamond\, \neg p \\
\neg\Diamond\, p & \longrightarrow & \Box\, \neg p \\
\neg\bigcirc p & \longrightarrow & \bigcirc \neg p \\
\neg(p \;\mathcal{U}\; q) & \longrightarrow & (\neg q)\; \mathcal{W}\; (\neg p \;\wedge\; \neg q) \\
\neg(p \;\mathcal{W}\; q) & \longrightarrow & (\neg q)\; \mathcal{U}\; (\neg p \;\wedge\; \neg q) \\
\neg\boxminus\, p & \longrightarrow & \diamondminus\, \neg p \\
\neg\diamondminus\, p & \longrightarrow & \boxminus\, \neg p \\
\neg(p \;\mathcal{S}\; q) & \longrightarrow & (\neg q)\; \mathcal{B}\; (\neg p \;\wedge\; \neg q) \\
\neg(p \;\mathcal{B}\; q) & \longrightarrow & (\neg q)\; \mathcal{S}\; (\neg p \;\wedge\; \neg q) \\
\neg\ominus p & \longrightarrow & \circledcirc \neg p \\
\neg\circledcirc p & \longrightarrow & \ominus \neg p
\end{array}
$$

Tableau Construction

The tableau T is a directed graph whose nodes are labeled by sets of formulas which are either sub-formulas of φ, or a formula of the form $\bigcirc p$ where $p \in \varphi$.

Initially, we place in T an initial node labeled by φ.

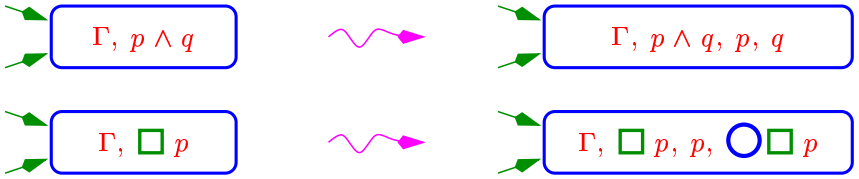


Next, alternately apply Steps 1 and 2 until they no longer affect the tableau:

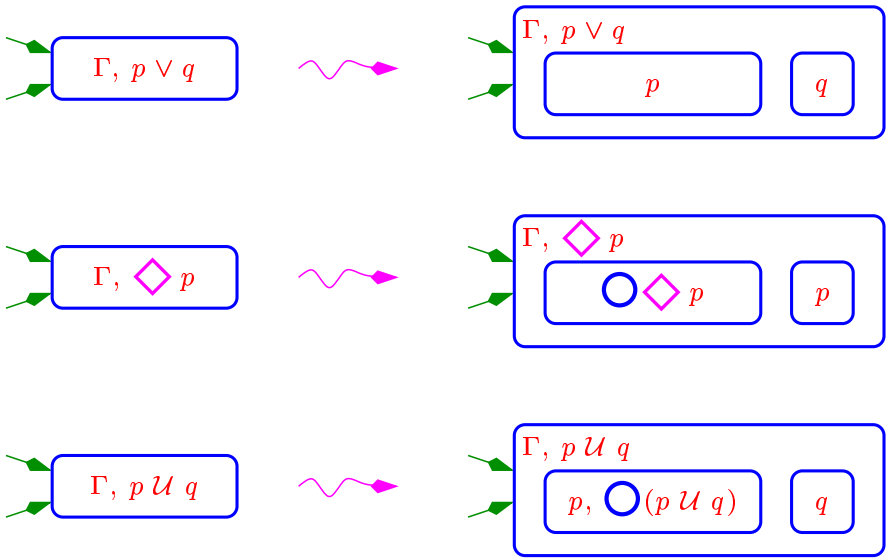
Step 1: Local Expansions

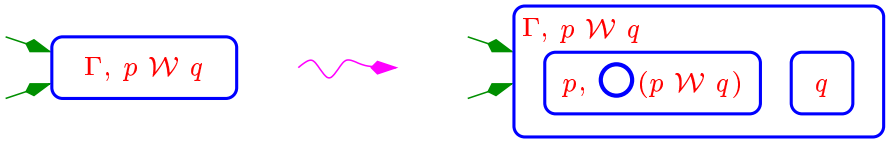
Repeatedly apply the following expansion rules until no further change:

- Conjunctive expansions



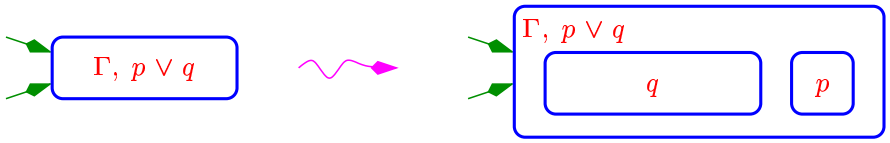
- Disjunctive expansions



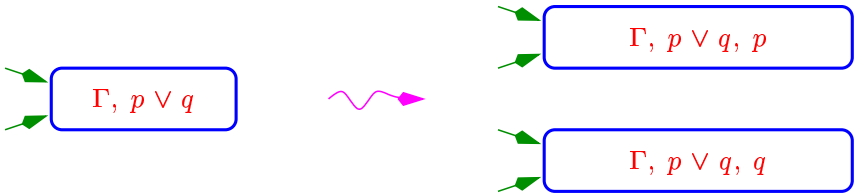


Statecharts Conventions

In the preceding rules, we made use of several statecharts conventions. Thus, the rewrite rule:



is an abbreviation for



Step 2: Next Expansion

Pick a node n to which the next expansion has not been applied yet. Assume that its label is of the form

$$p_1,\ \ldots,\ p_m;\ \bigcirc q_1,\ \ldots,\ \bigcirc q_k,$$

where the principal operator of the formulas $p_1, \ldots, p_m$ is other than $\bigcirc$.

Add to the tableau T a new node n' labeled by $q_1, \ldots, q_k$, if such a node does not already exists in T. In any case draw an edge connecting n to n'.

This will lead to the following structure within T:

Whenever in the construction we encounter a propositionally inconsistent node, i.e. a node whose label contains the formulas p and $\neg p$, such a node must be removed from the tableau.

Also, whenever we detect two nodes n_i and n_j which have been fully locally expanded, whose labels contain the same propositional formulas and the same $\bigcirc$-formulas, then n_i and n_j can be merged (identified).

Summing it Up

When the above construction terminates, it defines for us the set of reachable states and the succession relation within the FDS T_φ. Assume that the reachable states are $s_0, \ldots, s_m$ and let $E \subseteq [0..m] \times [0..m]$ be the set of pairs (i, j) such that there exists an edge in the tableau connecting s_i to s_j. Let $\lambda_0, \ldots, \lambda_m$ be the labels of the nodes (states) $s_0, \ldots, s_m$, respectively. Let Π be the set of propositions which appear in the formula φ.

For a node n_i, we denote by $prop_i$ the conjunction of the non-temporal formulas within λ_i. Note that $prop_i$ does not necessarily assign values to all the propositions in Π.

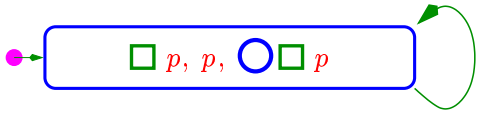
The FDS T_φ

We are now ready to define the FDS T_φ.

- For the state variables we take $V = \{\kappa : [0..m]\} \cup \Pi$. Thus, we take all the propositions appearing in φ plus a control variable κ which ranges over $[0..m]$.
- $\mathcal{O} = \Pi$. Only the propositions appearing in φ are observable.
- $\Theta : \bigvee_{\varphi \in \lambda_i} (prop_i \ \wedge \ \kappa = i)$. Thus, the initial states are all the states s_i which include φ in their label.
- $\rho : \bigvee_{(i,j) \in E} (prop_i \ \wedge \ \kappa = i \ \wedge \ prop_j' \ \wedge \ \kappa' = j)$. Thus, the possible transitions are determined by the edges connecting nodes within the tableau, and every state s_i imposes the valuation $prop_i$.
- For every sub-formula $\Diamond p \in \varphi$, $\mathcal{J}$ includes the requirement
  $J_{\Diamond p} : \bigvee_{p \in \lambda_i} (\kappa = i) \ \vee \ \bigvee_{\Diamond p \notin \lambda_j} (\kappa = j)$.
  For every sub-formula $p \,\mathcal{U}\, q \in \varphi$, $\mathcal{J}$ includes the requirement
  $J_{p\mathcal{U}q} : \bigvee_{q \in \lambda_i} (\kappa = i) \ \vee \ \bigvee_{p\mathcal{U}q \notin \lambda_j} (\kappa = j)$.

Example: A Tester for $\Box\, p$

Constructing the tableau for $\Box\, p$, we obtain

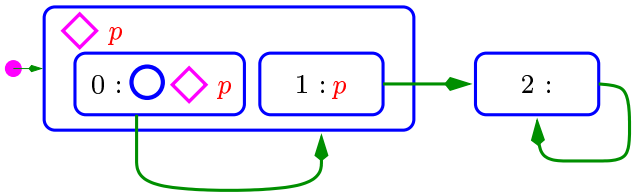


which leads to the following FDS $T_{\Box p}$:

$$
\begin{array}{ll}
V = \mathcal{O}: & p : \textbf{boolean} \\
\Theta: & p \\
\rho: & p \ \wedge \ p' \\
\mathcal{J} = \mathcal{C}: & \emptyset
\end{array}
$$

A Tester for $\Diamond\, p$

The tableau for $\Diamond\, p$ is:

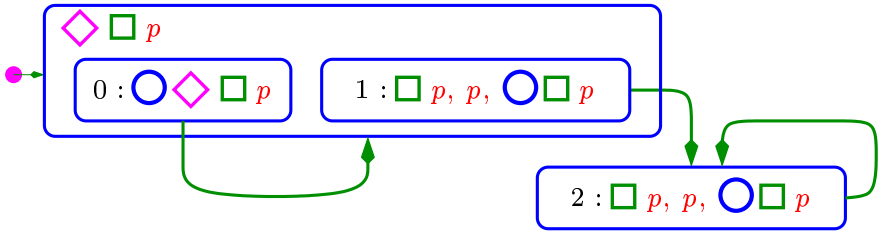


Leading to the FDS $T_{\Diamond p}$:

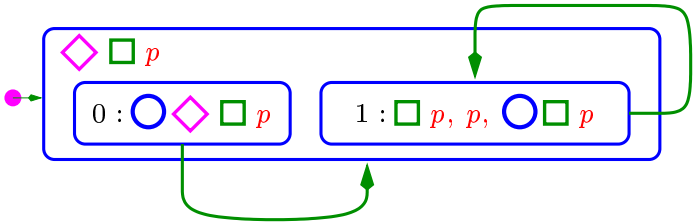
$$
\begin{array}{ll}
V: & \{\kappa : [0..2];\ p : \textbf{boolean}\} \\
\mathcal{O}: & p \\
\Theta: & \kappa = 0 \ \vee \ \kappa = 1 \wedge p \\
\rho: & \kappa = 0 \wedge (\kappa' = 0 \ \vee \ \kappa' = 1 \wedge p') \quad \vee \quad \kappa \in \{1, 2\} \wedge \kappa' = 2 \\
\mathcal{J}: & J_{\Diamond p} : \kappa \in \{1, 2\}
\end{array}
$$

A Tester for $\Diamond\,\Box\,p$

The tableau for $\Diamond\,\Box\,p$ is:



Observing that λ_1 and λ_2 agree on the set of propositional formulas ($\{p\}$) and the set of $\bigcirc$-formulas ($\{\bigcirc\,\Box\,p\}$), we identify n_2 with n_1. This leads to the tableau:



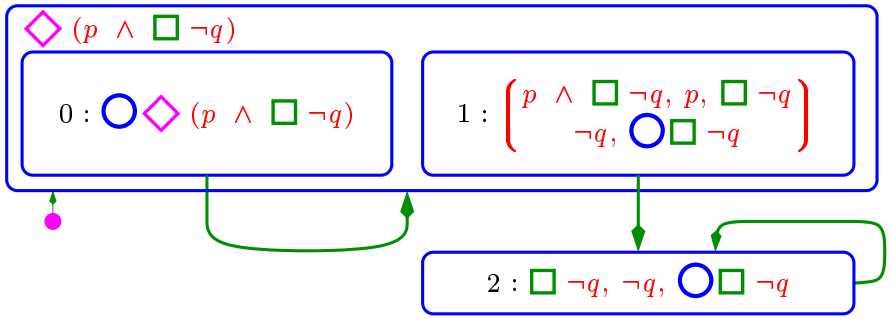
Whose corresponding FDS $T_{\Diamond\Box p}$ is:

$$
\begin{array}{ll}
V: & \{\kappa : [0..1];\ p : \textbf{boolean}\} \\
\mathcal{O}: & p \\
\Theta: & \kappa = 0 \ \vee\ \kappa = 1 \wedge p \\
\rho: & \kappa = 0 \wedge \kappa' = 0 \quad \vee \quad \kappa \in \{0,1\} \wedge \kappa' = 1 \wedge p' \\
\mathcal{J}: & J_{\Diamond\Box p} : \kappa = 1
\end{array}
$$

A Tester for $\Diamond\,(p\ \wedge\ \Box\,\neg q)$

For our final example, we construct a tester for the formula $\Diamond\,(p \wedge \Box\,\neg q)$. This formula is of interest because it is the negation of the formula $\Box\,(p \rightarrow \Diamond\,q)$.

The tableau for this formula is given by



Leading to the FDS:

$$
\begin{array}{ll}
V: & \{\kappa : [0..2];\ p, q : \textbf{boolean}\} \\
\mathcal{O}: & \{p,\ q\} \\
\Theta: & \kappa = 0 \ \vee\ \kappa = 1 \wedge p \wedge \neg q \\
\rho: & \left\{\begin{array}{ll} & \kappa = 0 \qquad \wedge\ (\kappa' = 0 \ \vee\ \kappa' = 1 \wedge p' \wedge \neg q') \\ \vee & \kappa \in \{1,2\} \wedge \kappa' = 2 \wedge \neg q' \end{array}\right\} \\
\mathcal{J}: & J_{\Diamond(p\wedge\Box\neg q)} : \kappa \in \{1,2\}
\end{array}
$$

Verifying Mutual Exclusion for MUX-SEM

We wish to verify that program MUX-SEM satisfies the property of mutual exclusion which can be specified by the formula

$$\psi: \quad \Box\, \neg(C_1 \ \wedge \ C_2)$$

The negation of this formula is given by

$$\varphi = \neg\psi: \quad \Diamond\, (C_1 \ \wedge \ C_2)$$

Following the tableau construction, we obtain the tester $T_{inclusion}$ given by:

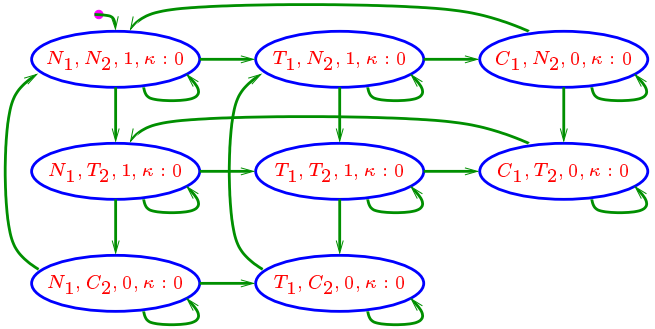
$$\begin{array}{ll} V: & \{\kappa : [0..2];\ C_1, C_2 : \textbf{boolean}\} \\ \mathcal{O}: & \{C_1,\ C_2\} \\ \Theta: & \kappa = 0 \ \vee \ \kappa = 1 \wedge C_1 \wedge C_2 \\ \rho: & \left\{ \begin{array}{ll} & \kappa = 0 \qquad \wedge (\kappa' = 0 \ \vee \ \kappa' = 1 \wedge C_1' \wedge C_2') \\ \vee & \kappa \in \{1,2\} \wedge \kappa' = 2 \end{array} \right\} \\ \mathcal{J}: & J_{\Diamond(C_1 \wedge C_2)} : \kappa \in \{1,2\} \end{array}$$

MUX-SEM Satisfies Mutual Exclusion

The state-transition graph for

$$\text{MUX-SEM} \quad \| \quad T_{inclusion}$$

is given by



Applying Algorithm FAIR-SUB to this graph yields the empty set since the justice requirement $\kappa \in \{1,2\}$ is not satisfied by any state.

We conclude:

$$\text{MUX-SEM} \models \Box\, \neg(C_1 \wedge C_2)$$

Verifying Accessibility for MUX-SEM

The property of accessibility for process P_2 of program MUX-SEM can be expressed by the temporal formula

$$\psi : \quad \Box \Diamond \neg T_2$$

It's negation is given by

$$\varphi = \neg\psi : \quad \Diamond \Box T_2$$

A tester $T_{\neg acc}$ for this formula is $T_{\Diamond\Box p}$, given by:

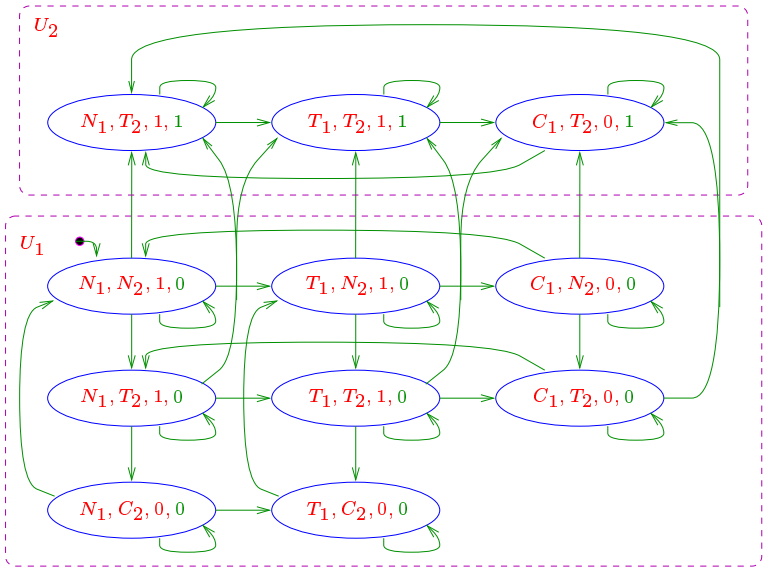
$$\begin{array}{ll} V: & \{\kappa : [0..1];\ T_2 : \textbf{boolean}\} \\ \mathcal{O}: & T_2 \\ \Theta: & \kappa = 0 \ \vee \ \kappa = 1 \wedge T_2 \\ \rho: & \kappa = 0 \wedge \kappa' = 0 \quad \vee \quad \kappa \in \{0,1\} \wedge \kappa' = 1 \wedge T_2' \\ \mathcal{J}: & J_{\Diamond\Box p} : \kappa = 1 \end{array}$$

MUX-SEM Satisfies Accessibility

The state-transition graph for

MUX-SEM ||| $T_{\neg acc}$

is given by



SCS U_1 is rejected because it is unjust towards $\kappa = 1$.

SCS U_2 is incompassionate towards $(T_2 \wedge y = 1,\ C_2)$. Eliminating the $(T_2 \wedge y = 1)$-states, this leaves us with $\langle C_1, T_2, 0, 1\rangle$ which is unjust towards C_1.

Temporal Testers for Formulas with Past

The previous incremental construction works only for future formulas. For formulas with past operators, we need a different construction, which we describe next.

Let φ be a formula in positive form with vocabulary U for which we wish to construct a temporal tester. A formula $p \in \varphi$ is called a principally temporal sub-formula if the main operator of p is temporal. Thus, the principally temporal sub-formulas of $\Box\,(p \rightarrow \Diamond\, q)$ are $\Box\,(p \rightarrow \Diamond\, q)$ and $\Diamond\, q$. Let $\mathcal{T}(\varphi)$ denote the set of principally temporal sub-formulas of φ.

Define a set of variables: $X_\varphi : \quad \{x_p \mid p \in \mathcal{T}(\varphi)\}$

For example, $X_{\Box(p \rightarrow \Diamond q)} \quad = \quad \{x_{\Box(p \rightarrow \Diamond q)},\ x_{\Diamond q}\}$

We introduce a stification transformation χ, mapping sub-formulas of φ into state formulas over $U \cup X_\varphi$, as follows:

$$\chi(\psi) \quad = \quad \begin{cases} \psi & \text{for } \psi \text{ a state formula} \\ \chi(p) \ \vee \ \chi(q) & \text{for } \psi = p \ \vee \ q \\ \chi(p) \ \wedge \ \chi(q) & \text{for } \psi = p \ \wedge \ q \\ x_\psi & \text{for } \psi \in \mathcal{T}(\varphi) \end{cases}$$

Construction Continued

For example, application of χ to the sub-formulas of $\Box\,(p \rightarrow \Diamond\, q)$ (equivalently $\Box\,(\neg p \ \vee \ \Diamond\, q)$) yields

$\psi \in \varphi$	$\chi(\psi)$
$\Box\,(p \rightarrow \Diamond\, q)$	$x_{\Box(p \rightarrow \Diamond q)}$
$p \rightarrow \Diamond\, q$	$p \rightarrow x_{\Diamond q}$
p	p
$\Diamond\, q$	$x_{\Diamond q}$
q	q

The tester T_φ is given by

$$T_\varphi \quad = \quad \left(\mathcal{D}_0 \quad ||| \quad \Big|||_{\psi \in \mathcal{T}(\varphi)} \mathcal{D}[\psi] \right) \Downarrow_U$$

We proceed to show how to construct $\mathcal{D}[\psi]$ for the various temporal formulas, recalling that the basic temporal operators for positive form formulas are

$$\{\bigcirc,\ \mathcal{U},\ \mathcal{W},\ \ominus,\ \circledcirc,\ \mathcal{S},\ \mathcal{B}\}$$

$\mathcal{D}[\bigcirc p]$, $\mathcal{D}[p\,\mathcal{U}\,q]$, and $\mathcal{D}[p\,\mathcal{W}\,q]$

The FDS $\mathcal{D}[\psi]$ for $\psi = \bigcirc p$ is given by

$$\begin{array}{rcl} V = \mathcal{O} & : & U \cup X_\psi \\ \Theta & : & 1 \\ \rho & : & x_\psi = \chi(p)' \\ \mathcal{J} = \mathcal{C} & : & \emptyset \end{array}$$

The FDS $\mathcal{D}[\psi]$ for $\psi = p\,\mathcal{U}\,q$ is given by

$$\begin{array}{rcl} V = \mathcal{O} & : & U \cup X_\psi \\ \Theta & : & 1 \\ \rho & : & x_\psi = \chi(q) \vee (\chi(p) \wedge x'_\psi) \\ \mathcal{J} & : & \neg x_\psi \vee \chi(q) \\ \mathcal{C} & : & \emptyset \end{array}$$

The FDS $\mathcal{D}[\psi]$ for $\psi = p\,\mathcal{W}\,q$ is given by

$$\begin{array}{rcl} V = \mathcal{O} & : & U \cup X_\psi \\ \Theta & : & 1 \\ \rho & : & x_\psi = \chi(q) \vee (\chi(p) \wedge x'_\psi) \\ \mathcal{J} = \mathcal{C} & : & \emptyset \end{array}$$

$\mathcal{D}[\ominus p]$ and $\mathcal{D}[\circledcirc p]$

The FDS $\mathcal{D}[\psi]$ for $\psi = \ominus p$ is given by

$$\begin{array}{rcl} V = \mathcal{O} & : & U \cup X_\psi \\ \Theta & : & \neg x_\psi \\ \rho & : & x'_\psi = \chi(p) \\ \mathcal{J} = \mathcal{C} & : & \emptyset \end{array}$$

Thus, the initial value of x_ψ for $\psi = \ominus p$ is always 0.

The FDS $\mathcal{D}[\psi]$ for $\psi = \circledcirc p$ is given by

$$\begin{array}{rcl} V = \mathcal{O} & : & U \cup X_\psi \\ \Theta & : & x_\psi \\ \rho & : & x'_\psi = \chi(p) \\ \mathcal{J} = \mathcal{C} & : & \emptyset \end{array}$$

Thus, the initial value of x_ψ for $\psi = \circledcirc p$ is always 1.

$\mathcal{D}[p\ \mathcal{S}\ q]$, $\mathcal{D}[p\ \mathcal{B}\ q]$, and $\mathcal{D}_0$

The FDS $\mathcal{D}[\psi]$ for $\psi = p\ \mathcal{S}\ q$ is given by

$$\begin{array}{rcl} V = \mathcal{O} & : & U \cup X_\psi \\ \Theta & : & x_\psi = \chi(q) \\ \rho & : & x'_\psi \quad = \quad \chi(q)' \ \vee\ (\chi(p)' \ \wedge\ x_\psi) \\ \mathcal{J} = \mathcal{C} & : & \emptyset \end{array}$$

The FDS $\mathcal{D}[\psi]$ for $\psi = p\ \mathcal{B}\ q$ is given by

$$\begin{array}{rcl} V = \mathcal{O} & : & U \cup X_\psi \\ \Theta & : & x_\psi = \chi(q)\ \vee\ \chi(p) \\ \rho & : & x'_\psi \quad = \quad \chi(q)' \ \vee\ (\chi(p)' \ \wedge\ x_\psi) \\ \mathcal{J} = \mathcal{C} & : & \emptyset \end{array}$$

Thus, $\mathcal{D}[p\ \mathcal{S}\ q]$ and $\mathcal{D}[p\ \mathcal{B}\ q]$ differ in their initial values which are $\chi(q)$ and $\chi(p)\ \vee\ \chi(q)$, respectively.

Finally, $\mathcal{D}_0$ is given by

$$\begin{array}{rcl} V = \mathcal{O} & : & U \cup X_\varphi \\ \Theta & : & \chi(\varphi) \\ \rho & : & 1 \\ \mathcal{J} = \mathcal{C} & : & \emptyset \end{array}$$

Example: $T_{\Box(p \rightarrow \Diamond q)}$

Taking $U = \{p, q, x_\Box, x_\Diamond\}$, the tester $T_{\Box(p \rightarrow \Diamond q)}$ is given by:

$$\left(\begin{array}{llll} \langle V = \mathcal{O} = U, & \Theta : x_\Box, & \rho : 1, & \mathcal{J} : \emptyset, \quad \mathcal{C} : \emptyset \quad \rangle \quad ||| \\ \langle V = \mathcal{O} = U, & \Theta : 1, & \rho : x_\Box \leftrightarrow (p \rightarrow x_\Diamond) \wedge {x_\Box}', & \\ & \mathcal{J} : \emptyset, & \mathcal{C} : \emptyset \quad \rangle \quad ||| & \Downarrow_{\{p,q\}} \\ \langle V = \mathcal{O} = U, & \Theta : 1, & \rho : x_\Diamond \leftrightarrow q \vee {x_\Diamond}', & \\ & \mathcal{J} : \neg x_\Diamond \vee q, & \mathcal{C} : \emptyset \quad \rangle & \end{array}\right)$$

Explanation and Motivation for the Construction

Consider first the simple case of a formula $\varphi = \Diamond q$, where q is a state formula. The partial tester for this formula according to the prescribed recipe is

$$\mathcal{D}[\Diamond q] : \left\{ \begin{array}{rcl} V = \mathcal{O} & : & \{q, x_\Diamond\} \\ \Theta & : & 1 \\ \rho & : & x_\Diamond \;=\; (q \;\vee\; x'_\Diamond) \\ \mathcal{J} & : & \neg x_\Diamond \;\vee\; q \\ \mathcal{C} & : & \emptyset \end{array} \right\}$$

We can prove the following:

Claim 13. *Let $\sigma : s_0, s_1, \ldots$ be a computation of $\mathcal{D}[\Diamond q]$ and $j \geq 0$ be a position. If $s_j[x_\Diamond] = 1$ then $(\sigma, j) \models \Diamond q$.*

Proof: Assume that $s_j[x_\Diamond] = 1$. Applying ρ to position j, we obtain that either $s_j \models q$ or $s_{j+1}[x_\Diamond] = 1$. Continuing in this manner to positions $j+1, j+2, \ldots$, we obtain that either there exists a $k \geq 0$ such that $s_k \models p$ or $s_i[x_\Diamond] = 1$ and $s_i \not\models q$ for all $i \geq j$. Since the second case violates the justice requirement $\neg x_\Diamond \vee q$, we are guaranteed that $s_k \models q$ for some $k \geq 0$ which, by the definition of $\Diamond$, implies $(s, j) \models \Diamond q$. ∎

Satisfaction Implies Computation of $\mathcal{D}[\Diamond q]$

Claim 13 showed that, within a computation, $\Diamond q$ holds whenever $x_\Diamond = 1$. The following claim establishes the other direction, namely, that a sequence in which $x_\Diamond = 1$ at precisely the positions which satisfy $\Diamond q$ is a computation of $\mathcal{D}[\Diamond q]$.

Claim 14. *Let $\sigma : s_0, s_1, \ldots$ be a $\{q, x_\Diamond\}$-sequence in which $s_j[x_\Diamond] = 1$ iff $(\sigma, j) \models \Diamond q$. Then, σ is a computation of $\mathcal{D}[\Diamond q]$.*

Proof: Since the formula $\Diamond q$ satisfies the expansion axiom

$$\Diamond q \iff q \;\vee\; \bigcirc \Diamond q,$$

it is obvious that $x_\Diamond$ satisfies the transition relation $x_\Diamond = q \;\vee\; x'_\Diamond$. It only remains to show that $x_\Diamond$ also satisfies the justice requirement $\neg x_\Diamond \;\vee\; q$.

We consider two cases. First assume that σ contains infinitely many q-positions (states satisfying q). Since $\Diamond q$ holds at each of these positions, we are guaranteed of having infinitely many positions at which $x_\Diamond = 1$.

In the other case, there are only finitely many q-positions. In this case, there exists a $j \geq 0$ such that there is no q-position beyond (or at) j. It follows that $\Diamond q$ is false at all positions beyond j and, therefore, there are infinitely many positions at which $x_\Diamond = 0$. ∎

A Tester for $\Box\, r$

Next, consider the case of a formula $\Box\, r$ where, again, we assume that r is a state formula. Since $\Box\, r \sim r\ \mathcal{W}\ 0$, we construct the partial tester for $1\ \mathcal{W}\ r$. This leads to the following partial tester:

$$\mathcal{D}[\Box\, r] : \left\{ \begin{array}{rcll} V = \mathcal{O} & : & \{r, x_{\Box}\} & \\ \Theta & : & 1 & \\ \rho & : & x_{\Box} & = \quad (r\ \wedge\ x_{\Box}') \\ \mathcal{J} = \mathcal{C} & : & \emptyset & \end{array} \right\}$$

which satisfies

Claim 15. *Let $\sigma : s_0, s_1, \ldots$ be a computation of $\mathcal{D}[\Box\, r]$ and $j \geq 0$ be a position. If $s_j[x_{\Box}] = 1$ then $(\sigma, j) \models \Box\, r$.*

Proof: Assume that $s_j[x_{\Box}] = 1$. Applying ρ to positions $j, j+1, \ldots$, we obtain that $s_i[x_{\Box}] = s_i[r] = 1$, for all $i \geq j$. By the definition of $\Box$, it follows that $(s, j) \models \Box\, r$. ∎

Satisfaction Implies Computation of $\mathcal{D}[\Box\, r]$

The other direction of Claim 15 states that a sequence in which $x_{\Box} = 1$ at precisely the positions which satisfy $\Box\, r$ is a computation of $\mathcal{D}[\Box\, r]$.

Claim 16. *Let $\sigma : s_0, s_1, \ldots$ be an $\{r, x_{\Box}\}$-sequence in which $s_j[x_{\Box}] = 1$ iff $(\sigma, j) \models \Box\, r$. Then, σ is a computation of $\mathcal{D}[\Box\, r]$.*

Proof: Since the formula $\Box\, r$ satisfies the expansion axiom

$$\Box\, r \iff r\ \wedge\ \bigcirc \Box\, r,$$

it is obvious that $x_{\Box}$ satisfies the transition relation $x_{\Box} = r\ \wedge\ x_{\Box}'$.

A Tester for $\Box\,(p \rightarrow \Diamond\, q)$

Next, let us consider the formula $\varphi = \Box\,(p \rightarrow \Diamond\, q)$. The partial tester proposed by our recipe is equivalent to

$$\mathcal{D}[\Box\,(p \rightarrow \Diamond\, q)] : \left\{ \begin{array}{lcl} V = \mathcal{O} & : & \{p, q, x_{\Box}, x_{\Diamond}\} \\ \Theta & : & 1 \\ \rho & : & \left(\begin{array}{lcl} x_{\Diamond} & = & q \ \vee \ x'_{\Diamond} \\ \wedge \ x_{\Box} & = & (p \rightarrow x_{\Diamond}) \ \wedge \ x_{\Box}{}' \end{array} \right) \\ \mathcal{J} & : & (\neg x_{\Diamond} \ \vee \ q) \\ \mathcal{C} & : & \emptyset \end{array} \right\}$$

Note that $p \rightarrow x_{\Diamond} \ \sim \ \chi(p \rightarrow \Diamond\, q)$. The correctness of this constructions is stated by

Claim 17. *Let $\sigma : s_0, s_1, \ldots$ be a computation of $\mathcal{D}[\Box\,(p \rightarrow \Diamond\, q)]$ and $j \geq 0$ be a position. If $s_j[x_{\Box}] = 1$ then $(\sigma, j) \models \Box\,(p \rightarrow \Diamond\, q)$.*

Proof: Since $p \rightarrow x_{\Diamond}$ is a state formula, Claim 15 implies that $s_j[x_{\Box}] = 1$ implies $(\sigma, j) \models \Box\,(p \rightarrow x_{\Diamond})$. By Claim 13, $(\sigma, k) \models \Diamond\, q$ holds at all positions $k \geq j$ in which $s_k[x_{\Diamond}] = 1$. Combining these two facts, we get that $s_j[x_{\Box}] = 1$ implies $(\sigma, j) \models \Box\,(p \rightarrow \Diamond\, q)$. ∎

Satisfaction Implies Computation of $\mathcal{D}[\Box\,(p \rightarrow \Diamond\, q)]$

The other direction of Claim 17 states that a sequence in which $x_{\Box} = 1$ at precisely the positions which satisfy $\Box\,(p \rightarrow \Diamond\, q)$ is a computation of $\mathcal{D}[\Box\,(p \rightarrow \Diamond\, q)]$.

Claim 18. *Let $\sigma : s_0, s_1, \ldots$ be a $\{p, q, x_{\Box}, x_{\Diamond}\}$-sequence in which $s_j[x_{\Box}] = 1$ iff $(\sigma, j) \models \Box\,(p \rightarrow \Diamond\, q)$ and $s_j[x_{\Diamond}] = 1$ iff $(\sigma, j) \models \Diamond\, q$. Then, σ is a computation of $\mathcal{D}[\Box\,(p \rightarrow \Diamond\, q)]$.*

Proof: Due to the expansion formulas of $\Diamond$ and $\Box$ and the fact that $\Diamond\, q$ holds precisely when $x_{\Diamond} = 1$, the two clauses of the transition relation obviously hold at all positions. By an argument similar to that of Claim 14, we can show that the justice requirement $\neg x_{\Diamond} \ \vee \ q$ also holds. It follows that σ is a computation of $\mathcal{D}[\Box\,(p \rightarrow \Diamond\, q)]$. ∎

Adding $\mathcal{D}_0$

Finally, let us add the component $\mathcal{D}_0$, which imposes the initial condition $\chi(\varphi)$. For the case of $\varphi = \Box\,(p \rightarrow \Diamond\, q)$, this leads to the following full temporal tester:

$$T_{\Box(p\rightarrow \Diamond q)} : \left\{ \begin{array}{lcl} V = \mathcal{O} & : & \{p, q, x_{\Box}, x_{\Diamond}\} \\ \Theta & : & x_{\Box} \\ \rho & : & \left(\begin{array}{rcl} x_{\Diamond} & = & q \ \vee \ x'_{\Diamond} \\ \wedge \ x_{\Box} & = & (p \rightarrow x_{\Diamond}) \ \wedge \ x_{\Box}{}' \end{array} \right) \\ \mathcal{J} & : & (\neg x_{\Diamond} \ \vee \ q) \\ \mathcal{C} & : & \emptyset \end{array} \right\} \Downarrow_{\{p,q\}}$$

The correctness of this construction is stated by the following claim.

Claim 19. *The $\{p, q\}$-sequence $\sigma : s_0, s_1, \ldots$ is an observation of $T_{\Box(p\rightarrow \Diamond q)}$ iff $\sigma \models \Box\,(p \rightarrow \Diamond\, q)$.*

Proof: First, assume that $\sigma : s_0, s_1, \ldots$ is an observation of $T_{\Box(p\rightarrow \Diamond q)}$. It follows that there exists a $\{p, q, x_{\Box}, x_{\Diamond}\}$-sequence $\widetilde{\sigma} : \widetilde{s}_0, \widetilde{s}_1, \ldots$, such that $\widetilde{\sigma}$ is a computation of $T_{\Box(p\rightarrow \Diamond q)}$ and $\widetilde{\sigma}\Downarrow_{\{p,q\}} = \sigma$. Since the initial condition of $T_{\Box(p\rightarrow \Diamond q)}$ is $x_{\Box}$, it follows that $\widetilde{s}_0[x_{\Box}] = 1$. Clearly, $T_{\Box(p\rightarrow \Diamond q)}$ differs from $\mathcal{D}(\Box\,(p \rightarrow \Diamond\, q))$ only in its initial condition which implies the initial condition of $\mathcal{D}(\Box\,(p \rightarrow \Diamond\, q))$. Consequently, $\widetilde{\sigma}$ is also a computation of $\mathcal{D}(\Box\,(p \rightarrow \Diamond\, q))$ and, by Claim 17, $\widetilde{s}_0[x_{\Box}] = 1$ implies $(\widetilde{s}, 0) \models \Box\,(p \rightarrow \Diamond\, q)$, from which we can conclude $\sigma \models \Box\,(p \rightarrow \Diamond\, q)$.

In the other direction, assume that $\sigma : s_0, s_1, \ldots$ is a $\{p, q\}$-sequence satisfying $\Box\,(p \rightarrow \Diamond\, q)$. We extend σ into a $\{p, q, x_{\Box}, x_{\Diamond}\}$-sequence $\widetilde{\sigma} : \widetilde{s}_0, \widetilde{s}_1, \ldots$ by assigning to each state $\widetilde{s}_j$ an evaluation for $x_{\Box}, x_{\Diamond}$ as follows:

$$\begin{array}{lcl} \widetilde{s}_j[x_{\Diamond}] = 1 & \Longleftrightarrow & (\sigma, j) \models \Diamond\, q \\ \widetilde{s}_j[x_{\Box}] = 1 & \Longleftrightarrow & (\sigma, j) \models \Box\,(p \rightarrow \Diamond\, q) \end{array}$$

By Claim 18, $\widetilde{\sigma}$ is a computation of the partial tester $\mathcal{D}[\Box\,(p \rightarrow \Diamond\, q)]$. Since σ (and hence $\widetilde{\sigma}$) satisfy $\Box\,(p \rightarrow \Diamond\, q)$, it follows that $\widetilde{s}_0[x_{\Box}] = 1$ and therefore $\widetilde{\sigma}$ is also a computation of $T_{\Box(p\rightarrow \Diamond q)}$. It follows that σ is an observation of $T_{\Box(p\rightarrow \Diamond q)}$.

State-Transition Diagram for $T_{\Box(p \rightarrow \Diamond q)}$