Solution 5.3: The two formulas are not equivalent, and in fact, none is stronger than the other.

Consider a trace ρ such that the formula φ_1 holds at every position and the formula φ_2 holds at no position. Then the trace does not satisfy the until-formula $(\varphi_1 \,\mathcal{U}\, \varphi_2)$, and thus, satisfies $\neg(\varphi_1 \,\mathcal{U}\, \varphi_2)$. However, it does not satisfy the until-formula $(\neg\varphi_2)\,\mathcal{U}\,(\neg\varphi_1)$. This shows that $\neg(\varphi_1 \,\mathcal{U}\, \varphi_2)$ does not imply $(\neg\varphi_2)\,\mathcal{U}\,(\neg\varphi_1)$.

Conversely, consider a trace ρ such that at the first position φ_2 holds but φ_1 does not hold. By the semantics of the until-formulas, the trace satisfies both $(\neg\varphi_2)\,\mathcal{U}\,(\neg\varphi_1)$ and $(\varphi_1 \,\mathcal{U}\, \varphi_2)$. This shows that $(\neg\varphi_2)\,\mathcal{U}\,(\neg\varphi_1)$ does not imply $\neg(\varphi_1 \,\mathcal{U}\, \varphi_2)$. ■

Solution 5.4: The two formulas are equivalent. We will show that if a trace satisfies $\Box\Diamond(\varphi_1 \wedge \Diamond\varphi_2)$, then it must satisfy $\Box\Diamond(\varphi_2 \wedge \Diamond\varphi_1)$. By a symmetric argument, it follows that if a trace satisfies $\Box\Diamond(\varphi_2 \wedge \Diamond\varphi_1)$, then it must satisfy $\Box\Diamond(\varphi_1 \wedge \Diamond\varphi_2)$, establishing the desired equivalence.

Suppose the trace ρ satisfies $\Box\Diamond(\varphi_1 \wedge \Diamond\varphi_2)$. Then, there must be infinitely many positions j such that $(\rho, j) \models (\varphi_1 \wedge \Diamond\varphi_2)$. Then, there must be infinitely many positions j such that φ_1 holds at position j and φ_2 holds at some position $k \geq j$. It follows that there must be infinitely many positions k where φ_2 holds, and furthermore, at each such position k, there must be a position $j \geq k$ where φ_1 holds. Thus there are infinitely many positions k such that $(\rho, k) \models (\varphi_2 \wedge \Diamond\varphi_1)$. Hence, the trace ρ satisfies $\Box\Diamond(\varphi_2 \wedge \Diamond\varphi_1)$. ■

Solution 5.5: The desired requirement is expressed by the following Ltl-formula:

$$\Box\Diamond(\mathit{inc} = 1) \;\rightarrow\; \Box\Diamond\,(\,\mathit{out}_0 = 1 \;\wedge\; \mathit{out}_1 = 1 \;\wedge\; \mathit{out}_2 = 1\,).$$

The circuit `3BitCounter` does not satisfy this specification. Suppose in every round both the input variables *start* and *inc* equal 1. The counter remains at zero in response to such a sequence of inputs. In the resulting execution the input *inc* is repeatedly high but the counter is never at its maximum value. This is a counterexample to the desired requirement. ■

Solution 5.6: The desired requirement is expressed by the following Ltl-formula:

$$\Box\,[\,(\,\mathit{on} = 1 \wedge \Box\neg\,\mathit{cruise}? \wedge \Box\neg\,\mathit{inc}? \wedge \Box\neg\,\mathit{dec}?\,) \;\rightarrow\; \Diamond\Box\,(\,\mathit{speed} = \mathit{cruiseSpeed}\,)\,].$$

■

Solution 5.7: Consider a trace ρ and assume that it satisfies the formula φ_1. Then it satisfies either $\Box\Diamond\neg\,\mathit{Guard}(A)$ or $\Box\Diamond\,(\mathit{taken} = A)$. That is, there are infinitely many positions where either $\neg\,\mathit{Guard}(A)$ holds or $(\mathit{taken} = A)$ holds.

This implies that at every position j, there is a position $k \geq j$, where the disjunction $(\mathit{taken} = A) \vee \neg \mathit{Guard}(A)$ is satisfied. It follows that the always-formula φ_2 holds since to show that φ_2 is satisfied, we simply need to establish that every position j where the condition $\mathit{Guard}(A)$ holds is followed by a position $k \geq j$ where the disjunction $(\mathit{taken} = A) \vee \neg \mathit{Guard}(A)$ holds.

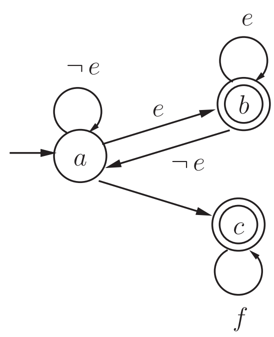
In the converse direction, suppose that a trace ρ satisfies the formula φ_2. To show that it also satisfies φ_1, assume that the trace also satisfies the antecedent $\Diamond \Box \mathit{Guard}(A)$ of φ_1. Then there exists a position j such that for all positions $k \geq j$, $\mathit{Guard}(A)$ holds at position k. Since the trace satisfies φ_2, it follows that for all positions $k \geq j$, the formula $\Diamond ((\mathit{taken} = A) \vee \neg \mathit{Guard}(A))$ holds. Since $\neg \mathit{Guard}(A)$ does not hold at any of these positions, it follows that for every position $k \geq j$, there is a later position where $(\mathit{taken} = A)$ is satisfied. This means that the trace satisfies the consequent $\Box \Diamond (\mathit{taken} = A)$ of φ_1. ■

Solution 5.8: 1. Along an execution where the input task A_z is executed at each step, the value of x stays stuck at 0, and thus, the process does not satisfy $\Diamond (\mathrm{x} > 5)$. If we assume weak fairness for the task A_x, since it is always enabled, we are guaranteed that it will be executed repeatedly along every fair execution. Thus, the value of x is guaranteed to be incremented repeatedly, and the process satisfies the eventuality property $\Diamond (\mathrm{x} > 5)$.

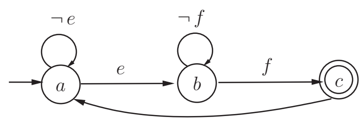
2. Along an execution where the task A_x is executed at each step, the value of y stays stuck at 0, and thus, the process does not satisfy $\Diamond (y > 5)$. In this specific execution, the value of z is also 0 at every step, and thus, the task A_y is never enabled and the execution is strongly-fair to the task A_y. Thus the specification is not satisfied even if we add fairness assumptions.

3. Along an execution where the input task A_z is executed at each step, the antecedent $\Box \Diamond (z = 1)$ is satisfied, but the value of y is stuck at 0. Thus, the specification $\Box \Diamond (z = 1) \rightarrow \Diamond (y > 5)$ is not satisfied. This execution is weakly-fair to the task A_y. However, if we assume strong fairness for the task A_y, then in every execution that satisfies the antecedent $\Box \Diamond (z = 1)$, the task A_y is repeatedly enabled, and is guaranteed to be executed repeatedly ensuring the satisfaction of $\Diamond (y > 5)$. Thus, the specification is satisfied assuming strong-fairness for the task A_y. ■

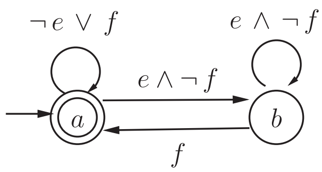
Solution 5.9: 1. The Büchi automaton is shown below. The transitions between the initial state a and the accepting state b are similar to the automaton of figure 5.5 for the formula $\Box \Diamond e$, and the nondeterministic switch to the accepting state c is similar to the automaton of figure 5.6 for the formula $\Diamond \Box f$. A trace is accepted exactly when either the state b is visited repeatedly, which can happen only when the trace contains infinitely many positions satisfying e, or when the state c is visited repeatedly, which can happen exactly when the property f holds persistently.



2. The Büchi automaton is shown below. In the initial state a, the automaton is waiting for an input where e holds, and on such an input, switches to the state b. In the state b, it is waiting for an input where f holds, and on such an input, switches to the accepting state c and then at the next step returns to the initial state. The accepting state is visited infinitely often precisely when the input trace satisfies e repeatedly and f repeatedly.



3. The Büchi automaton is shown below. The automaton switches to the state b when it encounters an input where e holds but f does not hold. In the state b, it is waiting to satisfy the formula $e\,\mathcal{U}\,f$: it returns to the initial accepting state a on an input that satisfies f and continues to wait as long as the inputs satisfy e.



■

Solution 5.10: Observe that the input e must be 1 at every step for the automaton to take a transition. In addition, the accepting state is visited repeatedly only if the input trace contains infinitely many positions where f equals 1. Thus a trace is accepted by the Büchi automaton of figure 5.9 precisely when it satisfies the LTL-formula $\Box\, e \wedge \Box \Diamond\, f$. ■

Solution 5.11: Suppose the Büchi automaton M_1 has states Q_1, initial states $Init_1$, accepting states F_1, and edges E_1, and the automaton M_2 has states Q_2, initial states $Init_2$, accepting states F_2, and edges E_2. We first construct the "product" automaton M_{12} over the input variables V as follows: the set initial states, and edges of the tableau M_φ are defined as before. In addition to the accepting sets corresponding to the always and eventuality subformulas, now for every until-subformula $\psi = \psi_1 \,\mathcal{U}\, \psi_2$ in the closure $Sub(\varphi)$, there is an accepting set F_ψ containing states q such that either $\psi_2 \in q$ or $\psi \notin q$.

The construction can be proved correct exactly as in the proof of proposition 5.2. Whenever an until-formula $\psi = \psi_1 \,\mathcal{U}\, \psi_2$ belongs to a state q along an accepting run of the tableau, the consistency condition ensures that either ψ_2 belongs to the state q (and in such a case, the satisfaction of ψ_2 suffices to ensure the satisfaction of ψ), or both ψ_1 and $\bigcirc\,\psi$ belong to q. In the latter case, the rules for adding edges in the tableau ensure that the next state along the execution contains ψ, and the accepting condition imposed by F_ψ ensures the eventual satisfaction of ψ_2 later in the execution. ■

Solution 5.15: Let $\varphi = (e \,\mathcal{U} \bigcirc f) \vee \neg\, e$ and let $\psi = (e \,\mathcal{U} \bigcirc f)$. Then

$$Sub(\varphi) \;=\; \{\, e, \neg\, e, f, \bigcirc f, \psi, \bigcirc\,\psi, \varphi \,\}.$$

The tableau has the following 16 states (that is, there are 16 consistent subsets of $Sub(\varphi)$):

$$\begin{array}{lcllcl}
q_0 & = & \{\, e, f, \bigcirc f, \bigcirc\,\psi, \psi, \varphi \,\}; & q_1 & = & \{\, e, f, \bigcirc f, \psi, \varphi \,\}; \\
q_2 & = & \{\, e, f, \bigcirc\,\psi, \psi, \varphi \,\}; & q_3 & = & \{\, e, f \,\}; \\
q_4 & = & \{\, e, \bigcirc f, \bigcirc\,\psi, \psi, \varphi \,\}; & q_5 & = & \{\, e, \bigcirc f, \psi, \varphi \,\}; \\
q_6 & = & \{\, e, \bigcirc\,\psi, \psi, \varphi \,\}; & q_7 & = & \{\, e \,\}; \\
q_8 & = & \{\, \neg\, e, f, \bigcirc f, \bigcirc\,\psi, \psi, \varphi \,\}; & q_9 & = & \{\, \neg\, e, f, \bigcirc f, \psi, \varphi \,\}; \\
q_{10} & = & \{\, \neg\, e, f, \bigcirc\,\psi, \varphi \,\}; & q_{11} & = & \{\, \neg\, e, f, \varphi \,\}; \\
q_{12} & = & \{\, \neg\, e, \bigcirc f, \bigcirc\,\psi, \psi, \varphi \,\}; & q_{13} & = & \{\, \neg\, e, \bigcirc f, \psi, \varphi \,\}; \\
q_{14} & = & \{\, \neg\, e, \bigcirc\,\psi, \varphi \,\}; & q_{15} & = & \{\, \neg\, e, \varphi \,\}.
\end{array}$$

Of these all states but q_3 and q_7 contain φ and are initial states. The edges are defined by the rules of the tableau. We give edges out of a few states as examples. The state q_0 has edges, all with the guard $e \wedge f$, to states that contain both f and ψ, that is, to states q_0, q_1, q_2, q_8, and q_9. The state q_{11} has edges, all with the guard $\neg\, e \wedge f$, to states that exclude both f and ψ, that is, to states q_7, q_{14}, and q_{15}. The state q_{13} has edges, all with the guard $\neg\, e \wedge \neg\, f$, to states that include f but exclude ψ, that is, to states q_3, q_{10} and q_{11}. The automaton M_φ has a single accepting set F_ψ: it contains all states that either include $\bigcirc f$ or exclude ψ, that is, all states except q_2 and q_6. ■

Solution 5.16: For the Büchi automaton shown below, on a given input trace, the automaton has an infinite execution (that is, the automaton can keep processing the inputs at each step) exactly when the input e has value 1 in every even position. If this is the case, the input trace is accepted since such an infinite execution contains the accepting state b repeatedly.