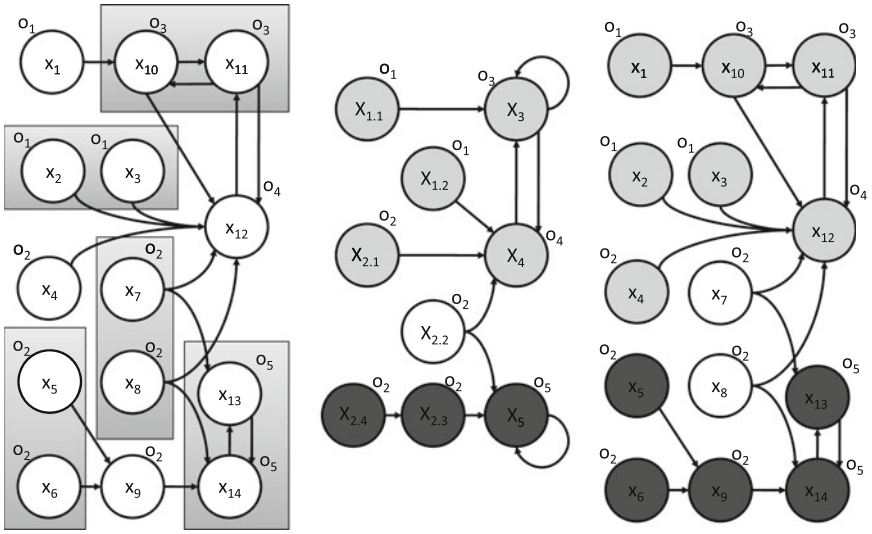
Studies in Systems, Decision and Control 89

Calin Belta
Boyan Yordanov
Ebru Aydin Gol

Formal Methods for Discrete-Time Dynamical Systems

Springer



(a) Equivalence classes $X/_{\approx}$. (b) Satisfying and violating states of the bisimulation quotient $T/_{\approx}$. (c) Satisfying and violating states of T.

Fig. 4.5 The bisimulation $\approx$ (**a**) of system T from Fig. 1.11a is obtained using Algorithm 1 by refining the equivalence classes $X/_{\sim}$ highlighted in Fig. 1.11a (the quotient $T/_{\sim}$ was shown in Fig. 1.11b). This allows the construction of the bisimulation quotient $T/_{\approx}$ (**b**). The largest satisfying (*light gray*), strictly violating (*dark gray*) and uncertain (*white*) regions of $T/_{\approx}$ for specification $\square\diamond o_3$ can then be identified using Algorithm 3. This allows for the computation of the largest satisfying, strictly violating and uncertain regions of T (**c**). For additional details, see Example 4.5

4.3 Iterative Strategies

In Sect. 4.1, we presented a model-checking-based strategy (Algorithm 3), which could be applied directly on T to provide a complete solution to Problem 4.1. However, this approach was computationally expensive when T included a large number of states and could not be used directly when T was infinite as will be discussed in subsequent chapters. Algorithm 3 could also be applied to the quotient $T/_{\sim}$ (constructed with the observational equivalence relation $\sim$) but in general this led only to an approximate solution to Problem 4.1, which was too conservative as illustrated in Sect. 4.2. Finally, Algorithm 3 could also be applied to the bisimulation quotient $T/_{\approx}$, thus providing a complete solution to Problem 4.1 as discussed in the previous section. However, such an analysis strategy required that the bisimulation algorithm (Algorithm 1) has terminated before applying model-checking techniques. While this approach guarantees that the largest satisfying region of the system is identified, it is not practical for certain systems (e.g., when many refinement steps are required to compute a bisimulation quotient) and cannot be applied directly to infinite systems, which will be our focus in subsequent chapters. As a compromise, instead of relying

saha.indra@gmail.com

on the termination of the bisimulation algorithm, the equivalence relation produced at each step of Algorithm 1 allows the construction of simulation quotients. Applying the analysis procedure (Algorithm 3) to these quotients leads to approximate solutions to Problem 4.1, which get increasingly better with subsequent iterations of Algorithm 1.

Formally, the equivalence relation $\sim_r$, initialized with the observational equivalence relation $\sim$ in Algorithm 1 and refined within the main loop of the algorithm, is used to construct quotient $T/_{\sim_r}$ at each step. Computing the largest satisfying and strictly violating regions $X^{\phi}_{T/_{\sim_r}}$ and $X^{\neg\phi}_{T/_{\sim_r}}$ of $T/_{\sim_r}$ through Algorithm 3 allows the computation of the subsets $con(X^{\phi}_{T/_{\sim_r}})$ and $con(X^{\neg\phi}_{T/_{\sim_r}})$ of the largest satisfying and strictly violating region of T. While the solution to Problem 4.1 obtained this way is not exact, a major advantage of the procedure is that an initial, rough approximation is obtained at little computational cost during the initial iterations and is improved through additional refinement (if more computational resources are available).

Let $\sim_1$ and $\sim_2$ denote the equivalence relation $\sim_r$ obtained during successive iterations of Algorithm 1 and consider the quotients $T/_{\sim_1} = (X/_{\sim_1}, \delta_{\sim_1}, O, o_{\sim_1})$ and $T/_{\sim_2} = (X/_{\sim_2}, \delta_{\sim_2}, O, o_{\sim_2})$. For any state $X_i \in X/_{\sim_2}$, $con(X_i) \subseteq con(X_j)$ for some $X_j \in X/_{\sim_1}$ (i.e., the equivalence classes of $X/_{\sim_1}$ are refined in $X/_{\sim_2}$). Given states $X_1, X_2 \in X/_{\sim_2}$ such that $con(X_1) \cup con(X_2) = con(X_i)$ for some $X_i \in X/_{\sim_1}$ (i.e., equivalence class X_i was refined into X_1 and X_2)

i. $o_{\sim_2}(X_1) = o_{\sim_2}(X_2) = o_{\sim_1}(X_i)$ (i.e., all subsets of a refined equivalence class inherit the observation of the parent) and
ii. for any $X_j \in X/_{\sim_2}$ such that $X_j \in \delta_{\sim_2}(X_1)$ or $X_j \in \delta_{\sim_2}(X_2)$, there exist an $X_k \in X/_{\sim_1}$ such that $con(X_j) \subseteq con(X_k)$ and $X_k \in \delta_{\sim_1}(X_i)$ (i.e., all transitions are preserved).

Since words can only be removed from the language of the quotient through refinement

$$\mathscr{L}_T \subseteq \mathscr{L}_{T/_{\sim_2}} \subseteq \mathscr{L}_{T/_{\sim_1}}. \tag{4.9}$$

and, from Eq. (4.9),

$$con(X^{\phi}_{T/_{\sim_1}}) \subseteq con(X^{\phi}_{T/_{\sim_2}}) \subseteq X^{\phi}_{T}. \tag{4.10}$$

Therefore, the largest satisfying region $X^{\phi}_{T/_{\sim_r}}$ of the quotient $T/_{\sim_r}$ obtained after more iterations of Algorithm 1 provides a better approximation of the largest satisfying region X^{ϕ}_{T} of T than the one obtained with less iterations. The same result holds for the largest violating region $X^{\neg\phi}_{T/_{\sim_r}}$.

The analysis strategy described in this section is summarized as Algorithm 4. While this illustrates the combination of quotient refinement and model checking, performing analysis after every single refinement step is prohibitive. To develop a solution more amenable to practical implementations, in the following section we present a number of improvements and optimizations to this method.

saha.indra@gmail.com

Algorithm 4 $[X^{\phi}, X^{\neg\phi}] =$ITERATIVEANALYSIS$(T, \phi)$: Given formula ϕ, compute the (under-approximations) $X^{\phi} \subseteq X_T^{\phi}$ and $X^{\neg\phi} \subseteq X_T^{\neg\phi}$.

Initialize $\sim_r := \sim$
while there exist equivalence classes $X_i, X_j \in X/_{\sim_r}$ such that
$\emptyset \subset con(X_i) \cap Pre(con(X_j)) \subset con(X_i)$ **do**
  Construct equivalence class X_1 such that $con(X_1) := con(X_i) \cap Pre(con(X_j))$
  Construct equivalence class X_2 such that $con(X_2) := con(X_i) \setminus Pre(con(X_j))$
  $X/_{\sim_r} := X/_{\sim_r} \setminus \{X_i\} \bigcup \{X_1, X_2\}$
  Construct quotient $T/_{\sim_r}$
  $X^{\phi}_{T/_{\sim_r}} :=$ANALYZE$(T/_{\sim_r}, X/_{\sim_r}, \phi)$
  $X^{\neg\phi}_{T/_{\sim_r}} :=$ANALYZE$(T/_{\sim_r}, X/_{\sim_r}, \neg\phi)$
  $X^{\phi} := con(X^{\phi}_{T/_{\sim_r}})$
  $X^{\neg\phi} := con(X^{\neg\phi}_{T/_{\sim_r}})$
end while

4.4 Conservative Quotient Refinement

If the initial steps of the analysis procedure outlined in the previous section fail to produce an adequate approximation of the solution to Problem 4.1, computation might become challenging due to the possible explosion in the number of states of $T/_{\sim_r}$ as refinement progresses. Therefore, minimizing the amount of refinement performed at each step is critical to the feasibility of the method. One possible strategy for controlling the number of states produced during the refinement involves refining and analyzing the quotient only at states where this can improve the solution (i.e., increase $X^{\phi}_{T/_{\sim_r}}$ and $X^{\neg\phi}_{T/_{\sim_r}}$). Targeting our computation in such a way requires identifying states where refinement might be beneficial and developing a refinement strategy that operates locally on particular states. Through the rest of this section we discuss both issues and use the results to develop an improved analysis procedure.

To understand where refinement should be targeted in the quotient $T/_{\sim_r}$, we first provide some additional intuition about why analysis of $T/_{\sim_r}$ leads only to a conservative solution to Problem 4.1 and how state refinement can help. Given a state $x \in X$, system T either satisfies formula ϕ from x (i.e., $T(x) \models \phi$), its negation $\neg\phi$ (i.e., $T(x) \models \neg\phi$), or x is uncertain in T with respect to the satisfaction of ϕ (i.e., $T(x) \not\models \phi$ and $T(x) \not\models \neg\phi$). Given two states $x_1, x_2 \in X$ such that $T(x_1) \models \phi$ and $T(x_2) \models \neg\phi$, if the states are equivalent (i.e., $x_1 \sim_r x_2$) and belong to equivalence class $X_i \in X/_{\sim_r}$ (i.e., $x_1, x_2 \in con(X_i)$), state X_i is uncertain in $T/_{\sim_r}$ with respect to the satisfaction of ϕ. However, refining X_i might separate states x_1 and x_2 and allow the resulting subsets of X_i to be characterized as satisfying or violating.

saha.indra@gmail.com

expand $X^{\phi}_{T/\sim_r}$ significantly since X_i is satisfying (i.e., $X_i \in X^{\phi}_{T/\sim_r}$), violating (i.e., $X_i \in X^{\neg\phi}_{T/\sim_r}$), uncertain (i.e., $X_i \in X^{\phi?}_{T/\sim_r}$), or small.

Even when no additional termination conditions are imposed and Algorithm 7 returns an exact solution to Problem 4.1, the quotient $T/\sim_r$ is, in general, not a bisimulation quotient (e.g., see Example 4.9). This motivates us to explore the conditions required to guarantee that an exact solution to Problem 4.1 is obtained in the following section and to use them later to develop a more efficient analysis strategy.

Example 4.9 Using the procedure from Algorithm 7, we analyze transition system T introduced originally in Example 1.10 (Fig. 1.11a).

In Example 1.10, we described the construction of the simulation quotient $T/\sim$ of T (Fig. 1.11b). However, in Example 4.4, we showed that simply analyzing $T/\sim$ to compute its largest satisfying and strictly violating regions $X^{\phi}_{T/\sim}$ and $X^{\neg\phi}_{T/\sim}$ (Fig. 4.4a) leads only to a conservative solution to Problem 4.1 and the computation of a subset of the largest satisfying and strictly violating regions X^{ϕ}_{T} and $X^{\neg\phi}_{T}$ of T (Fig. 4.4b).

In Example 4.5, we considered an approach where the bisimulation algorithm (Algorithm 1) was used to obtain a quotient $T/\approx$, bisimilar with T Fig. 4.5b with refined equivalence classes shown in Fig. 4.5a. By computing the largest satisfying and strictly violating regions of this bisimulation quotient, we were able to obtain the corresponding regions for T Fig. 4.5c (i.e., the solution to Problem 4.1 was exact). Furthermore, this allowed us to compute the uncertain region $X^{\phi?}_{T/\approx}$ of $T/\approx$ and use it to obtain the uncertain region $X^{\phi?}_{T}$ of T.

By applying the procedure implemented in Algorithm 7, we can also obtain an exact solution to Problem 4.1, without constructing a bisimulation quotient. Indeed, after the initial quotient $T/\sim$ is constructed and analyzed as in Example 4.4, the algorithm targets refinement to state $X_2 \in X/\sim$ only (see Fig. 4.4a). Applying the refinement procedure to that state as described in Example 4.8 leads to the construction of quotient $T/\sim_r$ (Fig. 4.5b). While $T/\sim_r$ is not bisimilar with T, computing regions $X^{\phi}_{T/\sim_r}$ and $X^{\neg\phi}_{T/\sim_r}$ (Fig. 4.8a with equivalence classes shown in Fig. 4.8b) provides an exact solution to Problem 4.1.

4.5 Formula-Equivalence

So far in this chapter, we described methods for analyzing transition systems based on the construction of finite quotients, which led to the analysis procedure summarized as Algorithm 7. While our discussion focused primarily on the analysis of potentially large, finite transition systems, the construction of finite quotients also makes these approaches suitable for infinite systems—such applications will be considered

saha.indra@gmail.com

in detail in subsequent chapters. Algorithm 7 combined state refinement inspired by the bisimulation algorithm (Algorithm 1) with model-checking-based analysis (Algorithm 3) in an iterative procedure. With a limited number of iterations, the algorithm was conservative and led to the computation of under-approximations of the largest satisfying and strictly violating regions (Definitions 4.1 and 4.3), which could be improved by performing additional iterations. When Algorithm 7 terminates, there could be states left that have not been assigned as either satisfying the specification or its negation and might be "too small" to undergo additional refinement. Even without a limit on the number of iteration or the size of regions that could undergo refinement the termination of the algorithm with an exact solution could not be guaranteed in general (e.g. when applied to an infinite system).

While the solution to Problem 4.1 returned by Algorithm 7 was, in general, only an approximation, we also encountered cases where an exact solution was obtained. In Sect. 1.3, we showed that the construction of a bisimulation quotient during refinement guarantees our solution was exact but Example 4.9 suggested that this condition was unnecessarily strong (i.e., bisimulation is a sufficient but not a necessary condition). The conditions from Propositions 4.1 and 4.2 could also guarantee that an exact solution was obtained but were too conservative unless only deterministic systems were considered. In the following, we explore more general conditions guaranteeing that an exact solution to Problem 4.1 is obtained (Eq. (4.8) holds). Specifically, we seek necessary and sufficient conditions weaker than bisimulation, guaranteeing that two transition systems (or a transition system and its quotient) are equivalent with respect to the satisfaction of a given LTL formula ϕ.

In Propositions 4.1 and 4.2 we described conditions guaranteeing that the largest satisfying and strictly violating regions of T were obtained by analyzing the quotient $T/_{\sim_r}$ (computed as part of Algorithm 7), even when the two systems are not bisimilar. While these conditions could be used to test whether such a solution to Problem 4.1 was exact, the approach could be applied only in specific cases (i.e., when T was deterministic). In the following, we generalize these results and formulate the conditions required to guarantee that two systems are equivalent with respect to the satisfaction of a specific LTL formula.

Definition 4.5 (*Formula equivalence*) Given a finite transition system T and an LTL formula ϕ, an observational equivalence relation $\sim$ is a ϕ-equivalence of T if and only if, for all states $x_1, x_2 \in X$ such that $x_1 \sim x_2$, we have

$$T(x_1) \vDash \phi \Leftrightarrow T(x_2) \vDash \phi$$

We denote a ϕ-equivalence relation as $\sim_\phi$ and refer to the quotient $T/_{\sim_\phi}$ as ϕ-equivalent quotient.

From Eq. (4.7) it follows that a bisimulation relation $\sim$ is a ϕ-equivalence for all LTL formulas ϕ. Bisimulation is a sufficient condition guaranteeing that Eq. (4.8) holds but since we are interested in the analysis of T for a specific LTL formula ϕ it can be too restrictive.

saha.indra@gmail.com

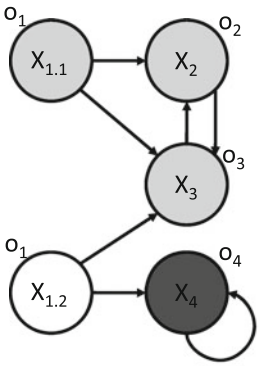
Proposition 4.3 *Given a transition system T and an LTL formula ϕ, the equation $X_T^\phi = con(X_{T/\sim}^\phi)$ is satisfied and an exact solution to Problem 4.1 is obtained by analyzing the quotient $T/\sim$ if and only if $\sim$ is a ϕ-equivalence of T.*

Proof Assume that $\sim$ is a ϕ-equivalence. From Definition 4.5 it follows that $\forall x \in X$ such that $T(x) \vDash \phi, x \in con(X_i), X_i \in T/\sim$ we have $T/\sim(X_i) \vDash \phi$. Then, $\forall x \in X, x \in X_T^\phi \Leftrightarrow x \in con(X_{T/\sim}^\phi)$ and therefore $X_T^\phi = con(X_{T/\sim}^\phi)$. Assume that $\sim$ is not a ϕ-equivalence. Then, $\exists x_1, x_2 \in X$ such that $x_1 \sim x_2, T(x_1) \vDash \phi$ and $T(x_2) \nvDash \phi$. Considering the equivalence class $X_i \in X/\sim$ such that $x_1, x_2 \in con(X_i)$ we have $T/\sim(X_i) \nvDash \phi$. Then $x_1 \in X_T^\phi$ but $x_1 \notin con(X_{T/\sim}^\phi)$ and therefore $X_T^\phi \neq con(X_{T/\sim}^\phi)$. ■

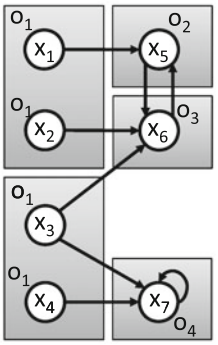
Proposition 4.3 shows that ϕ-equivalence is a necessary and sufficient condition guaranteeing that the largest satisfying region of T can be computed through the computation of the largest satisfying region of the quotient $T/\sim_\phi$ (i.e., $X_T^\phi = con(X_{T/\sim_\phi}^\phi)$). The condition $X_T^\phi = con(X_{T/\sim}^\phi)$ is similar to the one in Eq. (4.8) (left-hand side) but is formulated for a quotient $T/\sim$, rather than the bisimulation quotient $T/\approx$. Then, obtaining an exact solution to Problem 4.1 reduces to the computation of $con(X_{T/\sim_\phi}^\phi)$, where $T/\sim_\phi$ is a finite, ϕ-equivalent quotient for T (e.g., see Fig. 4.9a, b). As previously discussed, in order to obtain more information about T, the strictly violating region of T might also be computed as $X_T^{\neg\phi} = con(X_{T/\sim}^{\neg\phi})$, which reduces to the construction and analysis of the $\neg\phi$-equivalent quotient for T. More generally, to guarantee the exact computation of both the largest satisfying and strictly violating regions of T (Eq. (4.8)), a quotient that is both ϕ-equivalent and $\neg\phi$-equivalent must be computed. We denote such a quotient as $T/\approx_\phi$ (e.g., see Fig. 4.9c, d). Constructing $T/\approx_\phi$ also allows the computation of the uncertain region $X_T^{\phi?} = con(X_{T/\approx_\phi}^{\phi?})$ of T through the computation of the uncertain region $X_{T/\approx_\phi}^{\phi?} = X/\approx_\phi \setminus (X_{T/\approx}^{\neg\phi} \cup X_{T/\approx}^{\phi})$ of $T/\approx_\phi$.

Example 4.10 We are interested in analyzing the transition system T shown in Fig. 4.9b with specification $\phi = \Box\Diamond o_3$. The quotient $T/\sim_\phi$ (shown in Fig. 4.9a with equivalence classes highlighted in Fig. 4.9b) of T is not a bisimulation quotient (the characterization from Theorem 1.1 is violated at state $X_{1.1}$). Even so, the quotient is ϕ-equivalent and therefore the largest satisfying region $X_T^\phi = \{x_1, x_2, x_5, x_6\}$ can be computed as $X_T^\phi = con(X_{T/\sim_\phi}^\phi)$, where $X_{T/\sim_\phi}^\phi = \{X_{1.1}, X_2, X_3\}$. However, region $X_{T/\sim_\phi}^{\neg\phi} = \{X_4\}$ provides only an under-approximation $con(X_{T/\sim}^{\neg\phi}) = \{x_7\}$ of the strictly violating $X_T^{\neg\phi} = \{x_4, x_7\}$ of T.

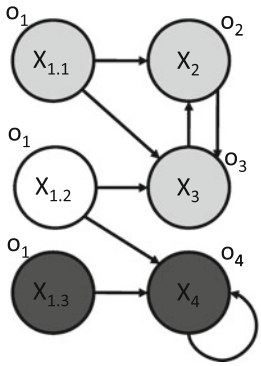
saha.indra@gmail.com



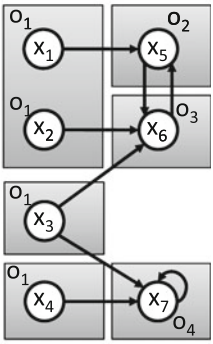
(a) Quotient $T/_{\sim_\phi}$.



(b) Equivalence classes of $T/_{\sim_\phi}$ from (a).



(c) Quotient $T/_{\approx_\phi}$.



(d) Equivalence classes of $T/_{\approx_\phi}$ from (c).

Fig. 4.9 Given LTL formula $\phi = \Box\Diamond o_3$, analysis of the ϕ-equivalent quotient $T/_{\sim_\phi}$ (**a**) of a transition system T allows the computation of the largest satisfying region (*light gray*) but only a subset of the strictly violating region (*dark gray*) of T. Analysis of the quotient $T/_{\approx_\phi}$ (**c**) that is both ϕ-equivalent and $\neg\phi$-equivalent allows the computation of both the largest satisfying (*light gray*) and strictly violating (*dark gray*) regions of T. For additional details, see Example 4.10

While quotient $T/_{\approx_\phi}$ (shown in Fig. 4.9c with equivalence classes highlighted in Fig. 4.9d) is still not a bisimulation quotient, it is both a ϕ-equivalent and an $\neg\phi$-equivalent of T. This allows the exact computation of the strictly violating region $X_T^{\neg\phi} = con(X_{T/_{\approx_\phi}}^{\neg\phi}) = \{x_4, x_7\}$ through the computation of region $X_{T/_{\approx_\phi}}^{\neg\phi} = \{X_{1.3}, X_4\}$. Constructing quotient $T/_{\approx_\phi}$ also leads to the computation of the uncertain region $X_T^{\phi?} = con(X_{T/_{\approx_\phi}}^{\phi?}) = \{x_3\}$ of T, where $X_{T/_{\approx_\phi}}^{\phi?} = X/_{\approx_\phi} \setminus (X_{T/_{\approx_\phi}}^{\neg\phi} \cup X_{T/_{\approx_\phi}}^{\phi}) = \{X_{1.2}\}$.

saha.indra@gmail.com

Motivated by the strategy for solving Problem 4.1 outlined above, in the following we develop a procedure for the computation of formula equivalent quotients. While the approach we described in Sect. 4.3 could, in principle, also lead to the construction of formula equivalent quotients through iterative analysis and refinement (as was the case for the system from Example 4.9), a large number of steps was required to achieve this. Furthermore, the optimizations introduced to control the number of states produced through refinement as part of this method resulted in an additional source of conservatism for general formulas. The method we develop next aims directly at the construction of formula equivalent quotients and is more efficient, while in addition, it overcomes some of the limitations of the previous procedure.

In the following, we develop an algorithm for the computation of ϕ-equivalent quotients of finite transitions systems, leveraging ideas from the bisimulation algorithm (Algorithm 1) and automata-based model checking. To simplify the presentation of our method, we assume that the LTL formula ϕ is translated into a deterministic Büchi automaton B_ϕ over the set of observations O—we discuss how this assumption is relaxed in Sect. 4.6.

Since the computation of ϕ-equivalent quotients is guided by formula ϕ, it is most natural to perform the computation in the product automaton $P = T/_\sim \otimes B_\phi$ (Definition 3.2), where both the structure of the system ($T/_\sim$) and the specification (B_ϕ) is captured. The computational techniques we employ in the following will be described in detail in Chap. 5 in the context of control transition systems. In this chapter we only give a brief overview, sufficient for the presentation of our analysis procedure.

Let $S_\top \subseteq S_P$ be the set of states of P, from which all runs are accepted (i.e., they visit a final state from the set F_P infinitely often). The set $S_\top$ can be computed efficiently using a method inspired by automata-theoretic model checking and games (to be discussed in detail in Chap. 5). We identify a subset $F_\top \subseteq F_P$ of accepting states of P from which infinitely many revisits to the set F_P are guaranteed. $S_\top$ is then a set of states from which a visit to $F_\top$ is guaranteed. The largest satisfying set $X^\phi_{T/_\sim}$ of $T/_\sim$ can be computed as the projection $\alpha(S_\top \cap S_{P0}) \subseteq X/_\sim$. Similarly, we can also identify a set of states $S_\bot \subseteq S_P$ of P from which no runs are accepting. The projection $\alpha(S_\bot \cap S_{P0}) \subseteq X/_\sim$ corresponds to $X^{\neg\phi}_{T/_\sim}$ (i.e., the largest set of states of $T/_\sim$ from which no runs satisfy ϕ).

The analysis approach we described in Sect. 4.3 required the construction of both the product automaton $P_\phi = T/_\sim \otimes B_\phi$ with the formula and $P_{\neg\phi} = T/_\sim \otimes B_{\neg\phi}$ with its negation in order to compute the largest satisfying and strictly violating regions $X^\phi_{T/_\sim}$ and $X^{\neg\phi}_{T/_\sim}$. Using the approach from this section, we avoid the construction of $P_{\neg\phi}$ and only perform computation on P_ϕ to find these sets.

Let $S_? = S_P \setminus (S_\top \cup S_\bot)$ be the subset of states, from which some but not all runs are accepting in P. The projection $X^{\phi?}_{T/_\sim} = \alpha(S_? \cap S_{P0}) \subseteq X/_\sim$ corresponds to the set of uncertain states of $T/_\sim$, where both runs satisfying ϕ and $\neg\phi$ originate. The ϕ-equivalence property (Definition 4.5) can only be violated at states from $X^{\phi?}_{T/_\sim}$. As previously discussed, when T is deterministic, each state $x \in X$ satisfies either ϕ or $\neg\phi$. Then, the existence of states in $X^{\phi?}_{T/_\sim}$ is only due to the construction of the

saha.indra@gmail.com

abstraction $T/_\sim$ (e.g., for two equivalent states $x_1, x_2 \in con(X_i)$ for some $X_i \in X/_\sim$, if $x_1 \vDash \phi$ and $x_2 \vDash \neg\phi$ then $X_i \in X^{\phi?}_{T/_\sim}$).

As in Sect. 4.3, we can attempt to separate satisfying and violating runs from a state in the set $S_?$ through refinement in order to separate the subsets of that state between sets $S_\top$ and $S_\perp$. Since the structure of P is completely determined by B_ϕ and $T/_\sim$ and B_ϕ is fixed and determined by the formula ϕ, states in P can only be refined through refinement of $T/_\sim$. We refine a state $(X_i, s) \in S_?$ by applying the procedure REFINE $(T/_\sim, \alpha(X_i, s))$ (Algorithm 5) Sect. 4.3.

To prevent unnecessary computation, changes made through refinement in the quotient $T/_\sim$ are projected to the product P by applying a function $P' =$ UPDATE $(P, T/_{\sim_r}, (X_i, s))$, rather than by recomputing it as $P' = T/_{\sim_r} \otimes B_\phi$. This further reduces the number of states in P' after refinement since, when a state X_i is refined in $T/_{\sim_r}$, not every state (X_i, s) is necessarily refined in P'. Due to the one-to-many correspondence between the states of the refined product P' and the refined quotient $T/_{\sim_r}$, after refinement the updated product automaton P' might include significantly fewer states than the recomputed product automaton $T/_{\sim_r} \otimes B_\phi$.

When T is nondeterministic, it is possible that for a state $X_i \in X/_\sim$, there exist both trajectories satisfying ϕ and $\neg\phi$ originating at all states $x \in con(X_i)$ of T. Then, state X_i is inherently uncertain in $T/_\sim$ and abstraction refinement can never separate satisfying and violating runs from X_i. We partition the set of states $S_?$ into subsets $S_\prec$ and $S_\div$, where states of $T/_\sim$ from the projection $\alpha(S_\prec \cap S_{P0})$ are inherently uncertain, while refinement should be targeted to states from $\alpha(S_\div \cap S_{P0})$. In the following, we provide a computational characterization of the states from $S_\prec$ (Proposition 4.4).

Proposition 4.4 *Given a state $(X_i, s) \in S_P$, we have $(X_i, s) \in S_\prec$ if and only if*

i. $(X_i, s) \in S_?$ *(i.e., both satisfying and violating runs originate there),*
ii. $\forall (X_j, s') \in \delta_P((X_i, s)), (X_j, s') \in S_\top$, $(X_j, s') \in S_\perp$ *or* $(X_j, s') \in S_\prec$ *(i.e., all successors states of (X_i, s) have been characterized in P),*
iii. $T/_\sim =$REFINE$(T/_\sim, X_i)$ *(i.e., state (X_i, s) cannot be refined further).*

Proof All successors of (X_i, s) are characterized in P and, therefore, none of the successors of $X_i = \alpha((X_i, s))$ will be considered for further refinement in $T/_\sim$. Since applying REFINE$(T/_\sim, X_i)$ does not affect X_i, then for all states X_j such that $X_j \in \delta_\sim(X_i)$ we have $\forall x \in con(X_i), \exists x' \in con(X_j)$ such that $x' \in \delta(x)$. Therefore, $(X_i, s) \in S_\prec$ and X_i is inherently uncertain. ■

Finally, in the following we derive the conditions guaranteeing that a ϕ-equivalent quotient has been computed based on the computation of sets $S_\top$, $S_\perp$, $S_\div$ and $S_\prec$ in P.

Proposition 4.5 *The equivalence relation $\sim$ is a ϕ-equivalence of T if and only if $S_\div \cap S_{P0} = \emptyset$.*

Proof For necessity, assume $S_\div \cap S_{P0} \neq \emptyset$ where (X_i, s) is a state of P such that $(X_i, s) \in S_\div$. Then, $X_i = \alpha((X_i, s))$ is a state of $T/_\sim$ such that $\exists x_1, x_2 \in con(X_i)$, where $T(x_1) \vDash \phi$ and $T(x_2) \vDash \neg\phi$ and, therefore, $\sim$ is not a ϕ-equivalence. For sufficiency, assume $S_\div \cap S_{P0} = \emptyset$. Then, $\forall X \in X/_\sim$ we have

saha.indra@gmail.com

i. $\forall x \in con(X_i), T(x) \models \phi$,
ii. $\forall x \in con(X_i), T(x) \models \neg\phi$ or
iii. $\forall x \in con(X_i), T(x) \not\models \phi$ and $T(x) \not\models \neg\phi$

and therefore $\sim$ is a ϕ-equivalence. ■

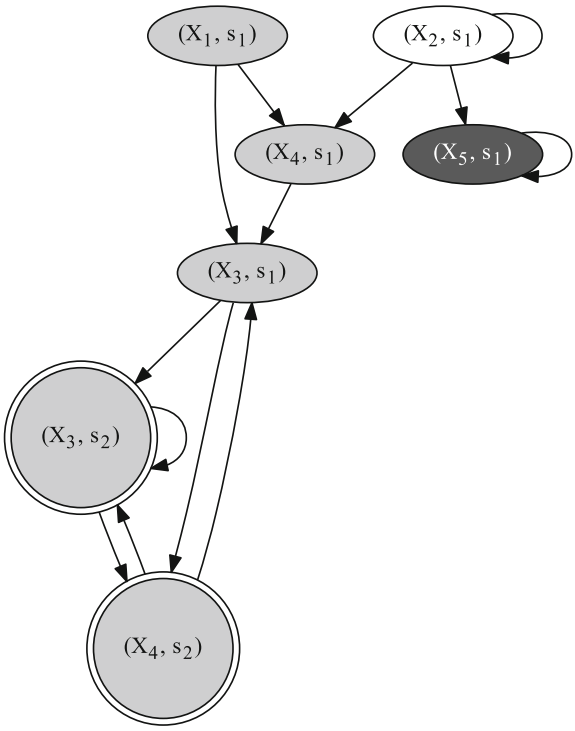
In general, the set $S_{\div}$ is nonempty but can be made empty if accepting and non-accepting runs from each state $(X_i, s) \in S_{\div}$ are separated through refinement as described above. Following from Proposition 4.5 and the discussion presented in Sect. 4.5, this provides a solution to Problem 4.1.

There are several additional optimizations that can be introduced in this analysis procedure. The product automaton P can be simplified by respectively replacing each set $S_\top, S_\bot$ and $S_\prec$ by a single dummy state s_1, s_2 or s_3, such that $\delta_P(s_1) = \{s_1\}, \delta_P(s_2) = \{s_2\}, \delta_P(s_3) = \{s_1, s_2\}$ and $s_1 \in F_P$. The motivation for this simplification comes from the fact that guaranteeing a visit to a state from (the previously identified) set $S_\top$ can be enforced from a state of P is equivalent to guaranteeing that the state belong to $S_\top$ and the same argument holds for set $S_\bot$. When simplifying P, for a state $s \in S_P \setminus (S_\top \cup S_\bot \cup S_\prec)$ a transition to a dummy state is included if a transition to a state from the corresponding set was present (e.g., if there existed a state $s' \in S_\top$ such that $s' \in \delta_P(s)$ then $s_1 \in \delta_P(s)$). This simplification is performed by the function $P' = \textsc{Simplify}(P)$, which reduces the number of states of P and leads to faster computation.

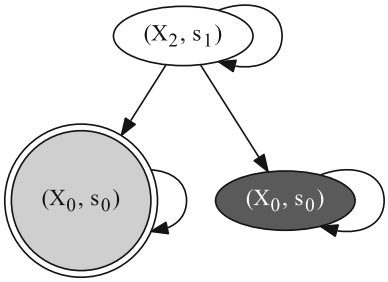
Example 4.11 We apply our formula-guided analysis procedure (Algorithm 8) in order to study transition system T, defined originally in Example 1.11 (Fig. 1.11a), against specification $\phi = \Box\Diamond o_3$.

i. The specification, given as the LTL formula ϕ, is translated into a deterministic Büchi automaton B_ϕ defined by $S = \{s_1, s_2\}$, $S_0 = \{s_1\}$, $O = \{o_1, \ldots, o_5\}$, $F = \{s_2\}$, $\delta(s_1, o_1) = \delta(s_1, o_2) = \delta(s_1, o_4) = \delta(s_1, o_5) = \{s_1\}, \delta(s_1, o_3) = \{s_2\}, \delta(s_2, o_3) = \{s_2\}, \delta(s_2, o_1) = \delta(s_2, o_2) = \delta(s_2, o_4) = \delta(s_2, o_5) = \{s_1\}$. Note that B_ϕ has a structure that resembles the automaton shown in Fig. 2.3d but with different transition labels and is indeed deterministic.
ii. The initial quotient $T/_\sim$ of T under the observational equivalence relation $\sim$ is constructed (the quotient $T/_\sim$ was already shown in Fig. 1.11b). Then, the product automaton $P = T/_\sim \otimes B_\phi$ shown in Fig. 4.10a is constructed (note that unreachable states in P are removed during the construction). The set of states of P, from which all runs are guaranteed to visit a final state of P infinitely often (all runs are accepted in P) are identified as $S_\top = \{(X_1, s_1), (X_4, s_1), (X_3, s_1), (X_3, s_2), (X_4, s_2)\}$. Similarly, the set of states from which no run visits a final state infinitely often is identified as $S_\bot = \{(X_5, s_1)\}$. Using this information, we can compute the largest

saha.indra@gmail.com



(a) Product automaton $P = T/_{\sim} \otimes B_{\phi}$.

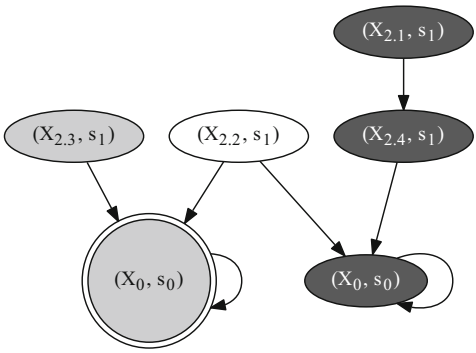


(b) Simplified product $P_1 = \textsc{Simplify}(P)$.

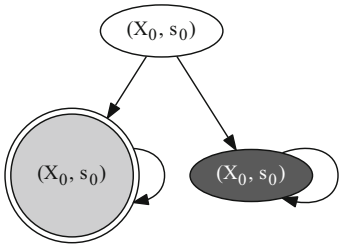
Fig. 4.10 Computing the set of states $S_{\top}$ and $S_{\perp}$ of the product automaton $P = T/_{\sim} \otimes B_{\phi}$, shown respectively in *light gray* and *dark gray* in (**a**), allows both the simplification of P into $P' = \textsc{Simplify}(P)$ (**b**) and the computation of the largest satisfying and strictly violating regions of $T/_{\sim}$, together with a set of states $S_{\div}$, where refinement is required. See Example 4.11 (steps i–iii) for details

satisfying and strictly violating regions of $T/_{\sim}$ through the projections $X^{\phi}_{T/_{\sim}} = \alpha(S_{\top} \cap S_{P0}) \subseteq X/_{\sim} = \{X_1, X_3, X_4\}$ and $X^{\neg\phi}_{T/_{\sim}} = \alpha(S_{\perp} \cap S_{P0}) = \{X_5\}$, which agrees with the previously computed sets from Example 4.4

saha.indra@gmail.com



(a) Product $P_2 = \textsc{Update}(P_1, \textsc{Refine}(T/_\sim, X_2), (X_2, s_2))$.



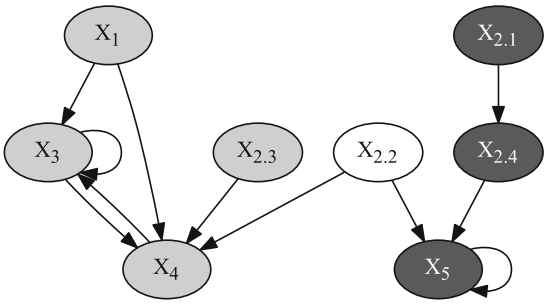
(b) Simplified product $P_3 = \textsc{Simplify}(P_2)$.

Fig. 4.11 The product P_2 is obtained by updating P_1 to capture the changes made to the quotient $T/_\sim$ through refinement. The sets of states $S_\top$ and $S_\bot$ of P_2 (shown respectively in *light gray* and *dark gray* in (**a**)) are computed and allows expanding the largest satisfying and strictly violating regions of T and the additional simplification of P_2 into $P_3 = \textsc{Simplify}(P_2)$ (**b**). See Example 4.11 (steps iv, v) for details

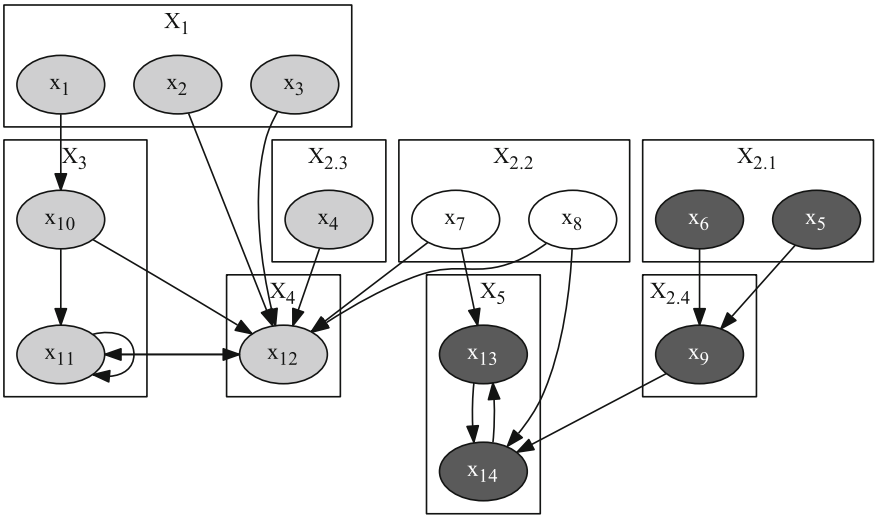
(Fig. 4.4a). In this case, no states are inherently uncertain in $T/_\sim$—in the following steps we will show that state (X_2, s_1) can be refined and, therefore, it fails the characterization from Proposition 4.4. Then, the set $S_\prec$ is empty and refinement is targeted to the set $S_\div = \{(X_2, s_1)\}$.

iii. While the sets of states $S_\top$ and $S_\bot$ computed at the previous step provide only an under-approximation of the largest satisfying and strictly violating sets of T (this approximation was already shown in Fig. 4.4b), they allow us to simplify the product automaton P into $P_1 = \textsc{Simplify}(P)$ shown in Fig. 4.10b. All states from the set $S_\top$ are replaced by a single dummy state (X_0, s_0) and the same simplification is performed for the set $S_\bot$ (note that one of the dummy states is accepting and the other one is not). Both dummy states have self loops and all transitions from the remaining states of P to a state from $S_\top$ or $S_\bot$ are replaced with a transition to the corresponding dummy state, thereby preserving the structure of P. Since, in this case, the set $S_\prec$ is empty, the third "inherently uncertain" dummy state is omitted.

saha.indra@gmail.com



(a) Formula equivalent quotient $T/_{\approx_\phi}$ = REFINE$(T/_\sim, X_2)$.



(b) Equivalence classes $X/_{\approx_\phi}$ of $T/_{\approx_\phi}$.

Fig. 4.12 The formula equivalent quotient $T/_{\approx_\phi}$ (**a**), constructed after refinement of $T/_\sim$, can be used to equivalently identify the largest satisfying, strictly violating and uncertain regions of T (**b**). See Example 4.11 (steps vi) for details

iv. The set of states from the projection $\alpha(S_\div) = \{X_2\}$ must be refined in $T/_\sim$. The refined quotient is constructed as $T/_{\sim_r}$ = REFINE$(T/_\sim, X_2)$ (see Fig. 4.12a), where state X_2 is partitioned into the set of subsets $\{X_{2.1}, X_{2.2}, X_{2.3}, X_{2.4}\}$ (i.e., $con(X_2) = con(X_{2.1}) \cup con(X_{2.2}) \cup con(X_{2.3}) \cup con(X_{2.4})$ in Fig. 4.12b). The refinement of $T/_\sim$ is projected to the product automaton

saha.indra@gmail.com

by applying the procedure $P_2 = \text{UPDATE}(P_1, T/_{\sim_r}, (X_2, s_2))$, rather than recomputing it as $P_2 = T/_{\sim_r} \otimes B_\phi$, which leads to a simpler structure.

v. Once the updated product P_2 is computed, the sets $S_\top$ and $S_\bot$ can be computed again (which now include the dummy states). State $(X_{2.2}, s_1)$ satisfies the characterization from Proposition 4.4. Therefore, the projection $\alpha((X_{2.2}, s_1)) = X_{2.2}$ corresponds to an inherently uncertain region of T (i.e., there exists both runs that satisfy ϕ and that violate it (and satisfy $\neg\phi$) that originate at each state $x \in con(X_{2.2})$). Simplifying the product as $P_3 = \text{SIMPLIFY}(P_2)$ results in an automaton containing dummy states only—now, an inherently uncertain dummy state with transitions to both the satisfying and rejecting dummy states is included.

vi. All states of P_2 have been partitioned between the sets $S_\top$, $S_\bot$ and $S_\prec$. Further refinement of states of $T/_{\sim_r}$ is not going to improve the solution and, in fact, a ϕ-equivalent quotient has been obtained. Furthermore, the computation from Algorithm 8 guarantees that a quotient that is both ϕ-equivalent and $\neg\phi$-equivalent has been obtained and, therefore, $T/_{\sim_r} = T/_{\approx_\phi}$ in Fig. 4.11a. This allows us to guarantee that the largest satisfying, strictly violating and inherently uncertain regions $X^{\phi}_{T/_{\approx_\phi}}$, $X^{\neg\phi}_{T/_{\approx_\phi}}$ and $X^{\phi?}_{T/_{\approx_\phi}}$ of $T/_{\approx_\phi}$ (Fig. 4.12a) can be used to compute the largest satisfying, strictly violating and inherently uncertain regions $X^{\phi}_T = con(X^{\phi}_{\hat{T}/_\sim})$, $X^{\neg\phi}_T = con(X^{\neg\phi}_{\hat{T}/_\sim})$ and $X^{\phi?}_T = con(X^{\phi?}_{\hat{T}/_\sim})$ of T (Fig. 4.12a), which provides an exact solution to Problem 4.1. Note that a coarser quotient that is only ϕ-equivalent but not $\neg\phi$-equivalent can be constructed by combining equivalence classes $X_{2.1}$ and $X_{2.2}$ into a single equivalence class—the corresponding quotient is still observation preserving, since $X_{2.1}$ and $X_{2.2}$ share the same observation o_2.

Optimizations based on the decomposition of the product P into strongly connected components (SCCs) and the subsequent construction of an SCC quotient graph, similar to the ones described for model checking in Chap. 3 are also possible. First, we make the following observations. Given states $s, s' \in S_P$, such that s' is reachable from s but s is not reachable from s', if $s' \in S_?$ then $s \in S_?$ but characterizing s as $S_?$, $S_\top$, or $S_\bot$ does not have any influence on characterization of s'. The product P is viewed as a directed graph $G_P = (S_P, E)$, where $(s, s') \in E$ if and only if $s' \in \delta_P(s)$. This graph is partitioned into maximal strongly connected components (SSCs), inducing the directed acyclic quotient graph $\mathbb{G}_P = (\mathbb{C}, \mathbb{E})$. The following properties are then guaranteed to hold:

i. for each SCC C we have $C \subseteq S_\top$, $C \subseteq S_\bot$, or $C \subseteq S_?$ (i.e., all states $s \in C$ are equivalent with respect to such characterization) and
ii. for all SCCs C, C', such that C' is reachable from C in $\mathbb{G}_P$ it holds that $C' \subseteq S_? \Rightarrow C \subseteq S_?$.

saha.indra@gmail.com

Converting the product automaton to a SCC quotient graph and processing the SCCs in bottom up manner allows for a more efficient characterization of states within each SCC and prevents unnecessary refinement.

Algorithm 8 $[X_T^{\phi},\ X_T^{\neg\phi},\ X_T^{\phi?}] = \text{FGANALYSIS}(T, \phi)$: Given an LTL formula ϕ and a large or infinite transition system T, compute of the largest satisfying and strictly violating regions of T.

Translate ϕ to deterministic Büchi automaton B_ϕ
Construct $T/_\sim$
Construct $P = T/_\sim \otimes B_\phi$
Initialize $T/_{\sim_r} := T/_\sim$
Compute $S_\top$ and $S_\bot$ in P
$S_? := S_P \setminus (S_\top \cup S_\bot)$
Compute $S_\prec \subseteq S_?$, $S_\div := S_? \setminus S_\prec$
repeat
  for all $(X_i, s) \in S_\div$ **do**
    if X not yet refined in $T/_{\sim_r}$ **then**
      $T/_{\sim_r} :=$REFINE$(T/_{\sim_r}, X_i)$
    end if
    $P :=$UPDATE $(P, T/_{\sim_r}, (X_i, s))$
  end for
  $P :=$SIMPLIFY(P)
until $S_\div = \emptyset$
$X^{\phi}_{T/\approx_\phi} = \alpha(S_\top \cap S_{P0})$, $X^{\neg\phi}_{T/\approx_\phi} = \alpha(S_\bot \cap S_{P0})$, $X^{\phi?}_{T/\approx_\phi} = \alpha(S_\prec \cap S_{P0})$
return $X_T^{\phi} = con(X^{\phi}_{\hat{T}/_\sim})$, $X_T^{\neg\phi} = con(X^{\neg\phi}_{\hat{T}/_\sim})$, $X_T^{\phi?} = con(X^{\phi?}_{\hat{T}/_\sim})$

The overall method discussed in this section is summarized in Algorithm 8. As before, this procedure cannot be guaranteed to terminate for general transitions system T returning the formula equivalent quotient $T/_\approx$, for which the set $S_\div$ is empty. Instead, termination can again be ensured by either limiting the number of iterations or by only refining states of $\hat{T}/_\sim$ that correspond to "large enough" regions of T. If computation is stopped because an iteration limit is reached or no additional "large enough" states remain, only an under-approximation of the solution to Problem 4.1 is obtained, which can be improved by adjusting these limits.

4.6 Notes

The analysis methods presented in this chapter were based on the construction, iterative refinement and verification of finite abstractions (quotients) of large or infinite systems. The verification strategy we used was based on LTL model checking [45] (discussed in Chap. 3), while our refinement procedures were inspired by bisimulation-based refinement [35, 99]. A related idea of developing iterative

saha.indra@gmail.com

procedures that combine model checking and refinement was used in [42] for verification from formulas in the universal fragment ACTL of CTL.

The abstractions we used could be sufficient, providing only approximate results to analysis problems, in which case they were constructed using simulation relations [45]. On the other hand, abstractions that were equivalent for all LTL specifications were constructed using the notion of bisimulation [131]. We also described a refinement procedure aimed at the constructions of abstractions that were equivalent to large or infinite systems only with respect to given LTL specifications. The related idea of defining CTL formula-specific equivalences coarser than bisimulation has been explored in [13] in the context of finite state systems.

Relying on a temporal logic formula to guide the refinement of an abstraction is also central to verification methods based on counterexample-guided refinement (CEGAR) [44]. The CEGAR loop involves the iterative generation and invalidation of counterexamples through model checking and abstraction refinement. Such a verification procedure might terminate if no additional counterexamples can be generated (the specification is satisfied), a valid counterexample is found (the specification is violated) or the computational resources are exhausted (results are inconclusive). Instead of performing many model checking steps, our method from Sect. 4.5 aimed directly at the construction of formula equivalent quotients.

The methods from this chapter provided more informative analysis results than simple Yes/No answers (as obtained by model-checking) by identifying satisfying and violating regions (i.e., regions of initial conditions from which all trajectories of the system are guaranteed to satisfy or violate the specification). In some cases, our analysis procedure revealed regions of initial conditions for which nothing can be guaranteed (i.e., trajectories of the system originating in such states might satisfy the specification) but the satisfaction of such states could be resolved through additional computation. Labeling sets of states by *true*, *false* or *maybe* with respect to the satisfaction of the LTL specification as described above was also central to our methods and is related to the construction and refinement of 3-valued abstractions [37, 41].

The analysis procedure from Sect. 4.5 was based on the computation of attractor sets of finite control transition systems, described in [104] and inspired by automata theoretic model checking and Büchi games. Here, we applied these methods only to transition systems without inputs but in Chap. 5 we will describe the algorithms in more detail and use them for the construction of control strategies—the original context in which these methods were developed. The approach presented in this chapter was enabled by the assumption that the LTL specification can be translated into a deterministic Büchi automaton, which simplified the presentation significantly. It is known that there exist LTL formulas that can only be translated to nondeterministic Büchi automata (i.e., the expressivity of specifications used as part of our method is currently restricted to only a fragment of LTL). However, any LTL formula can also be translated into a language-equivalent deterministic Rabin automaton [153], a translation that is associated with a higher computational cost but allows an extension of this method to arbitrary LTL formulas. We will describe such an extension based on the use of deterministic Rabin automata in Chap. 5.

saha.indra@gmail.com