Model refinement using bisimulation quotients

Roland Glück, Bernhard Möller, Michel Sintzoff

Angaben zur Veröffentlichung / Publication details:

Glück, Roland, Bernhard Möller, and Michel Sintzoff. 2011. "Model refinement using bisimulation quotients." *Lecture Notes in Computer Science* 6486: 76–91. https://doi.org/10.1007/978-3-642-17796-5_5.

Model Refinement using Bisimulation Quotients

R. Glück[1], B. Möller[1] and M. Sintzoff[2]

[1] Universität Augsburg, Germany
[2] Université catholique de Louvain, Belgium

Abstract. The paper shows how to refine large-scale or even infinite transition systems so as to ensure certain desired properties. First, a given system is reduced into a smallish, finite bisimulation quotient. Second, the reduced system is refined in order to ensure a given property, using any known finite-state method. Third, the refined reduced system is expanded back into an adequate refinement of the system given initially. The proposed method is based on a Galois connection between systems and their quotients. It is applicable to various models and bisimulations and is illustrated with a few qualitative and quantitative properties.

1 Introduction

This paper extends the work in [8]. There a generic method for refining system models was presented informally. Here it is defined constructively and is then used to ensure optimality properties and temporal properties.

Our aim is to refine large-scale system models so as to satisfy a required property. According to the Concise Oxford Dictionary, "to refine something" is "to free it from defects". In our context, refinement is the elimination of unsuitable transitions and states, and is thus a form of control refinement or control synthesis. To realize it, we use a well known approach to problem solving: reduce –or abstract– the problem, solve the reduced problem, and expand the obtained solution. These three steps are implemented as follows. First, the given model is reduced to a finite bisimulation quotient [1]; we use bisimulations because they preserve many properties, including quantitative ones, and they can be computed efficiently. Second, the design problem is solved for the reduced model by some known finite-state technique. Third, the refined reduced model is expanded back into a satisfactory refinement of the large-scale model given initially. The expansion function is a right-inverse of the quotient function that yields largest refined models. It results from a Galois connection.

Consider for instance the control problem of restricting a possibly infinite transition graph G in such a way that every path starting from any node is a path of shortest length leading into a set of given nodes. The reduced model is a smaller finite quotient graph of G. It is refined into an optimal sub-graph using a known finite-state algorithm. The small optimal sub-graph is then expanded back into a full-fledged optimal sub-graph of G. This problem has been treated by dedicated techniques [15]. Here we extend that work to a generic method that allows tackling different problems in a uniform way. The novelty lies in the

particular combination and adaptation of results known from control synthesis and abstraction-based verification to yield an abstraction-based technique for refinement that ensures qualitative as well as quantitative properties and works for a family of infinite-state systems.

In the sequel, first we present generic definitions for classes of usual models and classes of related bisimulations. Second the proposed method is elaborated with the help of a Galois connection. Third a few applications are summarized. At the end we compare with related work and conclude.

2 Model Classes and Bisimulation Classes

We introduce a general pattern for typical model and bisimulation classes. The quotient operation, used to collapse models, is also defined generically.

This preliminary section may be skimmed over in a first reading.

2.1 Prerequisites and Notational Conventions

Our technical development is based on binary and ternary relations; therefore we recall some definitions for them and fix some notation. The symbol $\subseteq$ denotes non-strict inclusion between sets. Instead of bi-implication $\Leftrightarrow$ we write $\equiv$. Definitional equality and equivalence are denoted by $\doteq$ and $\doteqdot$, resp.

- For relation $R \subseteq Q \times Q', x \in Q\,, x' \in Q'$, and $B, C \subseteq Q\,, \mathcal{B} \subseteq 2^Q$,
  $R^\circ \doteq \{(y,x) \mid (x,y) \in R\}$ is the *converse* of R,
  $R(x) \doteq \{x' \mid xRx'\}$, $R(B) \doteq \bigcup\{R(x) \mid x \in B\}$,
  $R(x,x') \doteqdot xRx' \doteqdot (x,x') \in R$,
  $\mathit{Im}(R) \doteq R(Q)$, $\mathit{Dom}(R) \doteq R^\circ(Q')$; R is *total* iff $\mathit{Dom}(R) = Q$.
- Domain restriction is defined as $R{\downarrow}B \doteq R \cap (B \times Q')$. Two properties are

$$(R{\downarrow}B){\downarrow}C = R{\downarrow}(B \cap C)\ , \qquad R{\downarrow}(\bigcup_{B\in\mathcal{B}}) = \bigcup_{B\in\mathcal{B}} (R{\downarrow}B)\ . \tag{1}$$

- The *composition* of two relations $R \subseteq Q \times Q', R' \subseteq Q' \times Q''$ is $R\,;R'$ where $x(R\,;R')x'' \doteqdot \exists x' \in Q' : xRx' \wedge x'R'x''$.
- Given sets S, S', we write $S - S'$ for the set difference of S, S'. Moreover, by S^2 we mean $S \times S$ and χ_S is some predicate that characterizes S.
- For the application of a function f to an argument e we may write fe instead of $f(e)$.
- The set of Booleans is $\mathsf{Bool} \doteq \{\mathsf{false}, \mathsf{true}\}$.

2.2 Model Classes

We present model structures that describe discrete-time dynamical systems. They comprise graphs and auxiliary components. The graph nodes and edges represent system states and transitions, resp. The auxiliary components may

be maps labelling states or transitions. Quotients of models are defined in a canonical way given equivalences between states.

Graphs. A *graph* is a pair (Q, T) composed of a set Q and an edge relation $T \subseteq Q^2$. A graph (Q', T') is a *subgraph* of a graph (Q, T) if $Q' \subseteq Q \wedge T' \subseteq T$. We then write $(Q', T') \subseteq (Q, T)$. All paths in a subgraph of G are paths in G. For a graph G let $SubGr(G)$ denote the set of all subgraphs of G.

The following result is straightforward from the definitions. It generalizes the supremum definition $(Q, T) \cup (Q', T') \doteq (Q \cup Q', T \cup T')$.

Lemma 2.1 (Complete Lattice of Subgraphs). *Given a graph G, the partial order $(SubGr(G), \subseteq)$ is a complete lattice where the supremum of a set of graphs is formed by taking componentwise union.*

Graph-Based Models. A *(graph-based) model* M is a structure G or (G, A) where G is a graph (Q, T), denoted by *graph* M, and A is a tuple $(A_1, \ldots, A_n)$. Each *(auxiliary) component* A_i is a mapping from Q or T into an external set S_i defined independently of G. If $S_i = \mathsf{Bool}$ then A_i represents a subset of Q or T. If $n = 1$ we omit the tuple parentheses around A and write $M = (G, A_1)$.

A model M is *infinite* if Q is infinite.

The considered models are relatively simple. The graph of a model can be seen as its skeleton. A subset-type component A_i may, e.g., represent a set of target nodes. Other mappings A_i can be used to describe edge labels, e.g. weights, or node labels, e.g. node inputs or outputs in an import network. Kripke structures are models which involve a mapping from nodes into a set of atomic propositions (e.g. [2]). Of course, a combination of both node and edge labels is possible.

For a decorated model identifier such as M' or $\tilde{M}$ it is understood that all its constituents carry the same decoration. E.g., when talking about a model M'' it is understood that $M'' = (G'', A'')$ and similarly for G'' and A''.

Various operations Ψ on models $M = ((Q, T), (A_1, \ldots, A_n))$ are determined by the definitions of $\Psi\, Q$, $\Psi\, T$, and $\Psi\, A_i$ for each component A_i, together with the *distribution rule* $\Psi\, M \doteq ((\Psi\, Q, \Psi\, T), (\Psi\, A_1, \ldots, \Psi\, A_n))$.

A *model class* $\mathcal{M}$ is some set of models which have the same structure, i.e., the same number and types of components. An *$\mathcal{M}$-model* is an element of $\mathcal{M}$.

Quotient Operation. Consider a model $M = ((Q, T), A)$ and an equivalence $E \subseteq Q^2$. The *quotient operation "/"* serves to construct quotient models and is defined first for states and transitions, while the auxiliary components A_i will be treated in the following paragraph.

For any $x, x' \in Q$, let $x/E \doteq E(x)$ and $(x, x')/E \doteq (x/E, x'/E)$. For any $U \subseteq Q$ or $U \subseteq T$, let $U/E \doteq \{u/E \mid u \in U\}$. Thus $T/E = \{(x/E, x'/E) \mid (x, x') \in T\} \subseteq (Q/E)^2$. We say that E is *finitary* if Q/E is finite.

Model Compatibility and Canonical Quotient Models. An equivalence E on a model $M = (G, A)$ is called *M-compatible* if each map A_i satisfies $\forall u, u' \in Dom(A_i) : u/E = u'/E \Rightarrow A_i(u) = A_i(u')$. Relationally this can be expressed as $E \cap Dom(A_i)^2 \subseteq A_i\,;\, A_i^\circ$.

Let $Compat(M)$ denote the set of M-compatible equivalences. For any $E \in Compat(M)$, we define the map A_i/E by $\forall u \in Dom(A_i) : (A_i/E)(u/E) = A_i(u)$.

The *(canonical) quotient (model) of M by E* is M/E as defined by the above distribution rule. A model class may contain models M and quotients M/E.

Labelled Transition Systems. The classical *labelled transition systems*, a.k.a. LTS, have the form $S = (Q, H, T_H)$ where H is a finite set of transition labels and $T_H \subseteq Q \times H \times Q$ [1, 5].

For each $h \in H$, let $T_h \doteq \{(x, x') \mid (x, h, x') \in T_H\}$. Then S is easily transformed into the model $M_S = ((Q, T), L)$, where $L : T \to 2^H$ and $\forall t \in T : L(t) = \{h \mid t \in T_h\}$. Thus S amounts to a structured presentation of the model M_S.

2.3 Bisimulation Classes

Bisimulations prove useful because they preserve many properties of system dynamics. Different classes of bisimulations are associated with different properties and different classes of models. Bisimulation equivalences may determine drastically reduced quotients and may even downsize some infinite models into finite ones. They can be computed in polynomial time in the case of finite models. Therefore we treat bisimulation equivalences in greater detail.

2.3.1 Bisimulations

We briefly review the classical definition of bisimulations, e.g. [14], and recall an equivalent form that is more suitable for our purposes.

A *(partial) simulation* between two models M, M' is a relation $R \subseteq Q \times Q'$ such that $\forall x, y \in Q,\ x' \in Q' : x\,T\,y \wedge x\,R\,x' \Rightarrow \exists y' \in Q' : x'\,T'\,y' \wedge y\,R\,y'$.

A *bisimulation* between models M, M' is a relation $R \subseteq Q \times Q'$ such that R and R° are simulations between M, M' and M', M, resp. For actually computing bisimulations the following equivalent characterisation is fundamental (see the preface in [13]): R is a bisimulation between M, M' iff $R \subseteq F_{basic}^{(M,M')}(R)$ where

$$F_{basic}^{(M,M')} \;\doteq\; \lambda R : Q \times Q' \,.\, \{\, (x, x') \mid (\forall y : xTy \Rightarrow \exists y' : x'T'y' \wedge yRy') \wedge (\forall y' : x'T'y' \Rightarrow \exists y : xTy \wedge yRy') \quad \} \;.$$

The above condition characterizes R as an expanded element, a.k.a. a post-fixpoint, of the isotone function $F_{basic}^{(M,M')}$. Since the set of all relations is a complete lattice under the $\subseteq$ ordering, by the Knaster-Tarski theorem that function has a greatest fixpoint. The latter is called the *coarsest* bisimulation between M, M', since it coincides with the union of all bisimulations between M, M'. It can be computed by iterating $F_{basic}^{(M,M')}$, starting with the universal relation.

2.3.2 Bisimulation Classes and Generators of Bisimulation Maps

We abstract a bit from the above definitions, since we aim at a generic model refinement technique. From now on, the bisimulations introduced as post-fixpoints in § 2.3.1 are called *basic bisimulations* and their defining function is $F_{basic}^{(M,M')}$.

Families and Classes of Bisimulations. Let M, M' be two models from an arbitrary model class. The set of all basic bisimulations for (M, M') is denoted by $\mathcal{B}_{basic}^{(M,M')}$. A *bisimulation family* $\mathcal{B}^{(M,M')}$ is a subset of $\mathcal{B}_{basic}^{(M,M')}$ that is closed under arbitrary unions. This subset may be defined by auxiliary constraints – e.g. equalities between node labels– which ensure the model compatibility of bisimulation equivalences (§ 2.2).

Given a model class $\mathcal{M}$ and a definition of the family $\mathcal{B}^{(M,M')}$ for all $M, M' \in \mathcal{M}$, the *($\mathcal{M}$-)bisimulation class* $\mathcal{B}$ is the set $\bigcup_{M,M' \in \mathcal{M}} \mathcal{B}^{(M,M')}$.

Bisimulation Maps. As recalled in § 2.3.1, basic bisimulation families can be defined using maps that generate basic bisimulations. A number of specialized bisimulation families can similarly be defined using functions between relations over states. Namely, a $\mathcal{B}^{(M,M')}$*-bisimulation map* is an isotone function $F_{\mathcal{B}}^{(M,M')} : 2^{Q \times Q'} \to 2^{Q \times Q'}$ such that $\mathcal{B}^{(M,M')} = \{R \mid R \subseteq F_{\mathcal{B}}^{(M,M')}(R)\}$.

The function $F_{\mathcal{B}}^{(M,M')}$ may be defined by a term $\lambda R : Q \times Q' . G$, where R, M, M' may occur free in G. Moreover the type of the parameter R may be strengthened as follows. Assume $G = W \cap G'$ where $W \subseteq Q \times Q'$ is defined independently of R. Then $F_{\mathcal{B}}^{(M,M')}$ may be given as $\lambda R : W \,.\, G'$. This formulation allows simplifying the computation of the greatest fixpoint of $F_{\mathcal{B}}^{(M,M')}$.

Iterative Construction of Coarsest Bisimulations. The coarsest basic bisimulation for M, M' is the greatest fixpoint of the map $F_{basic}^{(M,M')}$. It can be obtained as $GFP(F_{basic}^{(M,M')}) \doteq \bigcap_{n \in \mathbb{N}} (F_{basic}^{(M,M')})^n (Q \times Q')$; see § 2.3.1 and [5, 14]. Similarly, if $F_{\mathcal{B}}^{(M,M')} \doteq \lambda R : W \,.\, G$ is a $\mathcal{B}^{(M,M')}$-bisimulation map then the coarsest $\mathcal{B}^{(M,M')}$-bisimulation is $GPF(F_{\mathcal{B}}^{(M,M')}) = \bigcap_{n \in \mathbb{N}} (F_{\mathcal{B}}^{(M,M')})^n (W)$.

Map Generators. A simple additional generalization abstracts bisimulation maps from particular models and thus determines a generic bisimulation-map for a whole set of models having the same structure. Namely, given a model class $\mathcal{M}$ and a bisimulation class $\mathcal{B}$, an *($\mathcal{M}$-bisimulation) map generator* $F_{\mathcal{B}}$ is a function that assigns a $\mathcal{B}$-bisimulation map $F_{\mathcal{B}}^{(M,M')}$ to any $M, M' \in \mathcal{M}$.

Hence $F_{\mathcal{B}} \doteq \lambda M, M' : \mathcal{M} \,.\, F_{\mathcal{B}}^{(M,M')}$, viz. $F_{\mathcal{B}}(M, M') = F_{\mathcal{B}}^{(M,M')}$. So the characteristic predicate $\chi_{\mathcal{B}}$ can be given by $\chi_{\mathcal{B}}(R) \hat{=} \bigvee_{M,M' \in \mathcal{M}} R \subseteq F_{\mathcal{B}}(M, M')(R)$.

Examples for map generators will be presented in § 2.4 and § 4.

2.3.3 Conventions

We fix a few further notational conventions and abbreviations on the basis of classical notions. Let $\mathcal{B}$ be a bisimulation class and assume M, M' belong to the same model class.

- A relation R is a $\mathcal{B}$*-bisimulation between* M *and* M' if $R \in \mathcal{B}^{(M,M')}$.
- The models M, M' are $\mathcal{B}$*-bisimilar* if $\exists R : R \in \mathcal{B}^{(M,M')}$.
- We write $\mathcal{B}^{(M)}, F_{\mathcal{B}}^{(M)}, F_{\mathcal{B}}(M)$ for $\mathcal{B}^{(M,M)}, F_{\mathcal{B}}^{(M,M)}, F_{\mathcal{B}}(M, M)$, resp.

– A relation $R \subseteq Q^2$ is a *$\mathcal{B}$-bisimulation for M* if $R \in \mathcal{B}^{(M)}$.

2.3.4 Bisimulation Equivalences and Model Reductions

We recall two useful properties (see e.g. [1, 2]) and introduce quotient models based on coarsest bisimulations. One fixed but arbitrary model class containing models and their quotients is understood.

Lemma 2.2 (Bisimulation Equivalences). *Consider a model M, a bisimulation class $\mathcal{B}$, and $E \in Compat(M)$.*

1. *The coarsest $\mathcal{B}$-bisimulation for M is an equivalence.*
2. *If E is a $\mathcal{B}$-bisimulation for M then M/E and M are $\mathcal{B}$-bisimilar.*

For a specific bisimulation class, these properties are established by the proofs of Lemma 7.8 and Theorem 7.14, resp., in [1]. These proofs can be reused for other auxiliary components and thus for other bisimulation classes.

A bisimulation class $\mathcal{B}$ is *$\mathcal{M}$-compatible* if, for all $M \in \mathcal{M}$, the coarsest $\mathcal{B}$-bisimulation for M is M-compatible; see Part 1 above and § 2.2.

Reducers and Reductions of Models. The coarsest $\mathcal{B}$-bisimulation for M determines the least number of equivalence classes of any bisimulation in $\mathcal{B}^{(M)}$. It is called the *(strongest $\mathcal{B}$-)reducer of M* and is denoted by $Red_{\mathcal{B}}(M)$. Accordingly the quotient model $M/Red_{\mathcal{B}}(M)$ is called the *(strongest $\mathcal{B}$-)reduction of M*.

For sufficiently regular infinite models, called *well-structured* [10], the reducers are finitary and hence the reductions are finite.

2.4 Algorithms Generating Bisimulation Equivalences

A *bisimulation algorithm $BisimAlgo_{\mathcal{B}}$ for* an $\mathcal{M}$-map-generator $F_{\mathcal{B}}$ is an algorithm which, given $M \in \mathcal{M}$, yields the $\mathcal{B}$-reducer $Red_{\mathcal{B}}(M)$, if finitary. So it computes the function $\lambda M : \mathcal{M} \,.\, GFP(F_{\mathcal{B}}\, M)$; see § 2.3.2. A few such algorithms are briefly recalled hereafter; all are derived from a fundamental algorithm for labelled transition systems in which partitions represent equivalences.

In this section models are finite unless indicated otherwise.

1. Labelled Transition Systems. Let M_H be an LTS (Q, H, T_H) (§ 2.1). Recall that $xT_hy \equiv (x, h, y) \in T_H$. The basic bisimulation map generator F^{LTS}_{basic} is akin to F_{basic} (§ 2.3.2) and is defined as follows [5, 14]:

$$F^{LTS}_{basic} \doteq \lambda M_H \,.\, \lambda E : Q^2 . \{(x, x') \mid \forall h \in H : (\forall y : xT_hy \Rightarrow \exists y' : x'T_hy' \wedge yEy') \wedge (\forall y' : x'T_hy' \Rightarrow \exists y : xT_hy \wedge yEy') \ \}.$$

A bisimulation algorithm $BisimAlgo^{LTS}_{basic}$ is found in [11].

2. Basic Maps for Models. A bisimulation algorithm $BisimAlgo_{basic}$ is obtained in two steps. First, the graph (Q, T) of any model M is transformed into the LTS $M_H = (Q, \{h\}, \{(x, h, y) \mid xTy\})$, where $H = \{h\}$ consists of an arbitrary single

label h (§ 2.1). Thus the basic bisimulations for M are those for M_H. Second, Case 1 above is applied.

3. Maps with an Independent Auxiliary Equivalence. Consider a model class $\mathcal{M}$ and a map generator $F_{basic,W} \doteq \lambda M : \mathcal{M} \,.\, \lambda E : W \,.\, F_{basic}(E)$ (§ 2.3.2). A bisimulation algorithm $BisimAlgo_{basic,W}$ is obtained by applying $BisimAlgo_{basic}$ from Case 2 where the initial partition $\{Q\}$ is replaced by Q/W [1, 5].

4. Maps with a Dependent Auxiliary Equivalence. Consider the class $\mathcal{M}_g$ of models $((Q,T),g)$ where $g : T \to S$ for a given set S. Let the map generator be

$$F_g \doteq \lambda M : \mathcal{M}_g \,.\, \lambda E : Q^2 \,.\, W_g(E) \cap F_{basic}(E)$$
$$W_g(E) \doteq \{(x,x') \mid \forall y,y' : xTy \wedge x'Ty' \wedge yEy' \Rightarrow g(x,y) = g(x',y')\}.$$

Case 3 is inapplicable because W_g depends on E. A bisimulation algorithm $BisimAlgo_g$ can be obtained as follows. First, any $\mathcal{M}_g$-model M is transformed into the LTS $M_H = (Q, H, T_H)$ such that $H = Im(g)$ and $T_H = \{(x, g(x,y), y) \mid xTy\}$ (§ 2.1). Thus the bisimulations generated by F_g for M are the basic bisimulations for M_H. Second, Case 1 is applied.

Bisimulation algorithms can be defined similarly for other models which include mappings akin to g.

5. Well-Structured Infinite Models. Since their reducers are finitary equivalences (§ 2.3.4), a symbolic variant of Case 2, 3 or 4 can be used [10].

3 Generic Refinement using Finite Abstract Models

As outlined in § 1, a given model is abstracted into a finite quotient model and the latter is refined into a model that must be expanded back to a full-fledged model refining the given one. In this main section, first an adequate expansion operation is constructed. Second, we define a class of formulae that are preserved under expansion. Third, a refinement algorithm is presented.

As in § 2.3.4, one fixed but arbitrary model class $\mathcal{M}$ is understood. Likewise, an $\mathcal{M}$-compatible bisimulation class $\mathcal{B}$ is understood.

3.1 Construction of an Adequate Expansion Operation

To ensure consistency, expansion needs to be an approximate inverse of quotient, which is not invertible (§ 2.2). Moreover refined models should include a maximum of useful states and transitions. Expansion should thus generate the largest possible models. Therefore, to ensure invertibility and maximality, the quotient and expansion operations must be adequately restricted.

Fortunately, expansion, restricted quotient, and restricted expansion can easily be developed using a Galois connection between models and their quotients.

3.1.1 A Brief Reminder about Galois Connections

A pair (F, G) of total functions $F : A \to B$ and $G : B \to A$ between pre-orders $(A, \leq_A)$ and $(B, \leq_B)$ is called a *Galois connection* between A and B iff

$$\forall x \in A : \forall y \in B : \ F(x) \leq_B y \ \equiv \ x \leq_A G(y) \ . \tag{2}$$

Then F is called the *lower*, G the *upper adjoint* of (F, G). In particular a pair (F, F°) of mutually inverse functions forms a Galois connection.

We summarize the most important results about Galois connections for our purposes (see e.g. [4]), omitting the indices $_A, _B$ and some symmetric properties.

Proposition 3.1 (Closures and Adjoints).

1. *F preserves all existing suprema and G preserves all existing infima.*
2. *$G \circ F$ and $F \circ G$ are a closure and a kernel operator, resp. The set of $G \circ F$-closed elements, i.e., the set of fixpoints of $G \circ F$, is the image set $G(B)$.*
3. *The adjoints of a Galois connection determine each other uniquely.*

By this latter fact, one adjoint allows defining the other. A sufficient condition for the existence of an upper adjoint is given in the following central proposition, which also characterizes the upper adjoint in terms of the lower one.

Proposition 3.2 (Upper Adjoints and Image-Maximal Inverses).

1. *Assume that $(A, \leq_A)$ and $(B, \leq_B)$ are complete lattices. Then every function $F : A \to B$ that preserves all suprema has an upper adjoint G given by*

$$\forall y \in B : G(y) = \sup\{x \in A \mid F(x) \leq y\} \ . \tag{3}$$

2. *Consider a Galois connection (F, G) between preorders $(A, \leq_A)$ and $(B, \leq_B)$. Let $f \doteq F{\downarrow}G(B)$ and $g \doteq G{\downarrow}F(A)$ be the restrictions of F and G to the respective image sets. Then f and g are inverses of each other such that*

$$\textit{(Maximality)} \qquad \forall y \in F(A) : g(y) = \sup\{x \in A \mid F(x) = y\} \, . \tag{4}$$

The function inverse g is called *image-maximal* in order to highlight maximality.

We will apply Prop. 3.2 for the case where F is the quotient operation. The details of the proofs to follow may be skipped in a first reading.

3.1.2 The Complete Lattice of Submodels of a Model

An operation restricting models serves to define lattices of submodels on the basis of lattices of subgraphs (cf. § 2.2).

Model Restriction and Canonical Submodels. Consider a model M and a graph $G' \subseteq \mathit{graph}\, M$. The *restriction operation* "$\Downarrow$" extends domain restriction $\downarrow$ of functions (§ 2.1) to the case of models.

We first set $P{\Downarrow}\, G' \doteq P \cap Q'$ for $P \subseteq Q$ and $S{\Downarrow}\, G' \doteq S \cap T'$ for $S \subseteq T$. For a map A_i, let $A_i{\Downarrow}\, G' \doteq A_i{\downarrow}(\mathit{Dom}(A_i){\Downarrow}\, G')$. Then the *(canonical) submodel of* a model M *induced by* G' is $M{\Downarrow}\, G'$ as defined by the distribution rule of § 2.2.

We establish three useful properties of restriction.

Lemma 3.3 (Composition of Model Restrictions).

1. *$graph\,(M{\Downarrow}\,G') = G'$ and $M = M{\Downarrow}\,graph\,M$.*
2. *If $G' \subseteq graph\,M$ then $Dom(A_i{\Downarrow}\,G') = Dom(A_i){\Downarrow}\,G'$.*
3. *If $G' \subseteq graph\,M$ and $G'' \subseteq G'$ then $(M{\Downarrow}\,G'){\Downarrow}\,G'' = M{\Downarrow}\,G''$.*

Proof. In calculations "$\lhd$ reason" can be read as "*because (of)* reason".

1. The first assertion is immediate from the definition. For the second one we observe that $A_i{\Downarrow}\,(graph\,M) = A_i{\downarrow}(Dom(A_i){\Downarrow}\,(Q,T)) = A_i{\downarrow}Dom(A_i) = A_i$.
2. Immediate from the definition.
3. First, $(M{\Downarrow}\,G'){\Downarrow}\,G'' = M{\Downarrow}\,G''$ is well defined, since by the assumption and Part 1 we have $G'' \subseteq G' = graph\,(M{\Downarrow}\,G')$. Second, given $M = (G,A)$, we treat $G = (Q,T)$ and A in turn. For $P \subseteq Q$, we calculate $(P{\Downarrow}\,G'){\Downarrow}\,G'' = (P \cap Q') \cap Q'' = P \cap Q'' = P{\Downarrow}\,G''$. Likewise for $S \subseteq T$. Then, for $D_i \doteq Dom(A_i)$, $D'_i \doteq D_i{\Downarrow}\,G'$, and $A'_i \doteq A_i{\downarrow}D'_i$, we derive

$$\begin{array}{lll}
(A_i{\Downarrow}\,G'){\Downarrow}\,G'' = A'_i{\Downarrow}\,G'' & \lhd & \text{LHS of the thesis and def. of } {\Downarrow} \\
= A'_i{\downarrow}(D'_i{\Downarrow}\,G'') = A'_i{\downarrow}((D_i{\Downarrow}\,G'){\Downarrow}\,G'') & \lhd & \text{def. of } {\Downarrow} \text{ and Part 2} \\
= A'_i{\downarrow}(D_i{\Downarrow}\,G'') & \lhd & \text{above calculation for } P \text{ or } S \\
= A_i{\downarrow}((D_i{\Downarrow}\,G') \cap (D_i{\Downarrow}\,G'')) & \lhd & \text{def. of } A'_i \text{ and (1)} \\
= A_i{\downarrow}(D_i{\Downarrow}\,G'') = A_i{\Downarrow}\,G'' & \lhd & G'' \subseteq G' \text{ and def. of } {\Downarrow}. \qquad \square
\end{array}$$

Submodel Relation. Using restriction we define the *submodel relation* $\subseteq$ by

$$M' \subseteq M \;\;\doteq\;\; \exists G' \subseteq graph\,M : M' = M{\Downarrow}\,G'$$

where $M, M' \in \mathcal{M}$. If $M' \subseteq M$ we also say that M' *refines* M. For a model M let $Sub(M) \doteq \{M' \mid M' \subseteq M\}$. In case M is infinite, $Sub(M)$ is infinite too.

The submodel –or refinement– relation has pleasant properties.

Lemma 3.4 (Complete Lattices of Submodels). *Consider any $M \in \mathcal{M}$.*

1. *For $M', M'' \in Sub(M)$ we have $M' \subseteq M'' \;\equiv\; graph\,M' \subseteq graph\,M''$.*
2. *The relation $\subseteq$ between models in $Sub(M)$ is a partial order.*
3. *$(Sub(M), \subseteq)$ is a complete lattice where the supremum of a set of submodels is componentwise union.*

Proof.

1. Assume $M' = M{\Downarrow}\,G'$ and $M'' = M{\Downarrow}\,G''$ for some $G', G'' \subseteq graph\,M$. Then by Lemma 3.3.1 we have $graph\,M' = G'$ and $graph\,M'' = G''$.

$$\begin{array}{lll}
(\Rightarrow) \quad M' \subseteq M'' & \lhd & \text{LHS of the thesis} \\
\Rightarrow \;\; graph\,M' = \widetilde{G} & \lhd & \text{for some } \widetilde{G} \subseteq G'' \text{ by def. of } \subseteq \\
 & & \text{and Lemma 3.3.1} \\
\Rightarrow \;\; G' \subseteq G'' & \lhd & G' = graph\,M' = \widetilde{G} \subseteq G''. \\
(\Leftarrow) \quad G' \subseteq G'' & \lhd & \text{RHS of the thesis} \\
\Rightarrow \;\; M''{\Downarrow}\,G' = (M{\Downarrow}\,G''){\Downarrow}\,G' & \lhd & \text{unfold } M'' \\
= M{\Downarrow}\,G' = M' & \lhd & \text{Lemma 3.3.3 and fold } M' \\
\Rightarrow \;\; M' \subseteq M'' & \lhd & \text{def. of } \subseteq.
\end{array}$$

2. This is immediate from Part 1, since $\subseteq$ is a partial order on graphs.
3. By Parts 1 and 2 the partial orders $(\mathit{SubGr}(\mathit{graph}\, M), \subseteq)$ and $(\mathit{Sub}(M), \subseteq)$ are order-isomorphic. The former is a complete lattice by Lemma 2.1. □

3.1.3 Expansion as an Upper Adjoint of Quotient

Expansion is specified using Prop. 3.2.1 and then is expressed constructively.

Specification of the Expansion Operation. We first study the interaction between the submodels of a model M and those of a quotient M/E. The equivalence E need not be a bisimulation.

Lemma 3.5 (Sub-Quotients). *Let $E \in \mathit{Compat}(M)$.*

1. *$\forall M' \in \mathit{Sub}(M) : E \in \mathit{Compat}(M')$, i.e., $E \cap \mathit{Dom}(A_i')^2 \subseteq A_i' ; (A_i')^\circ$.*
2. *The quotient operation $/E : \mathit{Sub}(M) \to \mathit{Sub}(M/E)$ preserves all suprema.*

Proof. Let $D_i \doteq \mathit{Dom}(A_i)$ and $D_i' \doteq \mathit{Dom}(A_i')$.

1. Consider an $M' \in \mathit{Sub}(M)$, say $M' = M{\Downarrow}G'$ for some $G' \subseteq \mathit{graph}\, M$. By definition of $\Downarrow$ we have $A_i' \subseteq A_i$ and thus $D_i' \subseteq D_i$. Hence

$$\begin{array}{lll} E \cap (D_i')^2 = E \cap (D_i)^2 \cap (D_i')^2 & \lhd & \text{LHS of the thesis and } D_i' \subseteq D_i \\ \subseteq (A_i \,;\, A_i^\circ) \cap (D_i')^2 & \lhd & E \in \mathit{Compat}(M) \\ = (A_i{\downarrow}D_i') \,;\, (A_i{\downarrow}D_i')^\circ = A_i' \,;\, (A_i')^\circ & \lhd & \text{relation algebra and def. of } \Downarrow . \end{array}$$

2. By a straightforward calculation. □

Now we can specify the *expansion operation* $\setminus E$ as an upper adjoint, thanks to Part 2 and Prop. 3.2.1 where $A = \mathit{Sub}(M), B = \mathit{Sub}(M/E), F = /E, G = \setminus E$. Namely, for any $M \in \mathcal{M}$ and $E \in \mathit{Compat}(M)$,

$$\forall M' \in \mathit{Sub}(M), N \in \mathit{Sub}(M/E): \quad M' \subseteq N \setminus E \;\;\doteqdot\;\; M'/E \subseteq N \,. \qquad (5)$$

The set $\mathit{Sub}(M/E)\setminus E$ of *E-closed* submodels of M (see Prop. 3.1.2) is denoted by $\mathit{ClSub}_E(M)$. If $M' \in \mathit{ClSub}_E(M)$ then $\forall t \in T, t' \in T' : t'/E = t/E \Rightarrow t \in T'$.

Computable Form of the Expansion Operation. Equation (3) for $G = \setminus E$ is brought into a form that for finite M/E is computable symbolically (§ 2.3.4).

Lemma 3.6 (Constructive Expansion). *Given any $E \in \mathit{Compat}(M)$ and $M_E \in \mathit{Sub}(M/E)$, we have $M_E \setminus E = M{\Downarrow}\widehat{G}$ for the graph*

$$\widehat{G} = (\widehat{Q}, \widehat{T}) \doteq (\, \bigcup Q_E \;,\; \bigcup_{(X,Y) \in T_E} (X \times Y) \cap T \,) \,.$$

Proof. By (3) it suffices to show that $M{\Downarrow}\widehat{G} = \sup\{M' \in \mathit{Sub}(M) \mid M'/E \subseteq M_E\}$. Then Lemma 3.4.1 reduces this to proving $\widehat{Q} = \sup \mathcal{Z}$ where $\mathcal{Z} \doteq \{P \mid P/E \subseteq Q_E\}$, and similarly for $\widehat{T}$. Let us detail the calculation for $\widehat{Q}$; that for $\widehat{T}$ is analogous.

First, clearly $\widehat{Q}/E = (\bigcup Q_E)/E = Q_E \subseteq Q_E$ and hence $\widehat{Q} \in \mathcal{Z}$. Second, for any $P \in \mathcal{Z}$, we deduce $P \subseteq \widehat{Q}$:

$$\begin{array}{lll} & \bigcup(P/E) \subseteq \bigcup Q_E & \lhd \;\; P/E \subseteq Q_E \text{ and isotony of } \bigcup \\ \Rightarrow & P \subseteq \widehat{Q} & \lhd \;\; P = \bigcup(P/E) \text{ and def. of } \widehat{Q} \,. \end{array}$$

□

3.1.4 Restrictions of the Quotient and Expansion Operations

Using Prop. 3.2.2 we first derive f and g from $/E$ and $\setminus E$, resp. Second, we formalize M as a parameter and instantiate E to a coarsest bisimulation (cf. Lemma 2.2.1).

Restricted Expansion as Maximal Inverse of Restricted Quotient. We want to obtain an isomorphism between certain classes of models. To this end we define the following restrictions of $/E$ and $\setminus E$ for any $M \in \mathcal{M}\,, E \in \mathit{Compat}(M)$:

$$\forall M' \in \mathit{ClSub}_E(M):\ \ \mathit{Shrink}_{M,E}\ M' = M'/E\ ,$$
$$\forall N \in \mathit{Sub}(M/E):\ \ \mathit{Grow}_{M,E}\ N = N \setminus E\,,$$

and $\mathit{Shrink}_{M,E} : \mathit{ClSub}_E(M) \to \mathit{Sub}(M/E)\,, \mathit{Grow}_{M,E} : \mathit{Sub}(M/E) \to \mathit{ClSub}_E(M)\,.$ Indeed $\setminus E : B \to A$ entails $A \cap (B \setminus E) = B \setminus E = \mathit{ClSub}_E(M)$ and $M/E \in \mathit{Sub}(M)/E$ entails $B \cap (A/E) = \mathit{Sub}(M/E) \cap (\mathit{Sub}(M)/E) = \mathit{Sub}(M/E)\,.$

The following maximality property follows from (5) and Prop. 3.2.2.

Lemma 3.7. *$\mathit{Grow}_{M,E}$ is the image-maximal inverse of $\mathit{Shrink}_{M,E}$.*

Generic Forms of Restricted Quotient and Expansion. Let $\mathit{ClSub}(M) \doteq \mathit{ClSub}_{\mathit{Red}(M)}(M)$ and $\mathit{SubRed}(M) \doteq \mathit{Sub}(M/\mathit{Red}(M))$. Here $\mathit{Red}(M)$ stands for $\mathit{Red}_{\mathcal{B}}(M)$ (cf. § 2.3.4) given the understood bisimulation class $\mathcal{B}$. We define

$$\forall M \in \mathcal{M}, M' \in \mathit{ClSub}(M):\ \ (\mathit{Shrink}\, M)\, M' = M'/\mathit{Red}(M)\,, \tag{6}$$
$$\forall M \in \mathcal{M}, N \in \mathit{SubRed}(M):\ \ (\mathit{Grow}\, M)\, N = N \setminus \mathit{Red}(M)\,. \tag{7}$$

The typing is $\mathit{Shrink} : (M : \mathcal{M}) \to (\mathit{ClSub}(M) \to \mathit{SubRed}(M))\,, \mathit{Grow} : (M : \mathcal{M}) \to (\mathit{SubRed}(M) \to \mathit{ClSub}(M))\,.$ The reduction of any $M \in \mathcal{M}$ is $(\mathit{Shrink}\, M)\, M$.

Since $\mathit{Grow}\, M = \mathit{Grow}_{M,\mathit{Red}(M)}$ and $\mathit{Shrink}\, M = \mathit{Shrink}_{M,\mathit{Red}(M)}$, Lemma 3.7 entails a parametrized form of maximality.

Proposition 3.8 (Maximal Inverse). *For any $\mathcal{M}$-model M, the function $\mathit{Grow}\, M$ is the image-maximal inverse of the function $\mathit{Shrink}\, M$.*

Therefore $(\mathit{Grow}\, M)\, N = \sup\{M' \in \mathit{Sub}(M) \mid M'/\mathit{Red}(M) = N\}$; see (4). Moreover $M \in \mathit{ClSub}(M)$ since $M = (\mathit{Grow}\, M \circ \mathit{Shrink}\, M)\, M\,.$

3.2 Preservation of Satisfactory Refinements under Expansion

Abstract Model Classes. We assume that $\mathcal{M}$ is partitioned into $\mathcal{M}^{\downarrow}$ and $\mathcal{M}^{\uparrow}$ such that $\mathcal{M}^{\uparrow} = \{M/E \mid M \in \mathcal{M}^{\downarrow} \wedge E \in \mathit{Compat}(M)\}\,.$ Thus quotients of quotients are not considered. In this case $\mathcal{M}$ is called a *two-level* model class. Let the class $\mathcal{N}^{\uparrow} \subseteq \mathcal{M}^{\uparrow}$ of bisimulation quotients be $\mathcal{N}^{\uparrow} \doteq \bigcup_{M \in \mathcal{M}^{\downarrow}} \mathit{SubRed}(M)\,.$ It is said to *abstract* the quotient-free class $\mathcal{M}^{\downarrow}$.

Satisfactory Refinements. Let φ be a predicate over states in $\mathcal{M}$-models. The refinement –or submodel– relation $\subseteq\, : \mathcal{M}^2$ (§ 3.1.2) is strengthened to the *φ-(satisfactory) refinement relation* $\sqsubseteq_{\varphi}\, : (\mathcal{M}^{\downarrow})^2 \cup (\mathcal{N}^{\uparrow})^2$ given by

$$M' \sqsubseteq_{\varphi} M \ \doteqdot\ (M' \subseteq M) \wedge (M' \models \varphi)\,. \tag{8}$$

Certain satisfactory refinement relations can be expanded from $(\mathcal{N}^{\uparrow})^2$ to $(\mathcal{M}^{\downarrow})^2$.

Admissibility. The predicate φ is $\mathcal{M}$-*admissible* –see examples in § 4– if

$$\forall M \in \mathcal{M}^{\downarrow}, E \in \mathcal{B}^{(M)} : \ M/E \models \varphi \ \Rightarrow \ M \models \varphi \ . \tag{9}$$

Lemma 3.9. *For any $\mathcal{M}$-admissible φ, we have $\forall M \in \mathcal{M}^{\downarrow}$, $M' \in \mathit{ClSub}(M)$: $(\mathit{Shrink}\, M)\, M' \models \varphi \ \Rightarrow \ M' \models \varphi$.*

Proof. Let any $M \in \mathcal{M}^{\downarrow}$ and $M' \in \mathit{ClSub}_E(M)$ (§3.1.3) where $E = \mathit{Red}(M)$ and $M'/E \models \varphi$ (6). First we deduce $E \in \mathcal{B}^{(M')}$ by proving $\forall x, y, x' \in Q' : xT'y \wedge xEx' \Rightarrow (\exists y' \in Q' : x'T'y' \wedge yEy')$. Let any $x, y, x' \in Q'$ and $t' = (x, y)$ such that $t' \in T' \wedge xEx'$. For some $y' \in Q$ and $t'' = (x', y')$, we have

$$\begin{array}{lll} & t' \in T' \wedge xEx' & \lhd \ \text{hyp. about } x, y, x', t' \\ \Rightarrow & t' \in T' \wedge xEx' \wedge t'' \in T \wedge yEy' & \lhd \ M' \subseteq M \text{ and } E \in \mathcal{B}^{(M)} \\ \Rightarrow & t'' \in T' \wedge yEy' & \lhd \ t'/E = t''/E \text{ and hyp. about } M'. \end{array}$$

Now (9), $\mathit{Red}(M) \in \mathcal{B}^{(M')}$, and (6) entail $(\mathit{Shrink}\, M)\, M' \models \varphi \Rightarrow M' \models \varphi$. □

For $M' = (\mathit{Grow}\, M)\, N'$, the thesis above becomes $\forall M \in \mathcal{M}^{\downarrow}, N' \in \mathit{SubRed}(M) : N' \models \varphi \Rightarrow (\mathit{Grow}\, M)\, N' \models \varphi$. This yields a basic property by (8) and Lemma 3.5.

Proposition 3.10 (Expanding Abstract Refinements). *Given a two-level model class $\mathcal{M}$, an abstract model class $\mathcal{N}^{\uparrow}$, and an $\mathcal{M}$-admissible formula φ,*

$$\forall M \in \mathcal{M}^{\downarrow}, \ N, N' \in \mathit{SubRed}(M) : N' \sqsubseteq_{\varphi} N \Rightarrow (\mathit{Grow}\, M)\, N' \sqsubseteq_{\varphi} (\mathit{Grow}\, M)\, N \, .$$

The generalization to abstract model classes other than $\mathcal{N}^{\uparrow}$ should be studied. This issue is related to the use of diverse abstractions in abstract verification [3].

3.3 A Generic Algorithm for Model Refinement by Abstraction

As a result we can present the specification and construction of Algorithm *Refine*.

Preconditions. The context has to provide the following data:

- (H_1) A two-level model class $\mathcal{M}$, an $\mathcal{M}$-compatible bisimulation class $\mathcal{B}$, and a map generator $F_{\mathcal{B}}$.
- (H_2) A finite or well-structured infinite $\mathcal{M}^{\downarrow}$-model M for which the finitary $\mathcal{B}$-reducer is obtained by some known algorithm $\mathit{BisimAlgo}_{\mathcal{B}}$.
- (H_3) A set *Frml* of $\mathcal{M}$-admissible formulae.
- (H_4) An algorithm $\mathit{FiniteRefine} : \mathit{Frml} \to (\mathcal{M} \to \mathcal{M})$ with lowest known complexity, which given any $\varphi \in \mathit{Frml}$ and any finite $M \in \mathcal{M}$, produces a model M' that satisfies $M' \sqsubseteq_{\varphi} M \wedge ((M \models \varphi) \equiv M' = M)$.

Postcondition. For parameters $\varphi \in \mathit{Frml}$ and $M \in \mathcal{M}^{\downarrow}$, the result M' satisfies $M' \sqsubseteq_{\varphi} M \wedge (((\mathit{Shrink}\, M) M \models \varphi) \equiv M' = M)$.

Algorithm *Refine*. The constructive function $\mathit{Refine} : \mathit{Frml} \to (\mathcal{M}^{\downarrow} \to \mathcal{M}^{\downarrow})$ is defined as follows, given (6), (7), Lemma 3.6, and the Preconditions:

$$\forall \varphi \in \mathit{Frml}, M \in \mathcal{M}^{\downarrow} : \ (\mathit{Refine}\, \varphi)\, M = (\mathit{Grow}\, M \circ \mathit{FiniteRefine}\, \varphi \circ \mathit{Shrink}\, M)\, M \, .$$

Correctness follows from Prop. 3.8 and Prop. 3.10 where $N' = (\mathit{FiniteRefine}\ \varphi)\ N$, $N = (\mathit{Shrink}\ M)\ M$. For finite models, the complexity of *Refine* is polynomial if that of *FiniteRefine* is polynomial. For well-structured infinite models, symbolic forms of the functions *Shrink* and *Grow* can be used; see § 2.4.5 and Lemma 3.6.

Maximal Refinements. The image-maximal inverse $\mathit{Grow}\ M$ yields the largest possible model given Prop. 3.8. Hence *Refine* generates a maximal correctly refined model if *FiniteRefine* does it. However various finite-state refinement algorithms do not yield maximal models, e.g., they may exclude legitimate nondeterminism. The latter issue is outside the scope of the present work.

If *Shrink* and *Grow* are defined as bidirectional functions then $\mathit{Grow}\ M$ is a right-inverse of $\mathit{Shrink}\ M$ [6]. This does entail the Postcondition but not Prop. 3.8.

Abstract Verification. Let $\mathit{Check} : \mathit{Frml} \to (\mathcal{M}^{\downarrow} \to \mathsf{Bool})$ where $(\mathit{Check}\ \varphi)\ M = (\mathit{FiniteCheck}\ \varphi \circ \mathit{Shrink}\ M)\ M$ assuming $(\mathit{FiniteCheck}\ \varphi)\ M \equiv (M \models \varphi)$. Then $(\mathit{Check}\ \varphi)\ M \equiv (M \models \varphi)$. So *Check* allows verifying models; see e.g. [1]. The preconditions of *Check* are those of *Refine* except for simple changes in (H_4). The function $\mathit{FiniteCheck}\ \varphi$ may have a lower complexity than $\mathit{FiniteRefine}\ \varphi$.

4 A Summary of Typical Applications

Due to limited space we merely sketch a few applications; proofs are omitted.

Determining Minimum-Cost Paths. This example was considered in § 1.

Given a model $M = (G, A)$, an *M-path* is a path of G. For any $Z \subseteq Q$, the set of M-paths from x to some $z \in Z$ is denoted by $\mathit{Paths}(x, G, Z)$.

Consider two models M and M', a bisimulation class $\mathcal{B}$ and a $\mathcal{B}$-bisimulation relation R for (M, M'). Let x_0, x_0' be any states of M, M' such that $x_0 R x_0'$. Then clearly for every M-path $x_0 \dots x_i \dots x_n$, there is an R-bisimilar M'-path $x_0' \dots x_i' \dots x_n'$, i.e., $\bigwedge_{i=0\dots n} x_i R x_i'$, and for any M'-path from x_0' there is a corresponding R-bisimilar M-path starting in x_0. See also Lemma 7.5 in [1].

Let $\mathcal{M}_{mcp}$ be the set of models $(G, (Z, g, V))$ where $G = (Q, T)$ may be infinite, $Z : Q \to \mathsf{Bool}$ represents the subset $Q - \mathit{Dom}(T)$ (for brevity we therefore write $Z = Q - \mathit{Dom}(T)$), $g : T \to \mathbb{R}^+$ is a total edge-cost function such that $\mathit{Im}(g)$ is finite, and $V : Q \to \mathbb{R}^+$ is the *value function for* M such that V is total and $\forall x \in Q : V(x) \doteq \min\{\mathit{cost}(p) \mid p \in \mathit{Paths}(x, G, Z)\}$ where $\mathit{cost}(p)$ is the cumulative cost of p. Obviously, $V(x) = 0$ iff $Z\,x = \mathsf{true}$. In this application refinement is thus expressed by the removal of edges but not of nodes. Indeed Z is reachable from every $x \in Q$ since $V(Q) \subseteq \mathbb{R}^+$. If needed, Q is first replaced by the set of Q-nodes from which Z is reachable.

The $\mathcal{B}_{mcp}$-bisimulation equivalences are generated by the map

$$F_g \doteq \lambda M : \mathcal{M}_{mcp}\,.\,\lambda E : Q^2.\ W_g(E) \cap F_{basic}(E),$$
$$W_g(E) \doteq \{\ (x, x')\ \mid \forall y, y' : xTy \wedge x'Ty' \wedge yEy' \Rightarrow g(x, y) = g(x', y')\}.$$

The bisimulation algorithms in § 2.4.4-5 can be used. The $\mathcal{M}_{mcp}$-compatibility of $\mathcal{B}_{mcp}$ is proved using Lemma 2.2 and the above construction of bisimilar paths. Model quotients are independent of V since F_g is independent of V.

A model M is *optimal* iff $\forall x \in Q : \varphi_{mcp}\, x$ where φ_{mcp} is defined by $\varphi_{mcp}\, x \doteq \forall p \in \mathit{Paths}(x, G, Z) : \mathit{cost}\, p = V(x)$. Regarding submodels, $M' \subseteq M$ if $Q' = Q, T' \subseteq T, Z' = Z$. So $g' = g{\downarrow}T'$ and $V' = V{\downarrow}Q' = V$ given $Q' = Q$. By definition of $\mathcal{M}_{mcp}$-models, V' is the value function for M'. Hence the optimality of M' entails its *optimality w.r.t.* M since $\forall x \in Q : V'(x) = V(x)$.

The $\mathcal{M}_{mcp}$-admissibility of φ_{mcp} has been proved. So *Refine* φ_{mcp} is applicable. As well known, the complexity of *FiniteRefine* φ_{mcp} is polynomial.

Illustration. We illustrate the above *mcp*-application. Here $[a, b]$ stands for the closed interval $\{x \mid x \in \mathbb{R} \wedge a \leq x \leq b\}$. Thus the set $[a, b]$ is infinite when $a < b$. The (half-)open intervals $]a, b]$, $[a, b[$ and $]a, b[$ are defined analogously.

Consider $M = ((Q, T), (Z, g, V))$ where $Q = [0, 4]$, the map V may remain implicit, $T = \{(x, y) \mid x \in [0, 1] \wedge y = x + 3 \vee x \in [0, 3[\wedge y = x + 1\}$, $Z = [3, 4]$ and $g = (T \to \{10, 5\}) \cap \{((x, y), c) \mid y = x + 3 \wedge c = 10 \vee y = x + 1 \wedge c = 5\}$.

First, M is reduced to $N \doteq (\mathit{Shrink}\, M)\, M = M/\mathit{Red}(M)$ using F_g. Let $N = ((Q_N, T_N), (Z_N, g_N, V_N))$, $X_0 = \{0\}$, $X_1 =]0, 1[$, $X_2 = \{1\}$, $X_3 =]1, 2[$ and $X_4 = [2, 3[$. We obtain $Q_N = \{X_0, X_1, X_2, X_3, X_4, Z\}$, $Z_N = \{Z\}$ and

$T_N = \{(X_0,X_2),(X_0,Z),(X_1,X_3),(X_1,Z),(X_2,X_4),(X_2,Z),(X_3,X_4),(X_4,Z)\}$,
$g_N = (T_N \to \{10,5\}) \cap \{((X,Y),c) \mid (X \neq X_4 \wedge Y = Z) \equiv (c = 10)\}$,
$V_N = \{(X,c) \mid X = Z \wedge c = 0 \vee X = X_4 \wedge c = 5 \vee X \in \{X_0,X_1,X_2,X_3\} \wedge c = 10\}$.

Second, $\mathit{FiniteRefine}\, \varphi_{mcp}$ is applied to the finite quotient N. The result is $N' = ((Q_N, T'_N), (Z_N, g'_N, V_N))$ where $T'_N = T_N - \{(X_0, X_2), (X_1, X_3)\}$ and $g'_N = g_N{\downarrow}T'_N$. By construction we have $N' \sqsubseteq_{\varphi_{mcp}} N$.

Third, the optimal N' is expanded to $(\mathit{Grow}\, M)\, N' = N' \setminus \mathit{Red}(M)$. The result is $M' = ((Q, T'), (Z, g', V))$ where $T' = \{(x, y) \mid x \in [0, 1] \wedge y = x + 3 \vee x \in [1, 3[\wedge y = x + 1\}$ and $g' = g{\downarrow}T'$. Indeed $X_0 \cup X_1 \cup X_2 = [0, 1]$ and $X_2 \cup X_3 \cup X_4 = [1, 3[$. By construction, $M' \sqsubseteq_{\varphi_{mcp}} M$. Note $T'(1) = \{4, 2\}$. Thanks to the above expression for V_N it is easy to derive an expression of V, namely $V = \{(x, c) \mid x \in [3, 4] \wedge c = 0 \vee x \in [2, 3[\wedge c = 5 \vee x \in [0, 2[\wedge c = 10\}$.

The bisimulation map F_g could be replaced by another one such that the equivalence of any nodes x and y entails $V(x) = V(y)$ (see [7, 15]). This could help in obtaining finitary equivalences. However the function V should then be computed symbolically before reducing any given infinite model.

Generalization to a Family of Optimal Control Problems. A *(selective and complete) dioid* is a complete idempotent semiring $(S, \oplus, \otimes, 0, 1)$, where S is a set of measures of some sort, $\otimes$ accumulates measures, $\oplus$ selects optimal measures, 0 and 1 are the neutral elements of $\oplus$ and $\otimes$, $\otimes$ is distributive w.r.t. $\oplus$, and the natural order $a \leq b \mathrel{\dot{=}} a \oplus b = b$ is linear. Various optimization problems are formalized using dioids D, e.g., the shortest-paths problem where $D = (\mathbb{R}_0^+ \cup \{\infty\}, \min, +, \infty, 0)$, the maximum capacity problem where $D = (\mathbb{R} \cup \{\pm\infty\}, \max, \min, -\infty, \infty)$, and related problems for Markov chains. See [9].

The *mcp*-example has been generalized to the class $\mathcal{M}_{optim}$ of dioid-based models and the corresponding adaptation φ_{optim} of φ_{mcp}. Hence *Refine* φ_{optim} is applicable. Incidentally, *Check* φ_{optim} too is applicable; see § 3.3.

The complexity of *FiniteRefine* φ_{optim} is polynomial in the case of $\mathcal{M}_{optim}$-models based on dioids with 1 as greatest element wrt. $\leq$, see [9] again. This is equivalent to $a \leq b \Rightarrow a \otimes c \leq b$ for all a, b and c, so extending a path cannot improve its cost. For instance the dioids $(\mathbb{R}_0^+ \cup \{\infty\}, \min, +, \infty, 0)$ and $(\mathbb{R} \cup \{\pm\infty\}, \max, \min, -\infty, \infty)$ shown above have this property, contrary to the dioid $(\mathbb{R}_0^+ \cup \{\infty\}, \max, +, 0, 0)$ used in the longest-paths problem. However, this property is not a necessary condition for polynomial complexity.

Application to a Family of Temporal Properties. We consider the temporal properties expressed in the logic *CTL**. Let be given a finite set P of atomic propositions. The P-based class $\mathcal{M}_{temp}$ is the set of models $((Q,T),(Q_{init},L))$ where $\forall x \in Q : T(x) \neq \emptyset$, the map $Q_{init} : Q \to \mathsf{Bool}$ characterizes a set of initial states, and the map $L : Q \to 2^P$ is total [1, 2]. The $\mathcal{B}_{temp}$-bisimulation equivalences are determined by the auxiliary equivalence $W_{temp} \doteq Q^2 \cap \{(x,x') \mid L(x) = L(x')\}$. We assume that Q_{init} in the given M defines an E-closed set for any equivalence $E \in \mathcal{B}_{temp}^{(M)}$. The bisimulation algorithms in § 2.4.3 and § 2.4.5 can be used and the $\mathcal{M}_{temp}$-compatibility of $\mathcal{B}_{temp}$ is easily checked.

The good news is that all *CTL**-formulae are $\mathcal{M}_{temp}$-admissible (Thm. 14 in [2]). Hence for each *CTL**-formula φ, *Refine* φ is applicable. The bad news is that the time complexity of *FiniteRefine* φ is at least exponential in the size of φ, like that of φ-satisfiability. This complexity remains exponential if φ belongs to the less general logics *LTL* or *CTL*. See e.g. [1].

5 Related Work and Conclusion

Related Work. In [7], a problem of optimal stochastic control is tackled with the help of a dedicated bisimulation. In [12], a design method for supervisory control is developed. It uses bisimulations, is applied to qualitative properties, and involves polynomials as symbolic representations of sets. To study the applicability of our approach to these problems would be worthwhile.

Conclusion. The proposed method reduces the refinement of models to that of finite abstract ones. It involves a restricted expansion which maps refined abstract models back to maximal submodels. It can be used for quantitative or qualitative goals and for models which are very large but finite or infinite but well-structured. Its usefulness depends on various factors which need further examination: each considered design problem must be defined in terms of a model class $\mathcal{M}$, an $\mathcal{M}$-compatible bisimulation class, and an $\mathcal{M}$-admissible goal formula; very large models must collapse to drastically smaller quotient models; we should know efficient algorithms for solving finite-state problem instances.

References

1. Ch. Baier, J.-P. Katoen, *Principles of model checking*, MIT Press, 2008.
2. E.Clarke, O. Grumberg, D. Peled, *Model checking*, MIT Press, 3rd ed., 2001.

3. P. Cousot, R. Cousot, Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints. *4th Symp. Principles of Programming Languages*: 238–252. ACM, New York, 1977.
4. M. Erné, J. Koslowski, A. Melton and G. Strecker, A primer on Galois connections. In: S. Andima et al. (eds.), *Papers on general topology and its applications.* 7th Summer Conf. Wisconsin. Annals New York Acad. Sci. 704, New York (1994), 103–125.
5. J.-C. Fernandez, An implementation of an efficient algorithm for bisimulation equivalence; *Sci. Computer Programming* 13(2–3):219-236, 1989.
6. J.N. Foster, M.B. Greenwald, J.T. Moore, B.J. Pierce, A. Schmitt, Combinators for bidirectional tree transformations: a linguistic approach to the view-update problem; *ACM Trans. Programming Languages and Systems* 29(3): Article 17, 65 pages, 2007.
7. R. Givan, Th. Dean, M. Greig, Equivalence notions and model minimization in Markov decision processes, *Artificial Intell. J.* 147(1-2): 163-223, 2003.
8. R. Glück, B. Möller, M. Sintzoff, A semiring approach to equivalences, bisimulations and control, *Conf. Relational Methods in Computer Sci. and Applications of Kleene Algebra*, LNCS 5827: 134-149, Springer, 2009.
9. M. Gondran, M. Minoux, *Graphs, dioids and semirings: new models and algorithms,* Springer, 2008.
10. T.A. Henzinger, R. Majumdar, J.-F. Raskin, A classification of symbolic transition systems, *ACM Trans. Computational Logic* **6** (2005) 1–32.
11. P. Kanellakis, S. Smolka, CCS expressions, finite state processes, and three problems of equivalence, *Information and Computation* **86** (1990) 43–68.
12. H. Marchand, S. Pinchinat, Supervisory control problem using symbolic bisimulation techniques, *Proc. Amer. Control Conf.*, Vol.6, 4067–4071, 2000.
13. R. Milner, A calculus of communicating systems. Extended reprint of LNCS 92. University of Edinburgh, Laboratory for Foundations of Computer Science, Report ECS-LFCS-86-7, 1986
14. R. Milner, Operational and algebraic semantics of concurrent processes. In: J. van Leeuwen (ed.), *Formal models and semantics*, Handbook of Theoretical Computer Sci., Vol.B: 1201–1242, Elsevier, 1990.
15. M. Sintzoff, Synthesis of optimal control policies for some infinite-state transition systems, *Conf. Maths of Program Construction*, LNCS 5133: 336–359, Springer, 2008.