Compatibility of Refining and Controlling Plant Automata with Bisimulation Quotients

Roland Glück[1]

[1]Center for Lightweight Production Technology
German Aerospace Center

20th International Conference on Relational and Algebraic Methods Computer Science
Augsburg, April 3rd, 2023

Idea

- Bisimulations known as tool in model checking
- Properties can be checked on bisimilar (possibly smaller) models
- Here application of bisimulations to model refinement
- Restriction to LTL formulae
- shrink - refine - expand
- possible speed-up
- possible taming of infinite systems

Definition of Plant Automata

Definition

A *model* is a tuple $M = (V, E, g, a)$ such that

- (V, E) is a directed graph,
- $g : E \to 2^{\Sigma}$ is the *edge labelling* function and
- $a : V \to 2^{\Pi}$ is the *node labelling* function.
- Π and Σ are disjoint alphabets.

A *plant automaton* has a unique $v_0 \in V$ with $\mathrm{I} \in a(v) \Leftrightarrow v = v_0$.

- Models correspond to labelled transition systems.
- v_0 has the role of a starting node.

Refinement

Writing convention: $(v, \alpha, w) \in E \Leftrightarrow_{df} \alpha \in g(v, w)$

Definition

A model $M' = (V', E', g', a')$ is called a *refinement* of a model $M = (V, E, g, a)$ if the following conditions hold:

- $V' = V$
- $(v, \alpha, w) \in E' \Rightarrow (v, \alpha, w) \in E$
- $a'(v) = a(v)$

- Refinement changes behavior of a model.
- Refinement keeps node set.
- Every present transition can be disabled or kept.

Controller

Definition

Given a model $M = (V, E, g, a)$ a *controller* of M is a mapping $c : V \to 2^{\Sigma}$ such that for all $v \in V$ the inclusion $c(v) \subseteq \{\alpha \mid \exists w : (v, \alpha, w) \in E\}$ holds. The model $M|c =_{df} (V|c, E|c, g|c, a|c)$, also called *$M$ controlled by* c, is defined as follows:

- $V|c =_{df} V$
- $(v, \alpha, w) \in E|c \Leftrightarrow_{df} (v, \alpha, w) \in E \wedge \alpha \in c(v)$
- $(a|c)(v) =_{df} a(v)$

- Controller can be used to change a model's behavior.
- Allows only disabling of groups of transitions with common edge label.

Runs, Traces and LTL Properties

Definition

A *run* r is a finite or infinite sequence from $V(\Sigma V)^* \cup V(\Sigma V)^\omega$ such that $(v_i, \alpha_, v_{i+1}) \in E$ holds for all subsequences of r from $V\Sigma V$. A run is called a *trace* if it starts with v_0 (the unique node with $\mathsf{I} \in a(v_0)$). A plant automaton M is called *live* if it has at least one trace and for every finite trace $v_0\alpha_0 v_1\alpha_1 v_2\alpha_2 v_3 \ldots v_i$ of M there exist an $\alpha_i \in \Sigma$ and a $v_{i+1} \in V$ such that $(v_i, \alpha_i, v_{i+1}) \in E$ holds.

A plant automaton *satisfies* an LTL formula φ if it is live and every trace fulfills φ.

Refineability and Controllability

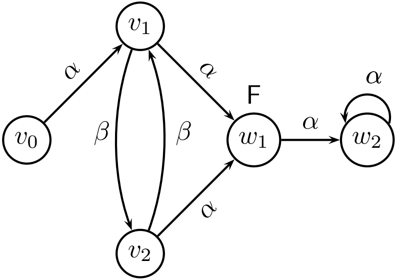
Definition

A plant automaton M is *refineable* with respect to an LTL formula φ if there is a refinement M' of M such that M' satisfies φ. It is *controllable* with respect to φ if there is a contoller c such that $M|c$ satisfies φ.

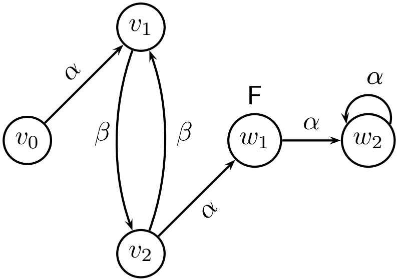
Goals:

- deciding refineabiliy/controllability
- computing an actual refinement/controller

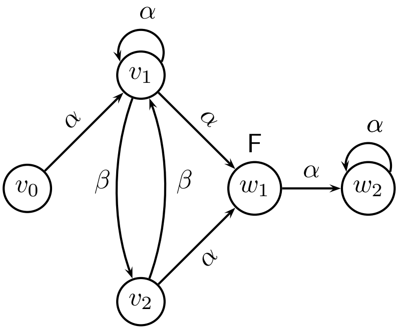
Example Plant Automaton



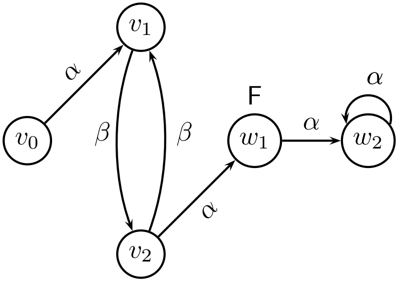
Refinement



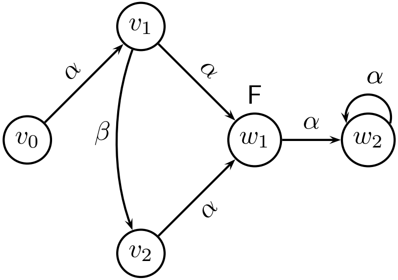
Another Plant Automaton



Another Plant Automaton, controlled



Runs, Traces and LTL formulae



- Run: $r = v_1 \beta v_2 \alpha w_1$
- Trace: $t = v_0 \alpha v_1 \alpha w_1 (\alpha w_2)^\omega$
- t fulfills $\Diamond$F
- even plant automaton satisfies $\Diamond$F

NP-hardness

Theorem

In general, it is NP-hard to decide whether a plant automaton is refineable (controllable) with respect to an LTL formula.

Proof:

- reduction from directed Hamilton cycle
- given $G = (V, E)$, pick an arbitrary $v_0 \in V$
- label all edges with a unique label
- set $a(v_0) = \mathsf{I}$ and $a(v) = \mathsf{F}$ for $v \neq v_0$
- $\varphi =_{df} (\bigwedge_{i=1}^{|V|-1} \bigcirc^i \mathsf{F}) \wedge \bigcirc^{|V|} \mathsf{I}$

Bisimulations

Definition

Given two models $M = (V, E, g, a)$ and $\hat{M} = (\hat{V}, \hat{E}, \hat{g}, \hat{a})$ we call a relation $B \subseteq V \times \hat{V}$ a *bisimulation* between M and $\hat{M}$ if B is both left and right total and fulfills the following conditions:

- $(v, \hat{v}) \in B \Rightarrow a(v) = \hat{a}(\hat{v})$
- $(v, \alpha, w) \in E \wedge (v, \hat{v}) \in B \Rightarrow \exists \hat{w} \in \hat{V} : (w, \hat{w}) \in B \wedge (\hat{v}, \alpha, \hat{w}) \in \hat{E}$
- $(\hat{v}, \alpha, \hat{w}) \in \hat{E} \wedge (v, \hat{v}) \in B \Rightarrow \exists w \in V : (w, \hat{w}) \in B \wedge (v, \alpha, w) \in E$

- Autobisimulation: bisimulation between M and itself
- Bisimulation equivalence: autobisimulation + equivalence
- Existence of a coarsest bisimulation equivalence

Quotients

Definition

Let B be a bisimulation equivalence for $M = (V, E, g, a)$. The *quotient* M/B is the model $(V/B, E/B, g/B, a/B)$, defined as follows:

- $V/B =_{df} \{v/B \mid v \in V\}$
- $(v/B, \alpha, w/B) \in E/B \Leftrightarrow_{df} \exists v' \in v/B, w' \in v/B : (v', \alpha, w') \in E$
- $(a/B)(v/B) =_{df} a(v)$

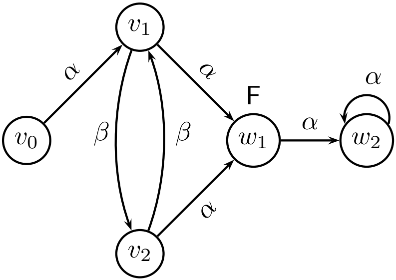
Expansion

Definition

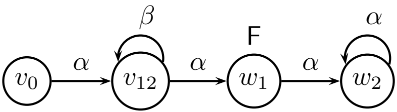
Given a model $M = (V, E, g, a)$, a bisimulation equivalence B for M and a refinement $(M/B)' = ((V/B)', (E/B)', (g/B)', (a/B)')$ of M/B we define the *expansion* $(M/B)' \backslash B = ((V/B)' \backslash B, (E/B)' \backslash B, (g/B)' \backslash B, (a/B)' \backslash B)$ as follows:

- $(V/B)' \backslash B = V$
- $(v, \alpha, w) \in (E/B)' \backslash B \Leftrightarrow (v, \alpha, w) \in E \wedge (v/B, \alpha, w/B) \in (E/B)'$
- $((a/B)' \backslash B)(v) = a(v)$

Quotient

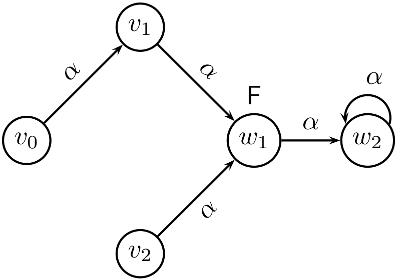


v_1
α
α
F
α
v_0
β
β
w_1
α
w_2
α
v_2

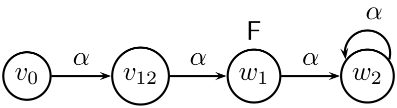


β
F
α
v_0
α
v_{12}
α
w_1
α
w_2

Expansion



v_1
α
α
F
α
v_0
w_1
α
w_2
α
v_2



α
F
v_0
α
v_{12}
α
w_1
α
w_2

Compatibility

Definition

Let φ be an LTL formula. We say that φ is *quotient compatible with respect to refinement (control)* if for all plant automata M and all bisimulation quotients M/B of M the equivalence

$$\begin{aligned} &M \text{ is refineable (controllable) wrt. } \varphi \Leftrightarrow \\ &M/B \text{ is refineable (controllable) wrt. } \varphi \end{aligned}$$

holds.

Refineability of M/B implies refineability of M by bisimilarity of $(M/B)'$ and $(M/B)' \backslash B$, analogously for controllability

Simple Case: F

DLR

Lemma

F *is quotient compatible with respect to refinement.*

Proof:

- consider refinement M' satisfying F
- pick arbitrary infinite trace $p = v_0\alpha_0 v_1\alpha_1 \dots$ in M'
- define $(M/B)'$ by:
  - $(V/B)' =_{df} V/B$
  - $(a/B)' =_{df} a/B$
  - $(v/B, \alpha, w/B) \in (E/B)' \Leftrightarrow_{df} \exists i : v \in v_i/B \wedge w \in v_{i+1}/B \wedge \alpha = \alpha_i$
- check properties (refinement, liveness, satisfaction)

Some More Interesting Cases

Lemma

$\bigcirc F$ *is quotient compatible with respect to refinement.*

Proof: similar to the previous case, $v_0/B = \{v_0\}$ makes live easy

Lemma

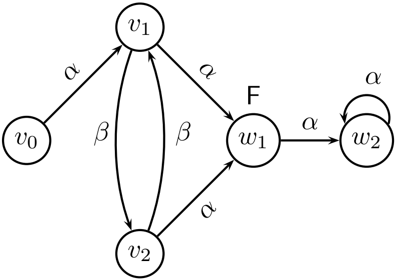
$\bigcirc\bigcirc F$ *is quotient compatible with respect to refinement.*

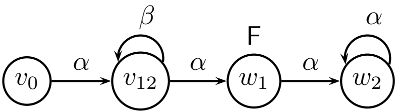
Proof: inconvenient, tedious case distinctions ($v_0 = v_2$, $v_1/B = v_2/B$, ...)

Lemma

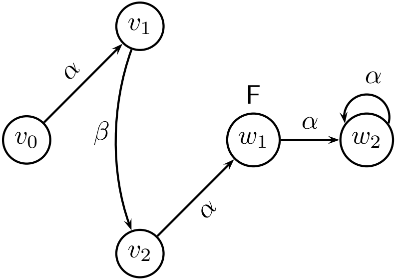
$\bigcirc\bigcirc\bigcirc F$ *is not quotient compatible with respect to refinement.*

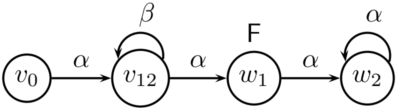
Refining for $\bigcirc\bigcirc\bigcirc F$





Refining for $\bigcirc \bigcirc \bigcirc F$





Two other examples

Lemma

F$\mathcal{U}$G *is quotient compatible with respect to refinement.*

Proof: consider trace in M, mind to remove cycles in the quotient

Lemma

$\bigcirc^i\Box$F *is quotient compatible with respect to refinement.*

Proof: similar to above, premature arrival at F doesn't hurt

Decidability and Computation

Ideas:

- look for deterministic refinements
- liveness has to be ensured
- use strongly connected components (SCC)
- every trace has to be trapped in an SCC (in the finite case)

Example with Proofs

Lemma

It can be decided in $\mathcal{O}(|V| + |E|)$ time whether M can be refined with respect to F*. A corresponding refinement can also be computed in $\mathcal{O}(|V| + |E|)$ time.*

Proof: test whether $\mathsf{F} \in a(v_0)$ holds and whether an SCC is reachable from v_0

Lemma

It can be decided in $\mathcal{O}(|V| + |E|)$ time whether M can be refined with respect to $\bigcirc$F*. A corresponding refinement can also be computed in $\mathcal{O}(|V| + |E|)$ time.*

Proof:

- $\mathsf{F} \in a(v_0)$ and $(v_0, v_0) \in E$ is obvious
- remove loop from v_0 and all edges (v_0, v_i) with $\mathsf{F} \notin a(v_i)$
- look for reachable SCCs in the emerging graph

Examples without Proofs

Lemma

For every formula φ of the following list it can be decided in $\mathcal{O}(|V| + |E|)$ time whether M can be refined with respect to φ. A corresponding refinement can also be computed in $\mathcal{O}(|V| + |E|)$ time.

- $\bigcirc \bigcirc \mathsf{F}$
- $\bigcirc^i \Box \mathsf{F}$ *for every i*
- $\Diamond \mathsf{F}$ *and* $\Box \mathsf{F}$
- $\Diamond \Box \mathsf{F}$ *and* $\Box \Diamond \mathsf{F}$
- $\mathsf{F} \mathcal{U} \mathsf{G}$

Running time and Controllers

- all examples till now have linear running time
- computation of coarsest quotient needs $\mathcal{O}(|E| + \log(|V|))$ time
- no speed-up using quotients for refinement (in considered cases)
- similar results for control
- also here dichotomy between $\bigcirc\bigcirc$F and $\bigcirc\bigcirc\bigcirc$F
- controlling with respect to $\Diamond$F and $\Box$F in $\mathcal{O}(|V|^2)$ time
- controlling with respect to $\Diamond\Box$F and $\Box\Diamond$F in $\mathcal{O}(|V|^3)$ time
- speed-up possible

Outlook and Further Research

- exact complexity of refinement (model checking is PSPACE-complete)
- consider more complex formulae (more variables)
- general criterion for compatibility
- explain gap between $\bigcirc\bigcirc F$ and $\bigcirc\bigcirc\bigcirc F$
- search for general/optimal refining/controlling algorithms

Questions?