Ready to Preorder

Get Your BCCSP Axiomatization for Free!

Luca Aceto[1], Wan Fokkink[2] and Anna Ingolfsdottir[1]

[1]Reykjavik University

[2]Vrije Universiteit Amsterdam

CALCO 2007, Bergen, 24 August 2007

Thanks to The Icelandic Research Fund for partial financial support.

The Role of Equalities Between Programs

Motto: In Computer Science, we use formal languages to communicate with machines and describe expected properties of computations.

Fact of Life: We often need to know when two syntactically different descriptions are describing the "same thing". Examples?

- Optimization in compilers.
- Program analysis/partial evaluation...
- Correctness: Is SPECification equivalent to IMPlementation?

Tenet: Equational logic can be used to capture "valid" equivalences.

(Finite) Complete Axiomatizations

The Challenge

Given some algebraic signature Σ, and some congruence $\sim$ over (closed) terms

Is there a finite set $\mathcal{E}$ of Σ-equations $s = t$ such that

$$t \sim u \quad \Leftrightarrow \quad \mathcal{E} \vdash t = u$$

for all (closed) Σ-terms t, u?

$\mathcal{E}$ is called a sound and (ground) complete axiomatization.

Why is This an Interesting Game?

Answer 1

The axiomatic method is a very powerful method of scientific analysis, so studying its power in Computer Science must be interesting!. . . And I like it!

Answer 2

An equational axiomatization

1. tells ye all ye need to know about your notion of program equivalence;
2. allows you to relate it to other types of program equivalence by simply looking at laws;
3. may form the basis for program verification tools based on theorem proving technology.

Why is This an Interesting Game?

Answer 1

The axiomatic method is a very powerful method of scientific analysis, so studying its power in Computer Science must be interesting!. . . And I like it!

Answer 2

An equational axiomatization

1. tells ye all ye need to know about your notion of program equivalence;
2. allows you to relate it to other types of program equivalence by simply looking at laws;
3. may form the basis for program verification tools based on theorem proving technology.

A Core Language: Basic CCSP

The Language

BCCSP nil $\mathbf{0}$ prefixing $a.t$
choice $t + u$ variables x

where a is an action drawn from a non-empty set A.

Its (Operational) Semantics

Given by transitions between terms of the form $t \xrightarrow{a} u$. These associate a loop-free finite automaton with each term. How?

$$\frac{}{ax \xrightarrow{a} x} \qquad \frac{x \xrightarrow{a} x'}{x + y \xrightarrow{a} x'} \qquad \frac{y \xrightarrow{a} y'}{x + y \xrightarrow{a} y'}$$

A Core Language: Basic CCSP

The Language

BCCSP nil $\mathbf{0}$ prefixing $a.t$
choice $t + u$ variables x

where a is an action drawn from a non-empty set A.

Its (Operational) Semantics

Given by transitions between terms of the form $t \xrightarrow{a} u$. These associate a loop-free finite automaton with each term. How?

$$\frac{}{ax \xrightarrow{a} x} \qquad \frac{x \xrightarrow{a} x'}{x + y \xrightarrow{a} x'} \qquad \frac{y \xrightarrow{a} y'}{x + y \xrightarrow{a} y'}$$

When Are Two BCCSP Processes Equivalent?

Answer of Programming Language Semantics

Two programs are "equivalent" when you can use one for the other in any program context without affecting the overall behaviour.

1. $\mathcal{O}(p)$, observations for p.
2. Programs p and q are equivalent iff $\mathcal{O}(C[p]) = \mathcal{O}(C[q])$ for each program context $C[\,]$.

Question: What is a "reasonable" notion of observation for concurrent processes? How many are there?

Lots, alas. . .

When Are Two BCCSP Processes Equivalent?

Answer of Programming Language Semantics

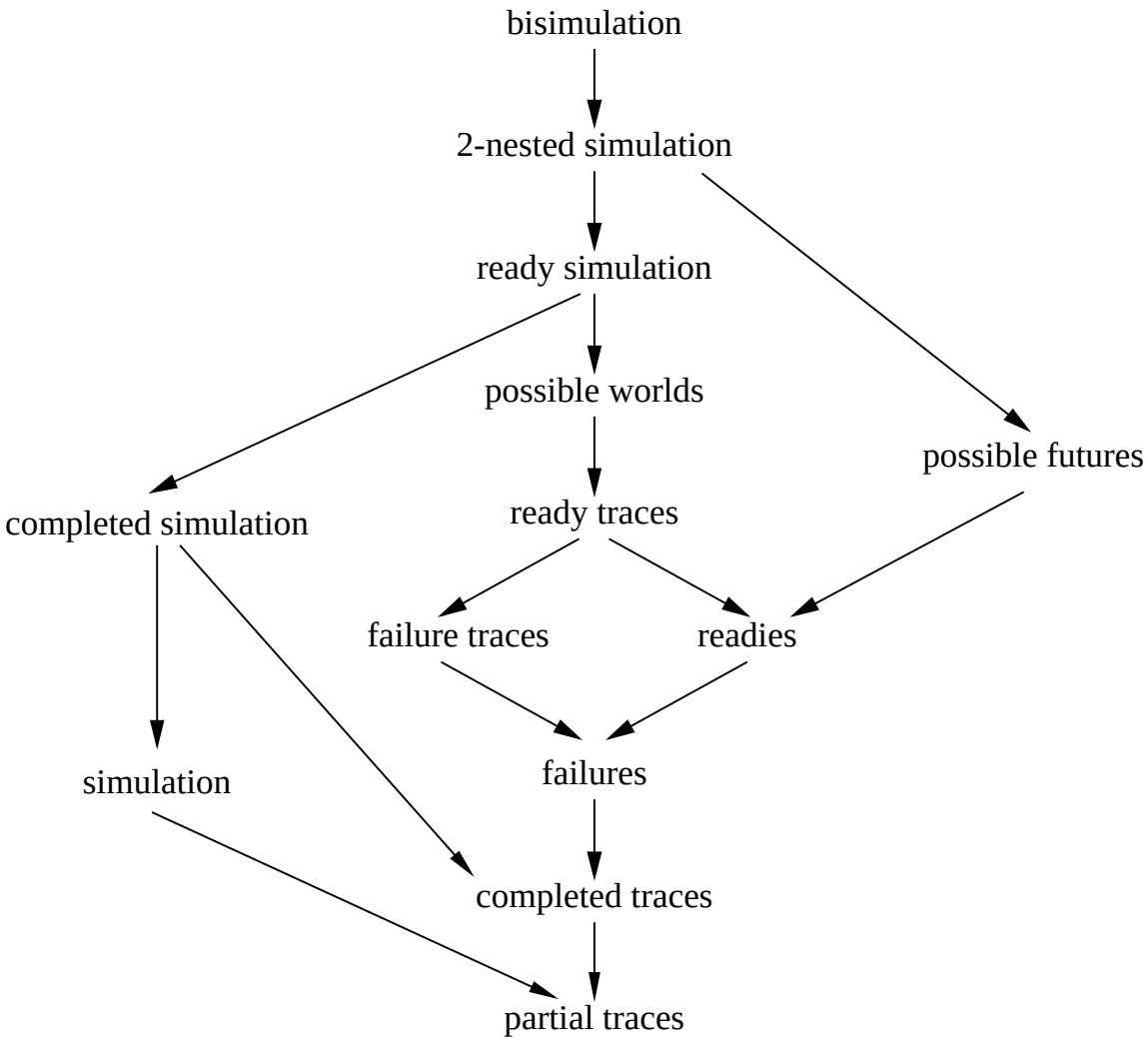
Two programs are "equivalent" when you can use one for the other in any program context without affecting the overall behaviour.

1. $\mathcal{O}(p)$, observations for p.
2. Programs p and q are equivalent iff $\mathcal{O}(C[p]) = \mathcal{O}(C[q])$ for each program context $C[\,]$.

Question: What is a "reasonable" notion of observation for concurrent processes? How many are there?

Lots, alas. . .

van Glabbeek's Spectrum



The Fragment of Interest in the Spectrum (van Glabbeek)

We focus on the ten semantics in the spectrum that include ready simulation and are included in partial traces.
This excludes bisimulation, 2-nested simulation and possible futures.

From Preorders to Equivalences

The semantics we consider are naturally defined in terms of preorders. Processes p and q are equivalent iff $p \precsim q$ and $q \precsim p$.

Overview of Known Results

	$\lvert A\rvert = 1$	$1 < \lvert A\rvert < \infty$	$\lvert A\rvert = \infty$
bisim	+	+	+
2-nes sim	−	−	−
poss futures	−	−	−
ready sim	+	−	+
sim	+	−	+
poss worlds	+	−	+
ready traces	+	−	−
failure traces	+	−	+
ready	+	−	+
failure	+	+	+
compl traces	+	+	+
part traces	+	+	+

Groovy! But, the proofs of the positive results for each preorder and its induced equivalence are different, and yet follow similar lines.

Overview of Known Results

	$\lvert A\rvert = 1$	$1 < \lvert A\rvert < \infty$	$\lvert A\rvert = \infty$
bisim	+	+	+
2-nes sim	−	−	−
poss futures	−	−	−
ready sim	+	−	+
sim	+	−	+
poss worlds	+	−	+
ready traces	+	−	−
failure traces	+	−	+
ready	+	−	+
failure	+	+	+
compl traces	+	+	+
part traces	+	+	+

Groovy! But, the proofs of the positive results for each preorder and its induced equivalence are different, and yet follow similar lines.

Our Motivating Question and its Answer

Our Question

Is there a general recipe for turning an axiomatization of a preorder in the spectrum into an axiomatization of its induced equivalence?

Our Answer: Yes!

Our procedure works for all of the semantics in the spectrum lying in between ready simulation and partial traces.
More generally, it works for any precongruence over BCCSP satisfying three criteria we isolate.

Our Motivating Question and its Answer

Our Question

Is there a general recipe for turning an axiomatization of a preorder in the spectrum into an axiomatization of its induced equivalence?

Our Answer: Yes!

Our procedure works for all of the semantics in the spectrum lying in between ready simulation and partial traces.
More generally, it works for any precongruence over BCCSP satisfying three criteria we isolate.

The Recipe

Input: A sound inequational axiomatization E for BCCSP modulo $\precsim$ that includes the four axioms for bisimilarity together with the defining inequational axioms for ready simulation equivalence for each $a \in A$:

$$ax \preccurlyeq ax + ay \quad .$$

Output: The axiomatization $\mathcal{A}(E)$ constructed as follows. The four axioms for bisimilarity are by default included in $\mathcal{A}(E)$. Furthermore, for each inequational axiom $t \preccurlyeq u$ in E, we add to $\mathcal{A}(E)$:

A. $t + u \approx u$; and

B. $b(t + x) + b(u + x) \approx b(u + x)$ (for all $b \in A$, and some x that does not occur in $t + u$).

Main Result: Correctness of the Recipe

Theorem

Let $\precsim$ be a preorder in the spectrum that includes the ready simulation preorder. Let E be a sound and ground-complete inequational axiomatization for BCCSP modulo $\precsim$. Then the equational axiomatization $\mathcal{A}(E)$ is sound and ground-complete for BCCSP modulo $\simeq$. Moreover, if E is ω-complete, then so is $\mathcal{A}(E)$.

The pudding is in the proof!

Step 1: Reduction to Cover Equations

A term has general form $\sum_{i \in I} a_i t_i + \sum_{j \in J} x_j$.

First Observations

1. $t \approx u$ holds iff so do $t + u \approx t$ and $t + u \approx u$.
2. $t_1 + t_2 + u \approx u$ iff $t_1 + u \approx u$ and $t_2 + u \approx u$.

Conclusion: It suffices to show that $\mathcal{A}(E)$ proves all valid equations of the form

$$\begin{aligned} at + u &\approx u \\ x + u &\approx u \ . \end{aligned}$$

These are the cover equations (Fokkink and Nain).

Step 2: Which Cover Equations?

Three Key Lemmas

- Lemma 1. If $t + x \precsim u$, and either $\precsim \subseteq \precsim_{\mathrm{CT}}$, or $\precsim \subseteq \precsim_{\mathrm{PT}}$ and $|A| > 1$, then x is a summand of u.
- Lemma 2. Let $\simeq$ be an equivalence in the spectrum. If $at + u + bv \simeq u + bv$ with $a \neq b$, then $at + u \simeq u$.
- Lemma 3. Let $\precsim$ be a preorder in the spectrum. If $t + x \precsim u + x$, and x is not a summand of $t + u$, then $t \precsim u$. (Lots of work! It takes 19 pages to prove this statement in the technical report.)

Step 3: Summing up

Conclusion

It suffices to show that $\mathcal{A}(E)$ proves all valid cover equations of the form

$$at + \sum_{i=1}^{n} au_i \quad \approx \quad \sum_{i=1}^{n} au_i$$

and, only for the case of partial traces semantics with $|A| = 1$, all sound equations of the form

$$x + u \approx u \ .$$

This we do by induction of the depth of the proof of $at \precsim \sum_{i=1}^{n} au_i$ and $x \precsim u$, respectively. (Non-trivial proof.)

Concluding Remarks

- The recipe produces a complete axiomatization for any precongruence over BCCSP that includes the ready simulation preorder and satisfies Lemmas 1–3.
- de Frutos-Escrig and Gregorio-Rodríguez generate an inequational axiomatization for preorders in the spectrum from equational axiomatizations for the corresponding equivalence.
- Nested simulation and possible futures semantics afford no finite ground-complete axiomatization over BCCSP even in the presence of a single action.
- Based on our work, de Frutos-Escrig recently gave a new proof that does not use the three key lemmas.

Thank You! Any Questions?

Concluding Remarks

- The recipe produces a complete axiomatization for any precongruence over BCCSP that includes the ready simulation preorder and satisfies Lemmas 1–3.
- de Frutos-Escrig and Gregorio-Rodríguez generate an inequational axiomatization for preorders in the spectrum from equational axiomatizations for the corresponding equivalence.
- Nested simulation and possible futures semantics afford no finite ground-complete axiomatization over BCCSP even in the presence of a single action.
- Based on our work, de Frutos-Escrig recently gave a new proof that does not use the three key lemmas.

Thank You! Any Questions?

Shameless Self-Promotion

- Submit your best work and/or workshop proposals to ICALP 2008, 6–13 July 2008, Reykjavik!
- Buy your copy of *Reactive Systems: Modelling, Specification and Verification* (Cambridge University Press) by Luca Aceto, Anna Ingolfsdottir, Kim G. Larsen and Jiri Srba at 20% discount now!