The Saga of the Axiomatization of Parallel Composition

Luca Aceto
Reykjavik University

18th CONCUR, Lisbon, 5 September 2007

Thanks to Taolue Chen, Wan Fokkink, Anna Ingolfsdottir, Bas Luttik, MohammadReza Mousavi and Sumit Nain for our joint work and to the Icelandic Research Fund for partial financial support.

Why This Talk at CONCUR?

My Tenet

(Theoretical) Computer Science and its forefather Mathematical Logic are pretty much unique in their development of mathematical methodology for proving negative results.

Christos Papadimitriou's Viewpoint

Negative results are the only possible self-contained theoretical results in Computer Science. Successful exploratory theoretical research is bound to produce predominantly negative results. (From "Database metatheory: Asking the big queries")

What kind of negative results is this talk about? Why are they of CONCUR interest?

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. In process algebra, the equational characterization of parallel composition is key.

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.

The Cold Shower

Main General Technical Message of the Talk

The life of a concurrency theorist is equationally hard.

In many situations, the collection of valid equivalences cannot be "captured" by means of a finite collection of equations. This holds true even for very simple languages!

Rest of the Talk

- Examples of (mostly) negative results of that type.
- This will give me an excuse to recount the saga of the equational axiomatization of parallel composition in process algebras.

And now for a little technical content!

The Cold Shower

Main General Technical Message of the Talk

The life of a concurrency theorist is equationally hard.

In many situations, the collection of valid equivalences cannot be "captured" by means of a finite collection of equations. This holds true even for very simple languages!

Rest of the Talk

- Examples of (mostly) negative results of that type.
- This will give me an excuse to recount the saga of the equational axiomatization of parallel composition in process algebras.

And now for a little technical content!

The Cold Shower

Main General Technical Message of the Talk

The life of a concurrency theorist is equationally hard.

In many situations, the collection of valid equivalences cannot be "captured" by means of a finite collection of equations. This holds true even for very simple languages!

Rest of the Talk

- Examples of (mostly) negative results of that type.
- This will give me an excuse to recount the saga of the equational axiomatization of parallel composition in process algebras.

And now for a little technical content!

A Core Language: CCS

The Language

CCS nil $\mathbf{0}$ prefixing at variables x
choice $t + u$ parallel $t \| u$

where a is an action drawn from a non-empty, finite set A. We assume A includes τ and complementary actions.

Its (Operational) Semantics (Sample Rules)

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

$$\frac{}{ax \xrightarrow{a} x} \qquad \frac{x \xrightarrow{a} x'}{x + y \xrightarrow{a} x'} \qquad \frac{x \xrightarrow{a} x'}{x \| y \xrightarrow{a} x' \| y} \qquad \frac{x \xrightarrow{a} x',\ y \xrightarrow{\bar{a}} y'}{x \| y \xrightarrow{\tau} x' \| y'}$$

A Core Language: CCS

The Language

CCS nil $\mathbf{0}$ choice $t + u$ prefixing at parallel $t \| u$ variables x

where a is an action drawn from a non-empty, finite set A. We assume A includes τ and complementary actions.

Its (Operational) Semantics (Sample Rules)

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

$$\frac{}{ax \xrightarrow{a} x} \qquad \frac{x \xrightarrow{a} x'}{x + y \xrightarrow{a} x'} \qquad \frac{x \xrightarrow{a} x'}{x \| y \xrightarrow{a} x' \| y} \qquad \frac{x \xrightarrow{a} x', \; y \xrightarrow{\bar{a}} y'}{x \| y \xrightarrow{\tau} x' \| y'}$$

Behavioural Equivalence

Throughout this talk, we consider only strong bisimilarity $\underline{\leftrightarrow}$ (Milner 1980, Park 1981).

Fact: Strong bisimilarity is a congruence over CCS and all of its extensions considered in this talk.

Motivating Question

Is there a (finite) collection of equations (valid with respect to $\underline{\leftrightarrow}$) that allows us to prove all the valid (ground) equivalences modulo $\underline{\leftrightarrow}$ over CCS?

Behavioural Equivalence

Throughout this talk, we consider only strong bisimilarity $\underline{\leftrightarrow}$ (Milner 1980, Park 1981).

Fact: Strong bisimilarity is a congruence over CCS and all of its extensions considered in this talk.

Motivating Question

Is there a (finite) collection of equations (valid with respect to $\underline{\leftrightarrow}$) that allows us to prove all the valid (ground) equivalences modulo $\underline{\leftrightarrow}$ over CCS?

An Axiom System $\mathcal{E}$ for Bisimilarity over CCS

$$\begin{aligned} x + y &= y + x \\ (x + y) + z &= x + (y + z) \\ x + x &= x \\ x + \mathbf{0} &= x \end{aligned}$$

$$\sum_{i \in I} a_i x_i \parallel \sum_{j \in J} b_j y_j = \sum_{i \in I} a_i (x_i \parallel y) + \sum_{j \in J} b_j (x \parallel y_j) + \sum_{i \in I, j \in J, a_i = \overline{b_j}} \tau(x_i \parallel y_j)$$

Soundness & completeness (Hennessy and Milner, circa 1980):
$t \underline{\leftrightarrow} u \Leftrightarrow \mathcal{E} \vdash t = u$, for all ground CCS terms t, u.
Groovy! But, can one obtain a finite axiomatization?

An Axiom System $\mathcal{E}$ for Bisimilarity over CCS

$$
\begin{aligned}
x + y &= y + x \\
(x + y) + z &= x + (y + z) \\
x + x &= x \\
x + \mathbf{0} &= x
\end{aligned}
$$

$$
\sum_{i \in I} a_i x_i \parallel \sum_{j \in J} b_j y_j =
$$

$$
\sum_{i \in I} a_i (x_i \parallel y) + \sum_{j \in J} b_j (x \parallel y_j) + \sum_{i \in I, j \in J, a_i = \overline{b_j}} \tau (x_i \parallel y_j)
$$

Soundness & completeness (Hennessy and Milner, circa 1980):
$t \underline{\leftrightarrow} u \Leftrightarrow \mathcal{E} \vdash t = u$, for all ground CCS terms t, u.
Groovy! But, can one obtain a finite axiomatization?

Bergstra and Klop's Auxiliary Operators

ID Card

Symbols: $⌊⌊$ (left merge), $|$ (communication merge).

Origin: Together for the first time in "Process Algebra for Synchronous Communication", Information & Control 60:109–137, 1984.

Operational Rules:

$$\frac{x \xrightarrow{a} x'}{x ⌊⌊ y \xrightarrow{a} x' \| y} \qquad \frac{x \xrightarrow{a} x', \; y \xrightarrow{\bar{a}} y'}{x \mid y \xrightarrow{\tau} x' \| y'}$$

Relationship with $\|$:

$$x \| y = (x ⌊⌊ y) + (y ⌊⌊ x) + (x \mid y)$$

Why Are $\llfloor$ and $|$ Good?

Theorem (Bergstra and Klop 1984, yours truly et al. ICALP 2006)

Bisimilarity affords a finite complete axiomatization over CCS with left and communication merge!

Key to the above result: $\llfloor$ and $|$ have a nice interplay with $+$, unlike $\|$. We can easily handle processes with large branching degree! (Completeness proof is tricky.)

$$
\begin{aligned}
(x+y) \llfloor z &= (x \llfloor z) + (y \llfloor z) \\
(x+y) \mid z &= (x \mid z) + (y \mid z)
\end{aligned}
$$

Cool! But, are auxiliary operators necessary?

Why Are $\mathbin{||\!\_}$ and $|$ Good?

Theorem (Bergstra and Klop 1984, yours truly et al. ICALP 2006)

Bisimilarity affords a finite complete axiomatization over CCS with left and communication merge!

Key to the above result: $\mathbin{||\!\_}$ and $|$ have a nice interplay with $+$, unlike $\|$. We can easily handle processes with large branching degree! (Completeness proof is tricky.)

$$
\begin{aligned}
(x + y) \mathbin{||\!\_}\, z &= (x \mathbin{||\!\_}\, z) + (y \mathbin{||\!\_}\, z) \\
(x + y) \mid z &= (x \mid z) + (y \mid z)
\end{aligned}
$$

Cool! But, are auxiliary operators necessary?

Auxiliary Operators Are Necessary!

Theorem (Moller)

No "reasonable" congruence affords a finite equational axiomatization over CCS. Bisimilarity is "reasonable".

Proof idea (for bisimilarity): No finite, sound axiom system $\mathcal{E}$ over CCS is powerful enough to "expand" the initial interleaving behaviour of a term of the form $a \| p$ when p has large branching degree.

Implementation: $\mathcal{E}$ cannot prove the sound equation

$$a \| \sum_{i=1}^{n} a^i = a(\sum_{i=1}^{n} a^i) + \sum_{i=2}^{n+1} a^i \qquad (n > \text{size}(\mathcal{E})) \quad .$$

Auxiliary Operators Are Necessary!

Theorem (Moller)

No "reasonable" congruence affords a finite equational axiomatization over CCS. Bisimilarity is "reasonable".

Proof idea (for bisimilarity): No finite, sound axiom system $\mathcal{E}$ over CCS is powerful enough to "expand" the initial interleaving behaviour of a term of the form $a\|p$ when p has large branching degree.

Implementation: $\mathcal{E}$ cannot prove the sound equation

$$a \| \sum_{i=1}^{n} a^i = a(\sum_{i=1}^{n} a^i) + \sum_{i=2}^{n+1} a^i \qquad (n > \text{size}(\mathcal{E})) \ .$$

Intermezzo: Techniques for Proving Negative Results

Technical Problem

How can one prove negative results like Moller's one?

1. Compactness theorem: Find an (infinite) sound and complete axiomatization for which no finite subset is complete.
2. Model-theoretic approach: For each finite sound axiomatization $\mathcal{E}$, find a model for $\mathcal{E}$, and a sound equation that fails in this model.
3. Proof-theoretic approach: For each finite sound axiomatization $\mathcal{E}$, find a property of equations that
   (A) is satisfied by all instantiations of axioms in $\mathcal{E}$,
   (B) is preserved by the rules of equational logic, and
   (C) fails for some sound equation.

Intermezzo: Techniques for Proving Negative Results

Technical Problem

How can one prove negative results like Moller's one?

1. Compactness theorem: Find an (infinite) sound and complete axiomatization for which no finite subset is complete.
2. Model-theoretic approach: For each finite sound axiomatization $\mathcal{E}$, find a model for $\mathcal{E}$, and a sound equation that fails in this model.
3. Proof-theoretic approach: For each finite sound axiomatization $\mathcal{E}$, find a property of equations that
   (A) is satisfied by all instantiations of axioms in $\mathcal{E}$,
   (B) is preserved by the rules of equational logic, and
   (C) fails for some sound equation.

The Positive Message in the Negative Result

Thou shalt add auxiliary operators to CCS!

Yes, but is there any alternative to Bergstra and Klop?

Hennessy's Merge (ID card)

Symbol: $|\!/$

Origin: "On the Relationship Between Time and Interleaving", CMA Preprint, 1981.

Operational Rules:

$$\frac{x \xrightarrow{a} x'}{x \,|\!/\, y \xrightarrow{a} x' \| y} \qquad \frac{x \xrightarrow{a} x', \; y \xrightarrow{\bar{a}} y'}{x \,|\!/\, y \xrightarrow{\tau} x' \| y'}$$

Relationship with $\|$: $x \| y = (x \,|\!/\, y) + (y \,|\!/\, x)$

The Positive Message in the Negative Result

Thou shalt add auxiliary operators to CCS!

Yes, but is there any alternative to Bergstra and Klop?

Hennessy's Merge (ID card)

Symbol: $\not|$

Origin: "On the Relationship Between Time and Interleaving", CMA Preprint, 1981.

Operational Rules:

$$\frac{x \xrightarrow{a} x'}{x \not| y \xrightarrow{a} x' \| y} \qquad \frac{x \xrightarrow{a} x', \; y \xrightarrow{\bar{a}} y'}{x \not| y \xrightarrow{\tau} x' \| y'}$$

Relationship with $\|$: $x \| y = (x \not| y) + (y \not| x)$

A New Theorem (At Last)

Question and an Old Conjecture

Does bisimilarity afford a finite equational axiomatization over CCS with $\lfloor\!\lfloor$?

> *It seems that $\lfloor\!\lfloor$ does not have a finite equational axiomatization. (Bergstra and Klop, 1984, p. 118)*

Theorem (Fokkink, Ingolfsdottir, Luttik and yours truly, 2005)

Bergstra and Klop were right! Bisimilarity affords no finite equational axiomatization over CCS with $\lfloor\!\lfloor$.

The pudding is in the proof!

A New Theorem (At Last)

Question and an Old Conjecture

Does bisimilarity afford a finite equational axiomatization over CCS with $\|$?

> *It seems that $\|$ does not have a finite equational axiomatization. (Bergstra and Klop, 1984, p. 118)*

Theorem (Fokkink, Ingolfsdottir, Luttik and yours truly, 2005)

Bergstra and Klop were right! Bisimilarity affords no finite equational axiomatization over CCS with $\|$.

The pudding is in the proof!

The Proof

Proof idea: No finite, sound axiom system $\mathcal{E}$ over CCS with $\lfloor$ is powerful enough to "expand" the initial synchronization behaviour of a term of the form $a \lfloor p$ when p has large branching degree.

Implementation: We show that $\mathcal{E}$ cannot prove the sound equation

$$a \lfloor \sum_{i=0}^{n} \bar{a}a^i = a(\sum_{i=0}^{n} \bar{a}a^i) + \sum_{i=0}^{n} \tau a^i \qquad (n > \mathrm{size}(\mathcal{E})) \ .$$

Message (anthropomorphically):

Hennessy cannot replace Bergstra and Klop!

Hennessy Strikes Back

What about other "reasonable" congruences over CCS with $|/$?

Theorem (Fokkink, Ingolfsdottir, Luttik and yours truly, 2005)

A non-interleaving equivalence like split-2 bisimilarity affords a finite ground-complete equational axiomatization over CCS with $|/$!

So a single binary operator may suffice to obtain a finite equational axiomatization for non-interleaving equivalences.

Conjecture: There is no single auxiliary binary operator that can be used to axiomatize bisimilarity over CCS.

Hennessy Strikes Back

What about other “reasonable” congruences over CCS with $|/$?

Theorem (Fokkink, Ingolfsdottir, Luttik and yours truly, 2005)

A non-interleaving equivalence like split-2 bisimilarity affords a finite ground-complete equational axiomatization over CCS with $|/$!

So a single binary operator may suffice to obtain a finite equational axiomatization for non-interleaving equivalences.

Conjecture: There is no single auxiliary binary operator that can be used to axiomatize bisimilarity over CCS.

TCCS: Adding Time to CCS (à la Wang Yi)

Syntax (d ranges over a "time domain", say the positive reals)

$$P \quad ::= \quad 0 \mid \epsilon(d).P \mid aP \mid P + P \mid P \| P$$

No τ or synchronization, for simplicity.

Semantics (SOS)

$$\frac{}{\epsilon(d).x \stackrel{\epsilon(d)}{\longrightarrow} x} \qquad \frac{}{\epsilon(d+e).x \stackrel{\epsilon(d)}{\longrightarrow} \epsilon(e).x} \qquad \frac{x \stackrel{\epsilon(e)}{\longrightarrow} y}{\epsilon(d).x \stackrel{\epsilon(d+e)}{\longrightarrow} y}$$

$$\frac{x_0 \stackrel{\epsilon(d)}{\longrightarrow} y_0 \quad x_1 \stackrel{\epsilon(d)}{\longrightarrow} y_1}{x_0 + x_1 \stackrel{\epsilon(d)}{\longrightarrow} y_0 + y_1} \qquad \frac{x_0 \stackrel{\epsilon(d)}{\longrightarrow} y_0 \quad x_1 \stackrel{\epsilon(d)}{\longrightarrow} y_1}{x_0 \| x_1 \stackrel{\epsilon(d)}{\longrightarrow} y_0 \| y_1}$$

TCCS: First Negative Result

Motivating Question (Reprise)

Is there a (finite) collection of equations (valid with respect to $\underline{\leftrightarrow}$) that allows us to prove all the valid (ground) equivalences modulo $\underline{\leftrightarrow}$ over TCCS?

Theorem (Ingolfsdottir, Mousavi and yours truly, 2007)

No! Moreover this holds true even if we add $\|\!\!\lfloor$ to TCCS.

Luca's demon asks: Sure, $\|\!\!\lfloor$ is time insensitive! Can one do better?

TCCS: First Negative Result

Motivating Question (Reprise)

Is there a (finite) collection of equations (valid with respect to $\underline{\leftrightarrow}$) that allows us to prove all the valid (ground) equivalences modulo $\underline{\leftrightarrow}$ over TCCS?

Theorem (Ingolfsdottir, Mousavi and yours truly, 2007)

No! Moreover this holds true even if we add $\|\!\!\lfloor$ to TCCS.

Luca's demon asks: Sure, $\|\!\!\lfloor$ is time insensitive! Can one do better?

TCCS: Bergstra and Klop Do Not Help (in Time)

Giving Bergstra & Klop a Sense of Time:

$$\frac{x_0 \stackrel{\epsilon(d)}{\longrightarrow} y_0 \quad x_1 \stackrel{\epsilon(d)}{\longrightarrow} y_1}{x_0 \mathbin{\|\!\lfloor} x_1 \stackrel{\epsilon(d)}{\longrightarrow} y_0 \mathbin{\|\!\lfloor} y_1}$$

Theorem: The above operator yields no finite axiomatization for TCCS!

- $a.x \mathbin{\|\!\lfloor} y = a.(x \parallel y)$ does not hold: $a \mathbin{\|\!\lfloor} \epsilon(d).a \stackrel{?}{=} a.(\epsilon(d).a)$.
- We show that no finite, sound $\mathcal{E}$ can prove

$$\mathcal{E} \vdash a \mathbin{\|\!\lfloor} \sum_{i=1}^{n} a.a^{\leq i} = a.(\sum_{i=1}^{n} a.a^{\leq i}) \quad (n > \text{size}(\mathcal{E}))$$

Main Message: Characterizing time insensitivity is equationally hard!

TCCS: Bergstra and Klop Do Not Help (in Time)

Giving Bergstra & Klop a Sense of Time:

$$\frac{x_0 \stackrel{\epsilon(d)}{\longrightarrow} y_0 \quad x_1 \stackrel{\epsilon(d)}{\longrightarrow} y_1}{x_0 \mathbin{|\!|\!\lfloor} x_1 \stackrel{\epsilon(d)}{\longrightarrow} y_0 \mathbin{|\!|\!\lfloor} y_1}$$

Theorem: The above operator yields no finite axiomatization for TCCS!

- $a.x \mathbin{|\!|\!\lfloor} y = a.(x \parallel y)$ does not hold: $a \mathbin{|\!|\!\lfloor} \epsilon(d).a \stackrel{?}{=} a.(\epsilon(d).a)$.
- We show that no finite, sound $\mathcal{E}$ can prove

$$\mathcal{E} \vdash a \mathbin{|\!|\!\lfloor} \sum_{i=1}^{n} a.a^{\leq i} = a.(\sum_{i=1}^{n} a.a^{\leq i}) \quad (n > \mathrm{size}(\mathcal{E}))$$

Main Message: Characterizing time insensitivity is equationally hard!

TCCS/TACS: Discrete Time and Bisimulations on Speed

Question

What about discrete-time TCCS? Faster-than preorders à la Moller-Tofts? Bisimulations on speed for Lüttgen's and Vogler's TACS?

Breaking News!

None of those affords a finite (in-)equational axiomatization.

We are developing a reduction-based proof technique that allows us to bootstrap non-finite axiomatizability proofs.

There's method in the madness! But this will be the subject for another talk. . .

TCCS/TACS: Discrete Time and Bisimulations on Speed

Question

What about discrete-time TCCS? Faster-than preorders à la Moller-Tofts? Bisimulations on speed for Lüttgen's and Vogler's TACS?

Breaking News!

None of those affords a finite (in-)equational axiomatization.

We are developing a reduction-based proof technique that allows us to bootstrap non-finite axiomatizability proofs.

There's method in the madness! But this will be the subject for another talk...

A Menagerie of Nasty Operations

Question

Is parallel composition the only operator that spoils finite axiomatizability?

No way!

The bestiary of "nasty operations" includes

- interrupt (Fokkink, Ingolfsdottir, Nain and yours truly),
- priority (Chen, Fokkink, Ingolfsdottir and yours truly),
- Kleene star and finite-state recursion (Sewell).

The life of a concurrency theorist is equationally hard, indeed...

A Menagerie of Nasty Operations

Question

Is parallel composition the only operator that spoils finite axiomatizability?

No way!

The bestiary of "nasty operations" includes

- interrupt (Fokkink, Ingolfsdottir, Nain and yours truly),
- priority (Chen, Fokkink, Ingolfsdottir and yours truly),
- Kleene star and finite-state recursion (Sewell).

The life of a concurrency theorist is equationally hard, indeed...

A Menagerie of Nasty Operations

Question

Is parallel composition the only operator that spoils finite axiomatizability?

No way!

The bestiary of "nasty operations" includes

- interrupt (Fokkink, Ingolfsdottir, Nain and yours truly),
- priority (Chen, Fokkink, Ingolfsdottir and yours truly),
- Kleene star and finite-state recursion (Sewell).

The life of a concurrency theorist is equationally hard, indeed...

Some Open Problems

- Can parallel composition be finitely axiomatized using a single auxiliary operator replacing Bergstra and Klop's left and communication merge?
- Find general sufficient conditions ensuring finite axiomatizability of bisimilarity over process algebras.
- What about negative results for congruences "abstracting from internal steps" in process behaviour? (Listen to the next talk!)
- What about other semantic equivalences? $\ldots \longrightarrow \infty$

A Pearl of Wisdom from Giorgio Parisi

The attraction of a scientific field depends a lot on fashion and on the story-telling ability of its expositors. In reality, each field has its own interesting and difficult problems, which are an intellectual challenge that may stimulate the interest of curious observers. (From *La Chiave, La Luce e L'Ubriaco*, p. 10, Di Renzo Editore)

Thank You!

Any Questions?

A Pearl of Wisdom from Giorgio Parisi

The attraction of a scientific field depends a lot on fashion and on the story-telling ability of its expositors. In reality, each field has its own interesting and difficult problems, which are an intellectual challenge that may stimulate the interest of curious observers. (From *La Chiave, La Luce e L'Ubriaco*, p. 10, Di Renzo Editore)

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!