Distributive Laws and Decidable Properties of SOS Specifications

Bartek Klin and Beata Nachyła
University of Warsaw

IFIP-WG 1.3 meeting, Sinaia, 02-03/09/2014

SOS

Streams: $\sigma \in A^\omega$

Stream systems: $\langle X,\ h : X \to A \times X \rangle$

SOS

Streams: $\sigma \in A^\omega$

Stream systems: $\langle X,\ h : X \to A \times X \rangle$

Stream SOS:

$$\frac{x \xrightarrow{a} x' \quad y \xrightarrow{b} y'}{\mathtt{alt}(x,y) \xrightarrow{a} \mathtt{alt}(y',x')}$$

$$\mathtt{alt}(\mathtt{a}_1\mathtt{a}_2\mathtt{a}_3\cdots,\mathtt{b}_1\mathtt{b}_2\mathtt{b}_3\cdots) = \mathtt{a}_1\mathtt{b}_2\mathtt{a}_3\mathtt{b}_4\cdots$$

SOS

Streams: $\sigma \in A^\omega$

Stream systems: $\langle X,\ h : X \to A \times X \rangle$

Stream SOS:

$$\frac{x \xrightarrow{a} x' \quad y \xrightarrow{b} y'}{\mathtt{alt}(x, y) \xrightarrow{a} \mathtt{alt}(y', x')}$$

$$\mathtt{alt}(\mathtt{a}_1\mathtt{a}_2\mathtt{a}_3 \cdots, \mathtt{b}_1\mathtt{b}_2\mathtt{b}_3 \cdots) = \mathtt{a}_1\mathtt{b}_2\mathtt{a}_3\mathtt{b}_4 \cdots$$

Defines:

- operations on streams
- a stream system on terms

GSOS

Things can go wrong:

$$\frac{}{\mathtt{C} \xrightarrow{a} \mathtt{q}(\mathtt{C})} \qquad \frac{x \xrightarrow{a} x' \xrightarrow{b} x''}{\mathtt{q}(x) \xrightarrow{b} \mathtt{q}(x'')} \ (\text{for } a, b \in A)$$

this does not induce a system.

GSOS:

$$\frac{x_1 \xrightarrow{a_{1,1}} y_{1,1} \quad x_1 \xrightarrow{a_{1,2}} y_{1,2} \quad \cdots \quad x_i \xrightarrow{a_{i,j}} y_{i,j} \quad \cdots \quad x_i \stackrel{b_{i,j}}{\not\longrightarrow} \quad \cdots}{\mathtt{f}(x_1, \ldots, x_k) \xrightarrow{b} \mathtt{t}}$$

Algebras and coalgebras

Terms form a Σ-algebra:

$$\Sigma X \to X \qquad\qquad \Sigma X = X^{r_1} + X^{r_2} + \cdots + X^{r_n}$$

Algebras and coalgebras

Terms form a Σ-algebra:

$$\Sigma X \to X \qquad\qquad \Sigma X = X^{r_1} + X^{r_2} + \cdots + X^{r_n}$$

Stream systems are F-coalgebras:

$$X \to FX \qquad\qquad FX = A \times X$$

Algebras and coalgebras

Terms form a Σ-algebra:

$$\Sigma X \to X \qquad\qquad \Sigma X = X^{r_1} + X^{r_2} + \cdots + X^{r_n}$$

Stream systems are F-coalgebras:

$$X \to FX \qquad\qquad FX = A \times X$$

Rules may form a distributive law of Σ over F:

$$\frac{x \xrightarrow{a} x' \quad y \xrightarrow{b} y'}{\texttt{alt}(x, y) \xrightarrow{a} \texttt{alt}(y', x')} \qquad\qquad \lambda : \Sigma F \Longrightarrow F\Sigma$$

Algebras and coalgebras

Terms form a Σ-algebra:

$$\Sigma X \to X \qquad \Sigma X = X^{r_1} + X^{r_2} + \cdots + X^{r_n}$$

Stream systems are F-coalgebras:

$$X \to FX \qquad FX = A \times X$$

Rules may form a distributive law of Σ over F:

$$\frac{x \xrightarrow{a} x' \quad y \xrightarrow{b} y'}{\mathtt{alt}(x, y) \xrightarrow{a} \mathtt{alt}(y', x')} \qquad \lambda : \Sigma F \Longrightarrow F\Sigma$$

Fact: such rules behave well.

Laws, laws, laws...

$$\lambda : \Sigma F \Longrightarrow F\Sigma$$

$$\frac{x \xrightarrow{a} x' \quad y \xrightarrow{b} y'}{\mathtt{alt}(x, y) \xrightarrow{a} \mathtt{alt}(y', x')}$$

Laws, laws, laws...

$$\lambda : \Sigma F \Longrightarrow F\Sigma$$

$$\frac{x \xrightarrow{a} x' \quad y \xrightarrow{b} y'}{\texttt{alt}(x, y) \xrightarrow{a} \texttt{alt}(y', x')}$$

$$\lambda : \Sigma(Id \times F) \Longrightarrow F\Sigma$$

$$\frac{x \xrightarrow{a} x' \quad y \xrightarrow{b} y'}{\texttt{zip}(x, y) \xrightarrow{a} \texttt{zip}(y, x')}$$

Laws, laws, laws...

$$\lambda : \Sigma F \Longrightarrow F\Sigma$$

$$\frac{x \xrightarrow{a} x' \quad y \xrightarrow{b} y'}{\texttt{alt}(x,y) \xrightarrow{a} \texttt{alt}(y',x')}$$

$$\lambda : \Sigma(Id \times F) \Longrightarrow F\Sigma$$

$$\frac{x \xrightarrow{a} x' \quad y \xrightarrow{b} y'}{\texttt{zip}(x,y) \xrightarrow{a} \texttt{zip}(y,x')}$$

$$\lambda : \Sigma F \Longrightarrow F(Id + \Sigma)$$

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

Laws, laws, laws...

$$\lambda : \Sigma F \Longrightarrow F\Sigma$$

$$\frac{x \stackrel{a}{\longrightarrow} x' \quad y \stackrel{b}{\longrightarrow} y'}{\texttt{alt}(x,y) \stackrel{a}{\longrightarrow} \texttt{alt}(y',x')}$$

$$\lambda : \Sigma(Id \times F) \Longrightarrow F\Sigma$$

$$\frac{x \stackrel{a}{\longrightarrow} x' \quad y \stackrel{b}{\longrightarrow} y'}{\texttt{zip}(x,y) \stackrel{a}{\longrightarrow} \texttt{zip}(y,x')}$$

$$\lambda : \Sigma F \Longrightarrow F(Id + \Sigma)$$

$$\frac{x \stackrel{a}{\longrightarrow} x'}{x + y \stackrel{a}{\longrightarrow} x'}$$

$$\lambda : \Sigma(Id \times F) \Longrightarrow F(Id + \Sigma)$$

$$\frac{}{a.x \stackrel{a}{\longrightarrow} x}$$

More laws...

GSOS: $\lambda : \Sigma(Id \times F) \Longrightarrow F\Sigma^*$

$$\frac{x_1 \xrightarrow{a_{1,1}} y_{1,1} \quad x_1 \xrightarrow{a_{1,2}} y_{1,2} \quad \cdots \quad x_i \xrightarrow{a_{i,j}} y_{i,j} \quad \cdots \quad x_i \stackrel{b_{i,j}}{\not\longrightarrow} \quad \cdots}{\mathtt{f}(x_1, \ldots, x_k) \xrightarrow{b} \mathtt{t}}$$

More laws...

GSOS: $\lambda : \Sigma(Id \times F) \Longrightarrow F\Sigma^*$

$$\frac{x_1 \xrightarrow{a_{1,1}} y_{1,1} \quad x_1 \xrightarrow{a_{1,2}} y_{1,2} \quad \cdots \quad x_i \xrightarrow{a_{i,j}} y_{i,j} \quad \cdots \quad x_i \stackrel{b_{i,j}}{\not\longrightarrow} \quad \cdots}{\mathtt{f}(x_1, \ldots, x_k) \xrightarrow{b} \mathtt{t}}$$

coGSOS: $\lambda : \Sigma F^\infty \Longrightarrow F(Id + \Sigma)$

$$\frac{x \xrightarrow{a} y \xrightarrow{b} z}{\mathtt{f}(x) \xrightarrow{c} \mathtt{g}(z)}$$

For $FX = A \times X$ we have $F^\infty X = (X \times A)^\omega$

The ultimate laws

Dist. laws of monads over comonads:

$$\lambda : \Sigma^* F^\infty \Longrightarrow F^\infty \Sigma^* \qquad \text{(+ axioms)}$$

The ultimate laws

Dist. laws of monads over comonads:

$$\lambda : \Sigma^* F^\infty \Longrightarrow F^\infty \Sigma^* \qquad \text{(+ axioms)}$$

Fact: these generalize all the other laws.

The ultimate laws

Dist. laws of monads over comonads:

$$\lambda : \Sigma^* F^\infty \Longrightarrow F^\infty \Sigma^* \qquad \text{(+ axioms)}$$

Fact: these generalize all the other laws.

Fact: the abstract theory still works:

- a law induces a unique system
- bisimilarity is a congruence

The ultimate laws

Dist. laws of monads over comonads:

$$\lambda : \Sigma^* F^\infty \Longrightarrow F^\infty \Sigma^* \qquad \text{(+ axioms)}$$

Fact: these generalize all the other laws.

Fact: the abstract theory still works:

- a law induces a unique system
- bisimilarity is a congruence

So: a very general format!

The ultimate laws

Dist. laws of monads over comonads:

$$\lambda : \Sigma^* F^\infty \Longrightarrow F^\infty \Sigma^* \qquad \text{(+ axioms)}$$

Fact: these generalize all the other laws.

Fact: the abstract theory still works:

- a law induces a unique system
- bisimilarity is a congruence

So: a very general format! But what is it?

In search for a format

Naive idea: allow both

- complex conclusions
- lookahead.

In search for a format

Naive idea: allow both

- complex conclusions
- lookahead.

Does not work:

$$\frac{}{\mathtt{C} \xrightarrow{a} \mathtt{q}(\mathtt{C})} \qquad \frac{x \xrightarrow{a} x' \xrightarrow{b} x''}{\mathtt{q}(x) \xrightarrow{b} \mathtt{q}(x'')} \ (\text{for } a, b \in A)$$

In search for a format

Naive idea: allow both

- complex conclusions
- lookahead.

Does not work:

$$\frac{}{\mathtt{C} \xrightarrow{a} \mathtt{q}(\mathtt{C})} \qquad \frac{x \xrightarrow{a} x' \xrightarrow{b} x''}{\mathtt{q}(x) \xrightarrow{b} \mathtt{q}(x'')} \text{ (for } a, b \in A)$$

But perhaps this can be cleverly restricted?

Our result

No.

Our result

No.

There is no expressive format for laws

$$\lambda : \Sigma^* F^\infty \Longrightarrow F^\infty \Sigma^* \qquad \text{(+ axioms)}$$

Our result, actually

There is no expressive format for laws

$$\lambda : \Sigma^* F^\infty \Longrightarrow F^\infty \Sigma^* \qquad \text{(+ axioms)}$$

Our result, actually

For

$$FX = A \times X \qquad \text{(streams)}$$

$$FX = \mathcal{P}(A \times X) \qquad \text{(LTSs)}$$

There is no expressive format for laws

$$\lambda : \Sigma^* F^\infty \Longrightarrow F^\infty \Sigma^* \qquad \text{(+ axioms)}$$

Our result, actually

For

$$FX = A \times X \qquad \text{(streams)}$$

$$FX = \mathcal{P}(A \times X) \qquad \text{(LTSs)}$$

There is no expressive format for laws

$$\lambda : \Sigma^* F^\infty \Longrightarrow F^\infty \Sigma^* \qquad \text{(+ axioms)}$$

Our result, actually

For

$$FX = A \times X \qquad \text{(streams)}$$

$$FX = \mathcal{P}(A \times X) \qquad \text{(LTSs)}$$

There is no expressive format for laws

$$\lambda : \Sigma^* F^\infty \Longrightarrow F^\infty \Sigma^* \qquad \text{(+ axioms)}$$

that would correctly classify mixed-GSOS specs.

Our result, actually

For

$$FX = A \times X \qquad \text{(streams)}$$

$$FX = \mathcal{P}(A \times X) \qquad \text{(LTSs)}$$

There is no expressive format for laws

$$\lambda : \Sigma^* F^\infty \Longrightarrow F^\infty \Sigma^* \qquad \text{(+ axioms)}$$

that would correctly classify mixed-GSOS specs.

(mixed-GSOS: GSOS and coGSOS rules allowed.)

Our result, actually

For

$$FX = A \times X \qquad \text{(streams)}$$

$$FX = \mathcal{P}(A \times X) \qquad \text{(LTSs)}$$

There is no format for laws

$$\lambda : \Sigma^* F^\infty \Longrightarrow F^\infty \Sigma^* \qquad \text{(+ axioms)}$$

that would correctly classify mixed-GSOS specs.

(mixed-GSOS: GSOS and coGSOS rules allowed.)

Our result, actually

There is no format for laws

$$\lambda : \Sigma^* F^\infty \Longrightarrow F^\infty \Sigma^* \qquad \text{(+ axioms)}$$

that would correctly classify mixed-GSOS specs.

Our result, actually

There is no format for laws

$$\lambda : \Sigma^* F^\infty \Longrightarrow F^\infty \Sigma^* \qquad \text{(+ axioms)}$$

that would correctly classify mixed-GSOS specs.

Our result, actually

There is no format for laws

$$\lambda : \Sigma^* F^\infty \Longrightarrow F^\infty \Sigma^* \qquad \text{(+ axioms)}$$

that would correctly classify mixed-GSOS specs.

Format = decidable property of specs.

Our result, actually

It is undecidable whether a mixed-GSOS spec induces a law

$$\lambda : \Sigma^* F^\infty \Longrightarrow F^\infty \Sigma^* \qquad \text{(+ axioms)}$$

Our result, actually

It is undecidable whether a mixed-GSOS spec **induces** a law

$$\lambda : \Sigma^* F^\infty \Longrightarrow F^\infty \Sigma^* \qquad \text{(+ axioms)}$$

Our result, actually

It is undecidable whether a mixed-GSOS spec induces a law

$$\lambda : \Sigma^* F^\infty \Longrightarrow F^\infty \Sigma^* \qquad \text{(+ axioms)}$$

Fact: every mixed-GSOS spec induces a law

$$\rho : \Sigma F^\infty \Longrightarrow F \Sigma^* \qquad \text{(no axioms)}$$

Our result, actually

It is undecidable whether a mixed-GSOS spec induces a law

$$\lambda : \Sigma^* F^\infty \Longrightarrow F^\infty \Sigma^* \qquad \text{(+ axioms)}$$

Fact: every mixed-GSOS spec induces a law

$$\rho : \Sigma F^\infty \Longrightarrow F \Sigma^* \qquad \text{(no axioms)}$$

Def: λ extends ρ :

$$\begin{array}{ccc} \Sigma F^\infty & \overset{\rho}{\Longrightarrow} & F\Sigma^* \\ {\scriptstyle \iota F^\infty}\Downarrow & & \Uparrow{\scriptstyle \pi \Sigma^*} \\ \Sigma^* F^\infty & \underset{\lambda}{\Longrightarrow} & F^\infty \Sigma^* \end{array}$$

Our result, actually

Theorem:

It is undecidable whether a mixed-GSOS spec extends to a unique law

$$\lambda : \Sigma^* F^\infty \Longrightarrow F^\infty \Sigma^* \qquad \text{(+ axioms)}$$

Our result, actually

Theorem:

It is undecidable whether a mixed-GSOS spec extends to a unique law

$$\lambda : \Sigma^* F^\infty \Longrightarrow F^\infty \Sigma^* \qquad \text{(+ axioms)}$$

Proof:

reduction from halting problem of queue machines.

Queue machines

Like pushdown automata, but:

- with a queue instead of a stack
- input word initially stored in the queue

Queue machines

Like pushdown automata, but:

- with a queue instead of a stack
- input word initially stored in the queue

Our variant: in each step,

- take 0, 1 or 2 letters
- put exactly 1 letter,

halt by taking 2 letters where there is 1.

Queue machines

Like pushdown automata, but:

- with a queue instead of a stack
- input word initially stored in the queue

$$\delta_0 : Q \to A \times Q$$

Our variant: in each step,

- take 0, 1 or 2 letters
- put exactly 1 letter,

$$\delta_1 : Q \times A \to A \times Q$$

$$\delta_2 : Q \times A^2 \to A \times Q$$

halt by taking 2 letters where there is 1.

Queue machines

Like pushdown automata, but:

- with a queue instead of a stack
- input word initially stored in the queue

$$\delta_0 : Q \to A \times Q$$
$$\delta_1 : Q \times A \to A \times Q$$
$$\delta_2 : Q \times A^2 \to A \times Q$$

Our variant: in each step,

- take 0, 1 or 2 letters
- put exactly 1 letter,

halt by taking 2 letters where there is 1.

Fact: halting problem undecidable.

From a QM to SOS

Take a QM $\langle Q, A, q_0, \delta_0, \delta_1, \delta_2 \rangle$.

From a QM to SOS

Take a QM $\langle Q, A, q_0, \delta_0, \delta_1, \delta_2 \rangle$.

Syntax Σ: constant C, unary q for each $q \in Q$.

From a QM to SOS

Take a QM $\langle Q, A, q_0, \delta_0, \delta_1, \delta_2 \rangle$.

Syntax Σ: constant C, unary q for each $q \in Q$.

$$\frac{}{C \xrightarrow{a_0} q_0(C)}$$

$$\frac{}{q(x) \xrightarrow{c} q'(x)}$$

(if $\delta_0(q) = (c, q')$)

$$\frac{x \xrightarrow{a} y}{q(x) \xrightarrow{c} q'(y)}$$

(if $\delta_1(q, a) = (c, q')$)

$$\frac{x \xrightarrow{a} y \xrightarrow{b} z}{q(x) \xrightarrow{c} q'(z)}$$

(if $\delta_2(q, a, b) = (c, q')$)

From a QM to SOS

Fact:

$$\text{the QM halts} \Updownarrow \text{the SOS does not extend to a law.}$$

From a QM to SOS

Fact:

the QM halts

$$\Updownarrow$$

the SOS does not extend to a law.

Key step:

finite runs of the QM

$$\parallel$$

initial segments of $\lambda_X(C)$

for each λ that extends the SOS.

The case of LTSs

Syntax Σ: constant C, unary q for each $q \in Q$.

$$\frac{}{C \xrightarrow{a_0} q_0(C)}$$

$$\frac{}{q(x) \xrightarrow{c} q'(x)}$$

(if $\delta_0(q) = (c, q')$)

$$\frac{x \xrightarrow{a} y}{q(x) \xrightarrow{c} q'(y)}$$

(if $\delta_1(q, a) = (c, q')$)

$$\frac{x \xrightarrow{a} y \xrightarrow{b} z}{q(x) \xrightarrow{c} q'(z)}$$

(if $\delta_2(q, a, b) = (c, q')$)

The case of LTSs

Syntax Σ: constant C, unary q for each $q \in Q$.

$$\frac{}{C \xrightarrow{a_0} q_0(C)}$$

$$\frac{}{q(x) \xrightarrow{c} q'(x)}$$

(if $\delta_0(q) = (c, q')$)

$$\frac{x \xrightarrow{a} y}{q(x) \xrightarrow{c} q'(y)}$$

(if $\delta_1(q, a) = (c, q')$)

$$\frac{x \xrightarrow{a} y \xrightarrow{b} z}{q(x) \xrightarrow{c} q'(z)}$$

(if $\delta_2(q, a, b) = (c, q')$)

$$\frac{x \xrightarrow{a} y \quad y \not\longrightarrow}{q(x) \xrightarrow{a} q(x)}$$

Conclusions

1. Distributive laws of monads over comonads:
   - a beautiful notion, but
   - will not give us a common generalization of GSOS and coGSOS.

Conclusions

1. Distributive laws of monads over comonads:
   - a beautiful notion, but
   - will not give us a common generalization of GSOS and coGSOS.

2. Other undecidable problems (for LTSs):
   - existence of a supported model
   - existence of a stable model

Conclusions

1. Distributive laws of monads over comonads:
   - a beautiful notion, but
   - will not give us a common generalization of GSOS and coGSOS.

2. Other undecidable problems (for LTSs):
   - existence of a supported model
   - existence of a stable model

3. Also related to work on stream productivity.