Algebraic Frameworks for Probabilistic and Concurrent Systems

Tahiry Rabehaja

Supervisor: A/Prof Annabelle McIver

Department of Computing
Macquarie University

June 13, 2012
Why algebra?

- Formal modelling: understanding how to design correct computer systems.
- Formal verification: prove correctness mathematically.
Why algebra?

- Formal modelling: understanding how to design correct computer systems.
- Formal verification: prove correctness mathematically.
- Algebra of programs: programs are mathematical object with their own theory.
- Algebras abstract complex interaction: more centred on structural properties.
- Algebras have simple and elegant proof systems.
- Model of executions in a first-order system: automated correctness proofs.
  → Study the algebras of probabilistic and concurrent systems.
A Simple Example

Assume a Probabilistic Vending Machine $M$:

- accept a coin
- flip a fair coin
- enable tea if head
- enable coffee if tail

Assume a user $U$ who wants tea:

- insert a coin
- choose tea (if enabled)
A Simple Example

The system:

\[ U \] run “concurrently” with \[ M. \]

The property:

\[ U \] drinks tea with “probability at least” \[ 1/2. \]

Goal:

Show that the system satisfies the property using algebras.
A Simple Example

The system:

\( U \) run “concurrently” with \( M \).

The property:

\( U \) drinks tea with “probability at least” \( \frac{1}{2} \).

Goal:

Show that the system satisfies the property using algebras.

Tools (algebraic):

- probabilistic Kleene algebra: No concurrency.
- concurrent Kleene algebra: No probability.

Algebra that captures probability and concurrency?
Nondeterminism

- Nondeterminism $+:$
  - unpredictable and “unquantifiable” choice,
  - can be used to model conditional in presence of guards.
- ex:

  $\tau_h \cdot tea + \tau_t \cdot coffee$

- where $\cdot$ is sequential execution,
- and $\tau_h$ and $\tau_t$ are internal actions and act as guards.
Nondeterminism: Algebraic Properties

- Usual properties of choice operator:
  - idempotence: \( x + x = x \),
  - commutativity: \( x + y = y + x \),
  - associativity: \( x + (y + z) = (x + y) + z \),
  - ...

- Interaction with other operators:
  - distribution of sequential:
    - \( x \cdot (y + z) = x \cdot y + x \cdot z \)
    - \( (x + y) \cdot z = x \cdot z + y \cdot z \)
Probability

Probabilistic choice: unpredictable but quantifiable choice.

- Explicit: From a state $s$ do an action $a$ and go to a distribution of states:

  $$s \xrightarrow{a} \frac{1}{2} \delta_{s_1} + \frac{1}{2} \delta_{s_2}$$

- Implicit: From a state $s$ do a probabilistic action:

  $$s \xrightarrow{\text{flip}_{\frac{1}{2}}} s_1$$

- ex:

  $$\text{flip}_{\frac{1}{2}} \cdot (\tau_h \cdot \text{tea} + \tau_t \cdot \text{coffee})$$
Algebraic properties of $p^{\oplus}$:

- **Explicit:**
  - quasi-commutativity: $x_{p^{\oplus}} y = y_{1-p^{\oplus}} x$,
  - distributivity: $x_{p^{\oplus}} (y + z) = x_{p^{\oplus}} y + x_{p^{\oplus}} z$,
  - ... 

- **Implicit:**
  - sub-distributivity: $x \cdot y + x \cdot z \leq x \cdot (y + z)$ where

$$x \leq y \quad \text{iff} \quad x + y = y.$$ 

The inequality is strict if $x$ contains probability.
Concurrency

- True-Concurrency:
  - Concurrency is realised from independent and non-conflicting events.

- Interleaving:
  - Concurrency is reduced to nondeterminism over all possible sequentialisations.

→ Concentrate on the Interleaving approach in the model.

- Ex: the Probabilistic Vending Machine and User are

\[ M = coin \cdot flip_{\frac{1}{2}} \cdot (\tau_h \cdot tea + \tau_t \cdot coffee) \]

and

\[ U = coin \cdot tea \]

The system is \( M_A \parallel U \) where \( A = \{coin, tea, coffee\} \).
Concurrency: Algebraic Properties

Algebraic properties of $\parallel$ (frame set $A$ is left implicit).

- Self restriction:
  - commutativity: $x \parallel y = y \parallel x$,
  - associativity: $x \parallel (y \parallel z) = (x \parallel y) \parallel z$,
  - ...  

- Interactions with other operators:
  - distributivity: $x \parallel (y + z) = x \parallel y + x \parallel z$,
  - exchange law: $(x \parallel u) \cdot (y \parallel v) \leq (x \cdot y) \parallel (u \cdot v)$,

\[
\begin{array}{cc}
\begin{array}{c}
\begin{array}{ccc}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}x\parallel u\parallel y\parallel v\end{array}
\end{array}
\end{array}
\end{array}
\end{array}
\end{array}
\]
\[
\begin{array}{cc}
\begin{array}{c}
\begin{array}{ccc}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}x\parallel u\parallel y\parallel v\end{array}
\end{array}
\end{array}
\end{array}
\end{array}
\end{array}
\end{array}
\]
Proving $M \parallel U$ Satisfies the Specification

Algebraic properties of the system:

- Synchronisation: $a \parallel a = a$ for $a \in \{\text{coin}, \text{tea}, \text{coffee}\}$,
- When a chosen action is not enabled, go away: $\text{tea} \parallel \text{coffee} = 1$ where 1 is the ineffectual process ($\text{Skip}$).

Theorem

We have $\text{coin} \cdot \text{flip}_{\frac{1}{2}} \cdot (\tau_h \cdot \text{tea} + \tau_t) \leq M \parallel U$.

Proof.

Key ingredient: exchange law and monotonicity.

→ Use automated tools (Prover9, Isabelle/HOL, . . .)

In the left hand side, $\text{tea}$ is enabled with probability at least $1/2$. 
The Algebra

Finite iteration: Kleene star

- is a (left) fixed point: \( x^* = 1 + x \cdot x^* \),
- is the least one: \( 1 + x \cdot y = y \Rightarrow x^* \leq y \).

weak concurrent Kleene algebra:

- Signature: \((K, +, \cdot, \|, *, 0, 1)\)
  - 1 ineffectual process \( 1 \cdot x = x \cdot 1 = 1 \),
  - 0 is the most deterministic process: \( 0 + x = x \),
    \( \rightarrow \) Probability is implicit!

- Axiom system: specific set of axioms derived from probabilistic and concurrent Kleene algebras.
Other Applications

- **Hoare Calculus:**

  \[ p \{ x \} q \iff p \cdot x \leq q \]

  where \( p, q \) are pre/post-computation.

  ex:

  \[ p \{ x \} q \land q \{ y \} q' \]

  \[ \frac{p \{ x \} q \land q \{ y \} q'}{p \{ x \cdot y \} q'} \]

- **Rely/Guarantee Calculus:**

  \( pr \{ x \} gq \iff p \{ r \parallel x \} q \land x \leq g \)

  where \( r, g \) are invariants.

  ex:

  \[ pr \{ x \} gq \land p' r' \{ x' \} g' q' \land g' \leq r \land g \leq r' \]

  \[ \frac{(p \sqcap p')(r \sqcap r') \{ x \parallel x' \}(g \parallel g')(q \sqcap q')}{r'} \]

  provided that \( q \sqcap q' \) exists.
Models and Soundness

How do we ensure that the axiom system is consistent i.e. we will not derive any contradiction from the axiom system?
How do we ensure that the axiom system is consistent i.e. we will not derive any contradiction from the axiom system?

Build mathematical models:
- Set of automata: \((P, \rightarrow, i, F)\)
  ex: automaton that does an action \(\text{flip}_p\) followed by \(b\), with probability \(p\), and \(c\), with probability \(1 - p\), is

\[
\begin{align*}
&S_2 \xrightarrow{b} S_3 \\
&S_2 \xrightarrow{\tau_p} S_1 \\
&S_0 \xrightarrow{\text{flip}_p} S_1 \\
&S_4 \xrightarrow{\tau_{1-p}} S_4 \\
&S_4 \xrightarrow{c} S_5
\end{align*}
\]
Models and Soundness

Let $P, Q$ be the sets of states of two automata.

- Rooted $\eta$-simulation equivalence: $R \subseteq P \times Q$
  - Initiality: $(i_P, i_Q) \in R,$
  - Inductiveness:

\[
\begin{array}{c}
s \xrightarrow{R} t \\
\downarrow \\
\downarrow \\
\downarrow \\
s' \xrightarrow{a} s \\
a \\
t_1 \\
t \\
t'
\end{array}
\]

- Finality: $(s, t) \in R \land s \in F_P \Rightarrow t \in F_Q,$
- Rootedness: $(i_P, t) \in R \Rightarrow t = i_Q.$
Models and Soundness

Let $P, Q$ be the sets of states of two automata.

- Rooted $\eta$-simulation equivalence: $R \subseteq P \times Q$
  - Initiality: $(i_P, i_Q) \in R$,
  - Inductiveness:

$$
\begin{array}{c}
\begin{array}{c}
  s \xrightarrow{R} t \\
  s' \xrightarrow{R} t'
  \end{array}
\end{array}

\begin{array}{c}
\begin{array}{c}
  s \xrightarrow{a} t_1 \\
  s' \xrightarrow{R} t'
  \end{array}
\end{array}

\begin{array}{c}
\begin{array}{c}
  s \xrightarrow{R} t \\
  s' \xrightarrow{R} t'
  \end{array}
\end{array}

\begin{array}{c}
\begin{array}{c}
  s \xrightarrow{a} t_1 \\
  s' \xrightarrow{R} t'
  \end{array}
\end{array}

- Finality: $(s, t) \in R \land s \in F_P \Rightarrow t \in F_Q$,
- Rootedness: $(i_P, t) \in R \Rightarrow t = i_Q$. 

Models and Soundness

- Programs are interpreted as (rooted and reachable) automata.
- $x = y$ means there are simulations from $x$ to $y$ and $y$ to $x$.

**Theorem (Soundness)**

*The set of automata modulo rooted $\eta$-simulation equivalence forms a weak concurrent Kleene algebra.*

This model
- insures consistency,
- provides a specification language.
Summary

- The algebra abstracts complex interactions into algebraic expressions:
  - synchronisation/concurrency is resolved with exchange law and distributivity,
  - existence of probabilities are abstracted.
  - ...

- Use of Automated Tools.
- The model insures consistency.
- The model can be used as a specification language (though probability is implicit).
Summary

- The algebra abstracts complex interactions into algebraic expressions:
  - synchronisation/concurrency is resolved with exchange law and distributivity,
  - existence of probabilities are abstracted.
  - ...

- Use of Automated Tools.
- The model insures consistency.
- The model can be used as a specification language (though probability is implicit).

- Outlook:
  - deeper understanding of the use of the algebra to Rely/Guarantee calculus.
  - construction of fully probabilistic models.
  - construction of “true-concurrency” models.