Endowing Concurrent Kleene Algebra with Communication Actions

14th International Conference on Relational and Algebraic Methods in Computer Science

Jason Jaskolka, Ridha Khedri, and Qinglei Zhang

Department of Computing and Software
Faculty of Engineering
McMaster University
Hamilton, Ontario, Canada
{jaskolj,khedri,zhangq33}@mcmaster.ca

April 28, 2014

Outline

Outline

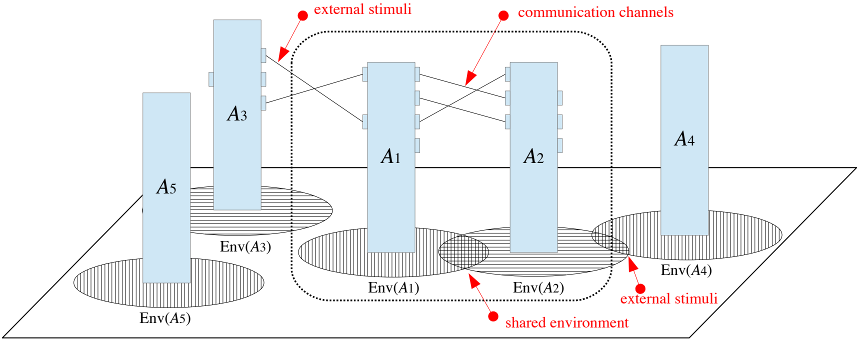
Motivating Question

Question

How can we mathematically formulate the potential for communication condition for covert channel existence in systems of communicating agents?

- We required a formalism that would:
  1. Provide a hybrid model for both communication and concurrency
  2. Lead to a mathematical formulation of the potential for communication

A Hybrid View of Agent Communication



What About Existing Formalisms?

- Looked at existing formalisms for communication and concurrency
  - Temporal Logics
  - Labelled Transition Systems
  - Petri Nets
  - Process Calculi (CCS, CSP, ACP, π-calculus)
- Interested in modelling the behaviour of a system in terms of:
  1. Properties of its states, *or*
  2. Observability of events
- Do not directly, if at all, provide a hybrid model of communication and concurrency that we are interested in

Is Concurrent Kleene Algebra the Answer?

- *Concurrent Kleene Algebra (CKA)* was perhaps the closest formalism to providing a hybrid model
- While CKA can be perceived as a hybrid model for concurrency, the same cannot be said for communication
- Communication in CKA is not directly captured
- CKA does not directly deal with describing how agent behaviours are influenced by external stimuli

Objectives

1. Specify communication in CKA without the need to articulate the state-based system of each action
   - i.e., at a convenient abstract level
2. Express the influence of external stimuli on agent behaviours resulting from the occurrence of external events from
   - Communication among agents
   - Environment of a particular agent

Outline

The Proposed Framework

- Propose a mathematical framework for communication and concurrency called **Communicating Concurrent Kleene Algebra (C^2KA)**
  - Extends the algebraic model of CKA
  - Captures communication and concurrency of agents at the abstract algebraic level
  - Captures the influence of external stimuli on agent behaviour as well as communication through shared environments
  - Presents a different view of communication and concurrency than what was found with existing formalisms

The Proposed Framework

- C^2KA allows for the separation of communicating and concurrent behaviour in a system and its environment
- Can think about concurrent and communicating systems from two different perspectives:
  1. **Behavioural Perspective**: influence of external stimuli as transformations of agent behaviours
  2. **External Event Perspective**: influence of agent behaviours as transformations of external stimuli

Stimuli and Induced Behaviours

Some Terminology

- Every external stimulus *invokes a response* from an agent
- An external stimulus *influences* the behaviour of an agent when the behaviour of then agent changes as a result of the response
- Set of possible influences that any given external stimulus may have on a particular agent are called the *induced behaviours* via external stimuli

A Simple Running Example: One-Place Buffer

- Suppose that a one-place buffer uses two flags to indicate its current status:
  - $flag_1$ denotes the empty/full status
  - $flag_2$ denotes the error status
- Assume that there are two basic system agents:
  - Agent **P** controls $flag_1$
  - Agent **Q** controls $flag_2$

Structure of Agent Behaviours

- Adopt the framework of CKA to describe agent behaviours

Definition (CKA)

A **concurrent Kleene algebra (CKA)** is a structure $(K, +, *, ;, ^{\circledast}, ^{\odot}, 0, 1)$ such that $(K, +, *, ^{\circledast}, 0, 1)$ and $(K, +, ;, ^{\odot}, 0, 1)$ are Kleene algebras linked by the *exchange axiom* given by $(a * b) ; (c * d) \leq_{\mathcal{K}} (b ; c) * (a ; d)$.

- $a \leq_{\mathcal{K}} b$ indicates that *a is a sub-behaviour of b* if and only if $a + b = b$

Structure of Agent Behaviours

Running Example: One-Place Buffer

- Consider the following set of events:

$$P_1 \stackrel{\text{def}}{=} (\mathit{flag}_1 := \mathit{off}) \qquad Q_1 \stackrel{\text{def}}{=} (\mathit{flag}_2 := \mathit{off})$$
$$P_2 \stackrel{\text{def}}{=} (\mathit{flag}_1 := \mathit{on}) \qquad Q_2 \stackrel{\text{def}}{=} (\mathit{flag}_2 := \mathit{on})$$

- K is generated by the set of basic behaviours $\{P_1, P_2, Q_1, Q_2, 0, 1\}$
  - *Inactive agent* 0 is interpreted as abort
  - *Idle agent* 1 is interpreted as skip

Structure of External Stimuli

- Each discrete, observable event introduced to a system is considered to be an *external stimulus* which invokes a response from each system agent

Definition (Stimulus Structure)

Let $\mathcal{S} \stackrel{\text{def}}{=} (S, \oplus, \odot, \mathfrak{d}, \mathfrak{n})$ be an idempotent semiring with a multiplicatively absorbing $\mathfrak{d}$ and identity $\mathfrak{n}$. We call $\mathcal{S}$ a **stimulus structure**.

- $s \leq_{\mathcal{S}} t$ indicates that *s is sub-stimulus of t* if and only if $s \oplus t = t$

Structure of External Stimuli

Running Example: One-Place Buffer

- Behaviour of each agent in the one-place buffer system is influenced by a number of external stimuli:
  - *in* places an item in the buffer
  - *out* removes an item from the buffer
  - *error* generates an error
- S is generated by the set of basic external stimuli $\{in, out, error, \mathfrak{d}, \mathfrak{n}\}$
  - *Deactivation stimulus* $\mathfrak{d}$ is interpreted as a kill signal
  - *Neutral stimulus* $\mathfrak{n}$ is interpreted as any stimulus with no influence that belongs to the complement of the set of external stimuli which may be introduced to a system

Communicating Concurrent Kleene Algebra (C^{2}KA)

Definition (C^{2}KA)

A **Communicating Concurrent Kleene Algebra (C^{2}KA)** is a system $(\mathcal{S}, \mathcal{K})$, where $\mathcal{S} = (S, \oplus, \odot, \mathfrak{d}, \mathfrak{n})$ is a stimulus structure and $\mathcal{K} = (K, +, *, ;, ^{\circledast}, ^{\odot}, 0, 1)$ is a CKA such that $({}_{\mathcal{S}}K, +)$ is a unitary and zero-preserving *left $\mathcal{S}$-semimodule* with mapping $\circ : S \times K \to K$ and $(S_{\mathcal{K}}, \oplus)$ is a unitary and zero-preserving *right $\mathcal{K}$-semimodule* with mapping $\lambda : S \times K \to S$, and where the following axioms are satisfied for all $a, b, c \in K$ and $s, t \in S$:

1. $s \circ (a \,;\, b) = (s \circ a) \,;\, (\lambda(s, a) \circ b)$
2. $c \leq_{\mathcal{K}} a \ \vee\ (s \circ a) \,;\, (\lambda(s, c) \circ b) = 0$
3. $\lambda(s \odot t, a) = \lambda(s, (t \circ a)) \odot \lambda(t, a)$

Communicating Concurrent Kleene Algebra (C^2KA)

- A C^2KA consists of two semimodules
  - $({}_{\mathcal{S}}K, +)$ describes how the stimulus structure $\mathcal{S}$ acts upon the CKA $\mathcal{K}$ via the *next behaviour mapping* $\circ$
  - $(S_{\mathcal{K}}, \oplus)$ describes how the CKA $\mathcal{K}$ acts upon the stimulus structure $\mathcal{S}$ via the *next stimulus mapping* λ
- Together $({}_{\mathcal{S}}K, +)$ and $(S_{\mathcal{K}}, \oplus)$ characterise the response invoked by an external stimulus on the behaviour of an agent as a next behaviour and a next stimulus

Initiating Agent Behaviours

- Agent behaviour can be initiated in two ways:
  1. **Reactivation**: A C^2KA is *with reactivation* if $s \circ 1 \neq 1$ for some $s \in S \backslash \{\mathfrak{d}\}$
     - Passive idle agent may be influenced to behave as any active agent
  2. **Stimulus Initiation**: $a \in K \backslash \{0, 1\}$ is a *stimulus initiator* if and only if $\lambda(\mathfrak{n}, a) \neq \mathfrak{n}$
     - May generate a new stimulus without outside influence

Isotonicity Laws

Proposition

Let $(\mathcal{S}, \mathcal{K})$ be a C^2KA. For all $a, b \in K$ and $s, t \in S$:

1. $a \leq_{\mathcal{K}} b \ \wedge \ s \leq_{\mathcal{S}} t \implies s \circ a \leq_{\mathcal{K}} t \circ b$
2. $a \leq_{\mathcal{K}} b \ \wedge \ s \leq_{\mathcal{S}} t \implies \lambda(s, a) \leq_{\mathcal{S}} \lambda(t, b)$

Corollary

In a C^2KA where the underlying CKA and stimulus structure are built up from quantales, the following laws hold:

1. $a \leq_{\mathcal{K}} b \implies s \circ a \leq_{\mathcal{K}} s \circ b$
2. $s \leq_{\mathcal{S}} t \implies s \circ a \leq_{\mathcal{K}} t \circ a$
3. $s \circ (a \,;\, b + b \,;\, a) \leq_{\mathcal{K}} s \circ (a * b)$
4. $s \circ a^{\circledS} \leq_{\mathcal{K}} s \circ a^{\circledast}$
5. $s \circ a^{\circledS} = +(n \mid n \geq 0 : s \circ a^n)$
6. $s \leq_{\mathcal{S}} t \implies \lambda(s, a) \leq_{\mathcal{S}} \lambda(t, a)$
7. $a \leq_{\mathcal{K}} b \implies \lambda(s, a) \leq_{\mathcal{S}} \lambda(s, b)$
8. $\lambda(s, (a \,;\, b + b \,;\, a)) \leq_{\mathcal{S}} \lambda(s, (a * b))$
9. $\lambda(s, a^{\circledS}) \leq_{\mathcal{S}} \lambda(s, a^{\circledast})$
10. $\lambda(s, a^{\circledS}) = \oplus(n \mid n \geq 0 : \lambda(s, a^n))$

A Comment on a Model for C^2KA

Structure of Agent Behaviours

$\big(PR(EV), \cup, *, ;, ^{\circledast}, ^{\circled{;}}, \emptyset, \{\emptyset\}\big)$ is a CKA.

- A CKA can be modelled as sets of programs and traces
- EV is a set of event occurrences
- A *trace* is a set of events and a *program* is a set of traces
- $TR(EV) \stackrel{\text{def}}{=} \mathcal{P}(EV)$ denotes the set of all traces over EV
- $PR(EV) \stackrel{\text{def}}{=} \mathcal{P}(TR(EV))$ denotes the set of all programs

A Comment on a Model for C^{2}KA
Structure of External Stimuli

$(\mathcal{P}(\Lambda), \cup, \bullet, \emptyset, \{\epsilon\})$ is a stimulus structure.

- A stimulus structure can be modelled by sets of strings
- Λ is a set of alphabet symbols
- $\bullet$ denotes set concatenation
- ϵ is the empty string

A Comment on a Model for C^{2}KA

$\big(Q, \Sigma, \Theta, F, G\big)$ is a C^{2}KA.

- A C^{2}KA can be modelled as a Mealy automaton
- The set of states Q is a subset of $PR(EV)$ (i.e., the set K)
- The input alphabet Σ and output alphabet Θ are given by the stimulus structure such that $\Sigma = \Theta = S$
- The transition function $F : \Sigma \times Q \to Q$ corresponds to the next behaviour mapping $\circ : S \times K \to K$
- The output function $G : \Sigma \times Q \to \Theta$ corresponds to the next stimulus mapping $\lambda : S \times K \to S$

A Comment on a Model for C^2KA

- Proposed model is also equipped with two operations:
  - Operation $;$ is associative
    - *Cascading Product* of Mealy automata
  - Operation $+$ is associative, idempotent, and commutative
    - *Full Direct Product* of Mealy automata

Specifying Systems of Communicating Agents with C^2KA

- Three levels of specification:
  1. Stimulus-Response Specification of Agents
  2. Abstract Behaviour Specification
  3. Concrete Behaviour Specification
- Context of the given problem helps to dictate at which level we need to work

Stimulus-Response Specification

Running Example: One-Place Buffer

$$\mathbf{P} \overset{\text{def}}{=} P_1 + P_2 \qquad\qquad \mathbf{Q} \overset{\text{def}}{=} Q_1 + Q_2$$

$\circ_{\mathbf{P}}$	$\mathfrak{n}$	*in*	*out*	*error*
P_1	P_1	P_2	P_1	P_1
P_2	P_2	P_2	P_1	P_2

$\circ_{\mathbf{Q}}$	$\mathfrak{n}$	*in*	*out*	*error*
Q_1	Q_1	Q_1	Q_1	Q_2
Q_2	Q_2	Q_2	Q_2	Q_2

$\lambda_{\mathbf{P}}$	$\mathfrak{n}$	*in*	*out*	*error*
P_1	$\mathfrak{n}$	$\mathfrak{n}$	*error*	$\mathfrak{n}$
P_2	$\mathfrak{n}$	*error*	$\mathfrak{n}$	$\mathfrak{n}$

$\lambda_{\mathbf{Q}}$	$\mathfrak{n}$	*in*	*out*	*error*
Q_1	$\mathfrak{n}$	$\mathfrak{n}$	$\mathfrak{n}$	$\mathfrak{n}$
Q_2	$\mathfrak{n}$	$\mathfrak{n}$	$\mathfrak{n}$	$\mathfrak{n}$

$$\forall(P_i, Q_i \mid 1 \leq i \leq 2 \;:\; \mathfrak{d} \circ P_i = 0 \;\wedge\; \mathfrak{d} \circ Q_i = 0 \;\wedge\; \lambda(\mathfrak{d}, P_i) = \mathfrak{d} \;\wedge\; \lambda(\mathfrak{d}, Q_i) = \mathfrak{d}\,)$$

$$\textbf{Buffer} \overset{\text{def}}{=} \mathbf{P} \,;\, \mathbf{Q} = (P_1 + P_2) \,;\, (Q_1 + Q_2)$$

Abstract Behaviour Specification

Running Example: One-Place Buffer

- Consider the following context:
  - Buffer can only behave as empty or full
  - Buffer may only be influenced by *in* and *out* stimuli
    - *error* is an uncontrollable stimulus

$$
\begin{aligned}
& (in \oplus out) \circ (P_1 \,;\, Q_1 + P_2 \,;\, Q_1) \\
= \;& in \circ (P_1 \,;\, Q_1) + out \circ (P_1 \,;\, Q_1) + in \circ (P_2 \,;\, Q_1) + out \circ (P_2 \,;\, Q_1) \\
= \;& (in \circ P_1) \,;\, \big(\lambda(in, P_1) \circ Q_1\big) + (out \circ P_1) \,;\, \big(\lambda(out, P_1) \circ Q_1\big) + \\
& (in \circ P_2) \,;\, \big(\lambda(in, P_2) \circ Q_1\big) + (out \circ P_2) \,;\, \big(\lambda(out, P_2) \circ Q_1\big) \\
= \;& P_2 \,;\, Q_1 + P_1 \,;\, Q_2 + P_2 \,;\, Q_2 + P_1 \,;\, Q_1
\end{aligned}
$$

Concrete Behaviour Specification

Running Example: One-Place Buffer

$$
\begin{array}{llllll}
\text{EMPTY} & \stackrel{\text{def}}{=} & P_1 \mathbin{;} Q_1 & = & (\mathit{flag}_1 := \mathit{off} \quad ; \quad \mathit{flag}_2 := \mathit{off}) \\
\text{FULL} & \stackrel{\text{def}}{=} & P_2 \mathbin{;} Q_1 & = & (\mathit{flag}_1 := \mathit{on} \quad ; \quad \mathit{flag}_2 := \mathit{off}) \\
\text{UNDERFLOW} & \stackrel{\text{def}}{=} & P_1 \mathbin{;} Q_2 & = & (\mathit{flag}_1 := \mathit{off} \quad ; \quad \mathit{flag}_2 := \mathit{on}) \\
\text{OVERFLOW} & \stackrel{\text{def}}{=} & P_2 \mathbin{;} Q_2 & = & (\mathit{flag}_1 := \mathit{on} \quad ; \quad \mathit{flag}_2 := \mathit{on})
\end{array}
$$

C^{2}KA and Orbits, Stabilisers, and Fixed Points

- Orbits, stabilisers, and fixed points allow us to:
  1. Perceive a kind of topology of a system
  2. Gain some insight into the communication channels that can be established
  3. Model the possible reactions of a system to a stimulus
  4. Alleviate the state explosion problem in model checking

C^2KA and Orbits, Stabilisers, and Fixed Points

- Two complementary notions of orbits, stabilisers, and fixed points
- Can think about concurrent and communicating systems from two different perspectives:
  1. **Behavioural Perspective**: action of external stimuli on agent behaviours described by $({}_{\mathcal{S}}K, +)$
  2. **External Event Perspective**: action of agent behaviours on external stimuli described by $(S_{\mathcal{K}}, \oplus)$

Orbits

Definition (Orbit)

The **orbit** of a in $\mathcal{S}$ is the set $\mathrm{Orb}(a) = \{s \circ a \mid s \in S\}$.

- Set of all possible behavioural responses from an agent to any external stimulus from $\mathcal{S}$
  - Set of all possible future behaviours
- **Running Example**:

$$\begin{aligned} \mathrm{Orb}(\textsc{empty}) &= \{\textsc{empty}, \textsc{full}, \textsc{underflow}, \textsc{overflow}\} \\ \mathrm{Orb}(\textsc{overflow}) &= \{\textsc{underflow}, \textsc{overflow}\} \end{aligned}$$

Another Interpretation of Orbits

Definition (Induced Behaviour)

Let $a, b \in K$ be agent behaviours such that $a \neq b$. We say that b **is induced by a via external stimuli** (denoted by $a \lhd b$) if and only if $\exists(s \mid s \in S : s \circ a = b)$.

- Equivalently, $a \lhd b \iff b \in \mathrm{Orb}(a)$ for $a \neq b$
- **Running Example**:
  - $\mathrm{EMPTY} \lhd \mathrm{UNDERFLOW}$ via the external stimulus *out*
  - $\mathrm{EMPTY} \lhd \mathrm{OVERFLOW}$ via the external stimulus $in \odot in$

Strong Orbits

Definition (Strong Orbit)

The **strong orbit** of a in $\mathcal{S}$ is the set
$\mathrm{Orb}_{\mathrm{S}}(a) = \{b \in K \mid \mathrm{Orb}(b) = \mathrm{Orb}(a)\}$.

- Two agents are in the same strong orbit ($a \sim_{\mathcal{K}} b$) if and only if their orbits are identical
- If $a \sim_{\mathcal{K}} b$, then $\exists(s, t \mid s, t \in S : s \circ a = b \land t \circ b = a)$
  - s and t can be perceived as *inverses* of one another
- **Running Example**: We have two strong orbits: $\{\mathrm{EMPTY}, \mathrm{FULL}\}$ and $\{\mathrm{UNDERFLOW}, \mathrm{OVERFLOW}\}$

Stabilisers

Definition (Stabiliser)

The **stabiliser** of a in $\mathcal{S}$ is the set $\mathrm{Stab}(a) = \{s \in S \mid s \circ a = a\}$.

- Set of external stimuli which have no observable influence (or act as neutral stimuli) on an agent
- **Running Example**: $\mathrm{Stab}(\textsc{empty})$ is generated by $\{error, in \odot out\}$

Fixed Point Behaviours

Definition (Fixed Point)

An element $a \in K$ is a **fixed point** if $\forall(s \mid s \in S\backslash\{\mathfrak{d}\} : s \circ a = a)$.

- Not influenced by any external stimulus other than the deactivation stimulus $\mathfrak{d}$
- May be any number of fixed points with respect to $\circ$
- When $a \in K$ is a fixed point, $\mathrm{Orb}(a) = \{0, a\}$ and $\mathrm{Stab}(a) = S\backslash\{\mathfrak{d}\}$
- **Running Example**: With regard to the specification, the behaviour Q_2 is a fixed point

Topological Insights and Induced Behaviours

Proposition

Let $a, b, c \in K$ be agent behaviours.

1. a is a fixed point $\implies \forall(b \mid b \in K \land b \neq 0 \land b \neq a : \neg(a \lhd b))$
2. $a \sim_{\mathcal{K}} b \implies a \lhd b \land b \lhd a$
3. $a \sim_{\mathcal{K}} b \implies (a \lhd c \iff b \lhd c)$

Outline

Conclusion

- C^2KA extends the algebraic setting of CKA to capture the influence of external stimuli on the behaviour of system agents
- C^2KA supports the ability to work in either a state-based or event-based model for both the specification of communicating and concurrent behaviour
- To the best of our knowledge, such a formalism does not currently exist in the literature
  - Required for studying the necessary conditions for covert channel existence

Current and Future Work

- Developed a formulation of the potential for communication condition for covert channels using C^2KA
- Prototype tool to support the automated computation and specification of systems of communicating agents using C^2KA
- Adapt C^2KA for solving interface equations
- Use C^2KA to capture and explain the influence of external stimuli on agent behaviour in social networking environments

Outline

Questions

Questions?