Concurrency Theory

Lecture 6: The Calculus of Communicating Systems (CCS)

Stephan Mennicke
Knowledge-Based Systems Group

May 16-17, 2023

CCS

$\mathcal{N} = \{a, b, c, \ldots\}$ . . . set of names ($\tau \notin \mathcal{N}$)

$\overline{\mathcal{N}} = \{\overline{\alpha} \mid \alpha \in \mathcal{N}\}$ . . . set of conames

$Act = \mathcal{N} \cup \overline{\mathcal{N}} \cup \{\tau\}$ (note, there is no $\overline{\tau}$ and for $\alpha \in Act \setminus \{\tau\}$, $\overline{\overline{\alpha}} = \alpha$)

The set of (CCS) processes *Pr* is defined by

$$P \quad ::= \quad \mathbf{0} \;\Big|\; \mu.P \;\Big|\; P + P \;\Big|\; P \mid P \;\Big|\; (\nu a)(P) \;\Big|\; K$$

where $\mu \in Act$, $a \in \mathcal{N}$, and $K \in \mathcal{K}$.

Define the language CCS parameterized over Act, $\mathcal{K}$, and $\mathcal{T}_{\mathcal{K}} \subseteq \mathcal{K} \times Act \times Pr$.

$$\text{CCS}(Act, \mathcal{K}, \mathcal{T}_{\mathcal{K}})$$

Structural Operational Semantics

CCS$(Act, \mathcal{K}, \mathcal{T}_\mathcal{K})$ specifies an LTS $(Pr, Act, \rightarrow \cup \mathcal{T}_\mathcal{K})$ where $\rightarrow \subseteq (Pr \setminus \mathcal{K}) \times Act \times Pr$ is the smallest relation satisfying the following rules:

$$\text{(Pref)}\ \frac{}{\mu.P \xrightarrow{\mu} P}$$

$$\text{(SumL)}\ \frac{P \xrightarrow{\mu} P'}{P + Q \xrightarrow{\mu} P'} \qquad \text{(SumR)}\ \frac{Q \xrightarrow{\mu} Q'}{P + Q \xrightarrow{\mu} Q'}$$

$$\text{(ParL)}\ \frac{P \xrightarrow{\mu} P'}{P \mid Q \xrightarrow{\mu} P' \mid Q} \qquad \text{(ParR)}\ \frac{Q \xrightarrow{\mu} Q'}{P \mid Q \xrightarrow{\mu} P \mid Q'}$$

$$\text{(Com)}\ \frac{P \xrightarrow{\mu} P' \qquad Q \xrightarrow{\overline{\mu}} Q'}{P \mid Q \xrightarrow{\tau} P' \mid Q'} \qquad \text{(Res)}\ \frac{P \xrightarrow{\mu} P'}{(\nu a)\,(P) \xrightarrow{\mu} (\nu a)\,(P')}\ \text{if } a \notin \{\mu, \overline{\mu}\}$$

Algebraic Properties of CCS (1/2)

Theorem 1
For CCS processes P, Q, R, the following equivalences hold

$$P \mid Q \quad \Leftrightarrow \quad Q \mid P \tag{1}$$

$$P \mid (Q \mid R) \quad \Leftrightarrow \quad (P \mid Q) \mid R \tag{2}$$

$$P \mid \mathbf{0} \quad \Leftrightarrow \quad P \tag{3}$$

Thus, for a finite family of CCS processes $(P_i)_{i \in I}$ we may write

$$\prod_{i \in I} P_i$$

for the parallel composition of all processes P_i ($i \in I$).

Algebraic Properties of CCS (2/2)

Theorem 2
For CCS processes P, Q, R, the following equivalences hold

$$P + Q \Leftrightarrow Q + P \tag{4}$$

$$P + (Q + R) \Leftrightarrow (P + Q) + R \tag{5}$$

$$P + \boldsymbol{0} \Leftrightarrow P \tag{6}$$

$$P + P \Leftrightarrow P \tag{7}$$

Thus, for a finite family of CCS processes $(P_i)_{i \in I}$ we may write

$$\sum_{i \in I} P_i$$

for the choice between all processes P_i ($i \in I$).

The Expansion Lemma

Definition 3 (Head Standard Form)
A process of the form $P = \sum_{i \in I} \mu_i \,.\, P_i$ is in *head standard form* (if $I = \emptyset$, $P = \mathbf{0}$).

Theorem 4 (Expansion Lemma)
If $P = \sum_{i \in I} \mu_i \,.\, P_i$ *and* $P' = \sum_{j \in J} \mu'_j \,.\, P'_j$, *then*

$$P \mid P' \;\; \Leftrightarrow \;\; \sum_{i \in I} \mu_i \,.\, P_i \mid P' + \sum_{j \in J} \mu'_j \,.\, P \mid P'_j + \sum_{\overline{\mu_i} = \mu'_j} \tau \,.\, P_i \mid P'_j. \qquad (8)$$

Compositionality of Bisimilarity (1/2)

Lemma 5
If $P \Leftrightarrow Q$, *then for all processes* R, $\mu \in Act$, *and* $a \in \mathcal{N}$,

$$P \mid R \quad \Leftrightarrow \quad Q \mid R \tag{9}$$

$$P + R \quad \Leftrightarrow \quad Q + R \tag{10}$$

$$(\nu a)\,()P \quad \Leftrightarrow \quad (\nu a)\,()Q \tag{11}$$

$$\mu\,.\,P \quad \Leftrightarrow \quad \mu\,.\,Q \tag{12}$$

Compositionality of Bisimilarity (2/2)

Definition 6 (CCS Context)
A *CCS context* is a process with a single occurrence of a hole $\bullet$ as a sub-expression. If C is a CCS context and P a CCS process, then $C[P]$ is the CCS process C with the hole replaced by process P. If C and C' are CCS contexts, then $C[C']$ is the CCS context C where the hole in C is replaced by C'.

Theorem 7 (Congruence)
In CCS, $\leftrightarrow$ *is a congruence relation.*

De Simone Format (1/2)

A transition rule is in *De Simone format* if it has the form

$$\frac{X_j \xrightarrow{\mu_j} Y_j (j \in J)}{f(X_1, \ldots, X_n) \xrightarrow{\mu} T}$$

where

1. f is an n-ary operator symbol in the language;
2. $J \subseteq \{1, \ldots, n\}$;
3. X_r $(1 \leq r \leq n)$, and Y_j $(j \in J)$ are distinct variables;
4. T is a term of the language possibly containing the variables $X'_1, \ldots, X'_n$, where for each $r \in \{1, \ldots, n\}$ we have $X'_r = Y_r$ if $r \in J$ and $X'_r = X_r$ otherwise; moreover, each X'_i $(1 \leq i \leq n)$ occurs at most once in T.

De Simone Format (2/2)

Theorem 8
If all operators of a process language have transition rules in de Simone format, then bisimilarity is a congruence.

It is important to note that the de Simone format requires a single transition relation type involved in transition rules following the format.

Expressivity of CCS

Theorem 9
There are Act, $\mathbb{C}$, *and* $\mathcal{T}_{\mathbb{C}}$, *so that* $CCS(Act, \mathbb{C}, \mathcal{T}_{\mathbb{C}})$ *is Turing-complete.*

$\rightsquigarrow$ bisimilarity of CCS processes is undecidable.

Proof Plan:

1. Pick a Turing-complete model $\rightsquigarrow$ Minsky machines
2. Encode computations by means of **CCS** using only finitely many actions, constants, and a finite constant transition relation per Minsky machine

1. Minsky Machine (or Counter Machine)

Definition 10
A *Minsky machine* is a pair $\mathcal{M} = (R, P)$, where $R = \{c_1, c_2, \ldots, c_n\}$ is a finite set of counters (or registers) and $P = \{l_0, l_1, \ldots, l_m\}$ is a finite set of *instructions* l_i $(i = 0, 1, \ldots, m)$ over $\mathcal{M}$, such that $l_i = \langle X_i, \texttt{inc}\ \ k\ :\ \ j\rangle$, $l_i = \langle X_i, \texttt{dec}\ \ k\ :\ \ j\ :\ \ j'\rangle$, and $l_m = \texttt{halt}$, where $i, j, j' \in \{0, 1, \ldots, m\}$ are line indizes and $k \in \{1, \ldots, n\}$ are counter indizes.

Definition 11
For Minsky machine $\mathcal{M} = (R, P)$ we call a pair $\langle i, \beta\rangle$ a *configuration of* $\mathcal{M}$ if $l_i \in P$ and $\beta : R \to \mathbb{N}$. A configuration $\langle 0, \beta\rangle$ is called an initial configuration. Define a step of $\mathcal{M}$ by $\langle i, \beta\rangle \rhd \langle j, \beta'\rangle$ if, and only if, (1) $l_i = \langle X_i, \texttt{inc}\ \ k\ :\ \ j\rangle$ and $\beta' = \beta[c_k \mapsto \beta(c_k) + 1]$, (2) $l_i = \langle X_i, \texttt{dec}\ \ k\ :\ \ j\ :\ \ j'\rangle$, $\beta(c_k) > 0$ and $\beta' = \beta[c_k \mapsto \beta(c_k) - 1]$, and (3) $l_i = \langle X_j, \texttt{dec}\ \ k\ :\ \ j'\ :\ \ j\rangle$ and $\beta(c_k) = 0$.

The *Halting Problem for Minsky Machines* is the language

$$\mathbf{L}_{\mathsf{HALT}} := \{\langle \mathcal{M}, \beta \rangle \mid \exists n \in \mathbb{N} : \langle 0, \beta \rangle \triangleright^{*} \langle n, \mathtt{halt} \rangle\}.$$

Theorem 12
$\mathbf{L}_{\mathsf{HALT}}$ *is undecidable, even if only two counters are used.*

Theorem 13
Minsky Machines are Turing-complete.

2. Implementing Minsky Machines in CCS

Construction: in two steps.

1. Implementing unbounded counters using finitely many actions and constants;
2. Implementing the program instructions

We do the second step first. As an interface to the counters c_1 and c_2, we assume action names $\overline{u^1}, \overline{d^1}, z^1$ to control the first counter and $\overline{u^2}, \overline{d^1}, z^2$ for the second. For each $l_i \in P$, $X_i \in \mathbb{C}$, which we translate using the following theme (assuming $k \in \{1, 2\}$):

1. $\langle X_i, \mathtt{inc}\ k\ :\ j\rangle \mapsto X_i$ with $X_i \xrightarrow{\overline{u^k}} X_j$;
2. $\langle X_i, \mathtt{dec}\ k\ :\ j\ :\ j'\rangle \mapsto X_i$ with $X_i \xrightarrow{\overline{d^k}} X_j$ and $X_i \xrightarrow{z^k} X_{j'}$;
3. $\langle X_i, \mathtt{halt}\rangle \mapsto X_i$ with $X_i \xrightarrow{h} \mathbf{0}$.

2.1 Implementing Counters

A single counter may be realized using constants $C, C_1, C_2 \in \mathbb{C}$ and actions $u, d, \overline{z} \in Act$.

1. Define $C \xrightarrow{\overline{z}} C$ and $C \xrightarrow{u} (\nu a)\,(C_1 \mid a.C)$;
2. Define $C_1 \xrightarrow{d} \overline{a}.\mathbf{0}$ and $C_1 \xrightarrow{u} (\nu b)\,(C_2 \mid b.C_1)$;
3. Define $C_2 \xrightarrow{d} \overline{b}.\mathbf{0}$ and $C_2 \xrightarrow{u} (\nu a)\,(C_1 \mid a.C_2)$.

For any process P, reachable from C, define $val(P)$ inductively:

Base: $val(P) = 0$ if $P = C$.

Step: For process Q with $val(Q) = n$ $(n > 0)$, $val(Q') = n + 1$ if $Q \xrightarrow{u} Q'$ and $val(Q') = n - 1$ if $Q \xrightarrow{d} \cdot \xrightarrow{\tau} Q'$.

For two processes P and Q, reachable from C, we get $val(P) = val(Q)$ iff $P \leftrightarrow Q$.

Putting Everything Together

Let $\mathcal{M} = (R, P)$ be a Minsky machine with $R = \{c_1, c_2\}$ and $P = \{l_0, l_1, \ldots, l_n\}$.

Our construction uses $Act = \{u^1, d^1, z^1, u^2, d^2, z^2, \tau, \overline{u^1}, \overline{d^1}, \overline{z^1}, \overline{u^2}, \overline{d^2}, \overline{z^2}\}$ and $\mathbb{C} = \{C_1^1, C_2^1, C^1, C_1^2, C_2^2, C^2, X_0, X_1, \ldots, X_n\}$, where n is the maximal line index of P. $\mathcal{T}_{\mathbb{C}}$ defined as before.

Theorem 14
For $\beta_0 = \{c_1 \mapsto 0, c_2 \mapsto 0\}$, $\langle 0, \beta_0 \rangle \triangleright^* \langle i, \beta \rangle$ *with* $\beta(c_1) = n_1$ *and* $\beta(c_2) = n_2$, *we get* $(\nu u^1, u^2, d^1, d^2, z^1, z^2)\,(X_0 \mid C^1 \mid C^2) \xrightarrow{\tau}{}^* (\nu u^1, u^2, d^1, d^2, z^1, z^2)\,(X_i \mid \underline{C^1} \mid \underline{C^2})$ *such that* $val(\underline{C_1}) = n_1$ *and* $val(\underline{C_2}) = n_2$.

$\rightsquigarrow$ halting problem for CCS is undecidable.

Outlook

- Alternative model: Carl Adam Petri and his Nets
- What is decidable about Petri nets?
- Enhancing CCS: the π-calculus