Preorder-Constrained Simulations for Program Refinement with Effects

Koko Muroya
(RIMS, Kyoto University)

Takahiro Sanada
(RIMS, Kyoto University)

Natsuki Urabe
(NII)

CMCS (Luxembourg), 6 April 2024

Today's topic

- a coinductive technique for
  quantitative equational reasoning on effectful programs

Quantitative equational reasoning

- "p behaves the same as p'

  and p' terminates with a less number of steps"

  - $p \Downarrow^n \implies p' \Downarrow^m \wedge \; n \geq m$

- (basic) *quantitative* notion of observational refinement

Quantitative equational reasoning

- "p behaves the same as p'

  and p' terminates with a certain number of steps"

  - $(p \Downarrow^{n} \implies p' \Downarrow^{m} \wedge \; n \, Q \, m) \quad \overset{\Delta}{\iff} \quad p \preceq^{Q} p'$

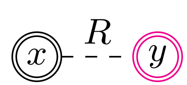
    given a "length preorder" $Q \subseteq \mathbb{N} \times \mathbb{N}$

- (basic) *quantitative* notion of observational refinement

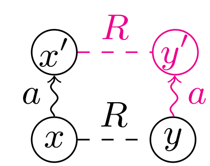
A coinductive approach

- stepwise reasoning on execution traces, using nondeterministic automata
- e.g. standard simulation
  - (FYI: simulation is the asymmetric version of bisimulation)
  - $p \preceq^{=} p' \overset{\Delta}{\iff} (p \Downarrow^n \implies p' \Downarrow^m \wedge \; n = m)$

    $\Longleftarrow \quad p \; R \; p'$ such that



(a) ***Final***

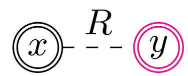


(b) ***Step***

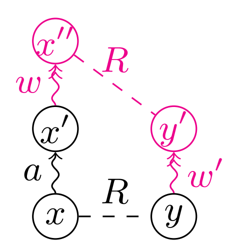
Counting simulation [M. 2020]

- stepwise reasoning on execution traces, using nondeterministic automata
- parameterised by a length preorder $Q \subseteq \mathbb{N} \times \mathbb{N}$
  - (FYI: simulation is the asymmetric version of bisimulation)
  - $p \preceq^{Q} p' \quad \overset{\Delta}{\Longleftrightarrow} \quad (p \Downarrow^{n} \Longrightarrow \; p' \Downarrow^{m} \wedge \; n \, Q \, m)$

    $\Longleftarrow \quad p \, R \, p'$ such that



(a) ***C-Final***

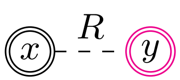


(c) ***C-Step*** (2) where $|aw|Q|w'|$

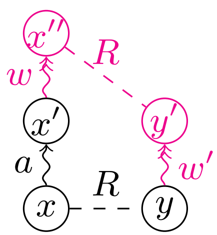
Counting simulation [M. 2020]

- stepwise reasoning on execution traces, using nondeterministic automata
  - $p \preceq^{Q} p' \quad \overset{\Delta}{\iff} \quad (p \Downarrow^{n} \implies p' \Downarrow^{m} \land \; n \, Q \, m)$

    $\impliedby \quad p \; R \; p'$ such that



(a) ***C-Final***

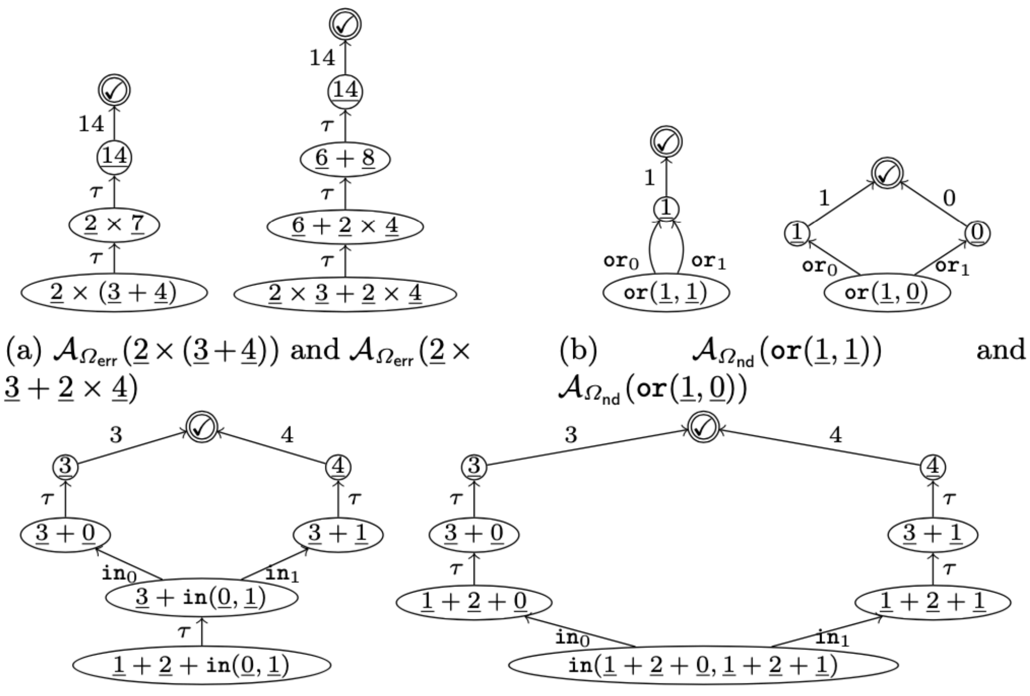


(c) ***C-Step*** (2) where $|aw| Q |w'|$

- soundness only for “deterministic” programs
  - or “branching-free” automata

Counting simulation [M. 2020]

- Today's topic: a coinductive technique for quantitative equational reasoning on effectful programs
- Goal: extend counting simulation to a wider class of effects



(a) $\mathcal{A}_{\Omega_{\mathsf{err}}}(\underline{2} \times (\underline{3} + \underline{4}))$ and $\mathcal{A}_{\Omega_{\mathsf{err}}}(\underline{2} \times \underline{3} + \underline{2} \times \underline{4})$

(b) $\mathcal{A}_{\Omega_{\mathsf{nd}}}(\mathtt{or}(\underline{1}, \underline{1}))$ and $\mathcal{A}_{\Omega_{\mathsf{nd}}}(\mathtt{or}(\underline{1}, \underline{0}))$

(c) $\mathcal{A}_{\Omega_{\mathsf{io}}}(\underline{1} + \underline{2} + \mathtt{in}(\underline{0}, \underline{1}))$ and $\mathcal{A}_{\Omega_{\mathsf{io}}}(\mathtt{in}(\underline{1} + \underline{2} + \underline{0}, \underline{1} + \underline{2} + \underline{1}))$

Fig. 3: Example pairs of NAs

Overview

- Goal: extend counting simulation to a wider class of effects
- Challenge 1:
    - Solution 1:
- Challenge 2:
    - Solution 2:
- Contribution:

Challenge 1: varying observation

- ✅ exception

termination only

$$p \leq^{Q} p' \overset{\Delta}{\iff} (p \Downarrow^{n} \implies p' \Downarrow^{m} \wedge \; n \, Q \, m)$$

- ❌ nondeterminism

result

$$p \leq^{Q} p' \overset{\Delta}{\iff} (p \Downarrow^{n} v \implies p' \Downarrow^{m} v \;\wedge\; n \, Q \, m)$$

- ❌ I/O

result, and trace of I/O values

$$p \leq^{Q} p' \overset{\Delta}{\iff} (p \Downarrow^{n} (v, tr) \implies p' \Downarrow^{m} (v, tr) \;\wedge\; n \, Q \, m)$$

Challenge 1: varying observation

- internal vs. external choice
  - nondeterminism: internal, unobservable choice

$$\mathtt{or}(1,1) \Downarrow 1 \implies 1 \Downarrow 1 \;\wedge\; 1 = 1$$

coincidence of results

  - input: external, observable choice

$$\mathtt{in}(1,1) \Downarrow (1,\mathtt{in}_i) \implies 1 \Downarrow (1,\varepsilon) \;\wedge\; 1 = 1$$

coincidence of results, but
no coincidence of I/O traces

Solution 1: "observation preorder" on traces

- program trace $tr \in \Sigma^*$

  where $\Sigma = \{\tau\} \cup \mathbb{N} \cup \overline{\Omega}$

  pure computation

  execution result

  effect traces (e.g. $\mathtt{or}_1\mathtt{in}_0$)

- examples:
  - $\mathrm{Tr}(\mathtt{or}(1,2)) = \{\mathtt{or}_0 1, \mathtt{or}_1 2\}$
  - $\mathrm{Tr}(\mathtt{in}(1,2)) = \{\mathtt{in}_0 1, \mathtt{in}_1 2\}$
  - $\mathrm{Tr}(1) = \{1\}$
  - $\mathrm{Tr}(1+1) = \{\tau 2\}$

Solution 1: “observation preorder” on traces

- program trace $tr \in \Sigma^*$

  where $\Sigma = \{\tau\} \cup \mathbb{N} \cup \overline{\Omega}$

  pure computation

  execution result

  effect traces (e.g. $\mathtt{or}_1\mathtt{in}_0$)

- in general: $p_0 \xrightarrow{l_0} p_1 \xrightarrow{l_1} \cdots \xrightarrow{l_k} \underline{n} \xrightarrow{n} \checkmark$

Solution 1: “observation preorder” on traces

- program trace $tr \in \Sigma^*$

  where $\Sigma = \{\tau\} \cup \mathbb{N} \cup \overline{\Omega}$

  pure computation

  execution result

  effect traces (e.g. $\mathtt{or}_1\mathtt{in}_0$)

- introducing “observation preorder” $\mathcal{Q} \subseteq \Sigma^* \times \Sigma^*$

Solution 1: "observation preorder" on traces

- program trace $tr \in \Sigma^*$

  where $\Sigma = \{\tau\} \cup \mathbb{N} \cup \overline{\Omega}$

  pure computation

  execution result

  effect traces (e.g. $\mathtt{or}_1\mathtt{in}_0$)

- introducing "observation preorder" $\mathcal{Q} \subseteq \Sigma^* \times \Sigma^*$

- e.g. lifted length preorder:

$$\text{given } Q \subseteq \mathbb{N} \times \mathbb{N}, \quad t\,\dot{Q}\,u \overset{\Delta}{\iff} |t|\,Q\,|u|$$

Solution 1: “observation preorder” on traces

- program trace $tr \in \Sigma^*$

  where $\Sigma = \{\tau\} \cup \mathbb{N} \cup \overline{\Omega}$

  pure computation

  execution result

  effect traces (e.g. $\mathtt{or}_1\mathtt{in}_0$)

- introducing “observation preorder” $\mathcal{Q} \subseteq \Sigma^* \times \Sigma^*$

- e.g. “filtered equality”

  given $\Sigma' \subseteq \Sigma$, $\quad t =_{(\mathrm{rem}_{\Sigma'})} u \overset{\Delta}{\iff}$ t and u are the same except for Σ'

  - $\tau a b \tau c \tau \tau =_{(\mathrm{rem}_{\{\tau\}})} abc$

Solution 1: “observation preorder” on traces

- program trace $tr \in \Sigma^*$

  where $\Sigma = \{\tau\} \cup \mathbb{N} \cup \overline{\Omega}$

  pure computation

  execution result

  effect traces (e.g. $\mathtt{or}_1\mathtt{in}_0$)

- introducing “observation preorder” $\mathcal{Q} \subseteq \Sigma^* \times \Sigma^*$

  Definition 1 ((quantitative) refinement). *Let Q be a preorder on $\mathbb{N}$ (dubbed* length preorder*).*

  1. *For Ω_{err}, $t \preceq^{Q}_{\mathsf{err}} u$ is defined by $\forall w.(t \overset{w}{\twoheadrightarrow} \checkmark \implies \exists w'.u \overset{w'}{\twoheadrightarrow} \checkmark \wedge |w|Q|w'|)$.*
  2. *For Ω_{nd}, $t \preceq^{Q}_{\mathsf{nd}} u$ is defined by $\forall w.(t \overset{w}{\twoheadrightarrow} \checkmark \implies \exists w'.u \overset{w'}{\twoheadrightarrow} \checkmark \wedge |w|Q|w'| \wedge w =_{\mathsf{rem}_{\{\tau\}\cup\overline{\Omega_{\mathsf{nd}}}}} w')$.*
  3. *For Ω_{io}, $t \preceq^{Q}_{\mathsf{io}} u$ is defined by $\forall w.(t \overset{w}{\twoheadrightarrow} \checkmark \implies \exists w'.u \overset{w'}{\twoheadrightarrow} \checkmark \wedge |w|Q|w'| \wedge w =_{\mathsf{rem}_{\{\tau\}}} w')$.*

Solution 1: “observation preorder” on traces

- program trace $tr \in \Sigma^*$

  where $\Sigma = \{\tau\} \cup \mathbb{N} \cup \overline{\Omega}$

  pure computation

  execution result

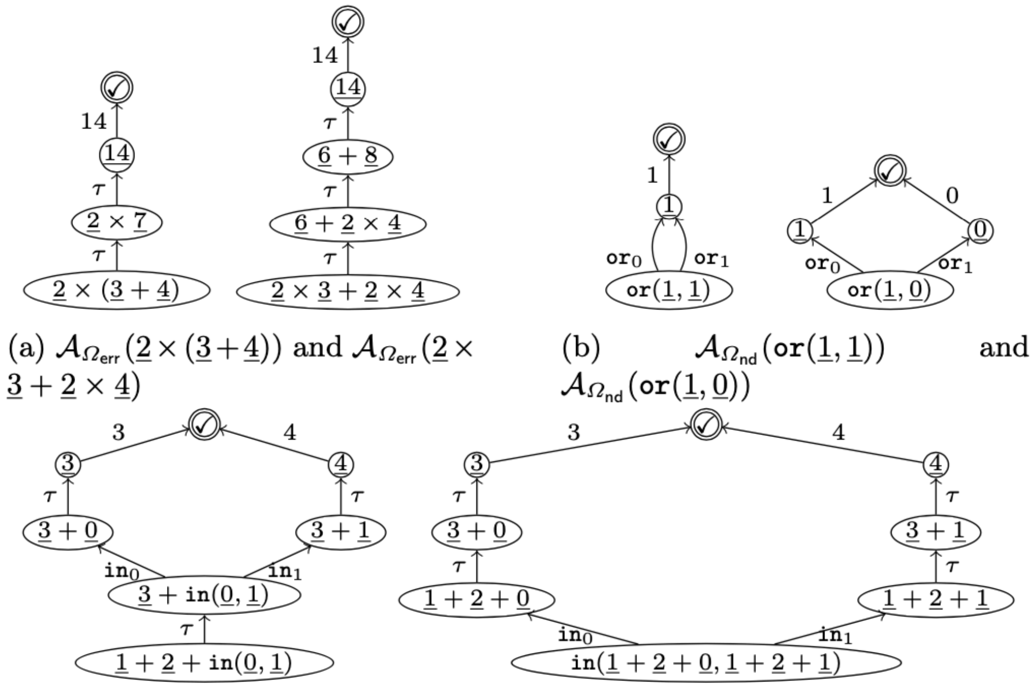
  effect traces (e.g. $\mathtt{or}_1 \mathtt{in}_0$)

- introducing “observation preorder” $\mathcal{Q} \subseteq \Sigma^* \times \Sigma^*$

$\dot{Q}$	refinement $\preceq^Q_{\mathsf{err}}$ for exception
$\dot{Q} \cap =_{\mathsf{rem}_{\{\tau\}\cup\overline{\Omega_{\mathsf{nd}}}}}$	refinement $\preceq^Q_{\mathsf{nd}}$ for nondeterminism
$\dot{Q} \cap =_{\mathsf{rem}_{\{\tau\}}}$	refinement $\preceq^Q_{\mathsf{io}}$ for I/O

Examples

- exhibit quantitative refinement $\preceq^{\leq}_{\mathrm{err}}$, $\preceq^{=}_{\mathrm{nd}}$, $\preceq^{=}_{\mathrm{io}}$



(a) $\mathcal{A}_{\Omega_{\mathrm{err}}}(\underline{2} \times (\underline{3} + \underline{4}))$ and $\mathcal{A}_{\Omega_{\mathrm{err}}}(\underline{2} \times \underline{3} + \underline{2} \times \underline{4})$

(b) $\mathcal{A}_{\Omega_{\mathrm{nd}}}(\mathtt{or}(\underline{1}, \underline{1}))$ and $\mathcal{A}_{\Omega_{\mathrm{nd}}}(\mathtt{or}(\underline{1}, \underline{0}))$

(c) $\mathcal{A}_{\Omega_{\mathrm{io}}}(\underline{1} + \underline{2} + \mathtt{in}(\underline{0}, \underline{1}))$ and $\mathcal{A}_{\Omega_{\mathrm{io}}}(\mathtt{in}(\underline{1} + \underline{2} + \underline{0}, \underline{1} + \underline{2} + \underline{1}))$

Fig. 3: Example pairs of NAs

Overview

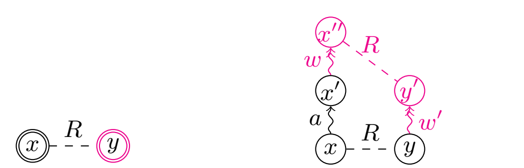
- Goal: extend counting simulation to a wider class of effects
  - Starting point: ✅ exception ❌ nondeterminism ❌ I/O
- Challenge 1: varying observation
  - Solution 1: “observation preorder” on traces
- Challenge 2:
  - Solution 2:
- Contribution:

Challenge 2: branching effects

- ✅ exception
- ❌ nondeterminism
- ❌ I/O

- unsoundness of counting simulation for branching effects
  - due to incomplete inspection of branches

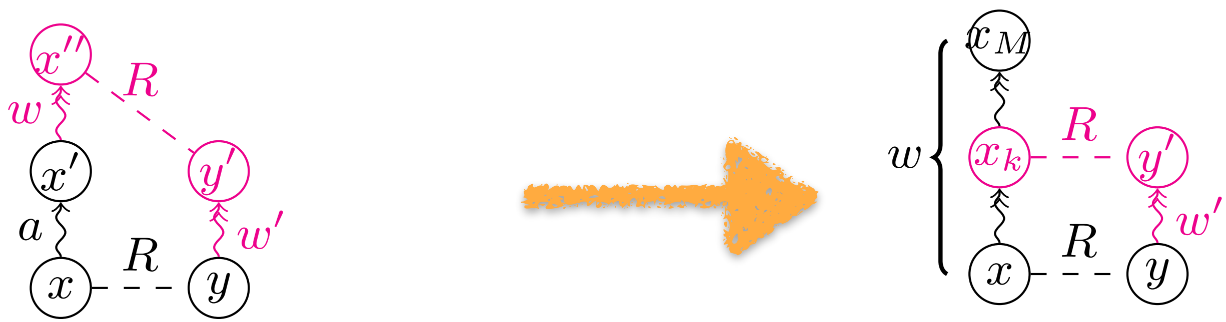
cf. definition of counting simulation



(a) ***C-Final*** **(c)** ***C-Step*** (2) where $|aw|Q|w'|$

Solution 2: limited ∃

- from unlimited ∃ to limited ∃



(c) ***C-Step*** (2) where $|aw|Q|w'|$

(b) ***Step***^{M} where $a_1 \cdots a_k \mathbf{Q} w'$

- enabling full inspection of branches

Overview

- Goal: extend counting simulation to a wider class of effects
  - Starting point: ✅ exception ❌ nondeterminism ❌ I/O
- Challenge 1: varying observation
  - Solution 1: “observation preorder” on traces
- Challenge 2: branching effects
  - Solution 2: limited ∃
- Contribution:

Contribution: $(M, \mathcal{Q})$-simulation

- parameterised by
  - "look-ahead bound" $M \in \mathbb{N}_+$
  - observation preorder $\mathcal{Q} \in \Sigma^* \times \Sigma^*$

Definition 3 **($(M, \mathbf{Q})$-simulations).** *For each $M \in \mathbb{N}_+$, a binary relation $R \subseteq X_1 \times X_2$ is an M*-bounded **$\mathbf{Q}$**-constrained simulation *($(M, \mathbf{Q})$*-simulation *in short) from $\mathcal{A}_1$ to $\mathcal{A}_2$ if, for any $(x, y) \in R$, the following* **Final**^{M} *and* **Step**^{M} *hold.*

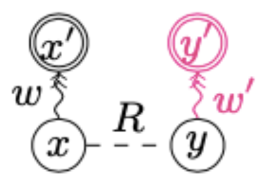
Final^M *For each $w = a_1 \ldots a_n \in \Sigma^*$ and $x_1 \ldots x_n \in X_1^*$ such that $n < M$, $x \stackrel{a_1}{\rightsquigarrow}_1 x_1 \cdots \stackrel{a_n}{\rightsquigarrow}_1 x_n$ and $x_n \in F_1$, there exist $w' \in \Sigma^*$ and $y' \in X_2$ such that $w \mathbf{Q} w'$, $y \stackrel{w'}{\rightsquigarrow\!\!\!\rightarrow}_2 y'$ and $y' \in F_2$.*

Step^M *For each $a_1 \ldots a_M \in \Sigma^M$ and $x_1 \ldots x_M \in X_1^M$ such that $x \stackrel{a_1}{\rightsquigarrow}_1 x_1 \cdots \stackrel{a_M}{\rightsquigarrow}_1 x_M$, there exist $k \in \{1, \ldots, M\}$, $w' \in \Sigma^*$ and $y' \in X_2$ such that $a_1 \cdots a_k \mathbf{Q} w'$, $y \stackrel{w'}{\rightsquigarrow\!\!\!\rightarrow}_2 y'$ and $x_k R y'$.*

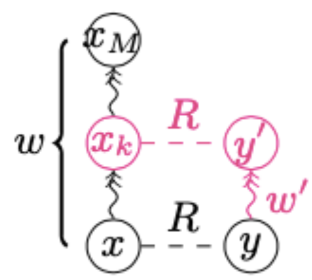
Contribution: $(M, \mathcal{Q})$-simulation

- parameterised by
  - "look-ahead bound" $M \in \mathbb{N}_+$
  - observation preorder $\mathcal{Q} \in \Sigma^* \times \Sigma^*$

Definition 3 **($(M, \mathbf{Q})$-simulations).** *For each $M \in \mathbb{N}_+$, a binary relation $R \subseteq X_1 \times X_2$ is an M-*bounded $\mathbf{Q}$-constrained simulation *($(M, \mathbf{Q})$-*simulation *in short) from $\mathcal{A}_1$ to $\mathcal{A}_2$ if, for any $(x, y) \in R$, the following* ***Final***^{M} *and* ***Step***^{M} *hold.*



(a) ***Final***^{M} where $|w| < M \ \wedge \ w\mathbf{Q}w'$



(b) ***Step***^{M} where $a_1 \cdots a_k \mathbf{Q} w'$

Contribution: $(M, \mathcal{Q})$-simulation

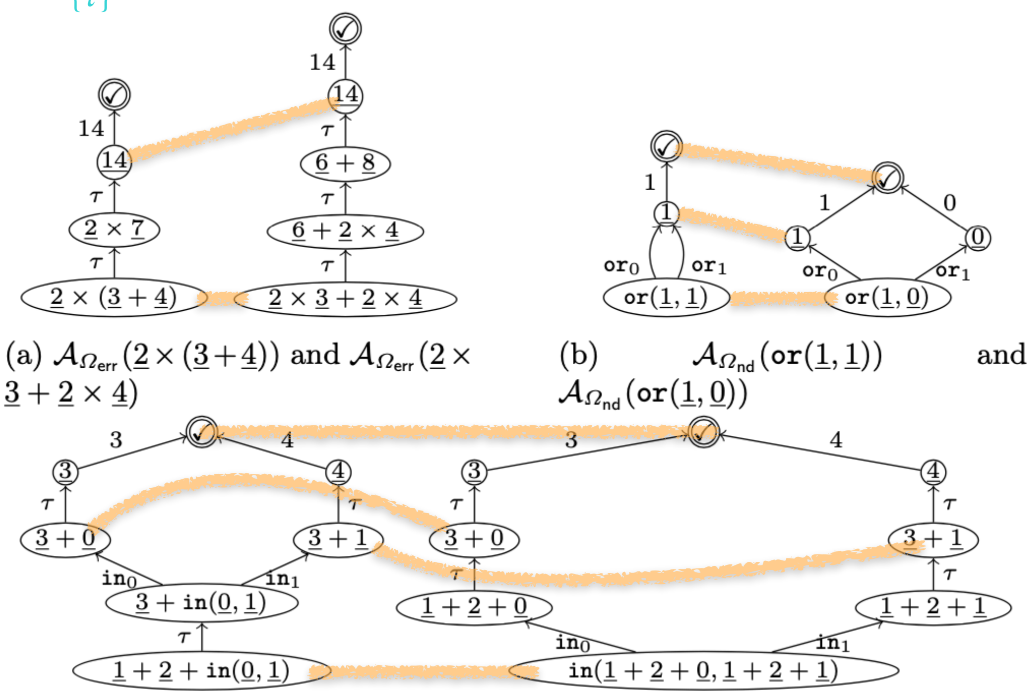
- parameterised by
  - "look-ahead bound" $M \in \mathbb{N}_+$
  - observation preorder $\mathcal{Q} \in \Sigma^* \times \Sigma^*$

Corollary 1 (correctness of $(M, \mathbf{Q})$-simulations wrt. refinement).

1. *For any* $M \in \mathbb{N}_+$ *and* $t, u \in \mathbf{T}_{\Omega_{\mathsf{err}}}$, $t \lesssim_{M,\dot{Q}} u \implies t \preceq^{Q}_{\mathsf{err}} u$.
2. *For any* $M \in \mathbb{N}_+$ *and* $t, u \in \mathbf{T}_{\Omega_{\mathsf{nd}}}$, $t \lesssim_{M,\dot{Q} \cap =_{\mathsf{rem}_{\{\tau\} \cup \overline{\Omega_{\mathsf{nd}}}}}} u \implies t \preceq^{Q}_{\mathsf{nd}} u$.
3. *For any* $M \in \mathbb{N}_+$ *and* $t, u \in \mathbf{T}_{\Omega_{\mathsf{io}}}$, $t \lesssim_{M,\dot{Q} \cap =_{\mathsf{rem}_{\{\tau\}}}} u \implies t \preceq^{Q}_{\mathsf{io}} u$. $\square$

Examples of (M, Q)-simulations

- $(2, \dot{\lesssim})$-simulation for (a)
- $(1, \doteq \cup =_{\mathrm{rem}_{\{\tau\}\cup\overline{\Omega}}})$-simulation for (b)
- $(2, \doteq \cup =_{\mathrm{rem}_{\{\tau\}}})$-simulation for (c)



(a) $\mathcal{A}_{\Omega_{\mathsf{err}}}(\underline{2} \times (\underline{3} + \underline{4}))$ and $\mathcal{A}_{\Omega_{\mathsf{err}}}(\underline{2} \times \underline{3} + \underline{2} \times \underline{4})$

(b) $\mathcal{A}_{\Omega_{\mathsf{nd}}}(\mathtt{or}(\underline{1}, \underline{1}))$ and $\mathcal{A}_{\Omega_{\mathsf{nd}}}(\mathtt{or}(\underline{1}, \underline{0}))$

(c) $\mathcal{A}_{\Omega_{\mathsf{io}}}(\underline{1} + \underline{2} + \mathtt{in}(\underline{0}, \underline{1}))$ and $\mathcal{A}_{\Omega_{\mathsf{io}}}(\mathtt{in}(\underline{1} + \underline{2} + \underline{0}, \underline{1} + \underline{2} + \underline{1}))$

Fig. 3: Example pairs of NAs

Overview

- Goal: extend counting simulation to a wider class of effects
  - Starting point: ✅ exception ❌ nondeterminism ❌ I/O
- Challenge 1: varying observation
  - Solution 1: “observation preorder” on traces
- Challenge 2: branching effects
  - Solution 2: limited ∃
- Contribution: $(M, \mathcal{Q})$-simulation
  - Result: ✅ exception ✅ nondeterminism ✅ I/O

Overview

- Goal: extend counting simulation to a wider class of effects
  - Starting point: ✅ exception ❌ nondeterminism ❌ I/O

- Challenge 1: varying observation
  - Solution 1: “observation preorder” on traces
- Challenge 2: branching effects
  - Solution 2: limited ∃

- Contribution: a *generative spectrum* of $(M, \mathcal{Q})$-simulations
  - Result: ✅ exception ✅ nondeterminism ✅ I/O

A generative spectrum of $(M, \mathcal{Q})$-simulations

$$
\begin{array}{ccccc}
(1,\mathbf{Q})\text{-similarity} & & (2,\mathbf{Q})\text{-similarity} & & \mathbf{Q}\text{-trace inclusion} \\
\lesssim_{1,\mathbf{Q}} & \longrightarrow & \lesssim_{2,\mathbf{Q}} & \longrightarrow \cdots \longrightarrow & \sqsubseteq_{\mathbf{Q}}
\end{array}
$$

Fig. 1: A generative spectrum, parameterised by the observation preorder $\mathbf{Q}$

observation preorder $\mathbf{Q}$	$(1, \mathbf{Q})$-simulation	$\mathbf{Q}$-trace inclusion $\sqsubseteq_{\mathbf{Q}}$
$=$	standard simulation	finite trace inclusion
$=_{\mathsf{rem}_{\{\tau\}}}$	weak simulation	weak trace inclusion
$\dot{Q}$	(new instances)	refinement $\preceq^{Q}_{\mathsf{err}}$ for exception
$\dot{Q} \cap =_{\mathsf{rem}_{\{\tau\} \cup \overline{\Omega_{\mathsf{nd}}}}}$		refinement $\preceq^{Q}_{\mathsf{nd}}$ for nondeterminism
$\dot{Q} \cap =_{\mathsf{rem}_{\{\tau\}}}$		refinement $\preceq^{Q}_{\mathsf{io}}$ for I/O

Table 1: Instances of the two ends of the generative spectrum (see Sec. $\boxed{4}$ for details)

Overview

- Goal: extend counting simulation to a wider class of effects
  - Starting point: ✅ exception ❌ nondeterminism ❌ I/O
- Challenge 1: varying observation
  - Solution 1: “observation preorder” on traces
- Challenge 2: branching effects
  - Solution 2: limited ∃
- Contribution: a *generative spectrum* of $(M, \mathcal{Q})$-simulations
  - Result: ✅ exception ✅ nondeterminism ✅ I/O

Future work 1: bunching branches

- ❌ probabilistic choice
  - a naive attempt yields a false refinement:

    $\mathtt{or}_{0.5}(1,1) \sqsubseteq_{\leq_+} \mathtt{or}_{0.5}(0,1)$
  - Idea: from nondeterministic automata to weighted automata?

Future work 2: efficient solving

- $p \overset{?}{\leq^{Q}} p' \;\Longleftarrow\; p \lesssim_{M,\mathcal{Q}} p'$ for nondeterministic automata $\mathscr{A}(p), \mathscr{A}(p')$

  that represent whole execution of p, p'

  $\Longleftarrow$ reachability in a graph "pairing" $\mathscr{A}(p)$ with $\mathscr{A}(p')$

- polynomial time solving, based on whole execution 😐
  - Idea: solving without executing programs
    - using TRS techniques? [M. & Hamana, FLOPS '24]

Overview

- Goal: extend counting simulation to a wider class of effects
  - Starting point: ✅ exception ❌ nondeterminism ❌ I/O

- Challenge 1: varying observation
  - Solution 1: “observation preorder” on traces
- Challenge 2: branching effects
  - Solution 2: limited $\exists$
- Contribution: a *generative spectrum* of $(M, \mathcal{Q})$-simulations
  - Result: ✅ exception ✅ nondeterminism ✅ I/O
  - (with a game-theoretic characterisation)
  - (with the up-to technique)