Environmental Bisimulations for Higher-Order Languages

Davide Sangiorgi
Naoki Kobayashi
Eijiro Sumii
Main Result

A bisimulation proof technique for various higher-order languages

- Pure $\lambda$-calculi (call-by-name/call-by-value)
- Cbv $\lambda$-calculus with higher-order store
- Higher-order $\pi$-calculus
  - Sound & complete (i.e., characterizes contextual equivalence) in each language
Talk Outline

- Background
  - Contextual equivalence
  - Bisimulation
  - Problems of bisimulation for higher-order languages
- Environmental bisimulation
- Up-to techniques
- Related work
Contextual Equivalence
[Morris 73]

Two programs $M, N$ are contextually equivalent

\[ M \equiv N \]

if they "behave the same" under any context

E.g., in pure lambda-calculi, $M \equiv N$ if

\[ \forall C. \ C[M] \downarrow \text{ iff } C[N] \downarrow \]

- Direct proof is hard because of "$\forall C$"

\[ \Rightarrow \text{ Proof technique is desired} \]
Bisimulation

Two programs M, N are bisimilar M ~ N if they can simulate each other's input/output behavior.

- Soundness: Bisimilar programs are contextually equivalent.
- Completeness: Vice versa.
  \[ \implies \text{Gives a proof technique for contextual equivalence} \]
Problem: Bisimulation for Higher-Order Languages (1/2)

\[ M \sim N \text{ if:} \]

1. If \( M \) outputs \( M_1 \) and becomes \( M' \), then \( N \) outputs \( N_1 \) and becomes \( N' \) with \( M' \sim N' \)

*What condition is needed for \( M_1 \) and \( N_1 \)?*

- "\( M_1 \sim N_1 \)" is too strong, because \( M_1 \) and \( M' \) (\( N_1 \) and \( N' \)) may share a "secret"

\[ \Rightarrow \text{Incomplete in impure languages} \]
Problem: Bisimulation for Higher-Order Languages (2/2)

\[ \mathbf{M} \sim \mathbf{N} \text{ if:} \]

2. If \( \mathbf{M} \) becomes \( \mathbf{M}' \) for input \( \mathbf{M}_1 \), then \( \mathbf{N} \) becomes \( \mathbf{N}' \) for input \( \mathbf{N}_1 \) with \( \mathbf{M}' \sim \mathbf{N}' \)

*What condition is needed for \( \mathbf{M}_1 \) and \( \mathbf{N}_1 \)?*

- "\( \mathbf{M}_1 \sim \mathbf{N}_1 \)" is ill-formed, because it appears in a negative position
  \[ \Rightarrow \text{Bisimilarity (\( \sim \)) may not exist} \]
Talk Outline

● Background
● **Environmental bisimulation**
  - Key idea
  - General definition
  - Specific definitions
● Up-to techniques
● Related work
Environmental Bisimulation

Key idea:
Use relation-indexed relation $\sim_R$ to represent the "changing world"

- $R$ is called an environment
- Accounts for the generativity of
  - Locations (in $\lambda$-calculus with store),
  - Channels (in higher-order $\pi$-calculus), etc.

- Complete also in impure languages
- Monotone (union-closed) and well-defined
General Definition (1/3)

X is an environmental simulation if $M X_R N$ implies:

1. If $M \rightarrow M'$, then $N \Rightarrow N'$ and $M' X_R N'$
2. If $M$ outputs $M_1$ and becomes $M'$, then $N$ outputs $N_1$ and becomes $N'$ with $M' X_R \cup \{(M_1, N_1)\} N'$
X is an environmental simulation if $M X_R N$ implies:

3. For all $M_1 R^* N_1$,
   if $M$ becomes $M'$ for input $M_1$,
   then $N$ becomes $N'$ for input $N_1$
   with $M' X_R N'$
   - $R^*$ is the context closure of $R$
   - Represents "synthesis of knowledge" by the context
X is an environmental bisimulation if both X and X\(^{-1}\) are environmental simulations
- X\(^{-1}\) is defined by \((X^{-1})_R = (X_R)^{-1}\)

\sim is the largest environmental bisimulation
Instance 1: Env. Bisim. for Higher-Order $\pi$-Calculus (Simplified)

$X$ is an environmental simulation if $P X_R Q$ implies:

1. If $P \rightarrow P'$, then $Q \Rightarrow Q'$ and $P' X_R Q'$
2. If $P = c!M.P'$, then $Q \Rightarrow c!N.Q'$ and $P' X_R \cup \{(M, N)\} Q'$
3. If $P = c?x.P'$, then $Q \Rightarrow c?x.Q'$ and $P'{P_1/x} X_R Q'{Q_1/x}$ for all $P_1 R^* Q_1$
4. $P | P_1 X_R Q | Q_1$ for all $P_1 R Q_1$
Instance 2: Env. Bisim. for Pure Call-by-Name $\lambda$-Calculus

$X$ is an environmental simulation if $M \xrightarrow{X_R} N$ implies:

1. If $M \rightarrow M'$, then $N \Rightarrow N'$ and $M' \xrightarrow{X_R} N'$

2. If $M = \lambda x. M'$, then $N \Rightarrow \lambda x. N'$ and $\lambda x. M' \xrightarrow{X_R} \cup \{ (\lambda x. M', \lambda x. N') \} \lambda x. N'$

- Moreover, $M'\{M_1/x\} \xrightarrow{X_R} N'\{N_1/x\}$ for all $M_1 \xrightarrow{R^*} N_1$
Simple Example (for Pedagogy)

\[ \text{M} = \lambda x. (\lambda y. y)x \text{ and } \text{N} = \lambda x. x \]

- Consider \( \mathcal{X}_0 = \{ (R, \text{M}, \text{N}) \} \) where \( R = \{(\text{M}, \text{N})\} \)
- For any \( \text{M}_1 R^* \text{N}_1 \),
  \( \text{M} \text{M}_1 \rightarrow (\lambda y. y)\text{M}_1 \rightarrow \text{M}_1 \)
  \( \text{N} \text{N}_1 \rightarrow \text{N}_1 \)
- Extend \( \mathcal{X}_0 \) to \( \mathcal{X} = \)
  \[ \{ (R^*, (\lambda y. y)\text{M}_1, \text{N}_1), (R^*, \text{M}_1, \text{N}_1) \mid \text{M}_1 R^* \text{N}_1 \} \]
- \( \mathcal{X} \) is an environmental bisimulation
Talk Outline

- Background
- Environmental bisimulation
- Up-to techniques
  - Big-step environmental bisimulation up to reduction and context
- Related work
X is a big-step environmental simulation up to reduction and context if \( M \xrightarrow{X_R} N \) implies:

- If \( M \Rightarrow \lambda x.M' \), then \( N \Rightarrow \lambda x.N' \) and for all \( M_1 R^* N_1 \),
  \[
  M'\{M_1/x\} \Rightarrow (X_R \cup \{ (\lambda x.M', \lambda x.N') \})^* \iff N'\{N_1/x\}
  \]
  - Recall \( R^* \) is the context closure of \( R \)
The Example Revisited

\[ M = \lambda x. (\lambda y. y)x \text{ and } N = \lambda x. x \]

- Take \( X = \{ (R, M, N) \} \) where \( R = \{ (M, N) \} \)
- For any \( M_1 R^* N_1 \),
  \[ M M_1 \Rightarrow M_1 \]
  \[ R R^* \quad R^* = (X_R)^* \]
  \[ N N_1 \Rightarrow N_1 \]
- \( X \) is a big-step environmental bisimulation up to reduction and context
  - The proof is now as easy as it should be!
In the paper

- Environmental bisimulations for
  - Pure cbv $\lambda$-calculus
  - Cbv $\lambda$-calculus with higher-order store
- Up-to techniques
  - Up-to environment / bisimilarity / reduction / expansion / contexts / full contexts
  - Combinations of the above
- Soundness and completeness proofs
- More examples
Talk Outline

- Background
- Environmental bisimulation
- Up-to techniques
  - Big-step environmental bisimulation up to reduction and context
- Related work
Applicative Bisimulation
[Abramsky 90]

\[ \lambda x. M \sim \lambda x. N \text{ if } (\lambda x. M)M_1 \sim (\lambda x. N)M_1 \]

for any closed term \( M_1 \)

- Soundness proof is hard [Howe 96]
- Unsound in languages with information hiding
  - Abstract types (\( \exists \alpha \)), generative names (\( \nu x \)), etc.

Reason:
The lhs and the rhs are "different worlds"
Normal Form Bisimulation
[Sangiorgi 94, Lassen et al.]

\[ \lambda x. M \sim \lambda x. N \text{ if } (\lambda x. M)y \sim (\lambda x. N)y \]

for a fresh variable \( y \)

- Easy to use: one argument suffices
- Complete only in languages with control (\( \mu \)), and state (\( := \)) [Lassen et al.]
Logical Bisimulation
[Sangorgi-Kobayashi-Sumii 07]

\[ \lambda x. M \sim \lambda x. N \text{ if} \]
\[ (\lambda x. M)C[M_1,\ldots,M_n] \sim (\lambda x. N)C[N_1,\ldots,N_n] \]
for all C with \( M_1,\ldots,M_n \sim N_1,\ldots,N_n \)

- Sound (and complete in pure \( \lambda \)-calculi)
- Not monotone, but works for pure \( \lambda \)-calculi
Previous "Environmental" Bisimulations

- For first-order languages
  - Polymorphic $\pi$-calculus [Pierce-Sangiorgi 00]
  - Spi calculus [Abadi-Gordon 98]
- For higher-order languages
  (with a few "built-in" up-to techniques)
  - $\lambda$-calculi with perfect encryption / existential types [Sumii-Pierce 04, 05]
  - Imperative $\lambda$-calculus / object calculi [Koutavas-Wand 06, 06, 07]
Conclusion

- Sound and complete bisimulations for
  - Pure $\lambda$-calculi (call-by-name/call-by-value)
  - Cbv $\lambda$-calculus with higher-order store
  - Higher-order $\pi$-calculus
- Up-to techniques for the bisimulations

Future work:
- "More formal" general framework
- More formal comparison with other proof techniques