# An Algebra of Alignment for Relational Verification

Timos Antonopoulos[1], Eric Koskinen[2], Ton Chanh Le[2], Ramana Nagasamudram[2], David A. Naumann[2], Minh Ngo[2]

[1]Yale University [2]Stevens Institute of Technology

POPL 2023

Supported by NSF CNS 1718713, CCF 2106845, CCF 2131476, and ONR N00014-17-1-2787

# Relational properties of programs

$$c_0 \mathrel{\widehat{=}} \begin{array}{l} i := 0; \\ z := 1; \\ \mathsf{while}\ i < n\ \mathsf{do} \\ \quad i := i + 1; \\ \quad z := z \times i; \\ \mathsf{od} \end{array} \qquad c_1 \mathrel{\widehat{=}} \begin{array}{l} i := 1; \\ z := 1; \\ \mathsf{while}\ i \leq n\ \mathsf{do} \\ \quad z := z \times i; \\ \quad i := i + 1; \\ \mathsf{od} \end{array}$$

Equiv: both programs compute the same value in $z$

# Relational properties of programs

$$c_0 \mathrel{\widehat{=}} \begin{array}{l} i := 0; \\ z := 1; \\ \mathsf{while}\ i < n\ \mathsf{do} \\ \quad i := i + 1; \\ \quad z := z \times i; \\ \mathsf{od} \end{array} \qquad c_1 \mathrel{\widehat{=}} \begin{array}{l} i := 1; \\ z := 1; \\ \mathsf{while}\ i \leq n\ \mathsf{do} \\ \quad z := z \times i; \\ \quad i := i + 1; \\ \mathsf{od} \end{array}$$

Equiv: both programs compute the same value in $z$

$c_0|c_1 : n = n' \approx\!> z = z'$ Any pair of terminating runs, from states related by $n = n'$, end in states related by $z = z'$

(*Note:* $n = n'$ *relates states: using* $'$ *for values in second state*)

# Relational properties of programs

$$c_0 \mathrel{\widehat{=}} \begin{array}{l} i := 0; \\ z := 1; \\ \mathsf{while}\ i < n\ \mathsf{do} \\ \quad i := i + 1; \\ \quad z := z \times i; \\ \mathsf{od} \end{array} \qquad c_1 \mathrel{\widehat{=}} \begin{array}{l} i := 1; \\ z := 1; \\ \mathsf{while}\ i \leq n\ \mathsf{do} \\ \quad z := z \times i; \\ \quad i := i + 1; \\ \mathsf{od} \end{array}$$

Equiv: both programs compute the same value in $z$

$c_0 | c_1 : n = n' \approx\!> z = z'$ Any pair of terminating runs, from states related by $n = n'$, end in states related by $z = z'$

(*Note:* $n = n'$ *relates states: using* $'$ *for values in second state*)

Examples: refinement, conditional equivalence, majorization
Example self relations: noninterference, monotonicity, continuity

# Establishing relational properties

$c_0 \;\widehat{=}\; i := 0;\ z := 1;\ \textsf{while}\ i < n\ \textsf{do}\ \ i := i+1;\ z := z \times i;\ \textsf{od}$

$c_1 \;\widehat{=}\; i := 1;\ z := 1;\ \textsf{while}\ i \le n\ \textsf{do}\ \ z := z \times i;\ i := i+1;\ \textsf{od}$

$c_0 | c_1 :\ n = n' \approx\!> z = z'$

Show $\{0 \le n\}\ c_0\ \{z = n!\}$ and $\{0 \le n\}\ c_1\ \{z = n!\}$
can show using unary techniques, but there's an easier way.

# Establishing relational properties

$c_0 \;\widehat{=}\; i := 0;\ z := 1;\ \mathsf{while}\ i < n\ \mathsf{do}\ \ i := i + 1;\ z := z \times i;\ \mathsf{od}$

$c_1 \;\widehat{=}\; i := 1;\ z := 1;\ \mathsf{while}\ i \leq n\ \mathsf{do}\ \ z := z \times i;\ i := i + 1;\ \mathsf{od}$

$c_0 | c_1 :\ n = n' \;\approx\!> \; z = z'$

Show $\{0 \leq n\}\ c_0\ \{z = n!\}$ and $\{0 \leq n\}\ c_1\ \{z = n!\}$
can show using unary techniques, but there's an easier way.

Reason in terms of a convenient alignment (here, *lockstep*)

| | iter. 0 | | iter. 1 | | iter. 2 | | | | iter. $n$ |
|---|---|---|---|---|---|---|---|---|---|
| $c_0$: | $i = 0$, $z = 1$ | $\mapsto^*$ | $i = 1$, $z = 1$ | $\mapsto^*$ | $i = 2$, $z = 2$ | $\mapsto^*$ | $\cdots$ | $\mapsto^*$ | $i = n$, $z = \ldots$ |
| $n = n'$ | | | | | | | | | $z = z'$ |
| $c_1$: | $i' = 1$, $z' = 1$ | $\mapsto^*$ | $i' = 2$, $z' = 1$ | $\mapsto^*$ | $i' = 3$, $z' = 2$ | $\mapsto^*$ | $\cdots$ | $\mapsto^*$ | $i' = n'+1$, $z' = \ldots$ |

# Establishing relational properties

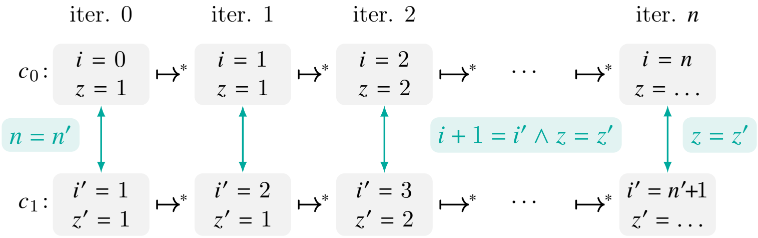
$c_0 \;\widehat{=}\; i := 0;\ z := 1;\ \mathsf{while}\ i < n\ \mathsf{do}\ \ i := i + 1;\ z := z \times i;\ \mathsf{od}$

$c_1 \;\widehat{=}\; i := 1;\ z := 1;\ \mathsf{while}\ i \leq n\ \mathsf{do}\ \ z := z \times i;\ i := i + 1;\ \mathsf{od}$

$c_0 | c_1 :\ n = n' \approx\!> z = z'$

Show $\{0 \leq n\}\ c_0\ \{z = n!\}$ and $\{0 \leq n\}\ c_1\ \{z = n!\}$
can show using unary techniques, but there's an easier way.

Reason in terms of a convenient alignment (here, *lockstep*)



# Representing Alignments

To establish a relational judgment

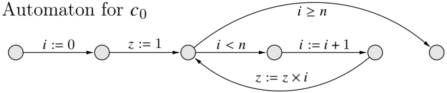
Pick an alignment

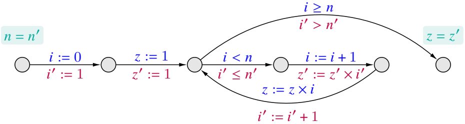
{pre $n = n'$}

$i := 0; z := 1; \qquad i' := 1; z' := 1;$
{invar $i + 1 = i' \wedge z = z'$}
while $i < n$ do
$\quad i := i + 1; \qquad z' := z' \times i';$
$\quad z := z \times i; \qquad i' := i' + 1$
od

{post $z = z'$}

Automaton for $c_0$




Product automaton for $c_0$ and $c_1$

Reason in terms of the alignment

# Representing Alignments

To establish a relational judgment

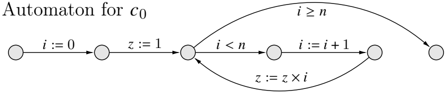
Pick an alignment

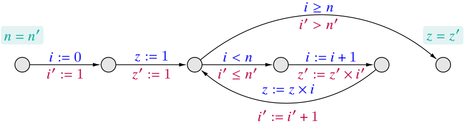
$\{\text{pre } n = n'\}$

$i := 0; z := 1; \quad i' := 1; z' := 1;$
$\{\text{invar } i + 1 = i' \wedge z = z'\}$
$\text{while } i < n \text{ do}$
$\quad i := i + 1; \quad z' := z' \times i';$
$\quad z := z \times i; \quad i' := i' + 1$
$\text{od}$

$\{\text{post } z = z'\}$

Automaton for $c_0$




Product automaton for $c_0$ and $c_1$

Reason in terms of the alignment

Why is this sound?

# Representing Alignments

To establish a relational judgment

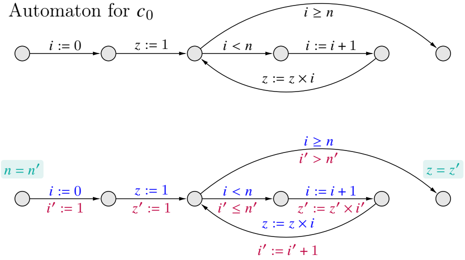
Pick an alignment

{pre $n = n'$}

$i := 0; z := 1; \quad i' := 1; z' := 1;$
{invar $i + 1 = i' \wedge z = z'$}
while $i < n$ do
$\quad i := i + 1; \quad z' := z' \times i';$
$\quad z := z \times i; \quad i' := i' + 1$
od

{post $z = z'$}

Automaton for $c_0$




Product automaton for $c_0$ and $c_1$

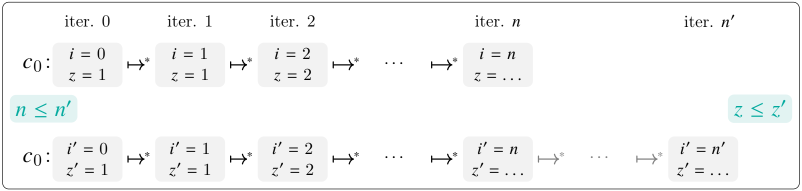
Reason in terms of the alignment

Why is this sound?
Product needs to be *adequate*

# Adequacy: covering all executions

$c_0 \;\widehat{=}\; i := 0;\ z := 1;\ \mathsf{while}\ i < n\ \mathsf{do}\ i := i+1;\ z := z \times i;\ \mathsf{od}$

$c_0 | c_0 : n \leq n' \approx\!> z \leq z'$ Lockstep alignment not adequate

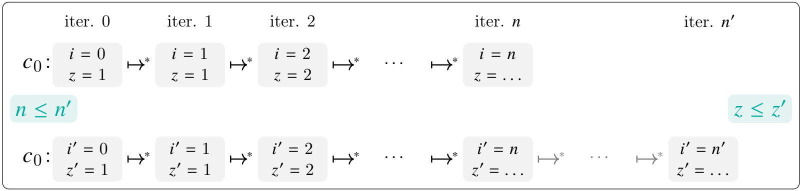


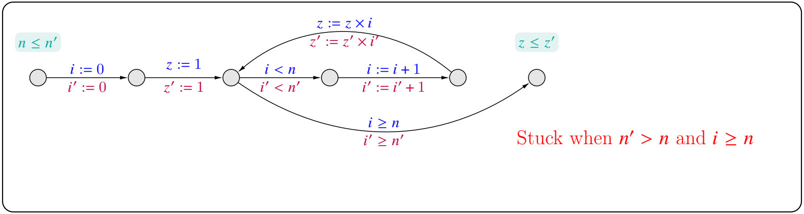
# Adequacy: covering all executions

$$c_0 \;\widehat{=}\; i := 0;\; z := 1;\; \mathsf{while}\ i < n\ \mathsf{do}\ i := i+1;\; z := z \times i;\; \mathsf{od}$$

$c_0 | c_0 :\ n \leq n' \approx\!> z \leq z'$

Lockstep alignment not adequate

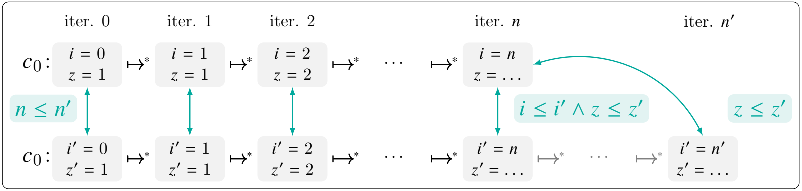


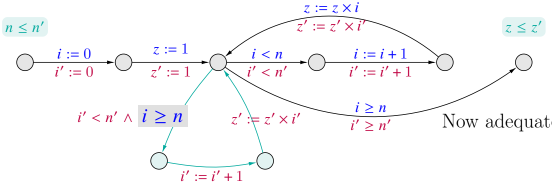



# Adequacy: covering all executions

$c_0 \;\widehat{=}\; i := 0;\; z := 1;\;$ while $i < n$ do $i := i + 1;\; z := z \times i;$ od

$c_0 | c_0 : n \leq n' \approx\!> z \leq z'$ Lockstep alignment not adequate



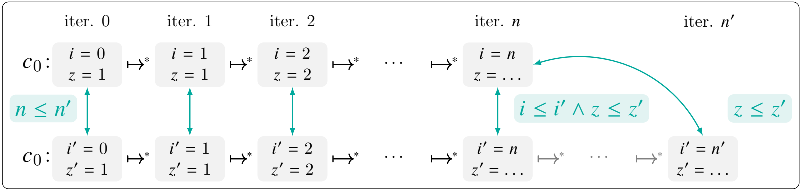

Now adequate under pre-relation $n \leq n'$

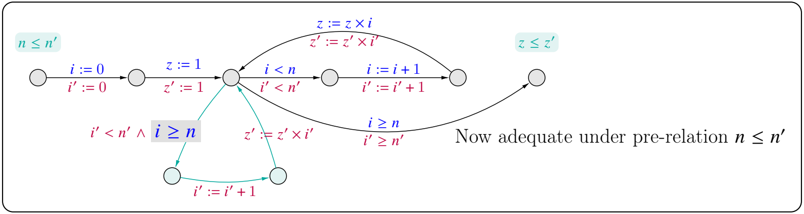
# Adequacy: covering all executions

$c_0 \;\widehat{=}\; i := 0;\; z := 1;\;$ while $i < n$ do $i := i + 1;\; z := z \times i;\;$ od

$c_0 | c_0 : n \le n' \approx\!\!> z \le z'$

Lockstep alignment not adequate



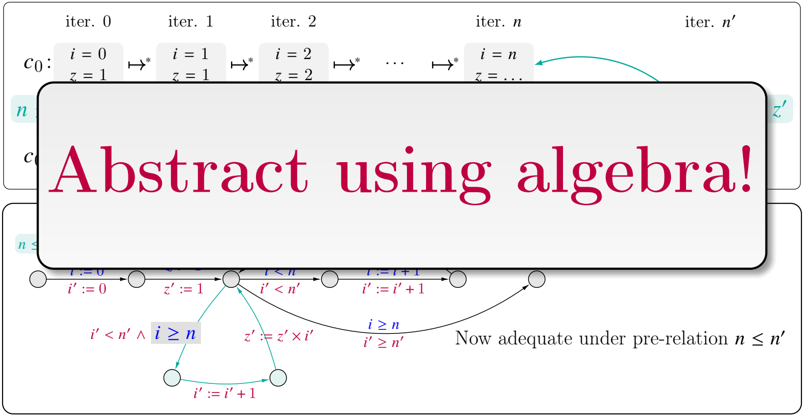

To establish $c|d : \mathcal{R} \approx\!\!> \mathcal{S}$, reason in terms of an $\mathcal{R}$-adequate alignment

Product is adequate if it covers all pairs of behaviors

# Adequacy: covering all executions

$c_0 \;\widehat{=}\; i := 0;\ z := 1;\ \text{while } i < n \text{ do } i := i+1;\ z := z \times i;\ \text{od}$

$c_0 | c_0 : n \leq n' \approx\!> z \leq z'$ Lockstep alignment not adequate




## Abstract using algebra!

To establish $c|d : \mathcal{R} \approx\!> \mathcal{S}$, reason in terms of an $\mathcal{R}$-adequate alignment

Product is adequate if it covers all pairs of behaviors

# Main contributions and outline of talk

- Introduce BiKAT, an algebra of alignment products
  - Equational proofs of adequacy
  - Equational proofs of correctness
- Derive proof rules for relational Hoare logics in BiKAT
  - Shows they hold in every model of BiKAT, including relations and traces
- Characterize forward/backward simulation, equationally
  - This characterization is used to derive new inference rules

# Recall: Kleene Algebra with Tests (KAT)

A KAT is $(\mathbb{A}, \mathbb{B}, +, ;, *, \neg, 1, 0)$ where $\mathbb{B} \subseteq \mathbb{A}$,

$(\mathbb{A}, +, ;, *, 1, 0)$ forms a Kleene algebra, and

$(\mathbb{B}, +, ;, \neg, 1, 0)$ forms a Boolean algebra

# Recall: Kleene Algebra with Tests (KAT)

A KAT is $(\mathbb{A}, \mathbb{B}, +, ;, *, \neg, 1, 0)$ where $\mathbb{B} \subseteq \mathbb{A}$,

$(\mathbb{A}, +, ;, *, 1, 0)$ forms a Kleene algebra, and

$(\mathbb{B}, +, ;, \neg, 1, 0)$ forms a Boolean algebra

Encoding programs

| | | |
|---|---|---|
| $c; d$ | $\longrightarrow$ | $c; d$ |
| if $e$ then $c$ else $d$ fi | $\longrightarrow$ | $e; c + \neg e; d$ |
| while $e$ do $c$ od | $\longrightarrow$ | $(e; c)^*; \neg e$ |

# Recall: Kleene Algebra with Tests (KAT)

A KAT is $(\mathbb{A}, \mathbb{B}, +, ;, *, \neg, 1, 0)$ where $\mathbb{B} \subseteq \mathbb{A}$,

$(\mathbb{A}, +, ;, *, 1, 0)$ forms a Kleene algebra, and

$(\mathbb{B}, +, ;, \neg, 1, 0)$ forms a Boolean algebra

Encoding programs

$$
\begin{array}{lcl}
c; d & \longrightarrow & c; d \\
\textsf{if } e \textsf{ then } c \textsf{ else } d \textsf{ fi} & \longrightarrow & e; c + \neg e; d \\
\textsf{while } e \textsf{ do } c \textsf{ od} & \longrightarrow & (e; c)^*; \neg e
\end{array}
$$

Encoding Hoare triples

$\{p\}\, c\, \{q\}$ expressed as $p; c \leq p; c; q$

where $x \leq y$ iff $x + y = y$

# Recall: Kleene Algebra with Tests (KAT)

A KAT is $(\mathbb{A}, \mathbb{B}, +, ;, *, \neg, 1, 0)$ where $\mathbb{B} \subseteq \mathbb{A}$,

$(\mathbb{A}, +, ;, *, 1, 0)$ forms a Kleene algebra, and

$(\mathbb{B}, +, ;, \neg, 1, 0)$ forms a Boolean algebra

Encoding programs

| | | |
|---|---|---|
| $c; d$ | $\longrightarrow$ | $c; d$ |
| if $e$ then $c$ else $d$ fi | $\longrightarrow$ | $e; c + \neg e; d$ |
| while $e$ do $c$ od | $\longrightarrow$ | $(e; c)^*; \neg e$ |

Encoding Hoare triples

$\{p\}\, c\, \{q\}$ expressed as $p; c \leq p; c; q$

where $x \leq y$ iff $x + y = y$

Relational model: $\mathbb{A}$ is relations on $\Sigma$, $\mathbb{B}$ coreflexives $(\subseteq id_\Sigma)$

Also: models based on traces

# Recall: Kleene Algebra with Tests (KAT)

A KAT is $(\mathbb{A}, \mathbb{B}, +, ;, *, \neg, 1, 0)$ where $\mathbb{B} \subseteq \mathbb{A}$,

$(\mathbb{A}, +, ;, *, 1, 0)$ forms a Kleene algebra, and

$(\mathbb{B}, +, ;, \neg, 1, 0)$ forms a Boolean algebra

Encoding programs

| | | |
|---|---|---|
| $c; d$ | $\rightarrow$ | $c; d$ |
| $\mathsf{if}\ e\ \mathsf{then}\ c\ \mathsf{else}\ d\ \mathsf{fi}$ | $\rightarrow$ | $e; c + \neg e; d$ |
| $\mathsf{while}\ e\ \mathsf{do}\ c\ \mathsf{od}$ | $\rightarrow$ | $(e; c)^*; \neg e$ |

Encoding Hoare triples

$\{p\}\, c\, \{q\}$ expressed as $p; c \leq p; c; q$

where $x \leq y$ iff $x + y = y$

Relational model: $\mathbb{A}$ is relations on $\Sigma$, $\mathbb{B}$ coreflexives ($\subseteq id_\Sigma$)

Also: models based on traces

Kozen: KAT subsumes propositional HL

# Introducing BiKAT

A BiKAT over a KAT $(\mathbb{A}, \mathbb{B}, +, ;, *, \neg, 1, 0)$ is

- itself a KAT $(\ddot{\mathbb{A}}, \ddot{\mathbb{B}}, \oplus, \mathbin{⨟}, \circledast, \ddot{\neg}, \ddot{1}, \ddot{0})$, along with two operations

  (*left* embed) $\langle \_ ] : \mathbb{A} \to \ddot{\mathbb{A}}$

  (*right* embed) $[ \_ \rangle : \mathbb{A} \to \ddot{\mathbb{A}}$
- that are homomorphic:

  $\langle x; y] = \langle x] \mathbin{⨟} \langle y]$; $\langle x + y] = \langle x] \oplus \langle y]$; $\langle x^*] = \langle x]^{\circledast}$
- and satisfy

  $\langle x] \mathbin{⨟} [y\rangle = [y\rangle \mathbin{⨟} \langle x]$ (left-right commute)

Models: relations over pairs, sets of trace pairs, sets of pair-traces

# Introducing BiKAT

A BiKAT over a KAT $(\mathbb{A}, \mathbb{B}, +, ;, *, \neg, 1, 0)$ is

- itself a KAT $(\ddot{\mathbb{A}}, \ddot{\mathbb{B}}, \oplus, \fatsemi, \circledast, \ddot{\neg}, \ddot{1}, \ddot{0})$, along with two operations
  (*left* embed) $\langle \_ ] : \mathbb{A} \to \ddot{\mathbb{A}}$
  (*right* embed) $[ \_ \rangle : \mathbb{A} \to \ddot{\mathbb{A}}$
- that are homomorphic:
  $\langle x; y] = \langle x] \fatsemi \langle y]$; $\langle x + y] = \langle x] \oplus \langle y]$; $\langle x^*] = \langle x]^{\circledast}$
- and satisfy
  $\langle x] \fatsemi [y\rangle = [y\rangle \fatsemi \langle x]$ (left-right commute)

Models: relations over pairs, sets of trace pairs, sets of pair-traces

Definition: $\langle c \,|\, d\rangle \mathrel{\widehat{=}} \langle c] \fatsemi [d\rangle$

# Introducing BiKAT

A BiKAT over a KAT $(\mathbb{A}, \mathbb{B}, +, ;, *, \neg, 1, 0)$ is

- itself a KAT $(\ddot{\mathbb{A}}, \ddot{\mathbb{B}}, \oplus, \ddot{;}, \circledast, \ddot{\neg}, \ddot{1}, \ddot{0})$, along with two operations
  $$(\textit{left}\text{ embed}) \quad \langle \_ ] : \mathbb{A} \to \ddot{\mathbb{A}}$$
  $$(\textit{right}\text{ embed}) \quad [ \_ \rangle : \mathbb{A} \to \ddot{\mathbb{A}}$$
- that are homomorphic:
  $$\langle x; y] = \langle x] \,\ddot{;}\, \langle y];\ \langle x + y] = \langle x] \oplus \langle y];\ \langle x^*] = \langle x]^{\circledast}$$
- and satisfy
  $$\langle x] \,\ddot{;}\, [y\rangle = [y\rangle \,\ddot{;}\, \langle x] \qquad \text{(left-right commute)}$$

Models: relations over pairs, sets of trace pairs, sets of pair-traces

Definition: $\langle c \mid d\rangle \mathrel{\widehat{=}} \langle c] \,\ddot{;}\, [d\rangle$

Example alignment and adequacy proof:

$$\langle a; b \mid c; d\rangle$$

# Introducing BiKAT

A BiKAT over a KAT $(\mathbb{A}, \mathbb{B}, +, ;, *, \neg, 1, 0)$ is

- itself a KAT $(\ddot{\mathbb{A}}, \ddot{\mathbb{B}}, \oplus, \fatsemi, \circledast, \ddot{\neg}, \ddot{1}, \ddot{0})$, along with two operations

  (*left* embed) $\langle\_] : \mathbb{A} \to \ddot{\mathbb{A}}$

  (*right* embed) $[\_\rangle : \mathbb{A} \to \ddot{\mathbb{A}}$
- that are homomorphic:

  $\langle x; y] = \langle x] \fatsemi \langle y]$; $\langle x + y] = \langle x] \oplus \langle y]$; $\langle x^*] = \langle x]^{\circledast}$
- and satisfy

  $\langle x] \fatsemi [y\rangle = [y\rangle \fatsemi \langle x]$ (left-right commute)

Models: relations over pairs, sets of trace pairs, sets of pair-traces

Definition: $\langle c \,|\, d\rangle \mathrel{\widehat{=}} \langle c] \fatsemi [d\rangle$

Example alignment and adequacy proof:

$$\langle a; b \,|\, c; d\rangle \quad = \quad \langle a; b] \fatsemi [c; d\rangle$$

# Introducing BiKAT

A BiKAT over a KAT $(\mathbb{A}, \mathbb{B}, +, ;, *, \neg, 1, 0)$ is

- itself a KAT $(\ddot{\mathbb{A}}, \ddot{\mathbb{B}}, \oplus, \fatsemi, \circledast, \ddot{\neg}, \ddot{1}, \ddot{0})$, along with two operations
  (*left* embed) $\langle \_ ] : \mathbb{A} \to \ddot{\mathbb{A}}$
  (*right* embed) $[ \_ \rangle : \mathbb{A} \to \ddot{\mathbb{A}}$
- that are homomorphic:
  $\langle x; y] = \langle x] \fatsemi \langle y]$; $\langle x + y] = \langle x] \oplus \langle y]$; $\langle x^*] = \langle x]^{\circledast}$
- and satisfy
  $\langle x] \fatsemi [y\rangle = [y\rangle \fatsemi \langle x]$ (left-right commute)

Models: relations over pairs, sets of trace pairs, sets of pair-traces

Definition: $\langle c \,|\, d\rangle \mathrel{\widehat{=}} \langle c] \fatsemi [d\rangle$

Example alignment and adequacy proof:

$$\begin{aligned} \langle a; b \,|\, c; d\rangle &= \langle a; b] \fatsemi [c; d\rangle \\ &= \langle a] \fatsemi \langle b] \fatsemi [c\rangle \fatsemi [d\rangle \end{aligned}$$

# Introducing BiKAT

A BiKAT over a KAT $(\mathbb{A}, \mathbb{B}, +, ;, *, \neg, 1, 0)$ is

- itself a KAT $(\ddot{\mathbb{A}}, \ddot{\mathbb{B}}, \oplus, \fatsemi, \circledast, \ddot{\neg}, \ddot{1}, \ddot{0})$, along with two operations

  (*left* embed) $\langle \_ ] : \mathbb{A} \to \ddot{\mathbb{A}}$

  (*right* embed) $[ \_ \rangle : \mathbb{A} \to \ddot{\mathbb{A}}$
- that are homomorphic:

  $\langle x; y] = \langle x] \fatsemi \langle y]$; $\langle x + y] = \langle x] \oplus \langle y]$; $\langle x^*] = \langle x]^\circledast$
- and satisfy

  $\langle x] \fatsemi [y\rangle = [y\rangle \fatsemi \langle x]$ (left-right commute)

Models: relations over pairs, sets of trace pairs, sets of pair-traces

Definition: $\langle c \,|\, d\rangle \mathrel{\widehat{=}} \langle c] \fatsemi [d\rangle$

Example alignment and adequacy proof:

$$\begin{aligned}\langle a; b \,|\, c; d\rangle &= \langle a; b] \fatsemi [c; d\rangle \\ &= \langle a] \fatsemi \langle b] \fatsemi [c\rangle \fatsemi [d\rangle \\ &= \langle a] \fatsemi [c\rangle \fatsemi \langle b] \fatsemi [d\rangle\end{aligned}$$

# Introducing BiKAT

A BiKAT over a KAT $(\mathbb{A}, \mathbb{B}, +, ;, *, \neg, 1, 0)$ is

- itself a KAT $(\ddot{\mathbb{A}}, \ddot{\mathbb{B}}, \oplus, \fatsemi, \circledast, \ddot{\neg}, \ddot{1}, \ddot{0})$, along with two operations
  (*left* embed) $\langle \_ ] : \mathbb{A} \to \ddot{\mathbb{A}}$
  (*right* embed) $[ \_ \rangle : \mathbb{A} \to \ddot{\mathbb{A}}$
- that are homomorphic:
  $\langle x; y] = \langle x] \fatsemi \langle y];\ \langle x + y] = \langle x] \oplus \langle y];\ \langle x^*] = \langle x]^{\circledast}$
- and satisfy
  $\langle x] \fatsemi [y\rangle = [y\rangle \fatsemi \langle x]$ (left-right commute)

Models: relations over pairs, sets of trace pairs, sets of pair-traces

Definition: $\langle c \mid d\rangle \mathrel{\widehat{=}} \langle c] \fatsemi [d\rangle$

Example alignment and adequacy proof:

$$\begin{aligned}\langle a; b \mid c; d\rangle &= \langle a; b] \fatsemi [c; d\rangle \\ &= \langle a] \fatsemi \langle b] \fatsemi [c\rangle \fatsemi [d\rangle \\ &= \langle a] \fatsemi [c\rangle \fatsemi \langle b] \fatsemi [d\rangle \quad = \quad \langle a \mid c\rangle \fatsemi \langle b \mid d\rangle\end{aligned}$$

# ∀∀ properties and adequacy in BiKAT

∀∀ property in relational model:

$$c \mid d : \mathcal{R} \approx\!\!> \mathcal{S} \qquad \forall \quad \begin{tikzcd} \sigma \arrow[r, mapsto, "c"] \arrow[d, "\mathcal{R}"'] & \tau \\ \sigma' \arrow[r, mapsto, "d"] & \tau' \end{tikzcd} \quad \Longrightarrow \quad \begin{tikzcd} \tau \arrow[d, "\mathcal{S}"] \\ \tau' \end{tikzcd}$$

Equivalent to $\hat{\mathcal{R}} \fatsemi \langle c \mid d \rangle \leq \hat{\mathcal{R}} \fatsemi \langle c \mid d \rangle \fatsemi \hat{\mathcal{S}}$ ($\hat{\mathcal{R}}$ is coreflexive lift[1])

[1] i.e., $\hat{\mathcal{R}} \mathrel{\widehat{=}} \{ ((\sigma, \sigma'), (\sigma, \sigma')) \mid (\sigma, \sigma') \in \mathcal{R} \}$

# ∀∀ properties and adequacy in BiKAT

∀∀ property in relational model:

$$c|d : \mathcal{R} \mathrel{\approx\!>} \mathcal{S} \qquad \forall \qquad \begin{tikzcd} \sigma \arrow[r, mapsto, "c"] \arrow[d, "\mathcal{R}"'] & \tau \\ \sigma' \arrow[r, mapsto, "d"] & \tau' \end{tikzcd} \qquad \Longrightarrow \qquad \begin{tikzcd} \tau \arrow[d, "\mathcal{S}"] \\ \tau' \end{tikzcd}$$

Equivalent to $\boxed{\hat{\mathcal{R}} \mathbin{;} \langle c \mid d \rangle \leq \hat{\mathcal{R}} \mathbin{;} \langle c \mid d \rangle \mathbin{;} \hat{\mathcal{S}}}$ ($\hat{\mathcal{R}}$ is coreflexive lift[1])

$\langle c \mid d \rangle$ is sequential alignment: stands for $\langle c] \mathbin{;} [d\rangle$
but we want to use other kinds of alignment

[1] i.e., $\hat{\mathcal{R}} \mathrel{\widehat{=}} \{ ((\sigma, \sigma'), (\sigma, \sigma')) \mid (\sigma, \sigma') \in \mathcal{R} \}$

# $\forall\forall$ properties and adequacy in BiKAT

$\forall\forall$ property in relational model:

$$c|d : \mathcal{R} \approx\!> \mathcal{S} \qquad \forall \quad \begin{array}{ccc} \sigma & \xmapsto{\;c\;} & \tau \\ {\scriptstyle\mathcal{R}}\downarrow & & \\ \sigma' & \xmapsto{\;d\;} & \tau' \end{array} \quad \Longrightarrow \quad \begin{array}{c} \tau \\ \downarrow{\scriptstyle\mathcal{S}} \\ \tau' \end{array}$$

Equivalent to $\boxed{\hat{\mathcal{R}} \,;\, \langle c \,|\, d\rangle \le \hat{\mathcal{R}} \,;\, \langle c \,|\, d\rangle \,;\, \hat{\mathcal{S}}}$ ($\hat{\mathcal{R}}$ is coreflexive lift[1])

$\langle c \,|\, d\rangle$ is sequential alignment: stands for $\langle c] \,;\, [d\rangle$
but we want to use other kinds of alignment

| | | |
|---|---|---|
| Thm [Adequacy] | If $\hat{\mathcal{R}} \,;\, \langle c \,|\, d\rangle \le \hat{\mathcal{R}} \,;\, B$ | ($B$ is $\mathcal{R}$-adequate) |
| | and $\hat{\mathcal{R}} \,;\, B \le \hat{\mathcal{R}} \,;\, B \,;\, \hat{\mathcal{S}}$ | ($B$ is correct) |
| | then $c|d : \mathcal{R} \approx\!> \mathcal{S}$ | ($c|d$ is correct) |

[1] i.e., $\hat{\mathcal{R}} \mathrel{\widehat{=}} \{\, ((\sigma,\sigma'),(\sigma,\sigma')) \mid (\sigma,\sigma') \in \mathcal{R} \,\}$

# Example revisited in BiKAT

$$c_0 \;\widehat{=}\; i := 0;\ z := 1;\ \mathsf{while}\ i < n\ \mathsf{do}\ i := i + 1;\ z := z \times i\ \mathsf{od}$$
$$c_1 \;\widehat{=}\; i := 1;\ z := 1;\ \mathsf{while}\ i \leq n\ \mathsf{do}\ z := z \times i;\ i := i + 1\ \mathsf{od}$$

Equiv: $c_0 \mid c_1 \ :\ n = n' \ \approx\!> \ z = z'$

As KAT terms:

$$k_0 \;\widehat{=}\; i := 0;\ z := 1;\ \big([i < n];\ i := i + 1;\ z := z \times i\big)^*;\ \neg[i < n]$$
$$k_1 \;\widehat{=}\; i := 1;\ z := 1;\ \big([i \leq n];\ z := z \times i;\ i := i + 1\big)^*;\ \neg[i \leq n]$$

Assignments as prim actions; $[e]$ for tests

Correctness judgment in BiKAT:

$$[n = n'] \mathbin{;} \langle k_0 \mid k_1 \rangle \leq [n = n'] \mathbin{;} \langle k_0 \mid k_1 \rangle \mathbin{;} [z = z']$$

# Example revisited in BiKAT

$$\Big[n = n'\Big] ⨟ \left\langle \begin{array}{l} i := 0; z := 1; ([i < n];\ i := i + 1;\ z := z \times i)^*; \neg[i < n] \\ \mid i := 1; z := 1; ([i \leq n];\ z := z \times i;\ i := i + 1)^*; \neg[i \leq n] \end{array} \right\rangle$$

# Example revisited in BiKAT

$$\left[n = n'\right] ⨾ \left\langle \begin{array}{l} i := 0; z := 1; ([i < n];\ i := i + 1;\ z := z \times i)^*; \neg[i < n] \\ \mid i := 1; z := 1; ([i \leq n];\ z := z \times i;\ i := i + 1)^*; \neg[i \leq n] \end{array} \right\rangle$$

1. Embeddings are homomorphic

$$\begin{array}{l} [n = n'] ⨾ \langle i := 0] ⨾ \langle z := 1] ⨾ \langle ([i < n]; \ldots)^*] ⨾ \langle \neg[i < n]] \\ \qquad ⨾ [i := 1\rangle ⨾ [z := 1\rangle ⨾ [([i \leq n]; \ldots)^*\rangle ⨾ [\neg[i \leq n]\rangle \end{array}$$

# Example revisited in BiKAT

$$\Big[n = n'\Big] ⨟ \Big\langle \begin{array}{l} i := 0; z := 1; ([i < n];\ i := i + 1;\ z := z \times i)^*; \neg[i < n] \\ \mid i := 1; z := 1; ([i \leq n];\ z := z \times i;\ i := i + 1)^*; \neg[i \leq n] \end{array} \Big\rangle$$

1. Embeddings are homomorphic

$$\begin{array}{l} [n = n'] ⨟ \langle i := 0] ⨟ \langle z := 1] ⨟ \langle ([i < n]; \ldots)^*] ⨟ \langle \neg[i < n]] \\ \quad ⨟ [i := 1\rangle ⨟ [z := 1\rangle ⨟ [([i \leq n]; \ldots)^*\rangle ⨟ [\neg[i \leq n]\rangle \end{array}$$

2. Commute embeddings and align initial points of interest

$$\Big[n = n'\Big] ⨟ \boxed{\begin{array}{l} \langle i := 0 \\ \mid i := 1\rangle \end{array}} ⨟ \boxed{\begin{array}{l} \langle z := 1 \\ \mid z := 1\rangle \end{array}} ⨟ \begin{array}{l} \langle ([i < n]; \ldots)^* \\ \mid ([i \leq n]; \ldots)^*\rangle \end{array} ⨟ \begin{array}{l} \langle \neg[i < n] \\ \mid \neg[i \leq n]\rangle \end{array}$$

# Example revisited in BiKAT

$$[n = n'] ⨾ \left\langle \begin{array}{l} i := 0; z := 1; ([i < n];\ i := i+1;\ z := z \times i)^*; \neg[i < n] \\ \mid i := 1; z := 1; ([i \leq n];\ z := z \times i;\ i := i+1)^*; \neg[i \leq n] \end{array} \right\rangle$$

1. Embeddings are homomorphic

$$[n = n'] ⨾ \langle i := 0] ⨾ \langle z := 1] ⨾ \langle ([i < n]; \ldots)^*] ⨾ \langle \neg[i < n]]$$
$$⨾ [i := 1\rangle ⨾ [z := 1\rangle ⨾ [([i \leq n]; \ldots)^*\rangle ⨾ [\neg[i \leq n]\rangle$$

2. Commute embeddings and align initial points of interest

$$[n = n'] ⨾ \left\langle \begin{array}{l} i := 0 \\ \mid i := 1 \end{array} \right\rangle ⨾ \left\langle \begin{array}{l} z := 1 \\ \mid z := 1 \end{array} \right\rangle ⨾ \left\langle \begin{array}{l} ([i < n]; \ldots)^* \\ \mid ([i \leq n]; \ldots)^* \end{array} \right\rangle ⨾ \left\langle \begin{array}{l} \neg[i < n] \\ \mid \neg[i \leq n] \end{array} \right\rangle$$

3. Introduce invariant using hypotheses about primitives

$$[n = n'] ⨾ \left\langle \begin{array}{l} i := 0 \\ \mid i := 1 \end{array} \right\rangle ⨾ \left\langle \begin{array}{l} z := 1 \\ \mid z := 1 \end{array} \right\rangle ⨾ \boxed{\mathcal{P}} ⨾ \left\langle \begin{array}{l} ([i < n]; \ldots)^* \\ \mid ([i \leq n]; \ldots)^* \end{array} \right\rangle ⨾ \left\langle \begin{array}{l} \neg[i < n] \\ \mid \neg[i \leq n] \end{array} \right\rangle$$

where $\mathcal{P} \mathrel{\widehat{=}} \boxed{[i+1 = i'] ⨾ [z = z'] ⨾ [n = n']}$

e.g., $\langle i := 0 \mid i := 1\rangle = \langle i := 0 \mid i := 1\rangle ⨾ [i+1 = i']$

$[n = n'] ⨾ \langle z := 1 \mid z := 1\rangle = [n = n'] ⨾ \langle z := 1 \mid z := 1\rangle ⨾ [n = n']$

# Example revisited in BiKAT

4. Use general lemmas to construct adequate alignments

$$\begin{bmatrix} n = n' \\ i{+}1 = i' \\ z = z' \end{bmatrix} ⨾ \langle ([i<n];\ldots)^* \mid ([i\leq n];\ldots)^* \rangle = \begin{bmatrix} n = n' \\ i{+}1 = i' \\ z = z' \end{bmatrix} ⨾ \left\langle [i<n];\ldots \middle| [i\leq n];\ldots \right\rangle^{\circledast}$$

If $\mathcal{R} \leq [e = e']$ then $\mathcal{R} ⨾ \langle (e;c)^* \mid (e';c')^* \rangle = \mathcal{R} ⨾ \langle e;c \mid e';c' \rangle^{\circledast}$

# Example revisited in BiKAT

4. Use general lemmas to construct adequate alignments

$$\begin{bmatrix} n = n' \\ i{+}1 = i' \\ z = z' \end{bmatrix} ⨾ \begin{array}{l}\langle\, ([i<n];\ldots)^* \\ \mid ([i\leq n];\ldots)^* \rangle\end{array} = \begin{bmatrix} n = n' \\ i{+}1 = i' \\ z = z' \end{bmatrix} ⨾ \Big\langle [i<n];\ldots \Big| [i\leq n];\ldots \Big\rangle^{\circledast}$$

If $\mathcal{R} \leq [e = e']$ then $\mathcal{R} ⨾ \langle (e;c)^* \mid (e';c')^* \rangle = \mathcal{R} ⨾ \langle e;c \mid e';c' \rangle^{\circledast}$

5. Reason about loop using standard KAT lemmas

$$\mathcal{P} ⨾ \langle [i<n];\ldots \mid [i\leq n];\ldots \rangle^{\circledast} = \boxed{\mathcal{P} ⨾ \langle [i<n];\ldots \mid [i\leq n];\ldots \rangle^{\circledast} ⨾ \mathcal{P}}$$

If $\mathcal{R} ⨾ B = \mathcal{R} ⨾ B ⨾ \mathcal{R}$ then $\mathcal{R} ⨾ B^{\circledast} = \mathcal{R} ⨾ B^{\circledast} ⨾ \mathcal{R}$

# Example revisited in BiKAT

4. Use general lemmas to construct adequate alignments

$$\begin{bmatrix} n = n' \\ i{+}1 = i' \\ z = z' \end{bmatrix} ⨟ \begin{array}{l}\langle\, ([i<n];\ldots)^* \\ \mid ([i\le n];\ldots)^*\rangle\end{array} = \begin{bmatrix} n = n' \\ i{+}1 = i' \\ z = z' \end{bmatrix} ⨟ \Big\langle [i<n];\ldots \,\Big|\, [i\le n];\ldots \Big\rangle^{\circledast}$$

If $\mathcal{R} \le [e = e']$ then $\mathcal{R} ⨟ \langle (e;c)^* \mid (e';c')^* \rangle = \mathcal{R} ⨟ \langle e;c \mid e';c' \rangle^{\circledast}$

5. Reason about loop using standard KAT lemmas

$$\mathcal{P} ⨟ \langle [i<n];\ldots \mid [i\le n];\ldots \rangle^{\circledast} = \mathcal{P} ⨟ \langle [i<n];\ldots \mid [i\le n];\ldots \rangle^{\circledast} ⨟ \mathcal{P}$$

If $\mathcal{R} ⨟ B = \mathcal{R} ⨟ B ⨟ \mathcal{R}$ then $\mathcal{R} ⨟ B^{\circledast} = \mathcal{R} ⨟ B^{\circledast} ⨟ \mathcal{R}$

6. Reason about post using boolean algebra

$$\begin{array}{rl} & \ldots ⨟ \langle [i<n];\ldots \mid [i\le n];\ldots \rangle^{\circledast} ⨟ \mathcal{P} ⨟ \langle \neg[i<n] \mid \neg[i\le n] \rangle \\ \le & \ldots ⨟ \langle [i<n];\ldots \mid [i\le n];\ldots \rangle^{\circledast} ⨟ \langle \neg[i<n] \mid \neg[i\le n] \rangle ⨟ [z = z'] \end{array}$$

# Example revisited in BiKAT

4. Use general lemmas to construct adequate alignments

$$\begin{bmatrix} n = n' \\ i{+}1 = i' \\ z = z \end{bmatrix} ; \langle ([i<n]; \ldots)^* \quad = \quad \begin{bmatrix} n = n' \\ i{+}1 = i' \end{bmatrix} ; \langle [i<n]; \quad | [i<n]; \ldots \rangle^{\circledast}$$

If $\mathcal{R} \leq [$

5. Reason

$\mathcal{P} \fatsemi \langle[ \quad \rangle^{\circledast} \fatsemi \mathcal{P}$

If $\mathcal{R} \fatsemi B$

$$\begin{array}{lll} & [n = n'] \fatsemi \langle k_0 \mid k_1 \rangle & \text{(original progs)} \\ = & \ldots & \\ = & [n = n'] \fatsemi B & (B \text{ is adequate}) \\ \leq & \ldots & \\ \leq & [n = n'] \fatsemi B \fatsemi [z = z'] & (B \text{ is correct}) \end{array}$$

6. Reason about post using boolean algebra

$$\begin{array}{ll} & \ldots \fatsemi \langle [i < n]; \ldots \mid [i \leq n]; \ldots \rangle^{\circledast} \fatsemi \mathcal{P} \fatsemi \langle \neg[i < n] \mid \neg[i \leq n] \rangle \\ \leq & \ldots \fatsemi \langle [i < n]; \ldots \mid [i \leq n]; \ldots \rangle^{\circledast} \fatsemi \langle \neg[i < n] \mid \neg[i \leq n] \rangle \fatsemi [z = z'] \end{array}$$

# Thm: Rules of RHL are derivable in BiKAT

$$\frac{\mathcal{R} \Rightarrow e = e' \qquad c|c' : \mathcal{R} \approx\!> \mathcal{S} \qquad d|d' : \mathcal{R} \approx\!> \mathcal{S}}{\text{if } e \text{ then } c \text{ else } d \mid \text{if } e' \text{ then } c' \text{ else } d' : \mathcal{R} \approx\!> \mathcal{S}}$$

# Thm: Rules of RHL are derivable in BiKAT

$$\frac{\mathcal{R} \Rightarrow e = e' \qquad c|c' : \mathcal{R} \approx\!> \mathcal{S} \qquad d|d' : \mathcal{R} \approx\!> \mathcal{S}}{\textsf{if } e \textsf{ then } c \textsf{ else } d \mid \textsf{if } e' \textsf{ then } c' \textsf{ else } d' : \mathcal{R} \approx\!> \mathcal{S}}$$

Assume: $\mathcal{R} \leq [e = e']$ $\quad \mathcal{R} \fatsemi \langle c \mid c' \rangle \leq \mathcal{R} \fatsemi \langle c \mid c' \rangle \fatsemi \mathcal{S}$ $\quad$ similar for $d, d'$

Prove:

$$\begin{aligned}
& \mathcal{R} \fatsemi \langle e; c + \neg e; d \mid e'; c' + \neg e'; d' \rangle \\
= \;& \mathcal{R} \fatsemi (\langle e; c \mid e'; c' \rangle + \langle e; c \mid \neg e'; d' \rangle + \langle \neg e; d \mid e'; c' \rangle + \ldots) \\
& \qquad \{ \textit{ distribute } \} \\
= \;& \mathcal{R} \fatsemi \langle e; c \mid e'; c' \rangle + \mathcal{R} \fatsemi \langle e; c \mid \neg e'; d' \rangle + \mathcal{R} \fatsemi \langle \neg e; d \mid e'; c' \rangle + \ldots \\
& \qquad \{ \textit{ use } \mathcal{R} \leq [e = e'] \textit{ to cancel out terms } \} \\
= \;& \mathcal{R} \fatsemi \langle e; c \mid e'; c' \rangle + \mathcal{R} \fatsemi \langle \neg e; d \mid \neg e'; d' \rangle \\
& \qquad \{ \textit{ assumptions about } c \textit{ and } d \} \\
\leq \;& \mathcal{R} \fatsemi \langle e; c \mid e'; c' \rangle \fatsemi \mathcal{S} + \mathcal{R} \fatsemi \langle \neg e; d \mid \neg e'; d' \rangle \fatsemi \mathcal{S}
\end{aligned}$$

# ∀∃ properties

For nondeterministic programs, possibilistic noninterference, possibilistic equivalence, refinement, co-termination …

Forward simulation (rel model): $c|d : \mathcal{R} \overset{\exists}{\approx\!\!>} \mathcal{S}$

$$
\forall \quad
\begin{tikzcd}
\sigma \arrow[r, mapsto, "c"] \arrow[d, "\mathcal{R}"'] & \tau \\
\sigma' &
\end{tikzcd}
\qquad \exists \qquad
\begin{tikzcd}
 & \tau \arrow[d, "\mathcal{S}"] \\
\sigma' \arrow[r, mapsto, "d"] & \tau'
\end{tikzcd}
$$

# ∀∃ properties

For nondeterministic programs, possibilistic noninterference, possibilistic equivalence, refinement, co-termination …

Forward simulation (rel model): $c|d : \mathcal{R} \overset{\exists}{\approx\!>} \mathcal{S}$

$$\begin{tikzcd} \forall & \sigma \arrow[r, mapsto, "c"] \arrow[d, "\mathcal{R}"'] & \tau & \exists & & \tau \arrow[d, "\mathcal{S}"] \\ & \sigma' & & & \sigma' \arrow[r, mapsto, "d"] & \tau' \end{tikzcd}$$

Backward simulation (rel model): $c|d : \mathcal{R} \overset{\exists\leftarrow}{\approx\!>} \mathcal{S}$

$$\begin{tikzcd} \forall & \sigma \arrow[r, mapsto, "c"] & \tau \arrow[d, "\mathcal{S}"] & \exists & \sigma \arrow[d, "\mathcal{R}"'] & \\ & & \tau' & & \sigma' \arrow[r, mapsto, "d"] & \tau' \end{tikzcd}$$

# Witness technique for ∀∃ properties

$c|d : \mathcal{R} \overset{\exists}{\approx\!\!>} \mathcal{S}$

$$\forall \quad \begin{tikzcd} \sigma \arrow[r, mapsto, "c"] \arrow[d, "\mathcal{R}"'] & \tau \\ \sigma' & \end{tikzcd} \qquad \exists \quad \begin{tikzcd} & \tau \arrow[d, "\mathcal{S}"] \\ \sigma' \arrow[r, mapsto, "d"] & \tau' \end{tikzcd}$$

Thm: $c|d : \mathcal{R} \overset{\exists}{\approx\!\!>} \mathcal{S}$ iff there exists a BiKAT **witness** term $W$,

$$\begin{array}{ll} \hat{\mathcal{R}} \mathbin{;} W \le \hat{\mathcal{R}} \mathbin{;} W \mathbin{;} \hat{\mathcal{S}} & \text{(witness ∀∀ correct)} \\ \hat{\mathcal{R}} \mathbin{;} \langle c] \le W \mathbin{;} [\mathbf{top}\rangle & \text{(witness overapproximates } c\text{)} \\ \hat{\mathcal{R}} \mathbin{;} W \le \langle \mathbf{top} \,|\, d\rangle & \text{(witness underapproximates } d\text{)} \end{array}$$

# Witness technique for ∀∃ properties

$c|d : \mathcal{R} \overset{\exists}{\approx\!>} \mathcal{S}$ $\quad \forall \quad$ $\begin{array}{ccc} \sigma & \overset{c}{\longmapsto} & \tau \\ \mathcal{R}\downarrow & & \\ \sigma' & & \end{array}$ $\quad \exists \quad$ $\begin{array}{ccc} & & \tau \\ & & \downarrow\mathcal{S} \\ \sigma' & \overset{d}{\longmapsto} & \tau' \end{array}$

Thm: $c|d : \mathcal{R} \overset{\exists}{\approx\!>} \mathcal{S}$ iff there exists a BiKAT **witness** term $W$,

$$\begin{array}{ll} \hat{\mathcal{R}} \fatsemi W \leq \hat{\mathcal{R}} \fatsemi W \fatsemi \hat{\mathcal{S}} & \text{(witness ∀∀ correct)} \\ \hat{\mathcal{R}} \fatsemi \langle c] \leq W \fatsemi [\mathbf{top}\rangle & \text{(witness overapproximates } c\text{)} \\ \hat{\mathcal{R}} \fatsemi W \leq \langle \mathbf{top} \mid d\rangle & \text{(witness underapproximates } d\text{)} \end{array}$$

Example: $\begin{array}{l} d_0 \mathrel{\widehat{=}} x := \mathit{any};\ z := x \\ d_1 \mathrel{\widehat{=}} y := \mathit{any};\ z := y{+}1 \end{array}$ $\qquad d_0|d_1 : \mathit{true} \overset{\exists}{\approx\!>} z = z'$

# Witness technique for ∀∃ properties

$$c|d : \mathcal{R} \overset{\exists}{\approx\!\!>} \mathcal{S} \qquad \forall \begin{tikzcd} \sigma \arrow[r, mapsto, "c"] \arrow[d, "\mathcal{R}"'] & \tau \\ \sigma' & \end{tikzcd} \qquad \exists \begin{tikzcd} & \tau \arrow[d, "\mathcal{S}"] \\ \sigma' \arrow[r, mapsto, "d"] & \tau' \end{tikzcd}$$

Thm: $c|d : \mathcal{R} \overset{\exists}{\approx\!\!>} \mathcal{S}$ iff there exists a BiKAT **witness** term $W$,

$$\begin{array}{ll} \hat{\mathcal{R}} \fatsemi W \leq \hat{\mathcal{R}} \fatsemi W \fatsemi \hat{\mathcal{S}} & \text{(witness ∀∀ correct)} \\ \hat{\mathcal{R}} \fatsemi \langle c] \leq W \fatsemi [\mathbf{top}\rangle & \text{(witness overapproximates } c\text{)} \\ \hat{\mathcal{R}} \fatsemi W \leq \langle \mathbf{top} \mid d \rangle & \text{(witness underapproximates } d\text{)} \end{array}$$

Example:

$$d_0 \mathrel{\widehat{=}} x := any;\ z := x$$
$$d_1 \mathrel{\widehat{=}} y := any;\ z := y{+}1 \qquad\qquad d_0|d_1 : true \overset{\exists}{\approx\!\!>} z = z'$$

$$W \mathrel{\widehat{=}} \langle x := any \mid y := any \rangle \fatsemi \boxed{[x{-}1 = y']} \fatsemi \langle z := x \mid z := y{+}1 \rangle$$

# Thm: forward simulation rules are derivable in BiKAT

$c|d : \mathcal{R} \overset{\exists}{\rightsquigarrow} \mathcal{S}$ $\quad\forall\quad$ $\begin{tikzcd} \sigma \arrow[r, mapsto, "c"] \arrow[d, "\mathcal{R}"'] & \tau \\ \sigma' & \end{tikzcd}$ $\quad\exists\quad$ $\begin{tikzcd} & \tau \arrow[d, "\mathcal{S}"] \\ \sigma' \arrow[r, mapsto, "d"] & \tau' \end{tikzcd}$

Thm: $c|d : \mathcal{R} \overset{\exists}{\rightsquigarrow} \mathcal{S}$ iff there is some BiKAT term $W$ such that

$\hat{\mathcal{R}} \fatsemi W \leq \hat{\mathcal{R}} \fatsemi W \fatsemi \hat{\mathcal{S}}$ (witness ∀∀ correct)

$\hat{\mathcal{R}} \fatsemi \langle c] \leq W \fatsemi [\mathbf{top}\rangle$ (witness overapproximates $c$)

$\hat{\mathcal{R}} \fatsemi W \leq \langle \mathbf{top} \mid d\rangle$ (witness underapproximates $d$)

$$\frac{c|c' : \mathcal{P} \overset{\exists}{\rightsquigarrow} \mathcal{R} \qquad d|d' : \mathcal{R} \overset{\exists}{\rightsquigarrow} \mathcal{Q}}{c; d \mid c'; d' : \mathcal{P} \overset{\exists}{\rightsquigarrow} \mathcal{Q}}$$

$$\frac{\mathcal{P} \Rightarrow [e'\rangle \Rightarrow \langle e] \qquad c|c' : \mathcal{P} \wedge \langle e \mid e'\rangle \overset{\exists}{\rightsquigarrow} \mathcal{P} \qquad c|\mathsf{skip} : \mathcal{P} \wedge \langle e] \overset{\exists}{\rightsquigarrow} \mathcal{P}}{\mathsf{while}\ e\ \mathsf{do}\ c\ \mathsf{od} \mid \mathsf{while}\ e'\ \mathsf{do}\ c'\ \mathsf{od} : \mathcal{P} \overset{\exists}{\rightsquigarrow} \mathcal{P} \wedge \neg\langle e] \wedge \neg[e'\rangle}$$

# Conclusion

Main contributions:

- Introduce BiKAT, an algebra of alignment products
- Derive ∀∀ proof rules of relational Hoare logic
- Characterize ∀∃ judgments and derive proof rules
- Some of this has been formalized in Coq

Also in the paper:

- Discussion of related work and expressiveness
- Equational theory of BiKAT is undecidable
- Backward simulation and connections with incorrectness logic
- Other quantifier alternations: ∃∃, ∃∀
- Other number of traces: *Tri*KAT

Future work and open problems:

- Automation: can BiKAT help in finding alignments?
- Complete models for equational theory