Note: For whatever reason, $fi$ ligatures print as British pound signs, £, in the pdf-translated version of these slides. Sorry!

# Current Kansas personnel

Torben Amtoft (just arrived from Heriot-Watt; does type-based analysis of process-based systems)

Anindya Banerjee (language-based security)

David Schmidt (abstract interpretation)

Gurvan le Guernic (PhD student, joint with Thomas Jensen, Univ. Rennes, France)

Élodie-Jane Sims (PhD student, joint with Radhia Cousot, École Polytechnique, France)

**On a parallel path:** Matthew Dwyer, John Hatcliff, multiple students (Bandera Java-model-checking project)

# Current activities

Banerjee/Naumann: “Heap-design patterns” for representation independence. Current case studies: storage encapsulation in ownership-transfer situations; storage sharing in the iterator design pattern.

Connections between access control (viz., stack inspection) and information ¤ow control.

le Guernic/Banerjee/Schmidt: Formalization of Myers and Liskov’s decentralized label (“declassi£cation”) model of information-¤ow control by typing rules and powerdomain semantics. Generalization to “security design patterns.”

Sims/Schmidt: Application of separation logic to alias analysis and extension of separation logic by £xed point operators.

Schmidt: Formalization of under- and over-approximating abstract interpretations with powerdomains.

# Abstract Models of Shape

## Branching- (and Linear-) Time

What do heap-shape analysis and process algebra have in common?

David Schmidt

Kansas State University

# Labelled Kripke transition systems

$\langle \Sigma, \{\tau_\ell \subseteq \Sigma \times \Sigma \mid \ell \in \mathsf{Label}\}, \mathcal{I}_\Sigma : \Sigma \to \mathcal{P}(\mathsf{Atom})\rangle$

$$\Sigma = \{s0, s1, s2\}$$

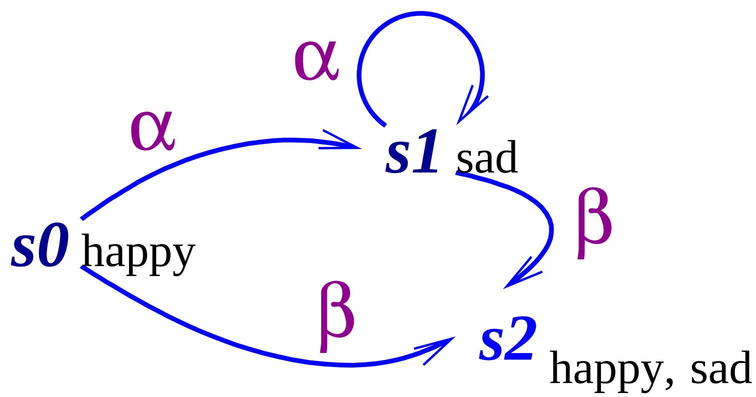
$$\tau_\alpha = \{(s0, s1), (s1, s1)\}$$

$$\tau_\beta = \{(s0, s2), (s1, s2)\}$$

$$\mathcal{I}_\Sigma(s0) = \{\mathsf{happy}\}$$

$$\mathcal{I}_\Sigma(s1) = \{\mathsf{sad}\}$$

$$\mathcal{I}_\Sigma(s2) = \{\mathsf{happy}, \mathsf{sad}\}$$




We might identify initial states, $\Sigma_0 \subseteq \Sigma$, also.

# Graph models apply to storage shapes

$\Sigma = \{c0, c1, c2\}$

$\tau_{head} = \{(c0, c0)\}$

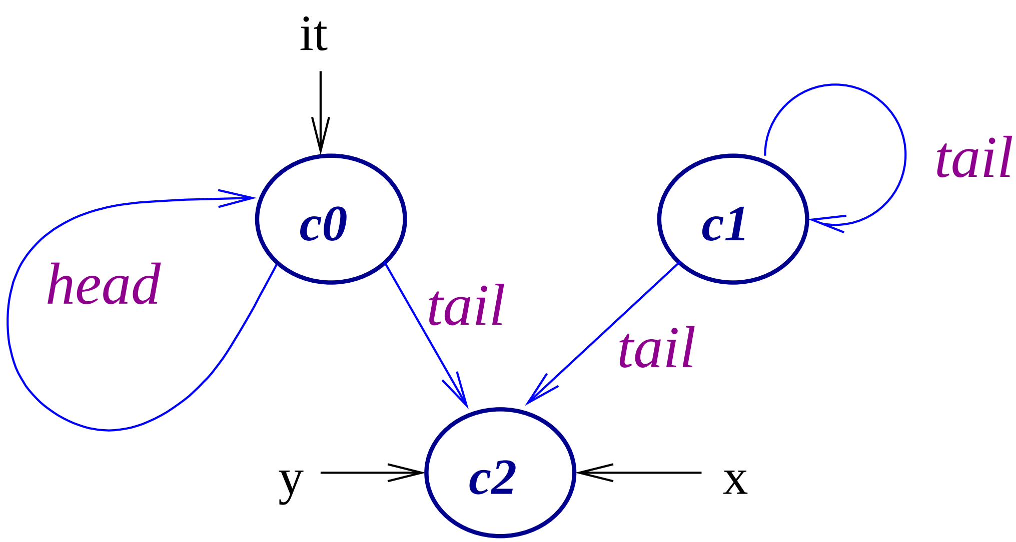
$\tau_{tail} = \{(c0, c2),$

$(c1, c1), (c1, c2)\}$

$\mathcal{I}_\Sigma(c0) = \{it\}$

$\mathcal{I}_\Sigma(c1) = \{\}$

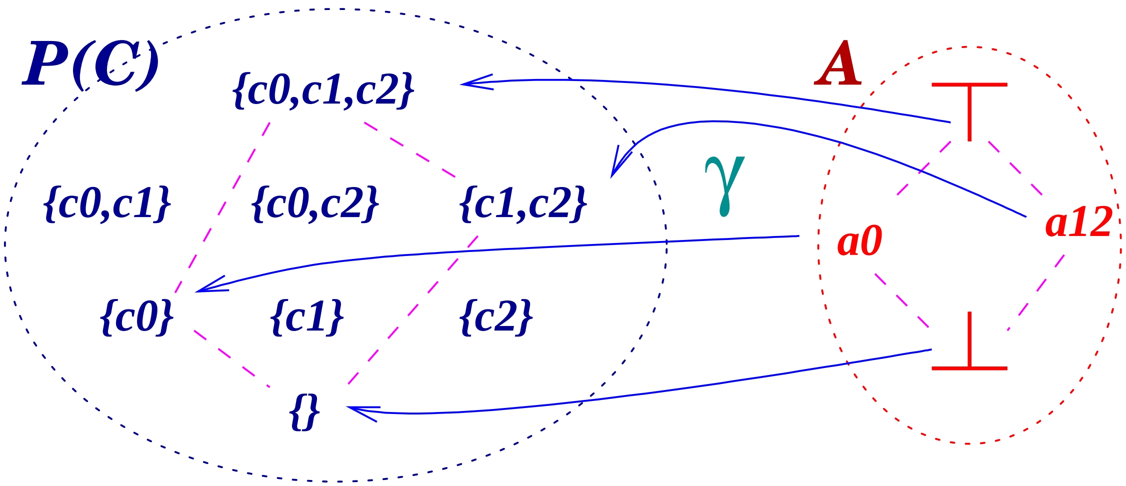
$\mathcal{I}_\Sigma(c2) = \{x, y\}$




Rather than states, the nodes now represent cells/objects.

# A Galois Connection abstracts the cells

Let $A$ be a £nite set of tokens that model sets of dynamically allocated storage cells:

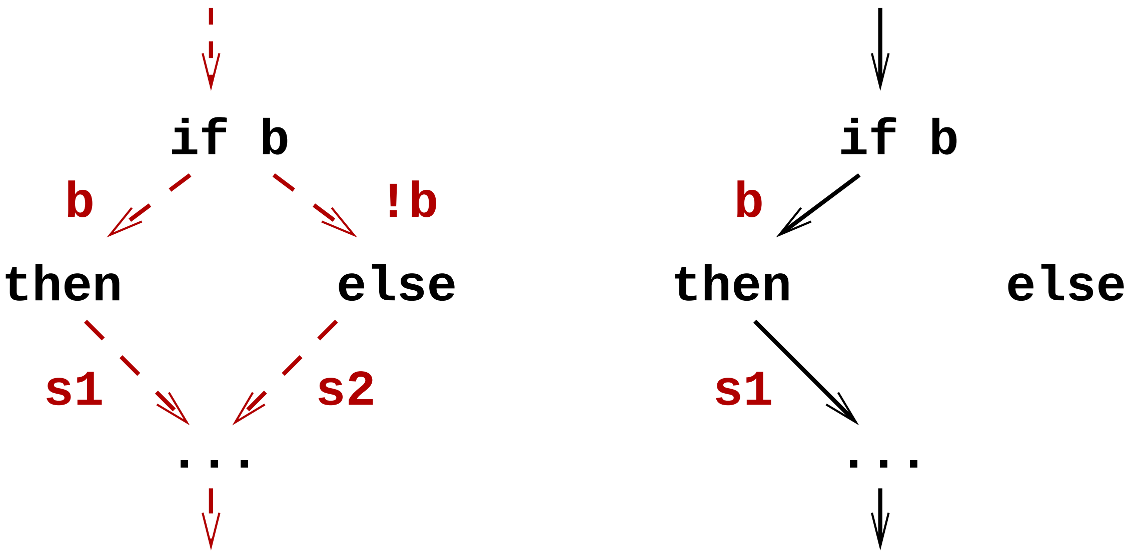



It's common to make $A$ into a complete lattice: $\perp$ denotes no cells at all, and $\top$ denotes all allocated cells.

$\gamma$ is the upper adjoint of a Galois connection; it indicates which cells are denoted by which tokens.

# What does an abstract transition denote?

what may possibly execute:




or what may possibly be pointed:

# What does "dashed" denote? *a0 - - ⇒ a1*

Overapproximation — $1/2$ — may/possibly: The corresponding concrete structure may or may not possess this transition, but all concrete transitions are "covered" by transitions of this form.

In operational semantics and process algebra, this is formalized as a simulation:

Given $\gamma : A \to \mathcal{P}(C)$, $K_C = \langle C, \tau_C, \mathcal{I}_C \rangle$, $K_A = \langle A, \tau_A, \mathcal{I}_A \rangle$, $K_C$ is $\gamma$-simulated by $K_A$ (written $K_C \lhd_\gamma K_A$)

iff for all $a \in A$, $c \in \gamma(a)$, $c' \in C$,

1. $\mathcal{I}_C(c) \subseteq \mathcal{I}_A(a)$
2. $c \to c'$ implies there exists $a' \in A$ such that $c' \in \gamma(a')$ and $a - - \to a'$.

That is, $K_A$ "mimicks" the transitions and atomic properties of $K_C$.

# Example over-approximation: $A = \{\bot, a0, a12, \top\}$

$\alpha\{\} = \bot$

$\alpha\{c0\} = a0$

$\alpha\{c1\} = a12 = \alpha\{c2\}$

$\alpha S = \top$, otherwise

$\gamma(\bot) = \{\}$

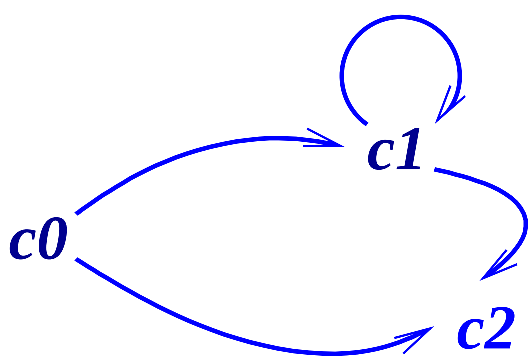
$\gamma(a0) = \{c0\}$

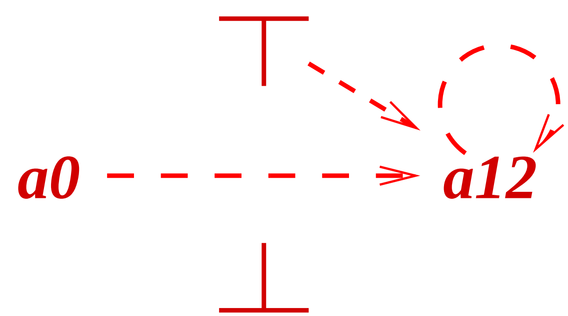
$\gamma(a12) = \{c1, c2\}$

$\gamma(\top) = \{c0, c1, c2\}$

$\mathcal{I}_A(a) = \cup\{\mathcal{I}_C(c) \mid c \in \gamma(a)\}$

$$\begin{array}{lcccl} \text{if } c & \varepsilon & \gamma(a) \\ \text{and} \downarrow & & \text{then} \downarrow \\ c' & \varepsilon & \gamma(a') \end{array}$$



# What properties can we safely check?

Aliasing — $a$ is possibly tail-aliased:

$$\mathit{isAliased}(a) = \exists x.\exists y.\tau_{tail}(x,a) \wedge \tau_{tail}(y,a) \wedge x \neq y$$

$$a \models (\exists \tau_{tail}^{-1}.\mathit{at}\ x) \wedge (\exists \tau_{tail}^{-1}.\mathit{at}\ y)$$

(recall $a \models \exists R.\phi$ iff exists $a'$ such that $R(a,a')$ and $a' \models \phi$)

$$\mathit{isAliased}(a) = \exists x.\exists y.(x \mapsto \_, a) * (y \mapsto \_, a) * \mathit{true}$$

that is, $\exists x.\exists y.\tau_{tail}(x,a) * \tau_{tail}(y,a) * \mathit{true}$

Reachability — $a$ is possibly reachable from $x$:

$$r_x(a) = \tau_{tail}^{*}(x,a)$$

$$a \models \mu Z.\mathit{at}\ x \vee \exists \tau_{tail}^{-1}.Z$$

$$r_x(a) =^{lfp} (x = a) \vee (\exists a'.\tau_{tail}(x,a') * r_{a'}(a))$$

We can refute such “possibility” properties.

Reachability — necessarily, all nodes reached from $a$ are "happy":

$$\mathrm{Happy}(a) = \forall y.\tau^*_{tail}(a, y) \supset happy \in \mathcal{I}_A(y)$$

$$a \models \nu Z.\mathrm{isHappy} \wedge \forall \tau_{tail}.Z$$

(Assumes that $\mathcal{I}_A(a) \subseteq \mathcal{I}_C(c)$, when $c \in \gamma(a)$.)

That is, there does not exist a reachable node/cell that lacks $happy$.

End cell — necessarily, there is no cell linked to $a$:

$$\mathrm{noTail}(a) = \forall y.\neg\tau_{tail}(a, y)$$

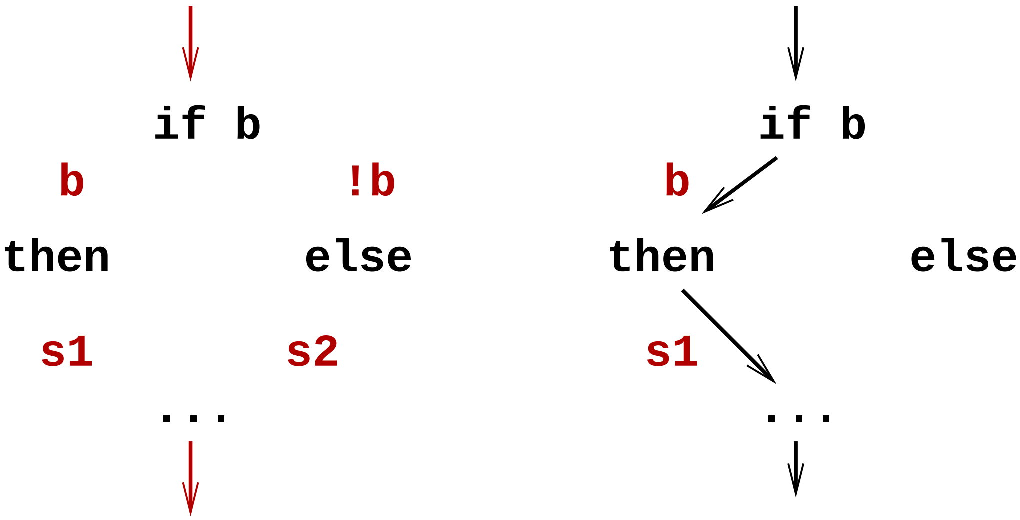
$$a \models \forall \tau_{tail}.\mathrm{false}$$

That is, there does not exist a $tail$-transition from $a$.

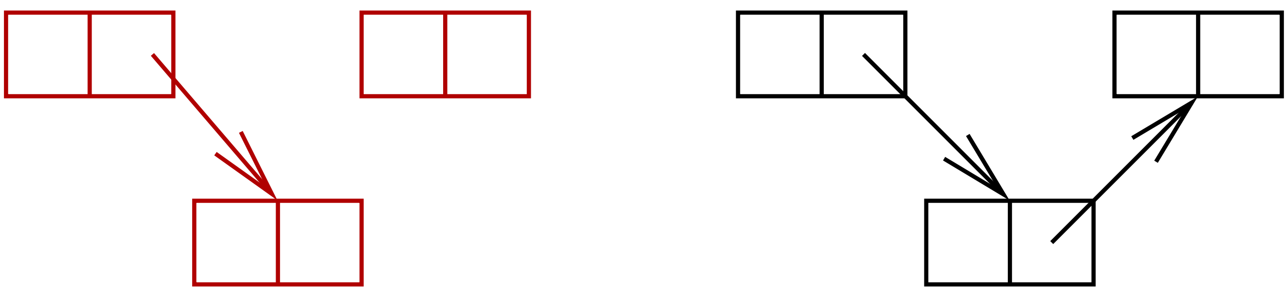
With an over-approximation model, we validate universal properties and refute existential ones.

# What does an abstract transition denote (2)?

what must necessarily execute:




what must necessarily be linked:



# What does "solid" denote? $\boldsymbol{a0} \longrightarrow \boldsymbol{a1}$

Underapproximation — 1 — must/necessarily: All corresponding concrete structures must possess this transition.

This is a (dual) simulation:

Given $\gamma : A \rightarrow \mathcal{P}(C)$, $K_C = \langle C, \tau_C, \mathcal{I}_C \rangle$, $K_A = \langle A, \tau_A, \mathcal{I}_A \rangle$, $K_A$ is dual-$\gamma$-simulated by $K_C$ (written $K_A \lhd_{\gamma}^{-1} K_C$)

iff for all $a \in A$, $c \in \gamma(a)$, $a' in A$,

1. $\mathcal{I}_C(c) \supseteq \mathcal{I}_A(a)$
2. $a \longrightarrow a'$ implies there exists $c' \in C$ such that $c' \in \gamma(a')$ and $c \longrightarrow c'$.

That is, $K_C$ "mimicks" the must-transitions and atomic must-properties of $K_A$.

# Example under-approximation: $A = \{\perp, \mathrm{a0}, \mathrm{a12}, \top\}$

$$\gamma(\perp) = \{\}$$

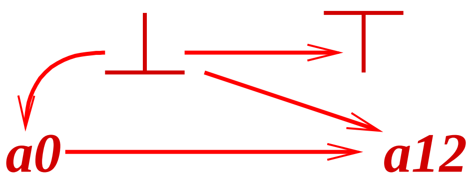
$$\alpha\{\} = \perp \qquad \gamma(\mathrm{a0}) = \{\mathrm{c0}\}$$

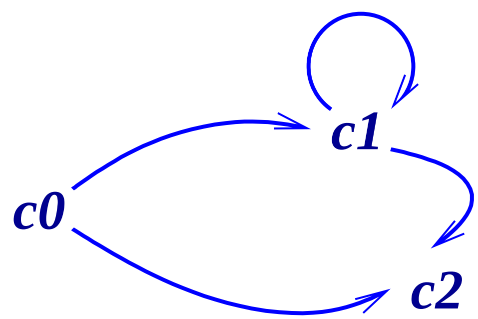
$$\alpha\{\mathrm{c0}\} = \mathrm{a0} \qquad \gamma(\mathrm{a12}) = \{\mathrm{c1}, \mathrm{c2}\}$$

$$\alpha\{\mathrm{c1}\} = \mathrm{a12} = \alpha\{\mathrm{c2}\} \qquad \gamma(\top) = \{\mathrm{c0}, \mathrm{c1}, \mathrm{c2}\}$$

$\alpha S = \top,$ otherwise

$$\mathcal{I}_A(\mathrm{a}) = \cap\{\mathcal{I}_C(\mathrm{c}) \mid \mathrm{c} \in \gamma(\mathrm{a})\}$$




(The transitions from $\perp$ are technically correct but are practically useless; you can ignore them.)

# What properties can we safely check?

$a$ is necessarily reachable from $x$:

$$r_x(a) = \tau^*_{\mathrm{tail}}(x, a)$$

$$a \models \mu Z.\mathrm{at}\ x \vee \exists \tau^{-1}_{\mathrm{tail}}.Z$$

$$r_x(a) =^{\mathit{lfp}} (x = a) \vee (\exists a'.\tau_{\mathrm{tail}}(x, a') * r_{a'}(a))$$

possibly, all cells reached from $a$ are “happy”:

$$\mathrm{isSafe}(a) = \forall y.\tau^*_{\mathrm{tail}}(a, y) \supset \mathrm{happy} \in \mathcal{I}_A(y)$$

$$a \models \nu Z.\mathrm{isHappy} \wedge \forall \tau_{\mathrm{tail}}.Z$$

(Assumes that $\mathcal{I}_A(a) \supseteq \mathcal{I}_C(c)$, when $c \in \gamma(a)$.)

That is, there does not exist a necessarily-reachable cell/node that lacks $\mathrm{happy}$ — the possibility that all reachable cells are happy still exists.

With an under-approximation model, we validate existential properties and refute universal ones.

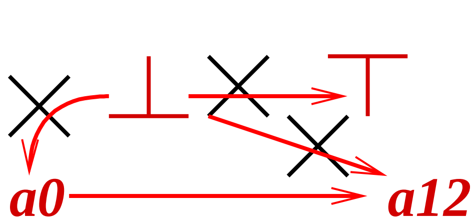
# Mixed and modal transition systems

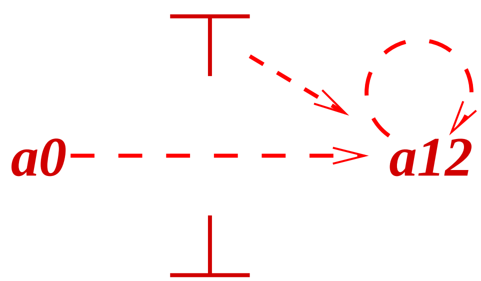
A mixed Kripke transition system is two systems, an under approximation and an over approximation, with the same cell/state set:

$$\langle \Sigma, \tau^{\mathrm{must}}, \tau^{\mathrm{may}}, \mathcal{I}^{\mathrm{must}}, \mathcal{I}^{\mathrm{may}} \rangle$$

When $\tau^{\mathrm{must}} \subseteq \tau^{\mathrm{may}}$ and $\mathcal{I}^{\mathrm{must}} \sqsubseteq \mathcal{I}^{\mathrm{may}}$, the system is modal.

When $\tau^{\mathrm{must}} = \tau^{\mathrm{may}}$ and $\mathcal{I}^{\mathrm{must}} = \mathcal{I}^{\mathrm{may}}$, the system is concrete — an ordinary Kripke transition system.



Simulation is replaced by refinement — simulations in two directions:

Given $M_C = \langle C, \tau_C^{must}, \tau_C^{may}, \mathcal{I}_C^{must}, \mathcal{I}_C^{may} \rangle$ and $M_A = \langle A, \tau_A^{must}, \tau_A^{may}, \mathcal{I}_A^{must}, \mathcal{I}_A^{may} \rangle$,

$M_C$ refines $M_A$ iff

$$\langle C, \tau_C^{may}, \mathcal{I}_C^{may} \rangle \lhd_\gamma \langle A, \tau_A^{may}, \mathcal{I}_A^{may} \rangle$$

and

$$\langle A, \tau_A^{must}, \mathcal{I}_A^{must} \rangle \lhd_\gamma^{-1} \langle C, \tau_C^{must}, \mathcal{I}_C^{must} \rangle$$

That is, $M_A$'s may-parts simulate $M_C$'s, and $M_C$'s must-parts dual-simulate $M_A$'s.

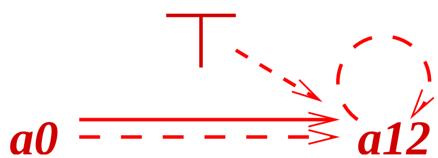
When $M_C$ refines $M_A$,

- $M_C$'s under-approximation is larger (more precise) than $M_A$'s
- $M_C$'s over-approximation is smaller (more precise) than $M_A$'s.

# We can validate a full predicate logic on a MTS

We validate universal subformulae on the upper-approximation and existential subformulae on the lower-approximation, jumping "back and forth" as needed.

We validate a negated formula by refuting it on the dual approximation.

Example: 


$$\mathsf{a0} \models^{\mathsf{under}} \exists\tau.\forall\tau.\neg\mathsf{at\_a0}$$

iff $\mathsf{a12} \models^{\mathsf{over}} \forall\tau.\neg\mathsf{at\_a0}$

iff $\mathsf{a12} \models^{\mathsf{over}} \neg\mathsf{at\_a0}$

iff $\mathsf{a12} \not\models^{\mathsf{under}} \mathsf{at\_a0}$

iff $\mathsf{true}$

For a MTS, where $\tau_{\mathrm{must}} \subseteq \tau_{\mathrm{may}}$, there are only three possible outcomes: $\phi$ necessarily holds, $\phi$ possibly holds, $\phi$ not possibly holds.

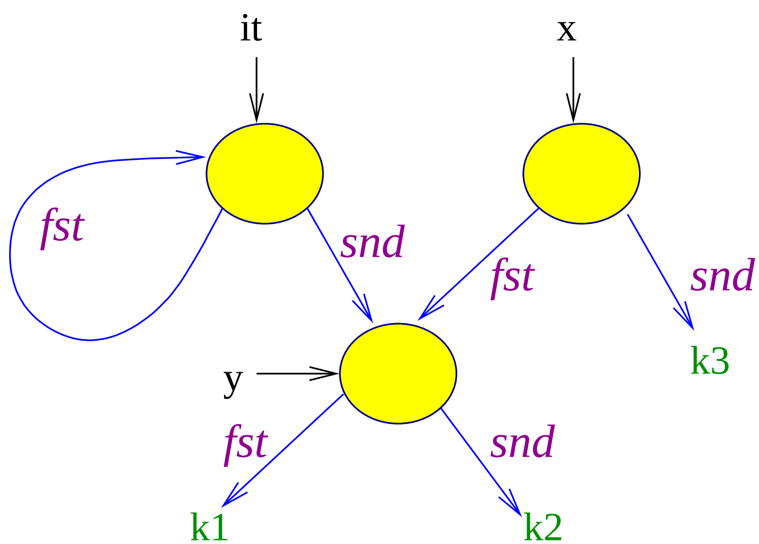
Sagiv/Reps/Wilhelm TVLA models have must-may cells such that *(i)* a must-cell can not be split (or merged) in a re£nement; *(ii)* a may-cell can not be merged in a re£nement. We might de£ne an extension of MTS with such cells. (We might also restrict $\gamma$!)

The re£nement relation, quotiented, is a partial ordering in a dcpo of modal transition systems. Given MTS, $M$, its re£nements form a Kripke model unto which we can apply a modal logic:

- $M \models \Box\phi$ — all re£nements satisfy $\phi$ (intuitionistic)
- $M \models \Box\Diamond\phi$ — always possible to re£ne to satisfy (dense)
- Generalized model checking examines only the limit points of $M$'s Kripke model. (*Aprés* Michael Huth, the three coincide.)

# "Store-less" models: Path sets

Jonkers and Deutsch proposed "storeless" (heap-less) models:




The heap shape is modelled by right-regular equivalence sets of paths from the "entry point," $\mathit{it}$:

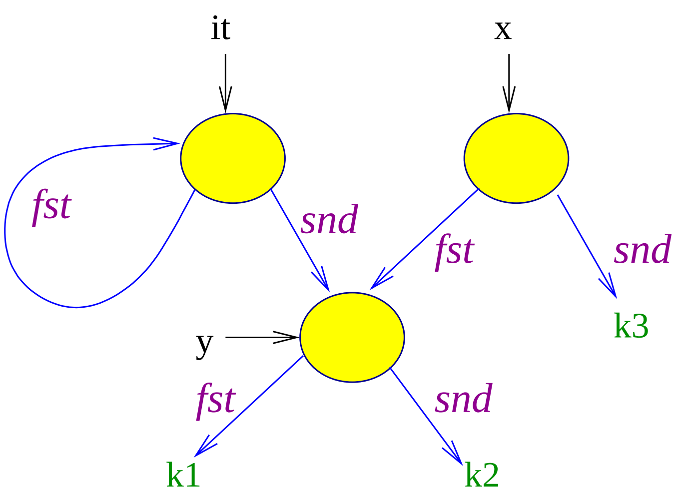
$$\{\mathit{fst}^i \mid i \geq 0\} \qquad \{\mathit{fst}^i.\mathit{snd} \mid i \geq 0\}$$

$$\{\mathit{fst}^i.\mathit{snd}.\mathit{fst} \mid i \geq 0\} \quad \{\mathit{fst}^i.\mathit{snd}^2 \mid i \geq 0\}$$

Deutsch developed clever fsa over-approximations of the equivalence classes.

# Blanchet's path models

Many questions regarding escapes, leaks, and aliases are answered by the paths from one object of interest to another, e.g., from a global variable to the heap's entry point:




$$\{y.snd^{-1}.fst^{i}.(fst^{-1})^{j} \mid i,j \geq 0\}$$
$$\cup \{x.fst.snd^{-1}.fst^{i}.(fst^{-1})^{j} \mid i,j \geq 0\}$$

The paths have been normalized by the cancellation law,

$$fst^{-1}.fst \equiv \epsilon$$

The cancellation law gives the paths a pleasant, regular format.

The paths are traces through the heap, and questions about the traces can be asked in the language of linear temporal logic. Let $\pi$ be a trace from variable x to it, the result/heap-entry.

We can ask standard questions:

- Is part of x embedded in the result? $\pi \models at\_x \wedge F(des^{-1})$
- Does x's cell itself escape in the result? $\pi \models at\_x \wedge G(des^{-1})$
- Is part of x aliased to y? $\pi \models F(at\_y)$
- Is x a cyclic structure? $\pi \models GF(at\_x)$

# Summary

- For analyses that deduce properties of paths, under- and over-approximation issues are crucial.
- Branching-time models of heap are widely used, but maybe Deutsch and Blanchet know better — end-users prefer linear-time logic over branching time; shouldn't we?
- Integration of spatial logics into heap abstraction and static analysis seems worth a try.

# References


1. This talk: `www.cis.ksu.edu/~schmidt/papers`
2. Blanchet PhD thesis
3. CousotCousot POPL 1979
4. Dams, Gerth, Grumberg ACM TOPLAS
5. Deutsch PhD thesis
6. Huth, Jagadeesan, Schmidt MSCS paper
7. Larsen Concur paper
8. Reps, Sagiv, Wilhelm POPL 1998
9. Vardi ETAPS 2000 paper