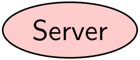
Ideal Abstraction for Well-Structured Transition Systems

Damien Zufferey[1] Thomas Wies[2] Thomas A. Henzinger[1]

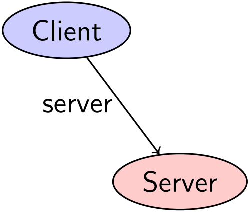
[1]IST Austria [2]New York University

VMCAI 2012

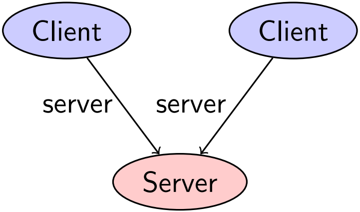
Example: client-server communication pattern



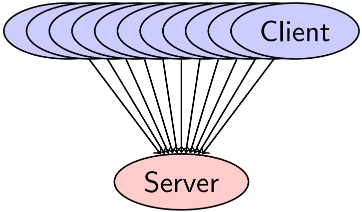
Example: client-server communication pattern



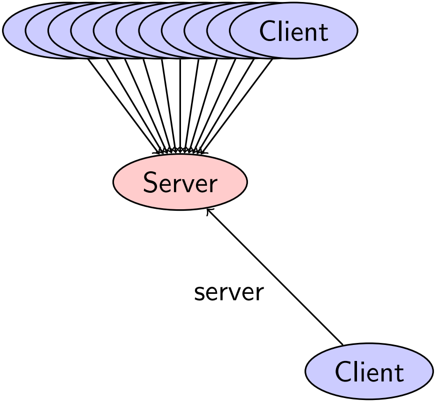
Example: client-server communication pattern



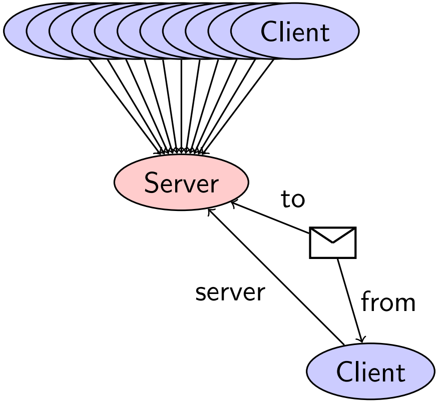
Example: client-server communication pattern



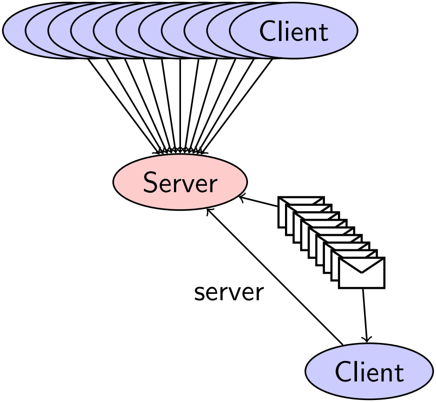
Example: client-server communication pattern



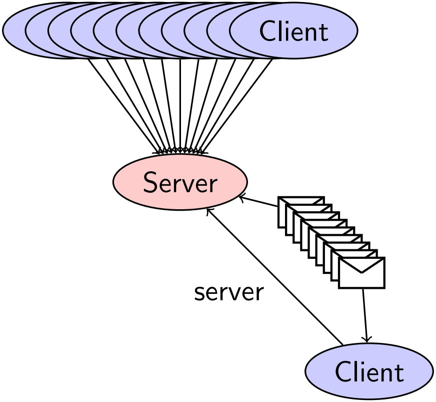
Example: client-server communication pattern



Example: client-server communication pattern



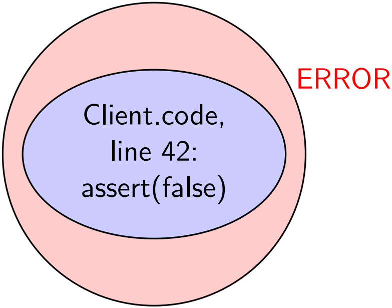
Example: client-server communication pattern



This example is a Depth-Bounded Process (DBP),
an instance of WSTS [Meyer, 2008, Wies et al., 2010].

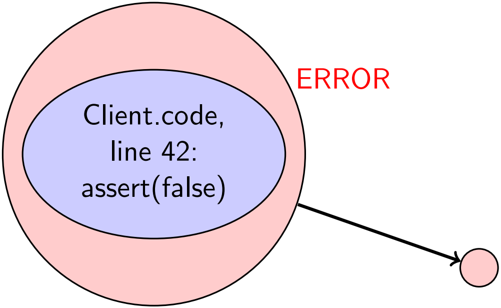
What kind of properties are we looking at ?

Safety properties, more precisely the control-state reachability problem (aka covering problem).



What kind of properties are we looking at ?

Safety properties, more precisely the control-state reachability problem (aka covering problem).



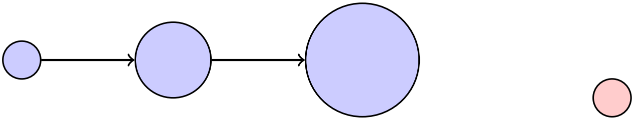
What kind of properties are we looking at ?

Safety properties, more precisely the control-state reachability problem (aka covering problem).

Safety properties, more precisely the control-state reachability problem (aka covering problem).

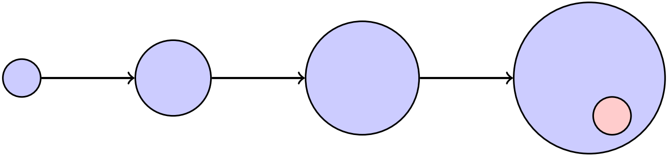
What kind of properties are we looking at ?

Safety properties, more precisely the control-state reachability problem (aka covering problem).



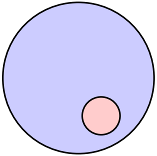
What kind of properties are we looking at ?

Safety properties, more precisely the control-state reachability problem (aka covering problem).

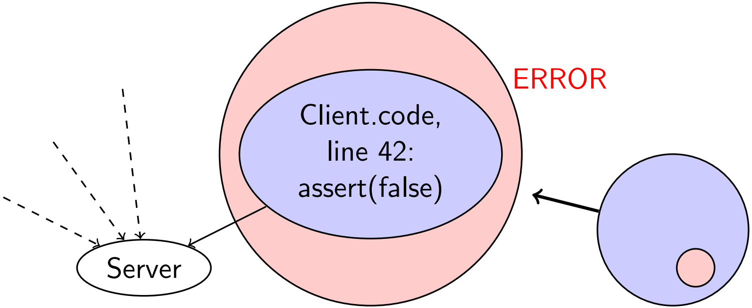


What kind of properties are we looking at ?

Safety properties, more precisely the control-state reachability problem (aka covering problem).



Safety properties, more precisely the control-state reachability problem (aka covering problem).



Outline

- Motivation
- Formalism
- Ideal abstraction
- Example of set-widening for ideal completion

Formal model: WSTS

A well-structured transition system (WSTS) is a transition system $\langle S, \rightarrow, \leq \rangle$ such that:

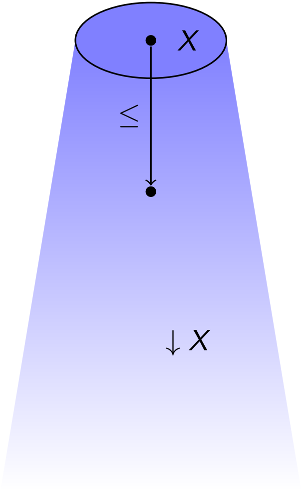
- $\leq$ is a well-quasi-ordering (wqo),
  i.e. well-founded + no infinite antichain.
- compatibility of $\leq$ w.r.t. $\rightarrow$

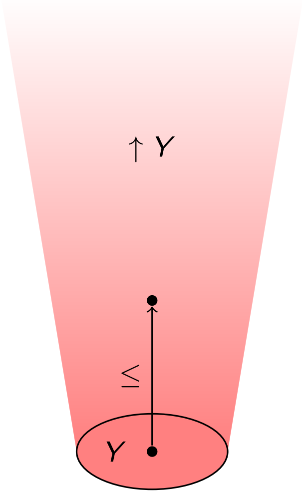
$$\forall \quad \begin{array}{ccc} t & \xrightarrow{*} & t' \\ \rotatebox{90}{$\geq$} & & \rotatebox{90}{$\geq$} \\ s & \longrightarrow & s' \end{array} \quad \exists$$

For more detail see:
[Finkel and Schnoebelen, 2001, Abdulla et al., 1996]

Downward and upward-closures

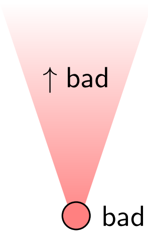
$$\downarrow X = \{\ x' \mid \exists x \in X.\ x' \leq x\ \}$$



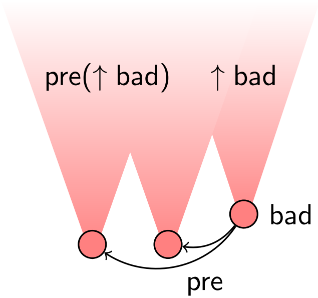


$$\uparrow Y = \{\ y' \mid \exists y \in Y.\ y \leq y'\ \}$$

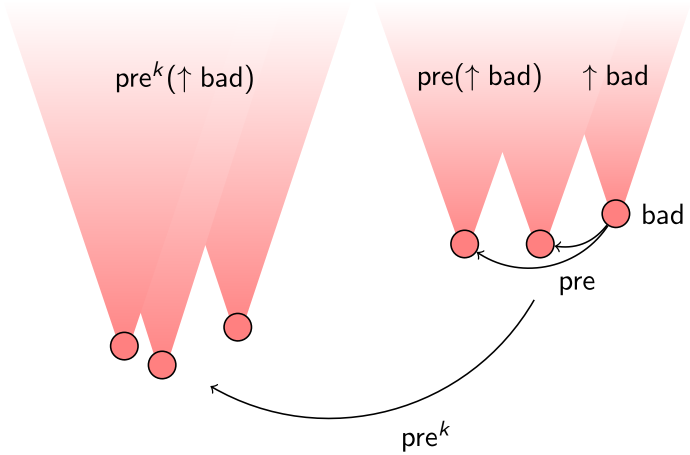
Backward algorithm for covering

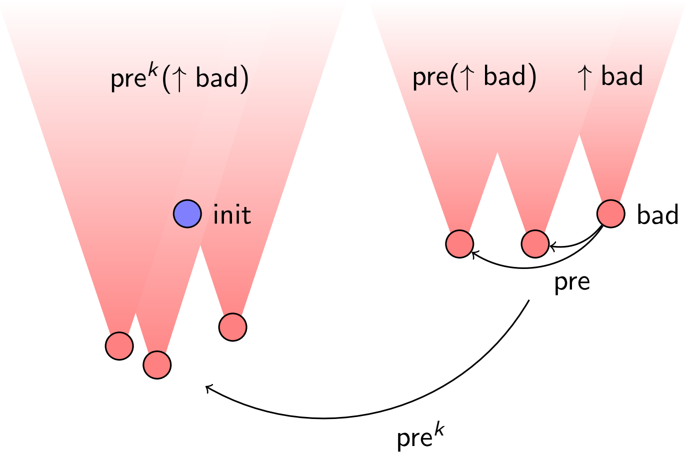


Backward algorithm for covering



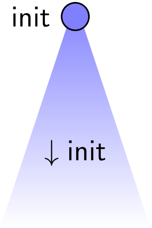
Backward algorithm for covering



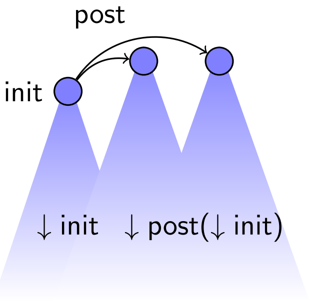


$\mathrm{pre}^k(\uparrow \mathrm{bad})$
init
$\mathrm{pre}(\uparrow \mathrm{bad})$
$\uparrow \mathrm{bad}$
bad
pre
pre^k

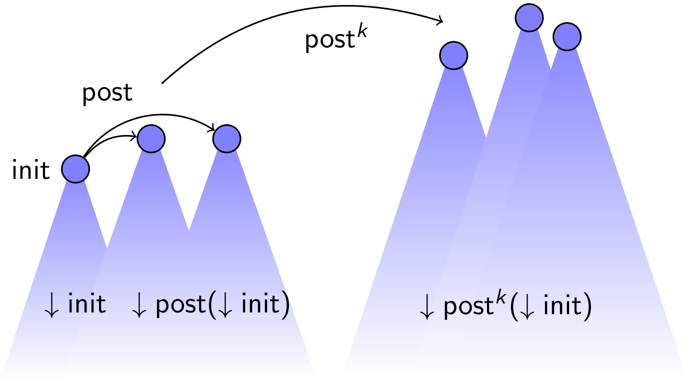
Forward algorithm for covering



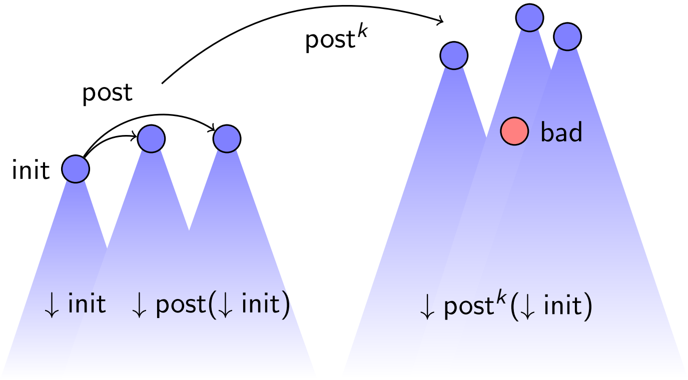
Forward algorithm for covering



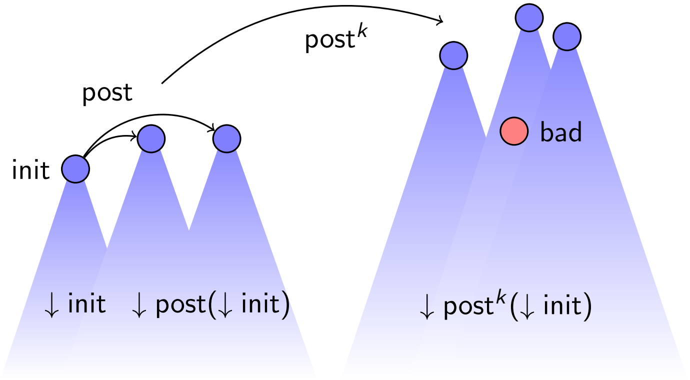
Forward algorithm for covering



Forward algorithm for covering



Forward algorithm for covering



Computes the covering set rather than answering only a single coverability query.

Representing downward-closed sets with ideals

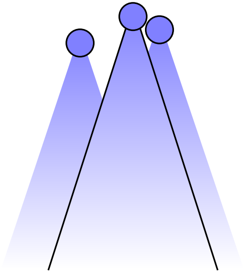
Given a wqo-set $(X, \leq)$.

A subset of X is directed if it is non-empty and closed under upper bounds.

An ideal of X is a directed downward-closed subset of X.

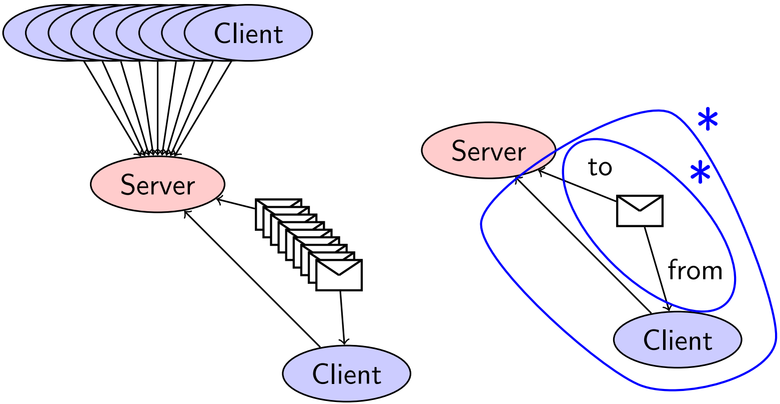
The ideal completion $Idl(X)$ of X is the set of all ideals of X.

A downward-closed subset is a finite union of ideals.



Ideal for representing downward-closed sets.

ADL: [Geeraerts et al., 2006]
Further developed in [Finkel and Goubault-Larrecq, 2009]
Applied to DBP in [Wies et al., 2010]



When does acceleration work ? (flat systems)

Forward algorithms are (usually) based on acceleration.
Acceleration *executes* loops infinitely many time (saturation).

Concretely, The algorithm terminates if the covering set can be generated by executing only simple loops. This condition is known as flattability [Bardin et al., 2005]. Acceleration *executes* traces of length $< \omega^2$.

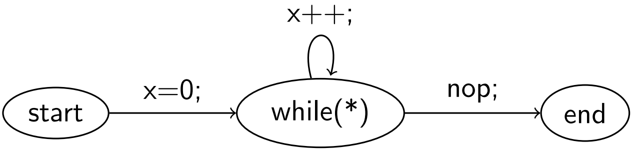
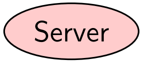
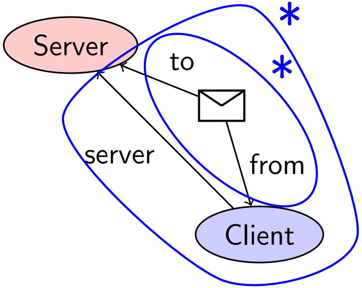


Figure: Example of a flat program

initial configuration:



covering set:

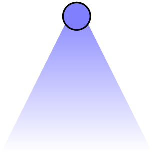


ω^2 steps from the initial state and the final state.

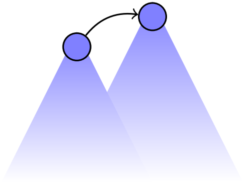
Nested loops are required to compute the covering set.

From acceleration to widening

Acceleration considers transitions - widening only states.

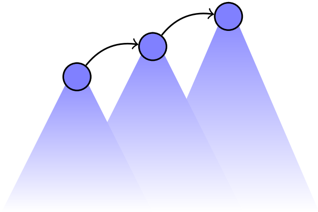


Acceleration considers transitions - widening only states.



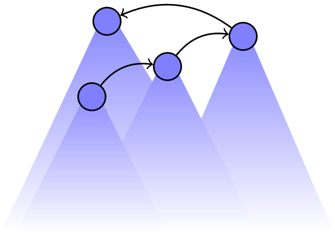
From acceleration to widening

Acceleration considers transitions - widening only states.

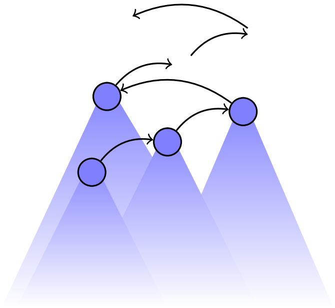


From acceleration to widening

Acceleration considers transitions - widening only states.

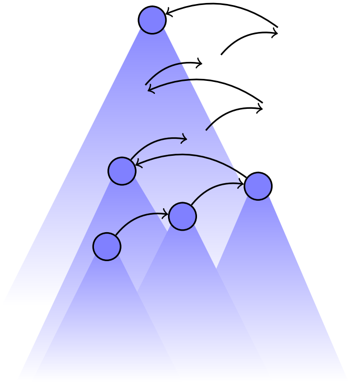


Acceleration considers transitions - widening only states.

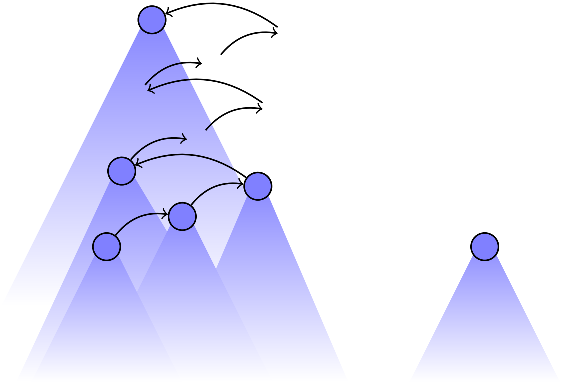


From acceleration to widening

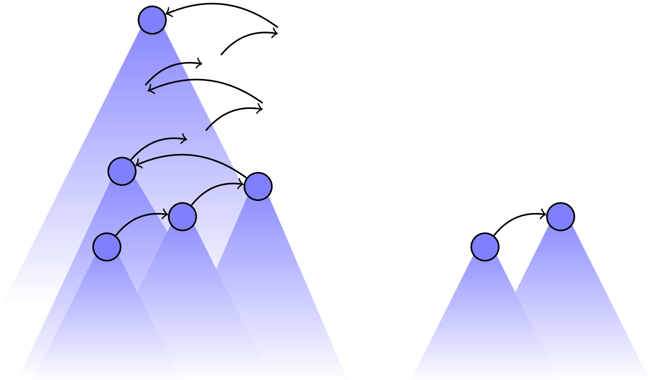
Acceleration considers transitions - widening only states.



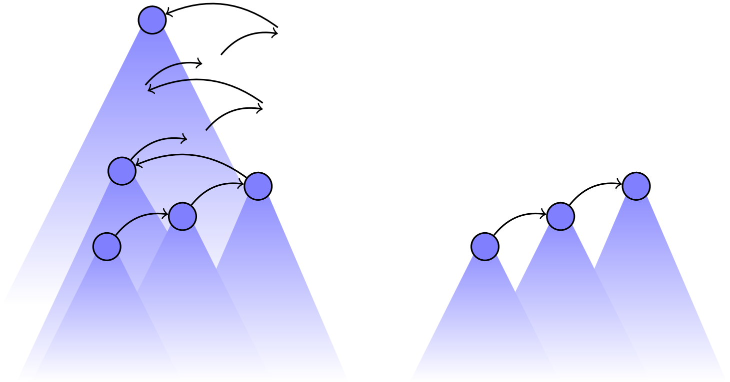
Acceleration considers transitions - widening only states.



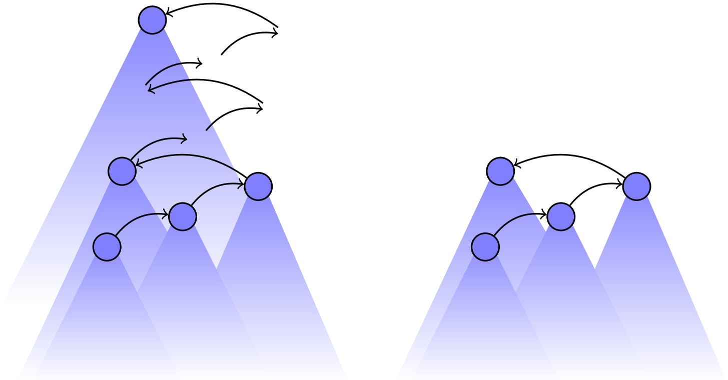
Acceleration considers transitions - widening only states.



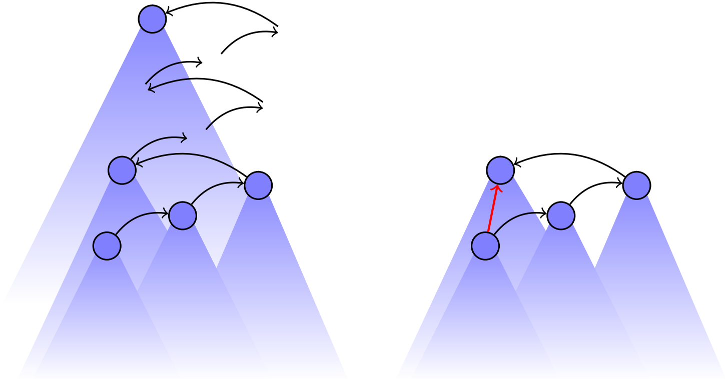
Acceleration considers transitions - widening only states.



Acceleration considers transitions - widening only states.

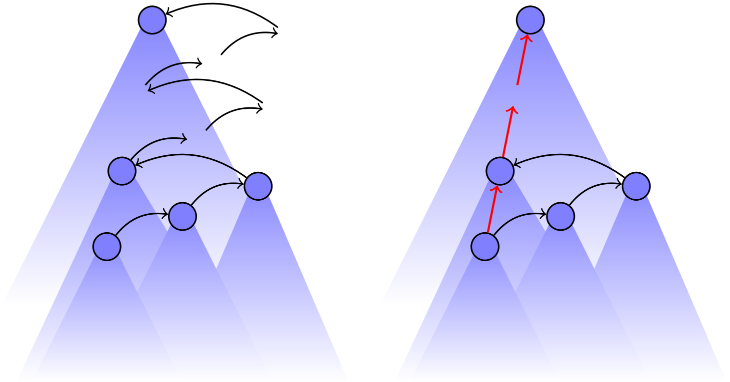


Acceleration considers transitions - widening only states.



From acceleration to widening

Acceleration considers transitions - widening only states.



Ideal abstraction

Rephrase analysis in terms of abstract interpretation [Cousot and Cousot, 1977].

- Concrete domain: sets of configurations ($\mathcal{P}(S)$)
- Abstract domain: ideal completion of $(S, \leq)$ ($\mathcal{P}_{\mathit{finite}}(\mathit{Idl}(S))$)

- Concretization function γ: identity
- Abstraction function α: downward-closure

(α, γ) is a Galois connection.

Covering set is abstract fixed point: $\mu X.\alpha(\mathit{init}) \cup \alpha \cdot \mathit{post} \cdot \gamma(X)$

To guarantee termination we need a widening operator.

Widening (1)

Goal: try to mimic acceleration (when possible), and force termination

A set-widening operator (∇) [Cousot and Cousot, 1992] for a poset S is partial function ($\mathcal{P}(S) \to S$) that satisfies:

Covering : for all $X \subseteq S$, $x \in X \Rightarrow x \leq \nabla(X)$;

Termination : widening of any ascending chain stabilizes.

Why set-widening rather than the usual pair-widening.
Reason of using a set-widening operator: we need the history.

Widening (2)

Keeping it simple: need only set-widening operator on $Idl(S)$.
It can be lifted to $\mathcal{P}_{finite}(Idl(S))$, using a general construction (details in the paper).

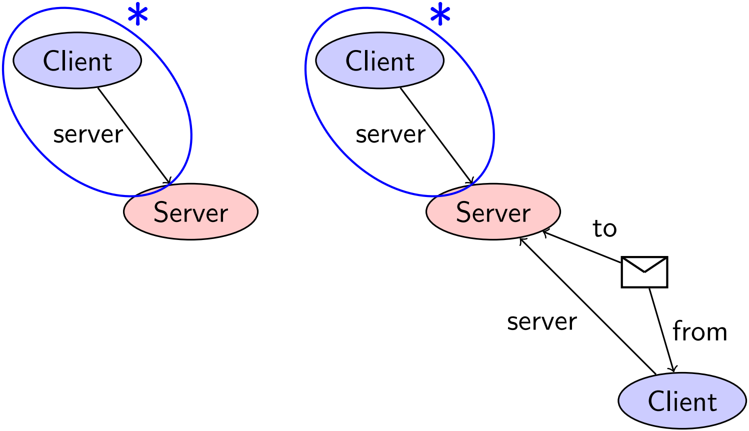
We assume that the ordering S is a better-quasi-ordering (bqo).
$\Rightarrow$ Thus, $Idl(S)$ is also a bqo.
$\Rightarrow$ No infinite antichain in $Idl(S)$.

We still need to define the widening on ideals.

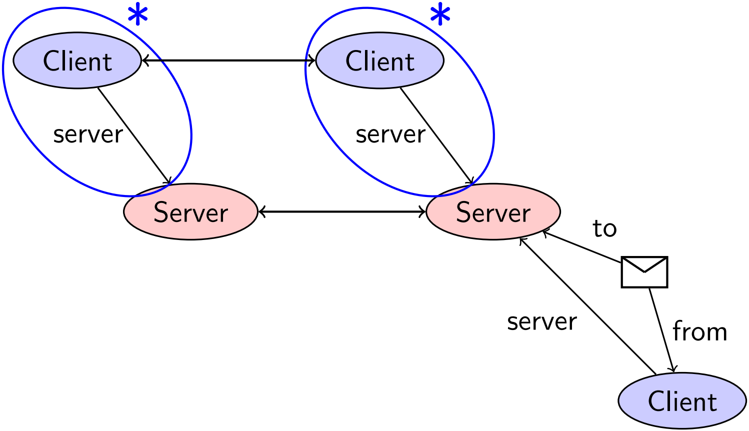
We provide concrete operators for

- Petri nets (and monotonic extensions),
- Lossy channel systems [Abdulla and Jonsson, 1993],
- Depth-bounded processes (DBP).

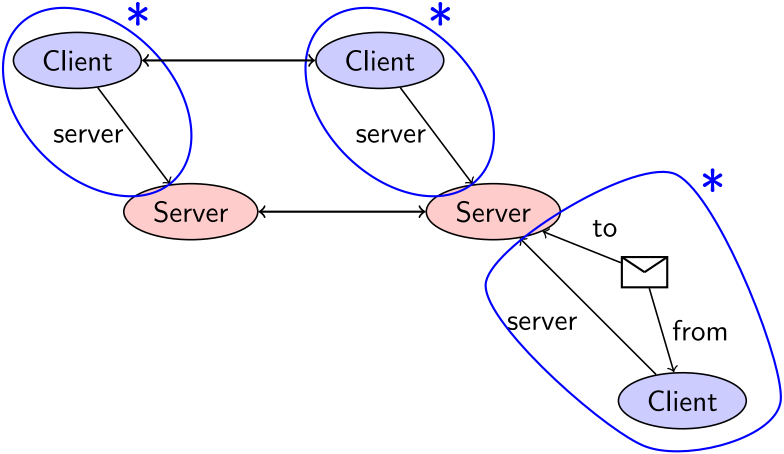
Set-widening for DBP (1)



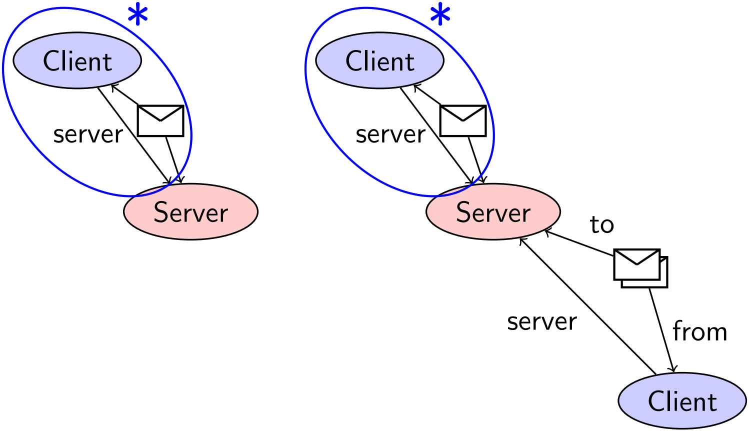
Set-widening for DBP (1)



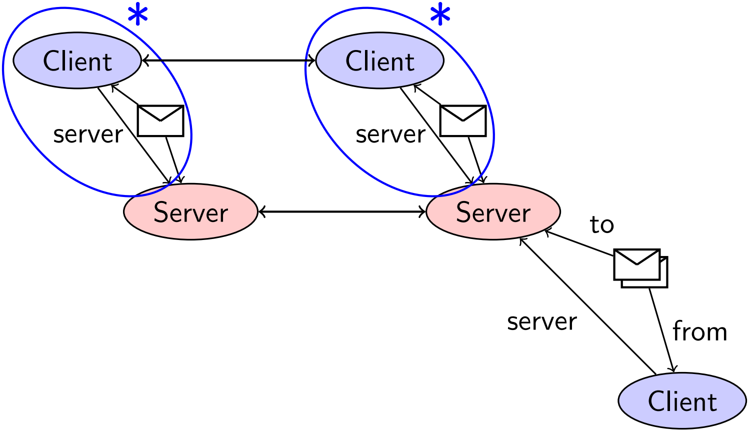
Set-widening for DBP (1)



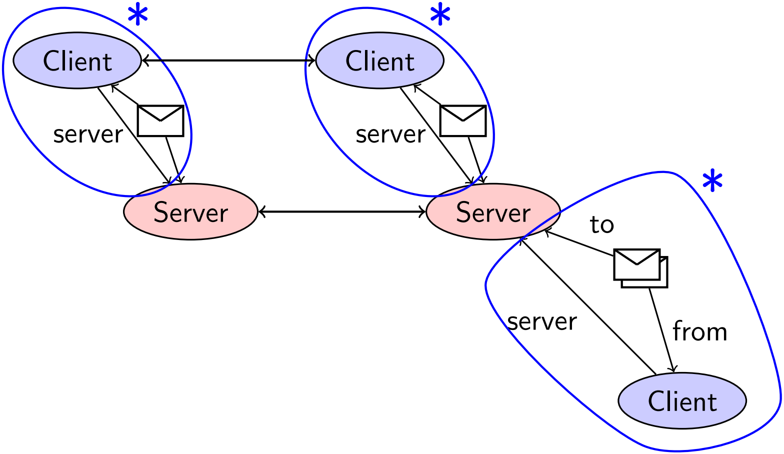
Set-widening for DBP (2)



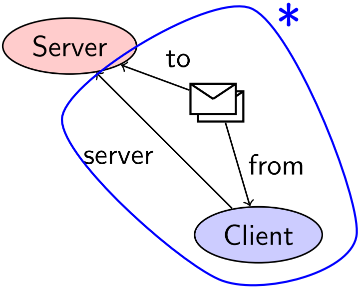
Set-widening for DBP (2)



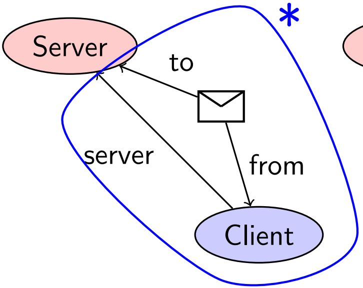
Set-widening for DBP (2)



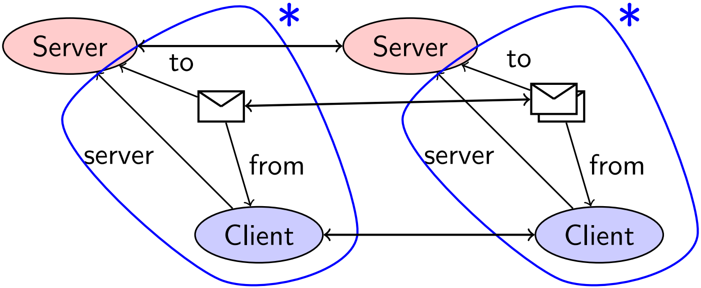
Set-widening for DBP (3)



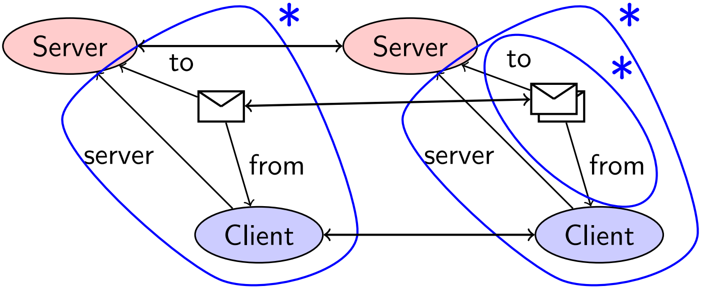
Set-widening for DBP (3)



Set-widening for DBP (3)



Set-widening for DBP (3)



Implementation: Picasso

Picasso: Pi-Calculus-based Static Software Analyzer

Picasso implements

- Ideal abstraction domain
- Widening for DBP
- Trace partitioning domain [Rival and Mauborgne, 2007] (similar to Karp-Miller Tree)

Target: Scala actor programs

Input: manually extracted models (soon a compiler plug-in)

Experimental results in the paper

Available at `http://pub.ist.ac.at/~zufferey/picasso/`.

Name	tree size	cov. set size	time
`ping-pong`	17	14	0.6 s
`client-server`	25	2	1.9 s
`client-server-with-TO`	184	5	12.8 s
`genericComputeServer`	57	4	4.6 s
`genericComputeServer-fctAsActor`	98	8	14.8 s
`liftChatLike`	1846	21	1830.9 s
`round_robin_2`	830	63	48.8 s
`round_robin_3`	3775	259	737.8 s

Further related work:

[Abdulla et al., 1996] Complete backward algorithm for coverability

[Geeraerts et al., 2006] Complete algorithm for coverability based on ideal completions

[Ganty et al., 2006] Complete algorithm for coverability based on abstract interpretation but not ideal completions

[Finkel and Goubault-Larrecq, 2009] Acceleration-based algorithm for computing covering sets

Conclusion

- Many verification problems for concurrent systems can be phrased in terms of coverability
- Coverability is decidable for well-structured transition systems but with high complexity
- Ideal Abstraction: a generic framework for computing an approximation of the covering set.
  - promises precise but more scalable analysis of WSTS
  - we provide instantiations of the framework for common classes of WSTS
  - Picasso: implementation of our framework for the analysis of Scala actor programs

References I

Abdulla, P. A., Cerans, K., Jonsson, B. and Tsay, Y.-K. (1996).
General Decidability Theorems for Infinite-State Systems.
In LICS pp. 313–321,.

Abdulla, P. A. and Jonsson, B. (1993).
Verifying Programs with Unreliable Channels.
In LICS pp. 160–170,.

Bardin, S., Finkel, A., Leroux, J. and Schnoebelen, P. (2005).
Flat Acceleration in Symbolic Model Checking.
In ATVA pp. 474–488,.

Cousot, P. and Cousot, R. (1977).
Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints.
In POPL pp. 238–252,.

Cousot, P. and Cousot, R. (1992).
Abstract Interpretation Frameworks.
Journal of Logic and Computation *2*, 511–547.

Finkel, A. and Goubault-Larrecq, J. (2009).
Forward Analysis for WSTS, Part I: Completions.
In STACS vol. 09001, of Dagstuhl Sem. Proc. pp. 433–444,.

Finkel, A. and Schnoebelen, P. (2001).
Well-structured transition systems everywhere!
Theor. Comput. Sci. *256*, 63–92.

References II

Ganty, P., Raskin, J.-F. and Begin, L. V. (2006).
A Complete Abstract Interpretation Framework for Coverability Properties of WSTS.
In VMCAI pp. 49–64,.

Geeraerts, G., Raskin, J.-F. and Van Begin, L. (2006).
Expand, Enlarge and Check: New algorithms for the coverability problem of WSTS.
J. Comput. Syst. Sci. *72*, 180–203.

Meyer, R. (2008).
On Boundedness in Depth in the pi-Calculus.
In IFIP TCS vol. 273, of IFIP pp. 477–489, Springer.

Rival, X. and Mauborgne, L. (2007).
The trace partitioning abstract domain.
ACM Trans. Program. Lang. Syst. *29*.

Wies, T., Zufferey, D. and Henzinger, T. A. (2010).
Forward Analysis of Depth-Bounded Processes.
In FoSSaCS 2010 vol. 4349, of LNCS pp. 94–108, Springer.