Example-Guided Abstraction Simplification

Francesco Ranzato
University of Padova

Abstraction Refinements

- Widely used paradigm in static analysis and verification, e.g. CEGAR

Abstraction Refinements

- Widely used paradigm in static analysis and verification, e.g. CEGAR

- Basic principles

Abstraction Refinements

- Widely used paradigm in static analysis and verification, e.g. CEGAR

- Basic principles

  - Identify when and how to refine the underlying abstraction, e.g. abstract domain

Abstraction Refinements

- Widely used paradigm in static analysis and verification, e.g. CEGAR

- Basic principles
  - Identify when and how to refine the underlying abstraction, e.g. abstract domain
  - Goal: remove some false alarms or spurious traces

Abstraction Simplifications

- Few examples in static analysis and verification

Abstraction Simplifications

- Few examples in static analysis and verification

- Basic principles

Abstraction Simplifications

- Few examples in static analysis and verification

- Basic principles
  - Identify when and how to simplify the underlying abstraction

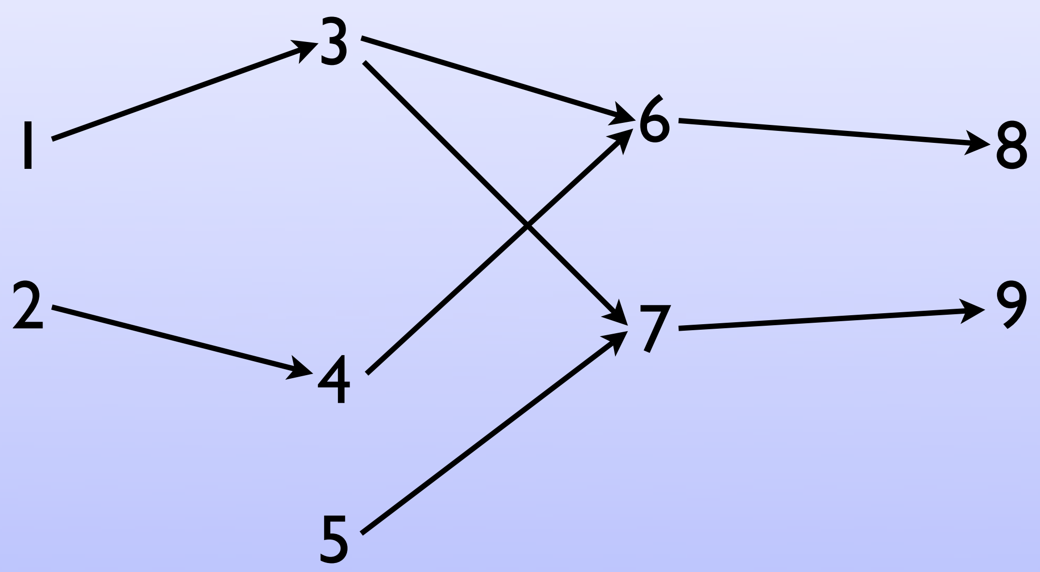
Abstraction Simplifications

- Few examples in static analysis and verification

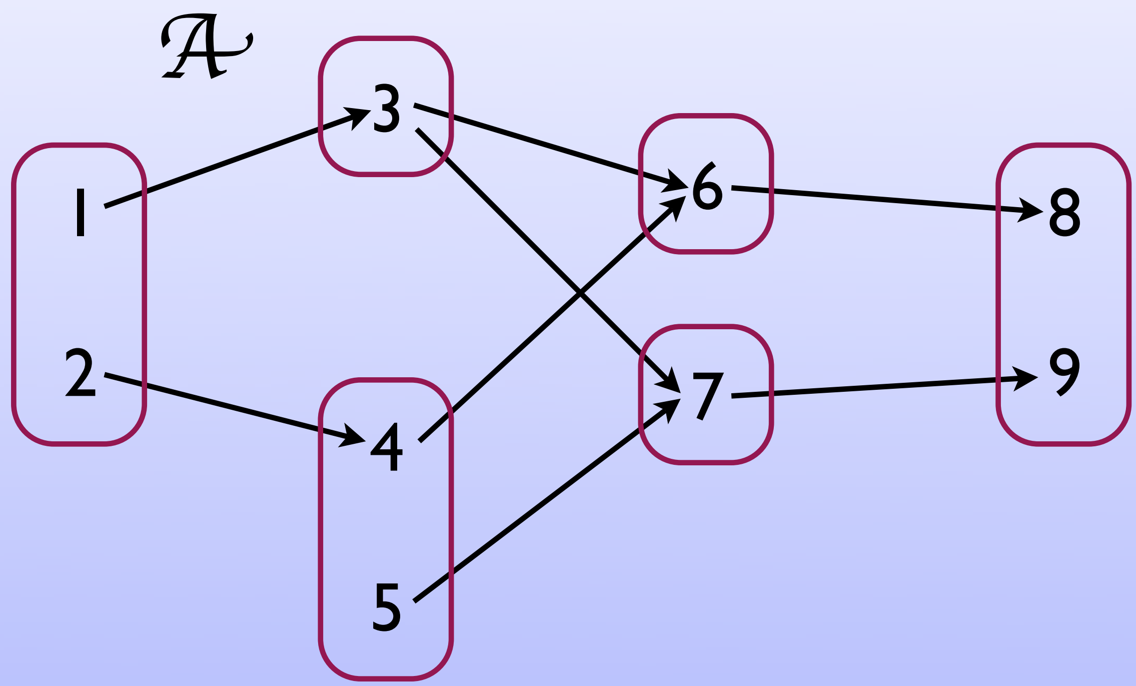
- Basic principles
  - Identify when and how to simplify the underlying abstraction
  - Goal: maintain the same approximate behaviour

Example in abstract model checking

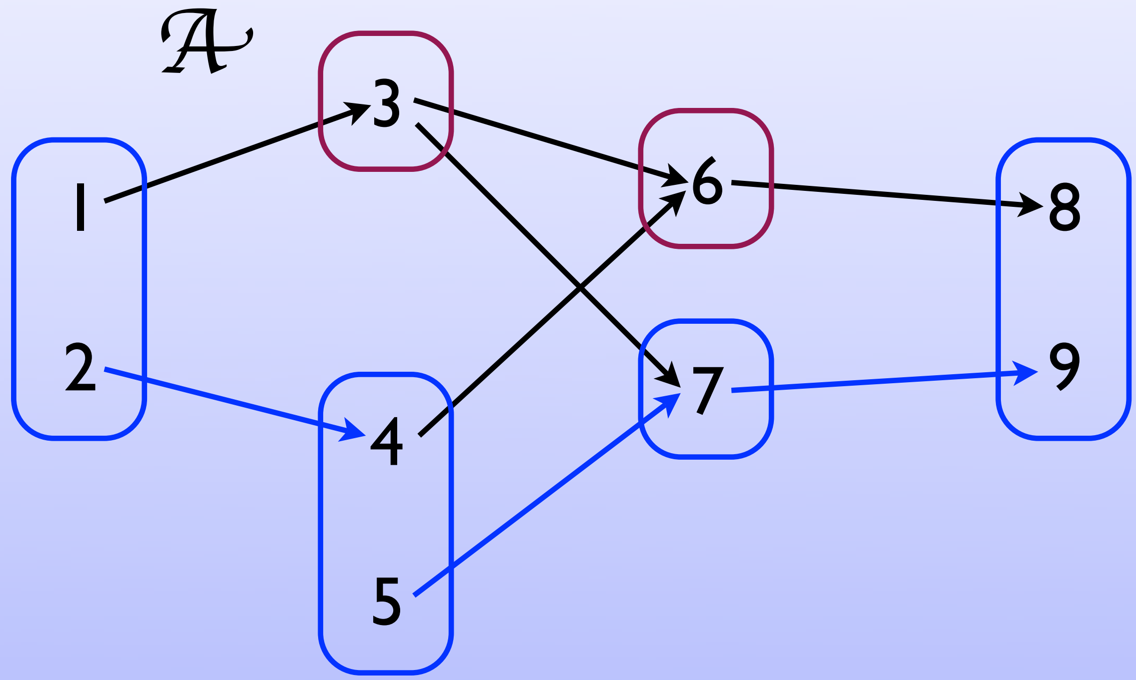
Example in abstract model checking



Example in abstract model checking

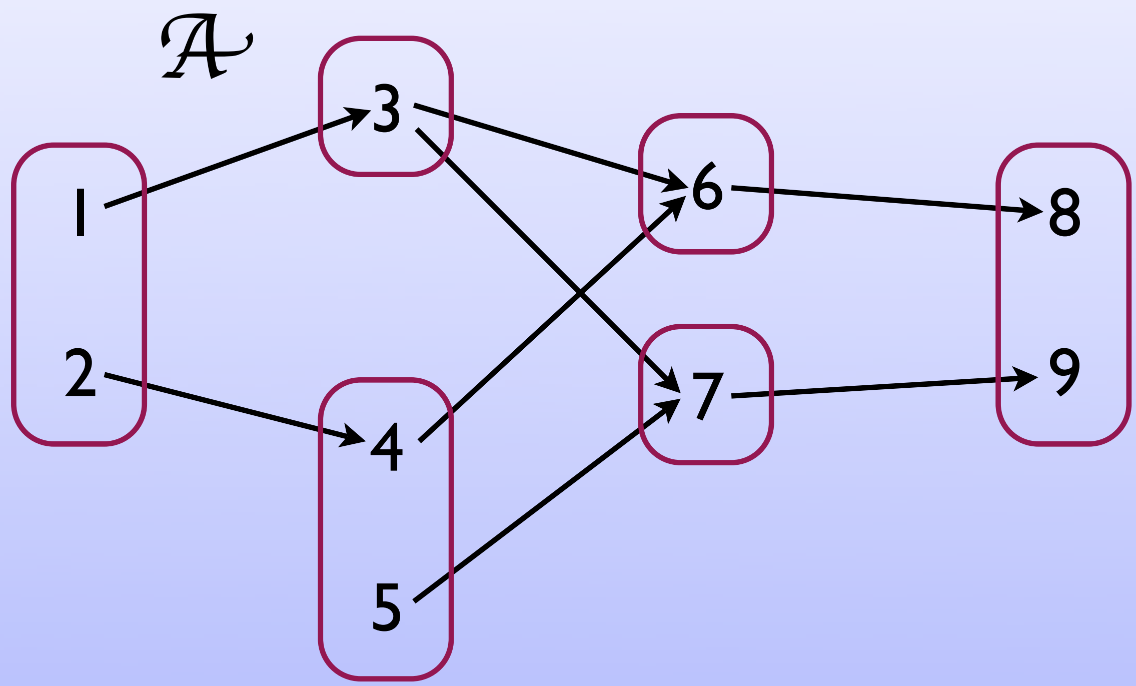


Example in abstract model checking

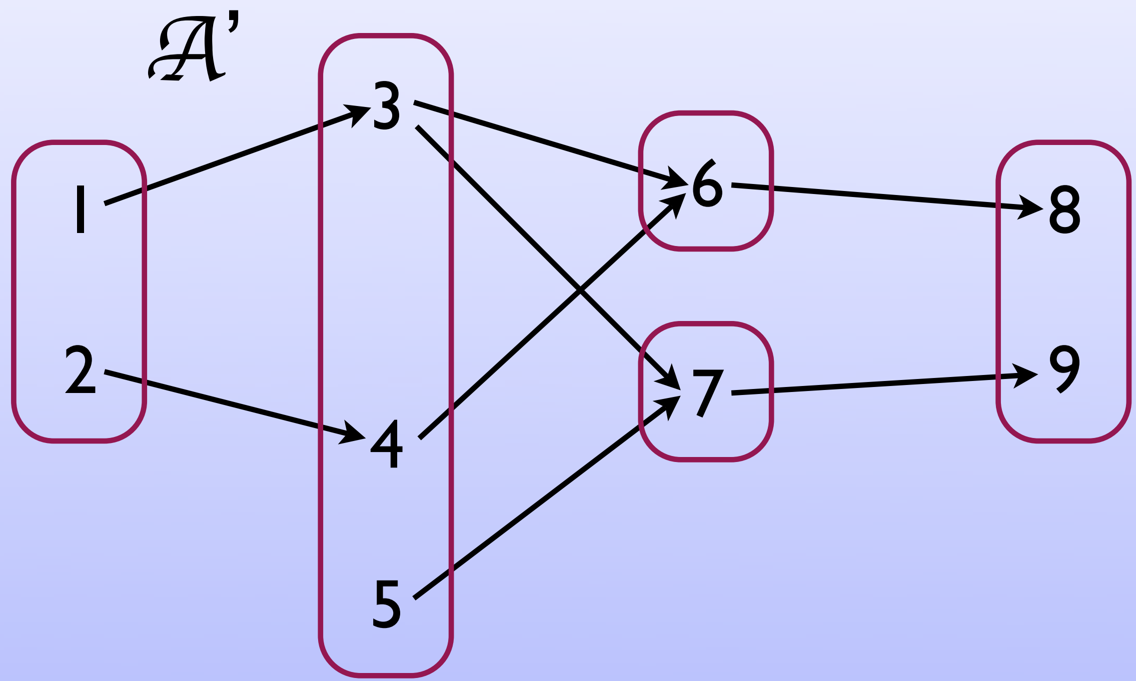


Spurious abstract path: [1,2] → [4,5] → [7] → [8,9]

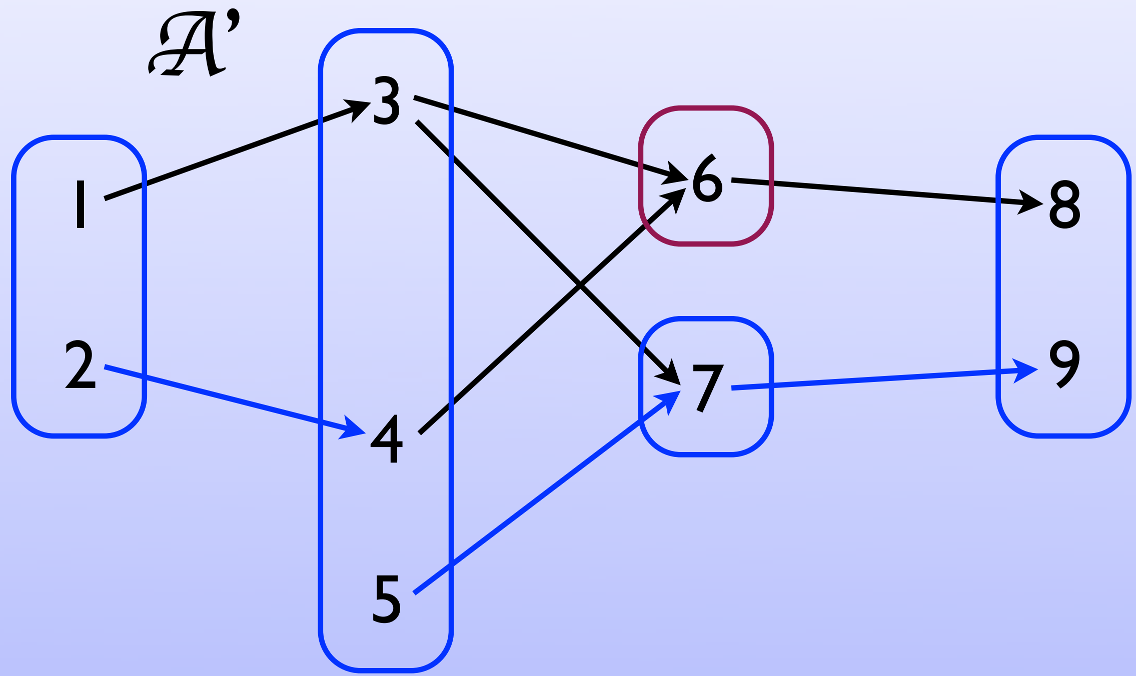
Example in abstract model checking



Example in abstract model checking

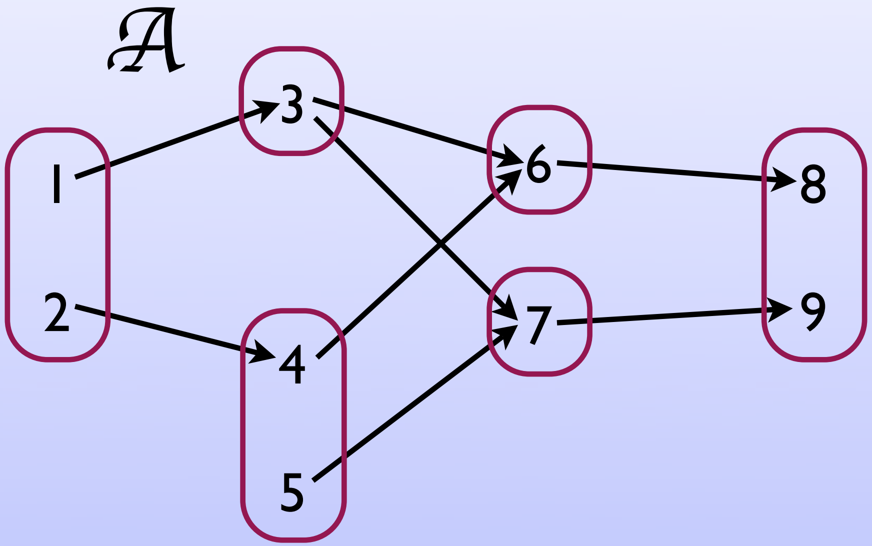


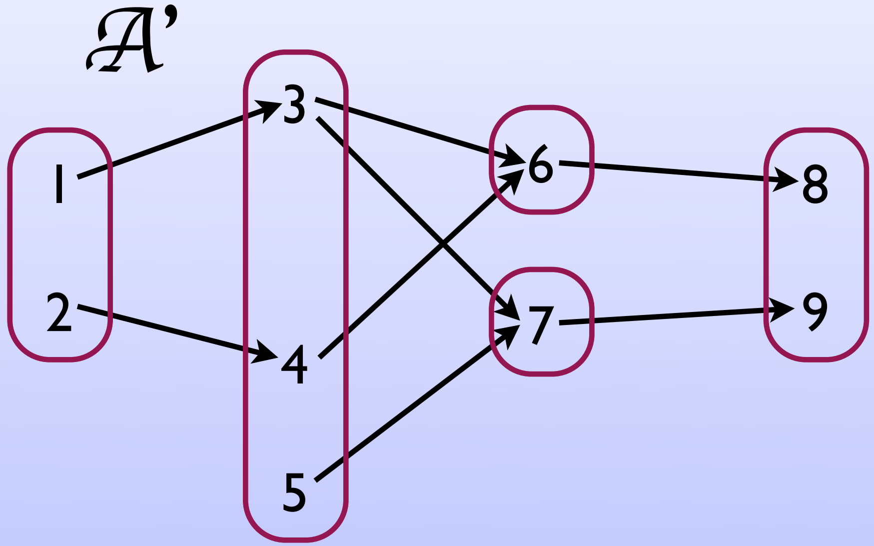
Example in abstract model checking



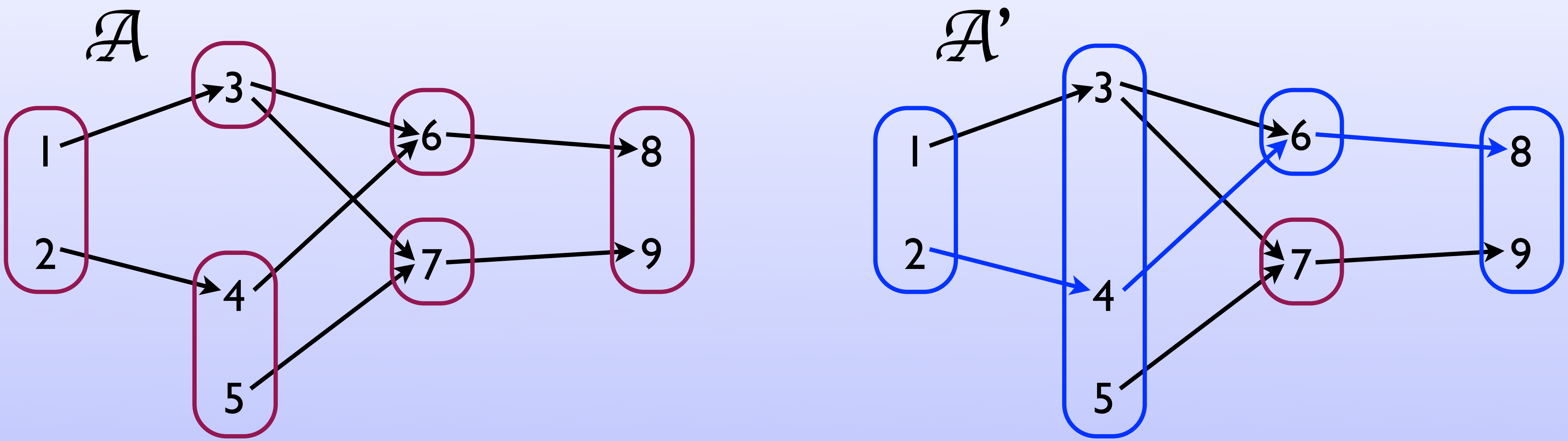
Spurious abstract path: [1,2] → [3,4,5] → [7] → [8,9]

Example in abstract model checking



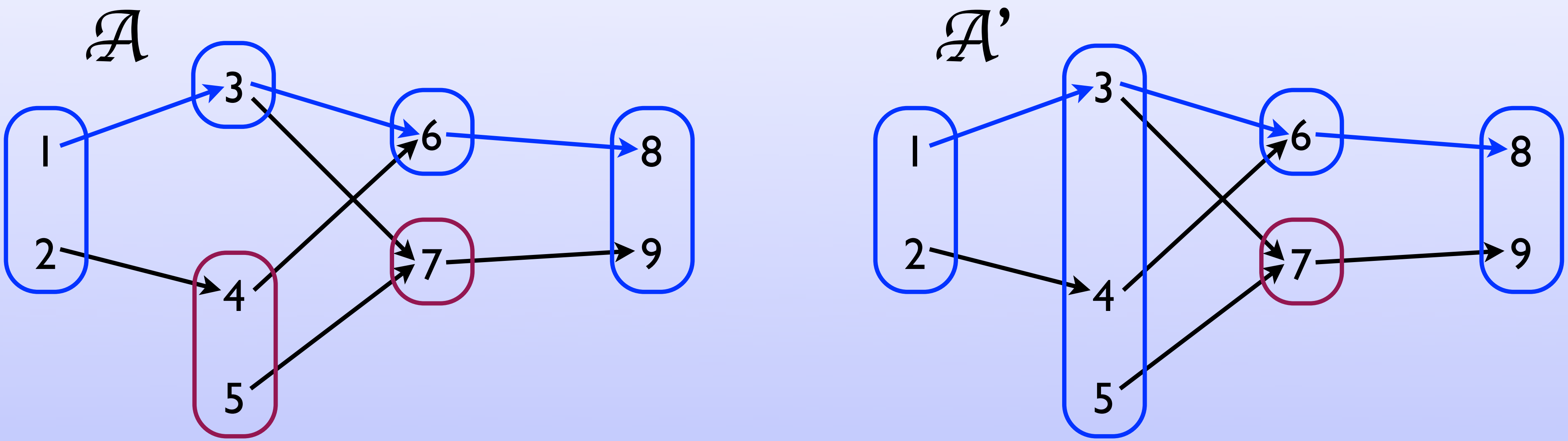


Example in abstract model checking



Not spurious abstract path in 𝒜': [1,2] → [3,4,5] → [6] → [8,9]

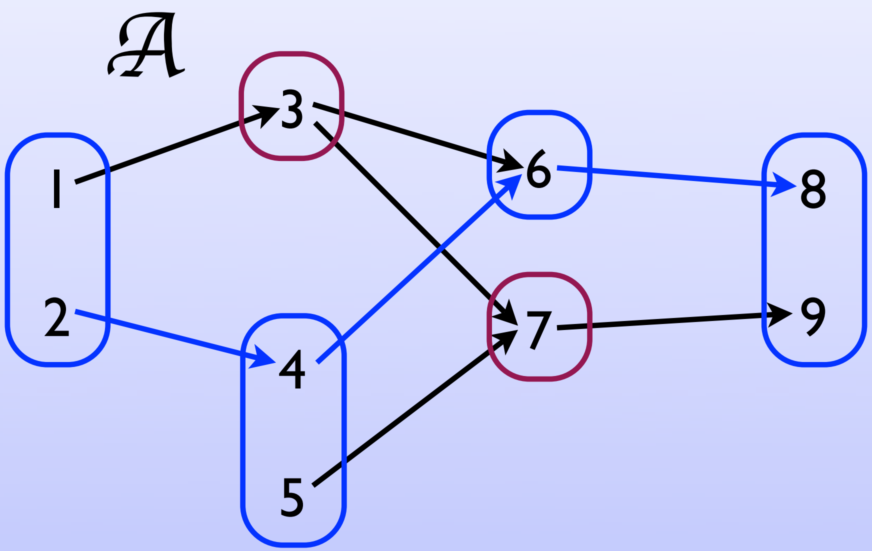
Example in abstract model checking

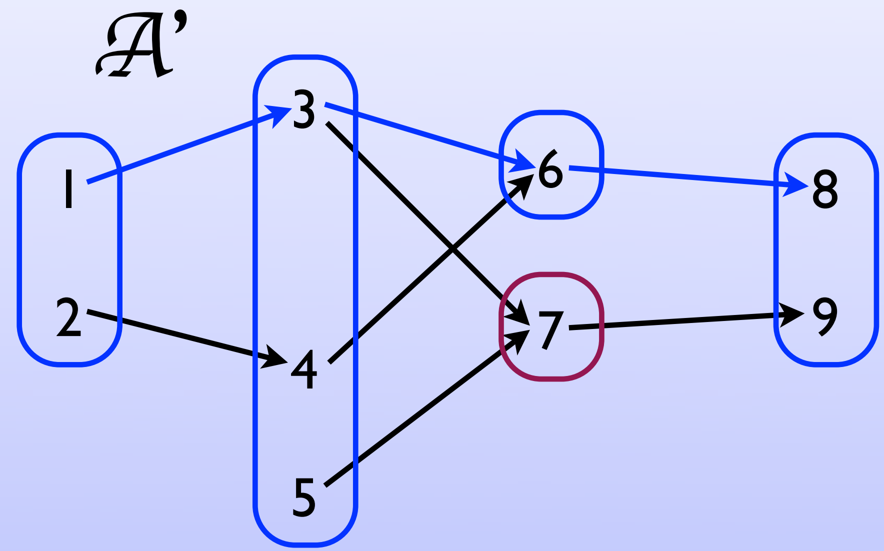


Not spurious abstract path in 𝒜': [1,2] → [3,4,5] → [6] → [8,9]

Not spurious abstract path in 𝒜: [1,2] → [3] → [6] → [8,9]

Example in abstract model checking



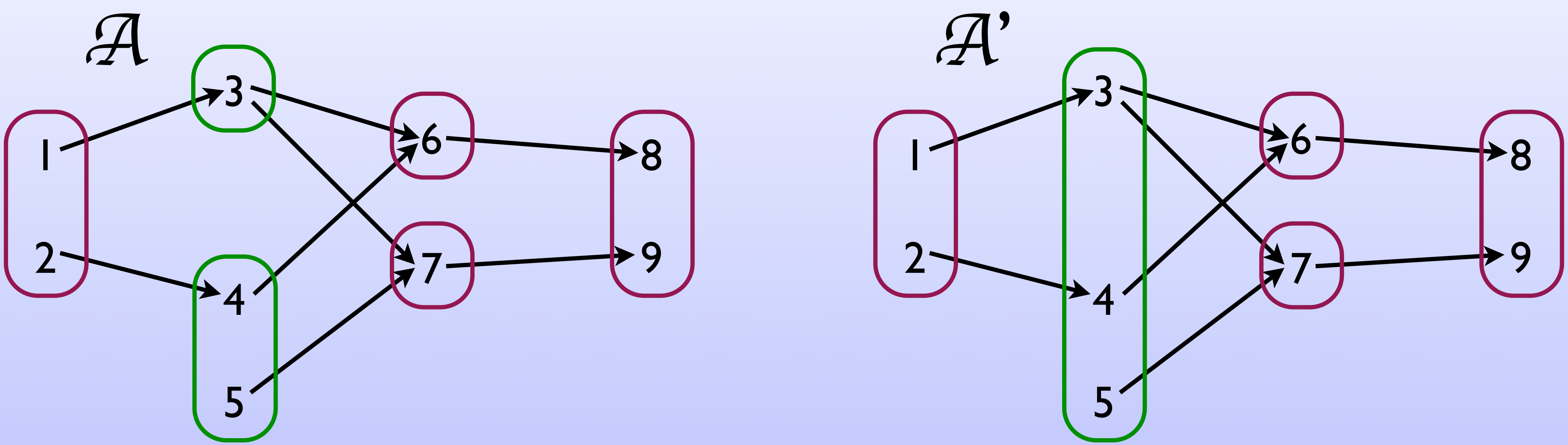


Not spurious abstract path in 𝒜': [1,2] → [3,4,5] → [6] → [8,9]

Not spurious abstract path in 𝒜: [1,2] → [3] → [6] → [8,9]

Not spurious abstract path in 𝒜: [1,2] → [4,5] → [6] → [8,9]

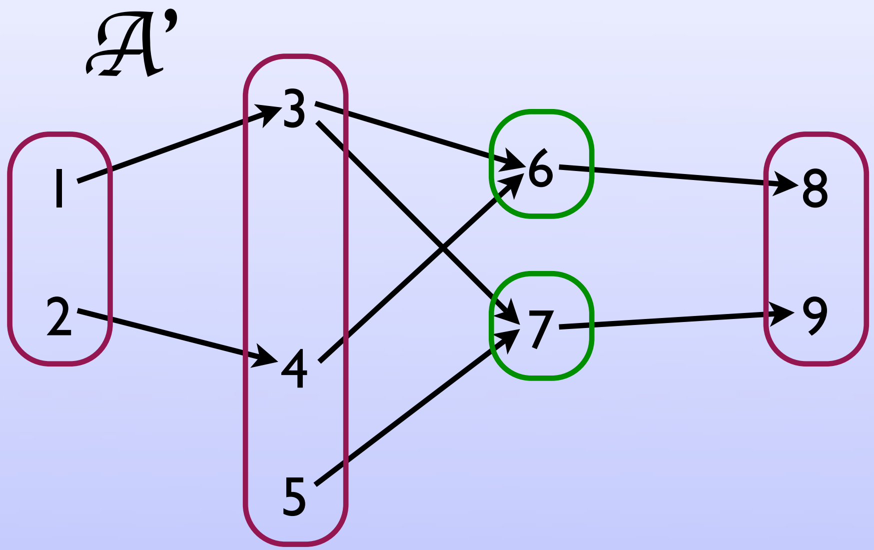
Example in abstract model checking

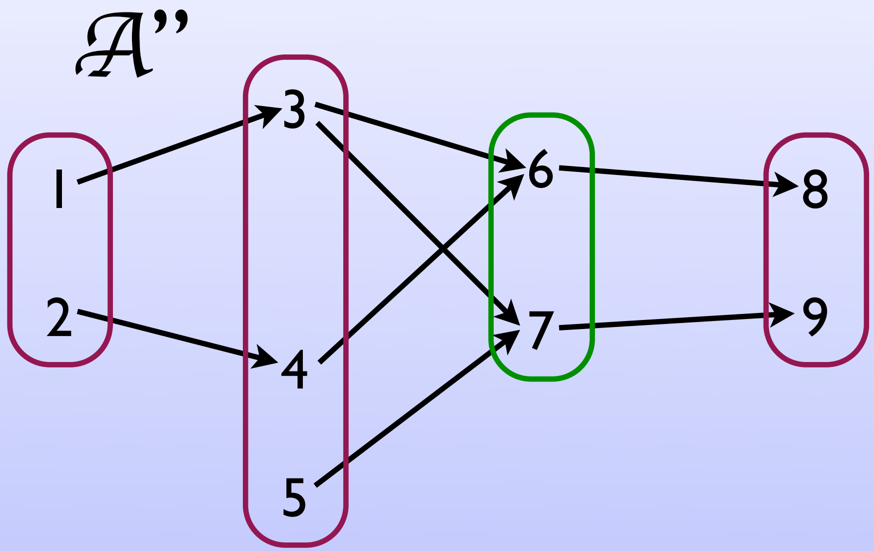


𝒜' keeps the same examples of 𝒜:

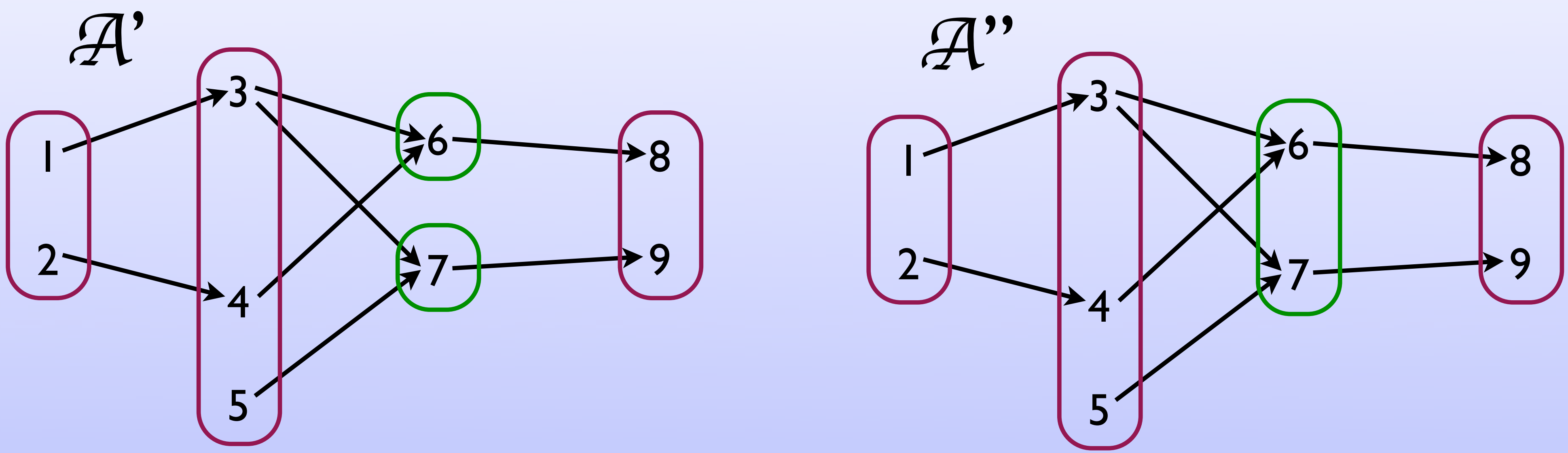
if π' is spurious in 𝒜' then there exists a spurious π in 𝒜 such that α(π)= π'

Example in abstract model checking



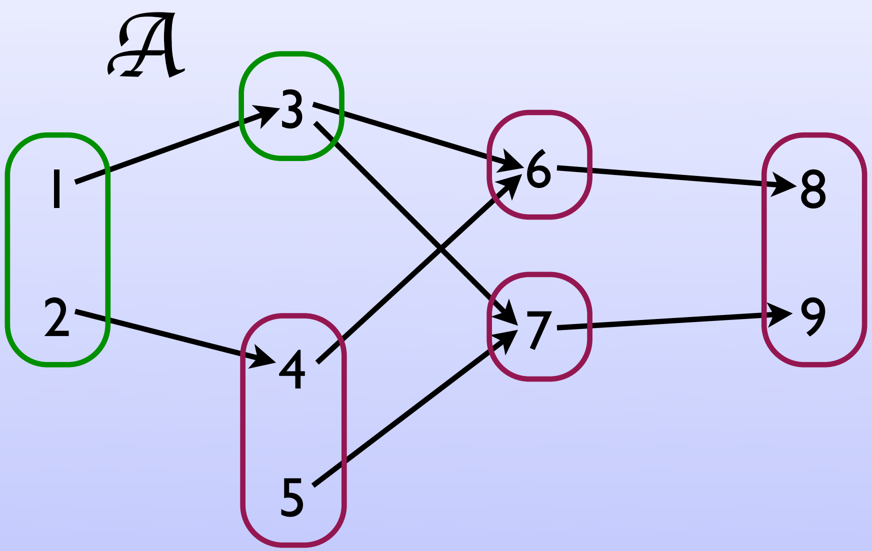


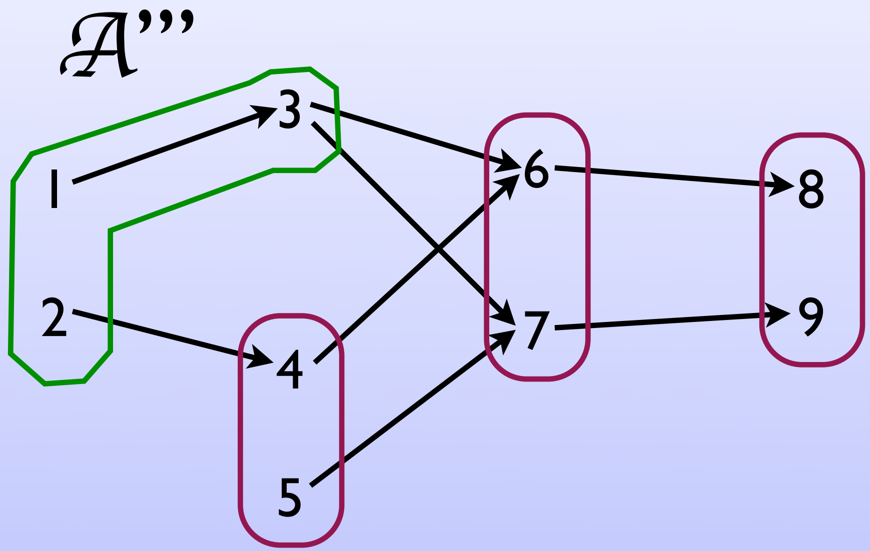
Example in abstract model checking



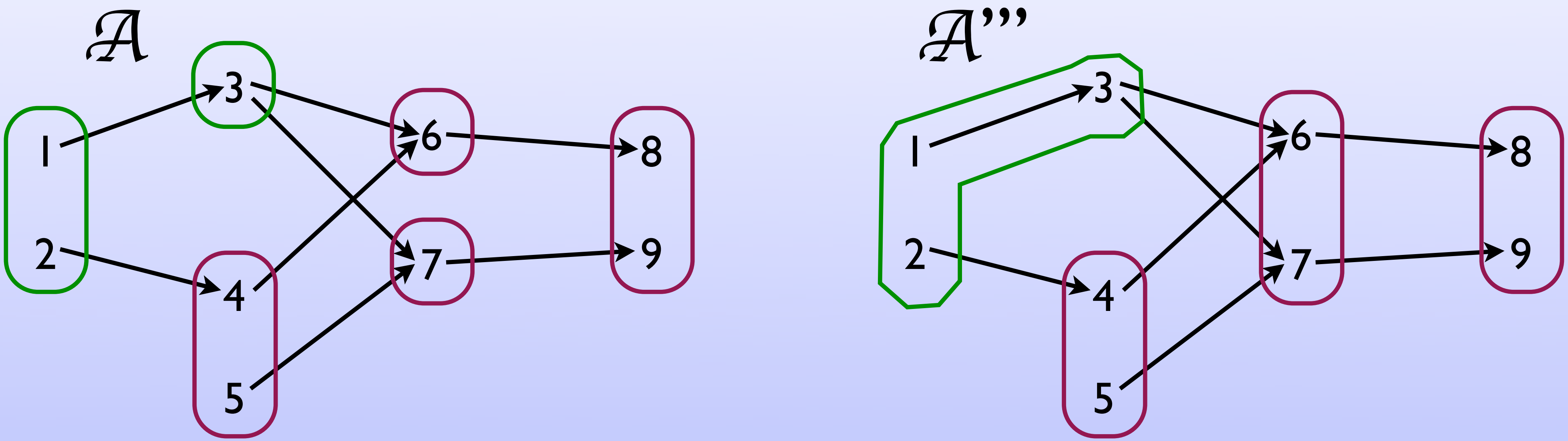
𝒜'' keeps the same examples of 𝒜'

Example in abstract model checking



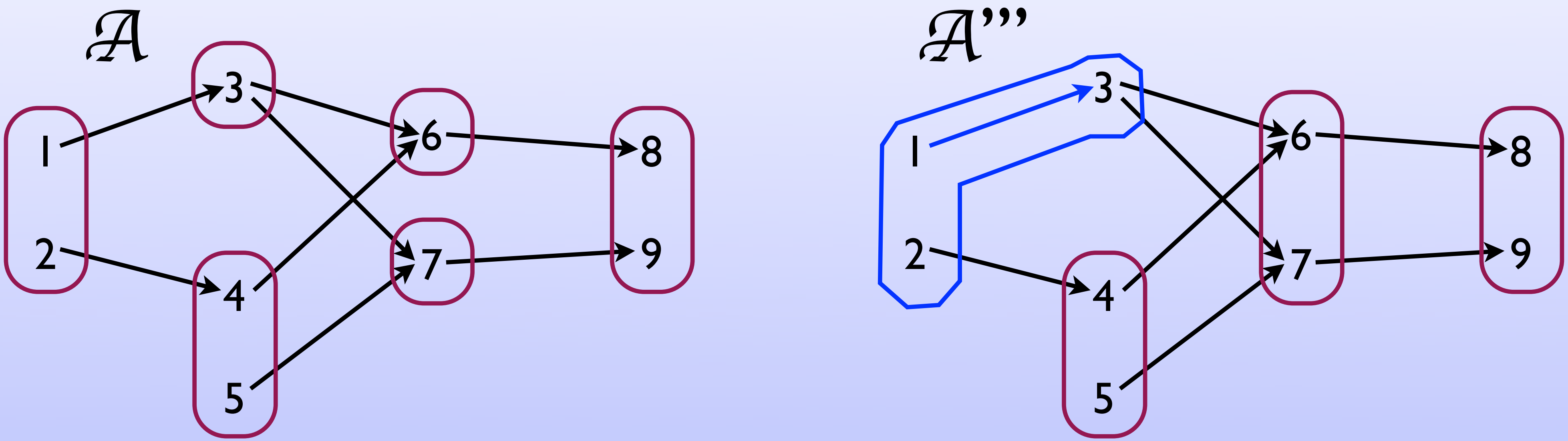


Example in abstract model checking



𝒜''' doesn't keep the same examples of 𝒜

Example in abstract model checking



𝒜''' doesn't keep the same examples of 𝒜

Spurious loop path in 𝒜''': [1,2,3] → [1,2,3] → [1,2,3] → ...

BUT no corresponding spurious path in 𝒜

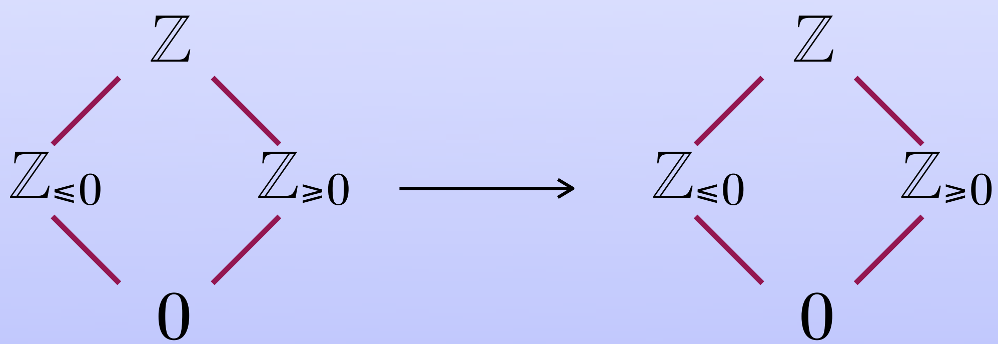
Example in abstract interpretation

$$x{+}{+} : \wp(\mathbb{Z}) \to \wp(\mathbb{Z})$$

Example in abstract interpretation

$$x{+}{+} : \wp(\mathbb{Z}) \rightarrow \wp(\mathbb{Z})$$

$x{+}{+}^{A_1}$:



$0{+}{+} = \mathbb{Z}_{\geq 0}$

$\mathbb{Z}_{\leq 0}{+}{+} = \mathbb{Z}$

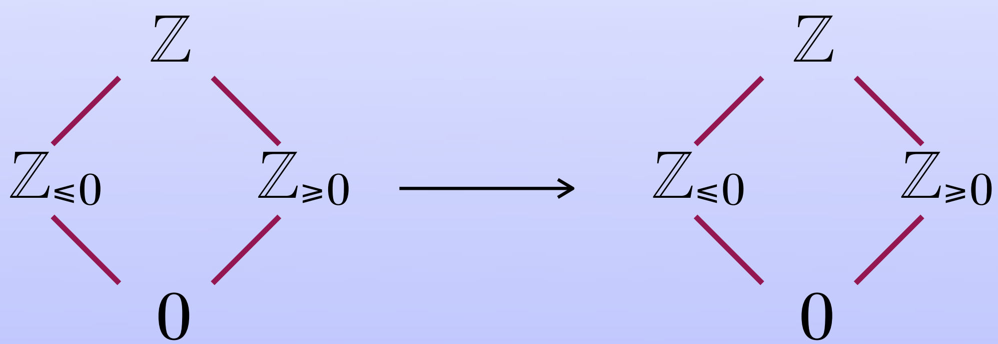
$\mathbb{Z}_{\geq 0}{+}{+} = \mathbb{Z}_{\geq 0}$

$\mathbb{Z}{+}{+} = \mathbb{Z}$

Example in abstract interpretation

$$x{+}{+} : \wp(\mathbb{Z}) \rightarrow \wp(\mathbb{Z})$$

$x{+}{+}^{A_1}$:



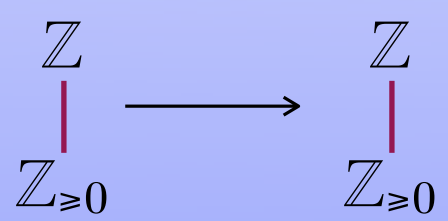
$0{+}{+} = \mathbb{Z}_{\geqslant 0}$

$\mathbb{Z}_{\leqslant 0}{+}{+} = \mathbb{Z}$

$\mathbb{Z}_{\geqslant 0}{+}{+} = \mathbb{Z}_{\geqslant 0}$

$\mathbb{Z}{+}{+} = \mathbb{Z}$

$x{+}{+}^{A_2}$:

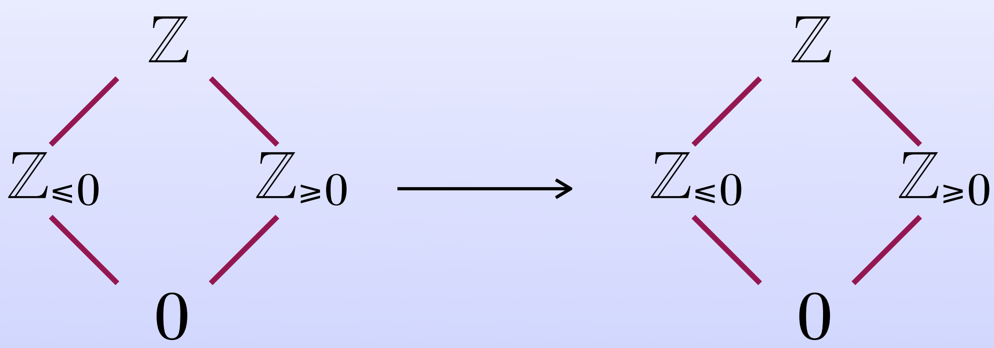


$\mathbb{Z}_{\geqslant 0}{+}{+} = \mathbb{Z}_{\geqslant 0}$

$\mathbb{Z}{+}{+} = \mathbb{Z}$

Example in abstract interpretation

$x{+}{+}^{A_1}$:



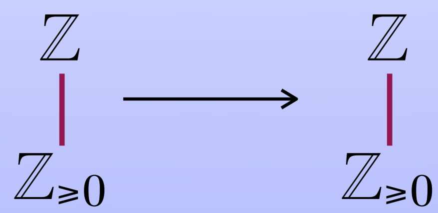
$0{+}{+} = \mathbb{Z}_{\geqslant 0}$

$\mathbb{Z}_{\leqslant 0}{+}{+} = \mathbb{Z}$

$\mathbb{Z}_{\geqslant 0}{+}{+} = \mathbb{Z}_{\geqslant 0}$

$\mathbb{Z}{+}{+} = \mathbb{Z}$

$x{+}{+}^{A_2}$:

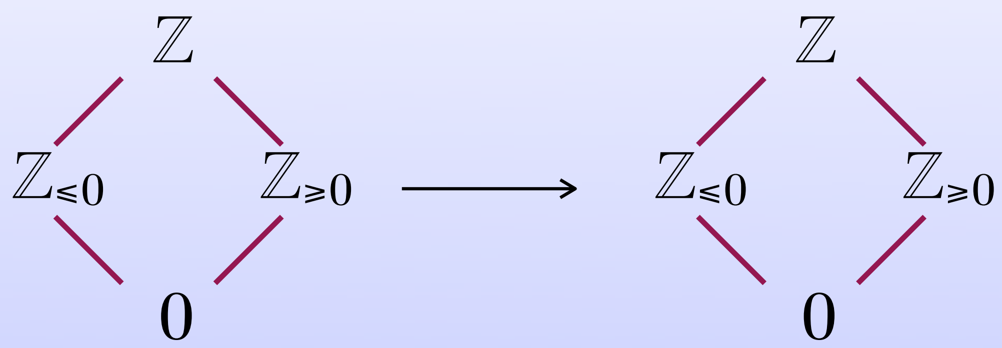


$\mathbb{Z}_{\geqslant 0}{+}{+} = \mathbb{Z}_{\geqslant 0}$

$\mathbb{Z}{+}{+} = \mathbb{Z}$

Example in abstract interpretation

$x{+}{+}^{A_1}$:



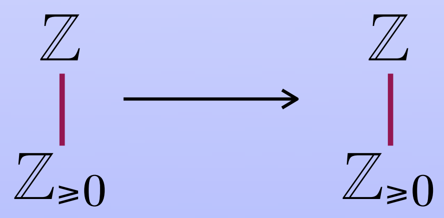
$0{+}{+} = \mathbb{Z}_{\geqslant 0}$

$\mathbb{Z}_{\leqslant 0}{+}{+} = \mathbb{Z}$

$\mathbb{Z}_{\geqslant 0}{+}{+} = \mathbb{Z}_{\geqslant 0}$

$\mathbb{Z}{+}{+} = \mathbb{Z}$

$x{+}{+}^{A_2}$:



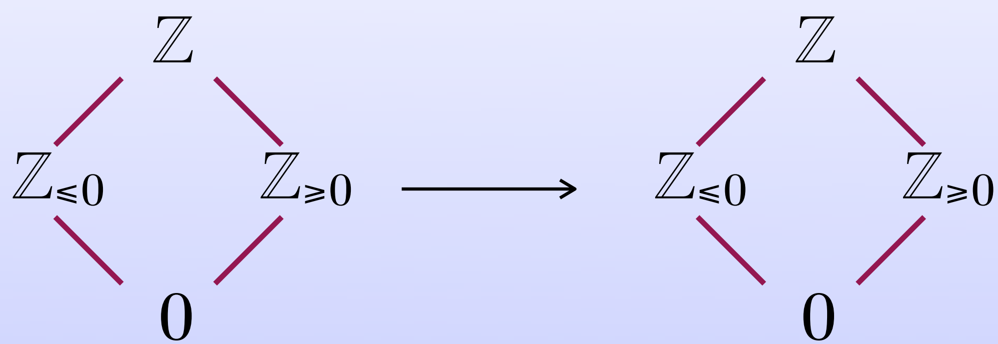
$\mathbb{Z}_{\geqslant 0}{+}{+} = \mathbb{Z}_{\geqslant 0}$

$\mathbb{Z}{+}{+} = \mathbb{Z}$

$x{+}{+}^{A_1},\ x{+}{+}^{A_2}$ encode the same function in $\wp(\mathbb{Z}) \to \wp(\mathbb{Z})$

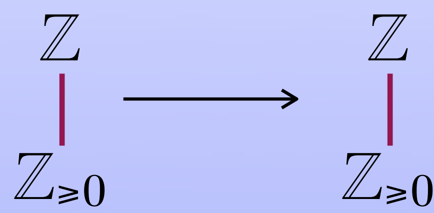
Example in abstract interpretation

$x{+}{+}^{A_1}$:



$0{+}{+} = \mathbb{Z}_{\geq 0}$
$\mathbb{Z}_{\leq 0}{+}{+} = \mathbb{Z}$
$\mathbb{Z}_{\geq 0}{+}{+} = \mathbb{Z}_{\geq 0}$
$\mathbb{Z}{+}{+} = \mathbb{Z}$

$x{+}{+}^{A_2}$:



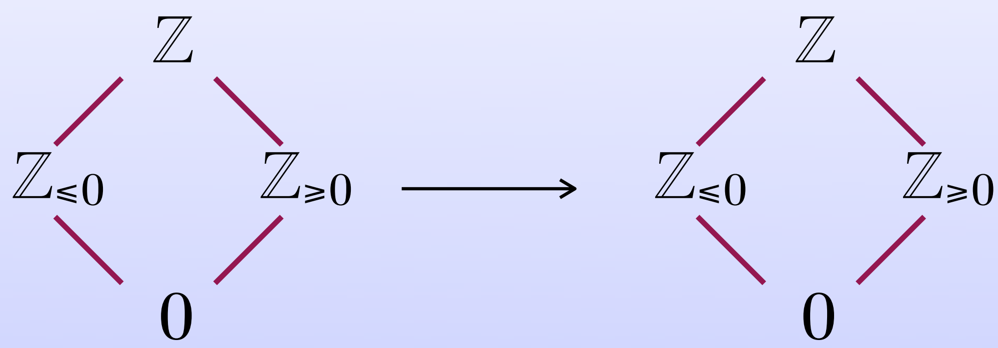
$\mathbb{Z}_{\geq 0}{+}{+} = \mathbb{Z}_{\geq 0}$
$\mathbb{Z}{+}{+} = \mathbb{Z}$

$x{+}{+}^{A_1}$, $x{+}{+}^{A_2}$ encode the same function in $\wp(\mathbb{Z}) \to \wp(\mathbb{Z})$

$$(\gamma_{A_1} \circ \alpha_{A_1}) \circ x{+}{+} \circ (\gamma_{A_1} \circ \alpha_{A_1}) = (\gamma_{A_2} \circ \alpha_{A_2}) \circ x{+}{+} \circ (\gamma_{A_2} \circ \alpha_{A_2})$$

Example in abstract interpretation

$x{+}{+}^{A_1}$:



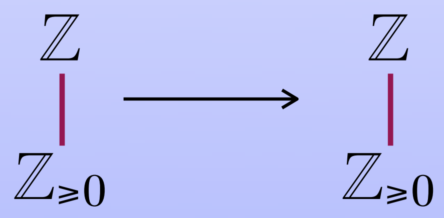
$0{+}{+} = \mathbb{Z}_{\geqslant 0}$

$\mathbb{Z}_{\leqslant 0}{+}{+} = \mathbb{Z}$

$\mathbb{Z}_{\geqslant 0}{+}{+} = \mathbb{Z}_{\geqslant 0}$

$\mathbb{Z}{+}{+} = \mathbb{Z}$

$x{+}{+}^{A_2}$:



$\mathbb{Z}_{\geqslant 0}{+}{+} = \mathbb{Z}_{\geqslant 0}$

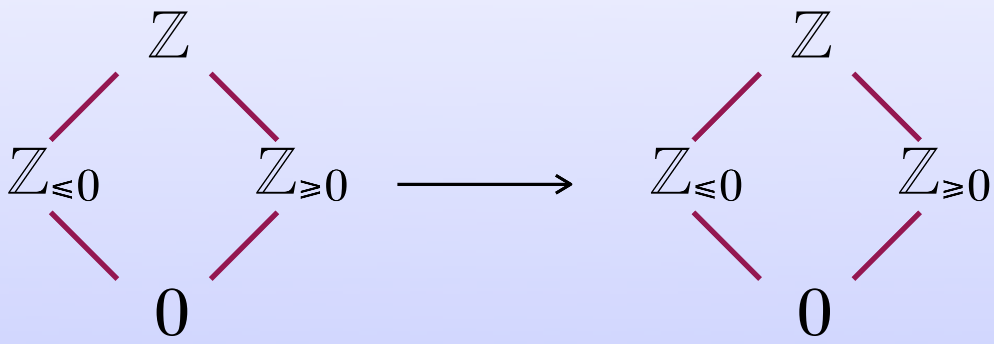
$\mathbb{Z}{+}{+} = \mathbb{Z}$

$x{+}{+}^{A_1},\ x{+}{+}^{A_2}$ encode the same function in $\wp(\mathbb{Z}) \to \wp(\mathbb{Z})$

0 and $\mathbb{Z}_{\leqslant 0}$ are "irrelevant" in A_1 for approximating $x{+}{+}$

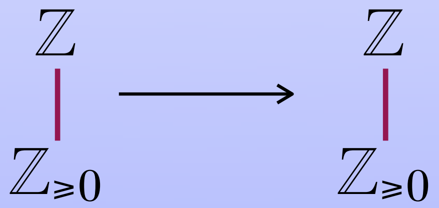
Example in abstract interpretation

$x{+}{+}^{A_1}$:



$0{+}{+} = \mathbb{Z}_{\geqslant 0}$
$\mathbb{Z}_{\leqslant 0}{+}{+} = \mathbb{Z}$
$\mathbb{Z}_{\geqslant 0}{+}{+} = \mathbb{Z}_{\geqslant 0}$
$\mathbb{Z}{+}{+} = \mathbb{Z}$

$x{+}{+}^{A_2}$:

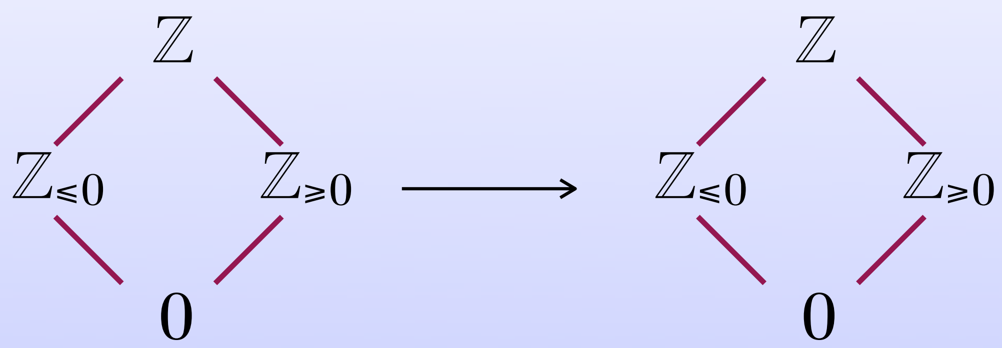


$\mathbb{Z}_{\geqslant 0}{+}{+} = \mathbb{Z}_{\geqslant 0}$
$\mathbb{Z}{+}{+} = \mathbb{Z}$

0 and $\mathbb{Z}_{\leqslant 0}$ are "irrelevant" in A_1 for approximating $x{+}{+}$

Example in abstract interpretation

$x{+}{+}^{A_1}$:



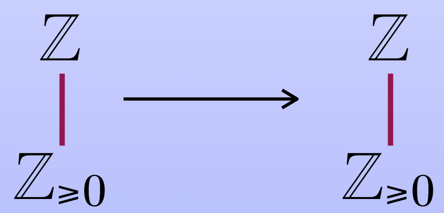
$0{+}{+} = \mathbb{Z}_{\geq 0}$

$\mathbb{Z}_{\leq 0}{+}{+} = \mathbb{Z}$

$\mathbb{Z}_{\geq 0}{+}{+} = \mathbb{Z}_{\geq 0}$

$\mathbb{Z}{+}{+} = \mathbb{Z}$

$x{+}{+}^{A_2}$:



$\mathbb{Z}_{\geq 0}{+}{+} = \mathbb{Z}_{\geq 0}$

$\mathbb{Z}{+}{+} = \mathbb{Z}$

0 and $\mathbb{Z}_{\leq 0}$ are "irrelevant" in A_1 for approximating $x{+}{+}$

$$\{0,-2,-7\} \xrightarrow{A_1} \mathbb{Z}_{\leq 0} \xrightarrow{++} \{\mathrm{x}\leq 1\} \xrightarrow{A_1} \mathbb{Z}$$

$$\{0,-2,-7\} \xrightarrow{A_2} \mathbb{Z} \xrightarrow{++} \mathbb{Z} \xrightarrow{A_2} \mathbb{Z}$$

Abstract interpretation

- Problem formalized in abstract interpretation

Abstract interpretation

- Problem formalized in abstract interpretation
- Main ingredients

Abstract interpretation

- Problem formalized in abstract interpretation
- Main ingredients
  - Approximation formalized by partial orders

Abstract interpretation

- Problem formalized in abstract interpretation
- Main ingredients
  - Approximation formalized by partial orders
  - Concrete domain $C_{\leqslant}$

Abstract interpretation

- Problem formalized in abstract interpretation
- Main ingredients
  - Approximation formalized by partial orders
  - Concrete domain $C_{\leqslant}$
  - Abstractions $A_{\leqslant}$ formalized by Galois connections α/γ

Abstract interpretation

- Problem formalized in abstract interpretation
- Main ingredients
  - Approximation formalized by partial orders
  - Concrete domain $C_{\leqslant}$
  - Abstractions $A_{\leqslant}$ formalized by Galois connections α/γ
  - Concrete objects c have best correct approximations $\alpha(c)$

Abstract interpretation

- Problem formalized in abstract interpretation
- Main ingredients
  - Approximation formalized by partial orders
  - Concrete domain $C_{\leqslant}$
  - Abstractions $A_{\leqslant}$ formalized by Galois connections α/γ
  - Concrete objects c have best correct approximations $\alpha(c)$
  - Semantic functions $f : C \to C$ have best correct approximations $f^A \stackrel{\text{def}}{=} \alpha \circ f \circ \gamma : A \to A$

Correctness Kernel

Concrete semantic function $f: C \rightarrow C$

Abstract domain $A \in Abs(C)$

$\alpha_A: C \rightarrow A \qquad \gamma_A: A \rightarrow C$

Abstract domain $B \in Abs(C)$

$\alpha_B: C \rightarrow B \qquad \gamma_B: B \rightarrow C$

Correctness Kernel

Concrete semantic function $f: C \to C$

Abstract domain $A \in Abs(C)$
$\alpha_A: C \to A \qquad \gamma_A: A \to C$

Abstract domain $B \in Abs(C)$
$\alpha_B: C \to B \qquad \gamma_B: B \to C$

$f^A = f^B$ when
$(\gamma_A\alpha_A) \circ f \circ (\gamma_A\alpha_A) = (\gamma_B\alpha_B) \circ f \circ (\gamma_B\alpha_B)$

That is, the best correct approximations of function f in A and B coincide when encoded within C

Correctness Kernel

$$f^A = f^B \text{ when}$$
$$(\gamma_A \alpha_A) \circ f \circ (\gamma_A \alpha_A) = (\gamma_B \alpha_B) \circ f \circ (\gamma_B \alpha_B)$$

Correctness Kernel

$f^A = f^B$ when

$(\gamma_A \alpha_A) \circ f \circ (\gamma_A \alpha_A) = (\gamma_B \alpha_B) \circ f \circ (\gamma_B \alpha_B)$

Correctness kernel $K_f(A)$ of A for f:

$K_f(A) \stackrel{\text{def}}{=}$ most abstract domain B such that $f^B = f^A$

Correctness Kernel

$$f^A = f^B \text{ when}$$
$$(\gamma_A \alpha_A) \circ f \circ (\gamma_A \alpha_A) = (\gamma_B \alpha_B) \circ f \circ (\gamma_B \alpha_B)$$

Main Technical Result

If $f \circ (\gamma_A \alpha_A)$ is continuous then $K_f(A)$ exists and

$$K_f(A) = img(f^A) \cup \bigcup_{y \in img(f^A)} \max(\{x \in A \mid f^A(x) = y\})$$

Correctness Kernel

$$f^A = f^B \text{ when}$$
$$(\gamma_A \alpha_A) \circ f \circ (\gamma_A \alpha_A) = (\gamma_B \alpha_B) \circ f \circ (\gamma_B \alpha_B)$$

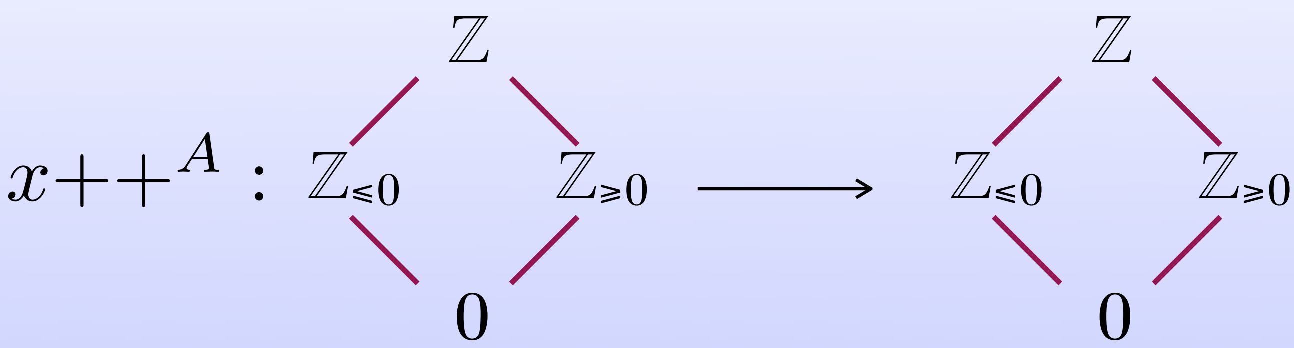
Main Technical Result

If $f \circ (\gamma_A \alpha_A)$ is continuous then $K_f(A)$ exists and

$$K_f(A) = img(f^A) \cup \bigcup_{y \in img(f^A)} \max(\{x \in A \mid f^A(x) = y\})$$

Proof relies on the notion of complete abstract interpretation

Example



$0{+}{+} = \mathbb{Z}_{\geqslant 0}$

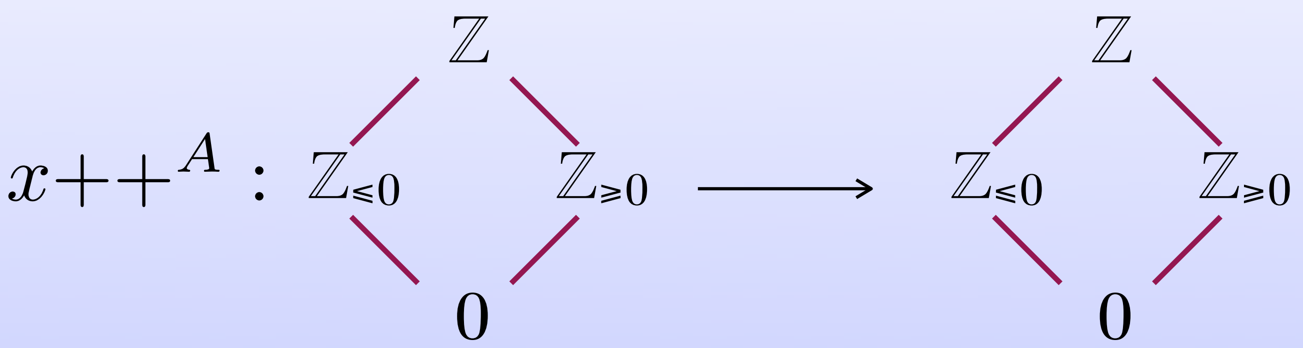
$\mathbb{Z}_{\leqslant 0}{+}{+} = \mathbb{Z}$

$\mathbb{Z}_{\geqslant 0}{+}{+} = \mathbb{Z}_{\geqslant 0}$

$\mathbb{Z}{+}{+} = \mathbb{Z}$

$$K_f(A) = \mathrm{img}(f^A) \cup \bigcup_{y \in \mathrm{img}(f^A)} \max(\{x \in A \mid f^A(x) = y\})$$

Example



$0{+}{+} = \mathbb{Z}_{\geq 0}$

$\mathbb{Z}_{\leq 0}{+}{+} = \mathbb{Z}$

$\mathbb{Z}_{\geq 0}{+}{+} = \mathbb{Z}_{\geq 0}$

$\mathbb{Z}{+}{+} = \mathbb{Z}$

$$K_f(A) = \mathrm{img}(f^A) \cup \bigcup_{y \in \mathrm{img}(f^A)} \max(\{x \in A \mid f^A(x) = y\})$$

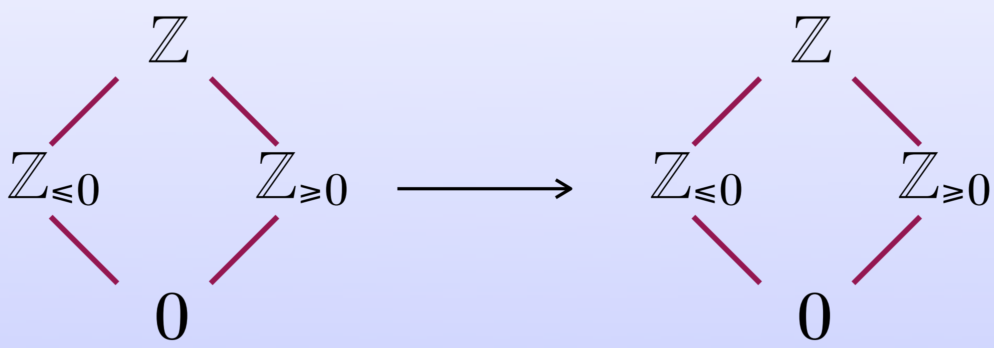
$$\mathrm{img}({+}{+}^A) = \{\, \mathbb{Z}, \mathbb{Z}_{\geq 0} \,\}$$

$$\max(\{x \in A \mid {+}{+}^A(x) = \mathbb{Z}\}) = \max(\{\mathbb{Z}_{\leq 0}, \mathbb{Z}\}) = \mathbb{Z}$$

$$\max(\{x \in A \mid {+}{+}^A(x) = \mathbb{Z}_{\geq 0}\}) = \max(\{0, \mathbb{Z}_{\geq 0}\}) = \mathbb{Z}_{\geq 0}$$

Example

$x{+}{+}^A$:



$0{+}{+} = \mathbb{Z}_{\geq 0}$

$\mathbb{Z}_{\leq 0}{+}{+} = \mathbb{Z}$

$\mathbb{Z}_{\geq 0}{+}{+} = \mathbb{Z}_{\geq 0}$

$\mathbb{Z}{+}{+} = \mathbb{Z}$

$$K_f(A) = \mathrm{img}(f^A) \cup \bigcup_{y \in \mathrm{img}(f^A)} \max(\{x \in A \mid f^A(x) = y\})$$

$$\mathrm{img}({+}{+}^A) = \{\, \mathbb{Z}, \mathbb{Z}_{\geq 0} \,\}$$

$$\max(\{x \in A \mid {+}{+}^A(x) = \mathbb{Z}\}) = \max(\{\mathbb{Z}_{\leq 0}, \mathbb{Z}\}) = \mathbb{Z}$$

$$\max(\{x \in A \mid {+}{+}^A(x) = \mathbb{Z}_{\geq 0}\}) = \max(\{0, \mathbb{Z}_{\geq 0}\}) = \mathbb{Z}_{\geq 0}$$

$$K_{++}(A) = \{\, \mathbb{Z}, \mathbb{Z}_{\geq 0} \,\}$$

Abstract Model Checking

Concrete Kripke structure $< \Sigma, \rightarrow, \ell >$

Abstract Model Checking

Concrete Kripke structure $<\Sigma, \rightarrow, \ell>$

Abstract state space P is a partition of Σ

Abstract Kripke structure $<P, \rightarrow^{\exists\exists}, \ell>$

$B \rightarrow^{\exists\exists} C$ iff there exist $x \in B$ and $y \in C$ s.t. $x \rightarrow y$

Abstract Model Checking

Concrete functions:

predecessor pre: $\wp(\Sigma) \rightarrow \wp(\Sigma)$

successor post: $\wp(\Sigma) \rightarrow \wp(\Sigma)$

Abstract Model Checking

Concrete functions:

predecessor pre: $\wp(\Sigma) \rightarrow \wp(\Sigma)$

successor post: $\wp(\Sigma) \rightarrow \wp(\Sigma)$

Partition P can be viewed as an abstraction of $\wp(\Sigma)$

Abstract Model Checking

Concrete functions:

predecessor pre: $\wp(\Sigma) \rightarrow \wp(\Sigma)$

successor post: $\wp(\Sigma) \rightarrow \wp(\Sigma)$

Partition P can be viewed as an abstraction of $\wp(\Sigma)$

What is the correctness kernel of P for pre and post?

Abstract Model Checking

What is the correctness kernel *K*(P) of P for pre and post?

Abstract Model Checking

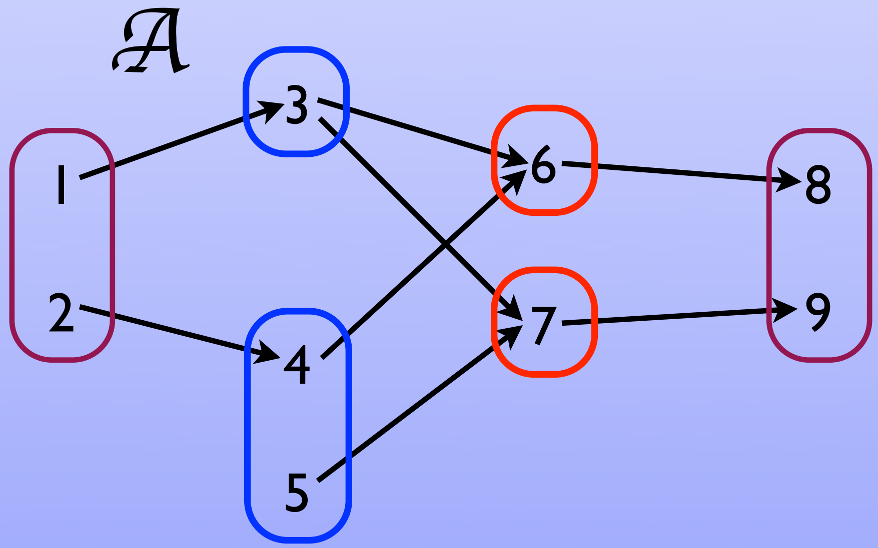
What is the correctness kernel $K(P)$ of P for pre and post?

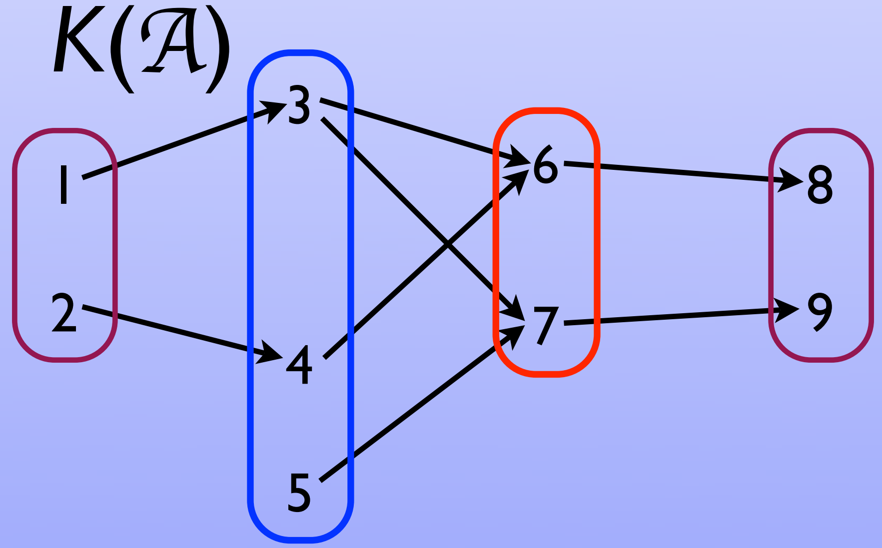
$K(P)$ merges two blocks B_1 and B_2 iff for any $A \in P$,
$A \rightarrow^{\exists\exists} B_1 \Leftrightarrow A \rightarrow^{\exists\exists} B_2$ and $B_1 \rightarrow^{\exists\exists} A \Leftrightarrow B_2 \rightarrow^{\exists\exists} A$

Abstract Model Checking

What is the correctness kernel $K(\mathrm{P})$ of P for pre and post?

$K(\mathrm{P})$ merges two blocks B_1 and B_2 iff for any $A \in P$,
$A \rightarrow^{\exists\exists} B_1 \Leftrightarrow A \rightarrow^{\exists\exists} B_2$ and $B_1 \rightarrow^{\exists\exists} A \Leftrightarrow B_2 \rightarrow^{\exists\exists} A$





EGAS

EGAS: Example-Guided Abstraction Simplification

EGAS

EGAS: Example-Guided Abstraction Simplification

Abstract Kripke structure $<P , \rightarrow^{\exists\exists}>$

Correctness Kernel $<K(P) , \rightarrow^{\exists\exists}>$

EGAS

EGAS: Example-Guided Abstraction Simplification

Abstract Kripke structure $<P , \rightarrow^{\exists\exists}>$

Correctness Kernel $<\mathit{K}(P) , \rightarrow^{\exists\exists}>$

Correctness kernels do not add spurious paths

if π is a spurious path in $\mathit{K}(P)$ then there exists a spurious path σ in P such that $\alpha(\sigma)= \pi$

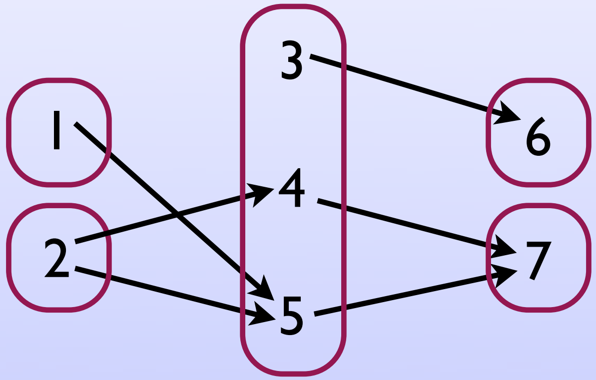
CEGAR

1) Model checker provides an abstract path (i.e. a counterexample)
$\pi = B_1 \rightarrow^{\exists\exists} B_2 \rightarrow^{\exists\exists} B_3 \ldots \rightarrow^{\exists\exists} B_n$

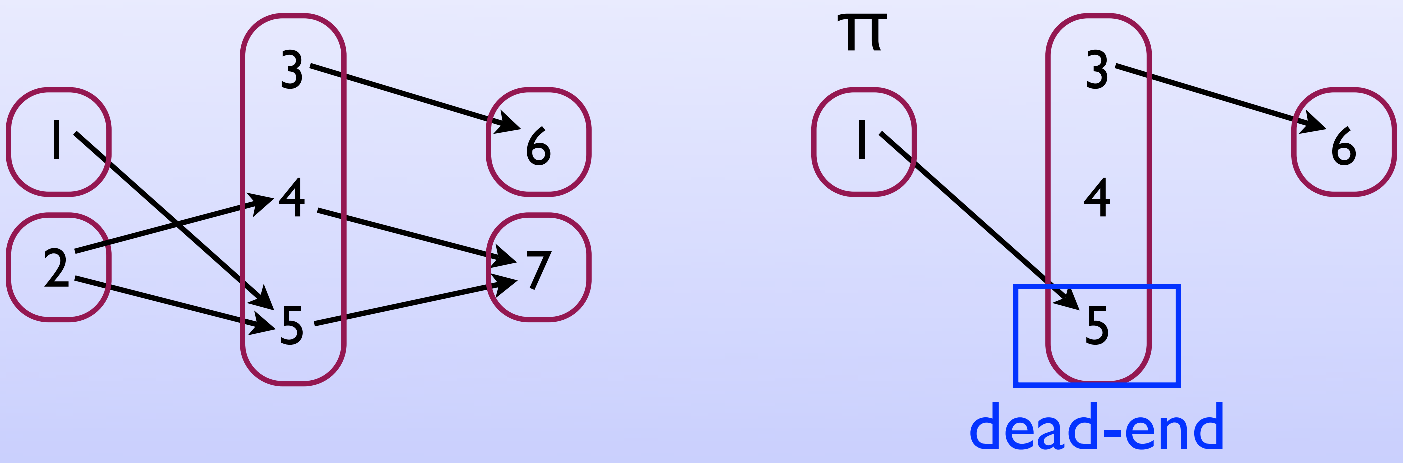
2) CEGAR determines whether π is spurious or not

3) Spuriousness of π depends on some block B_k of π with bad and dead-end states. Thus, CEGAR splits B_k in order to separate bad and dead-end states.

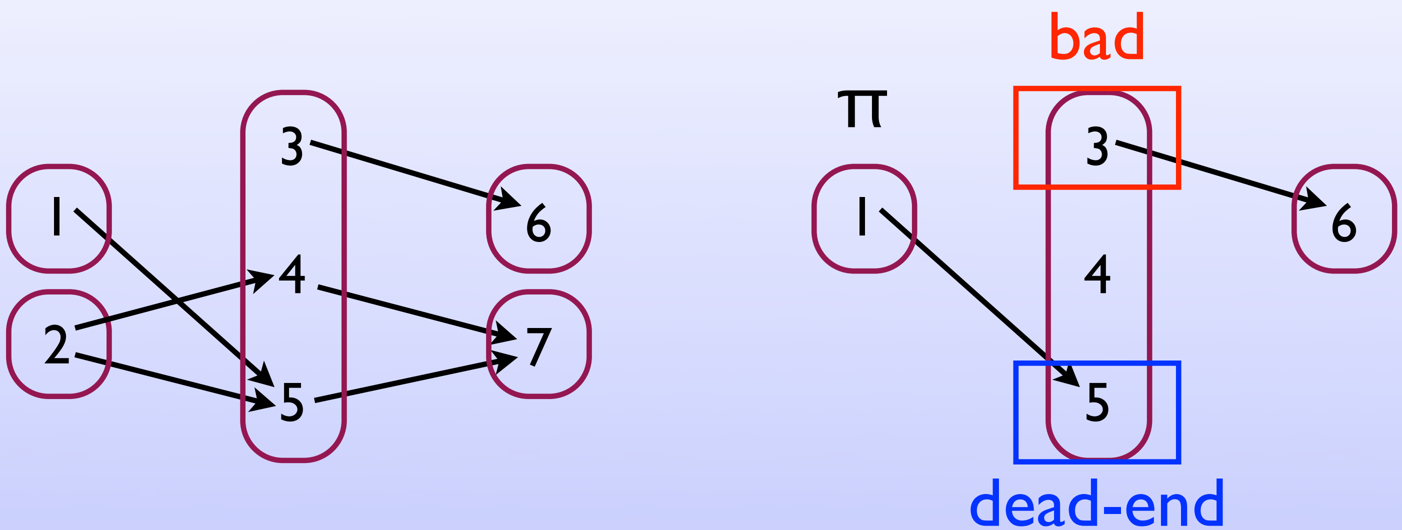
CEGAR



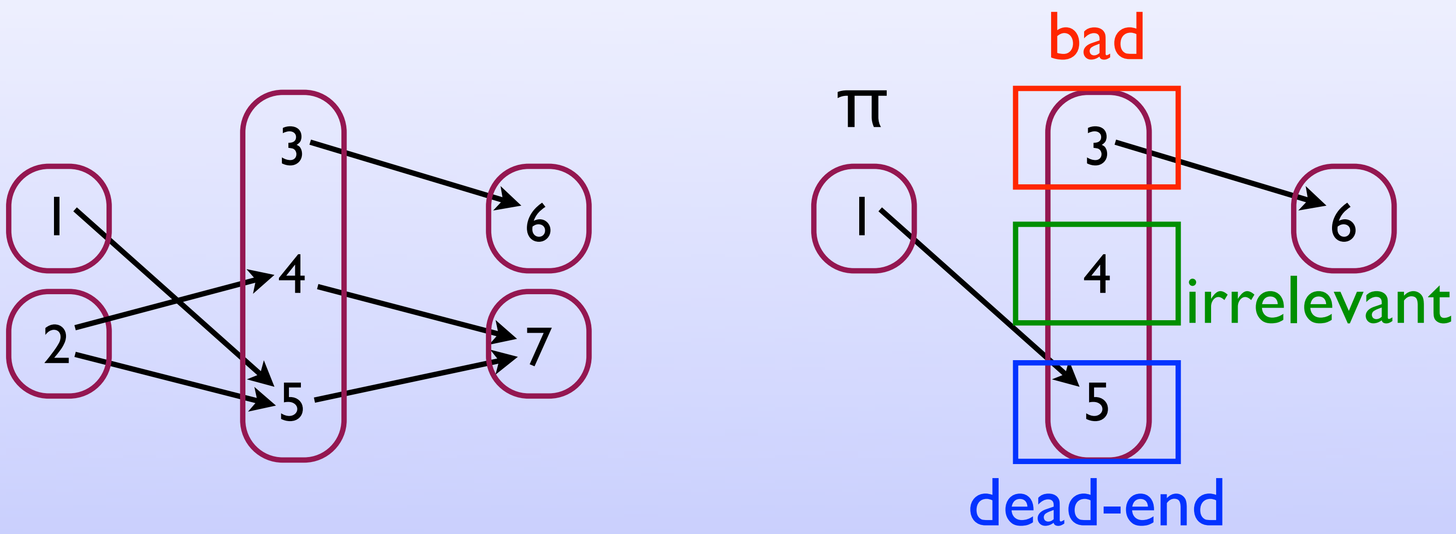
CEGAR



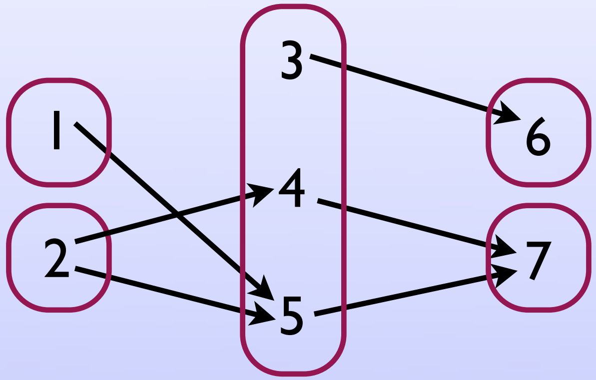
CEGAR

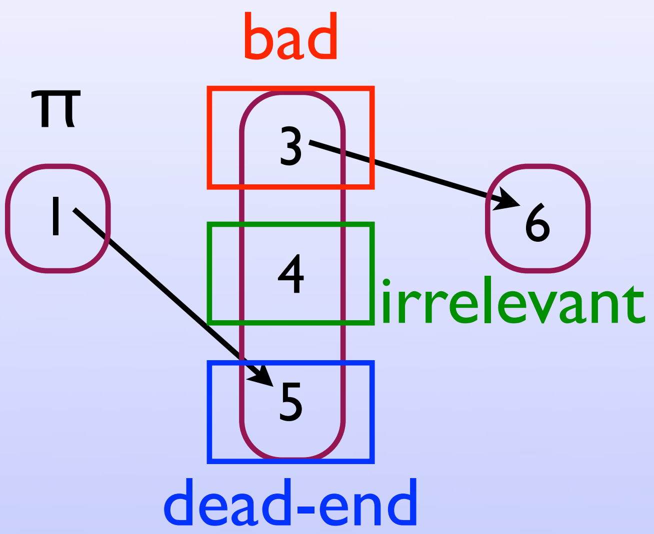


CEGAR



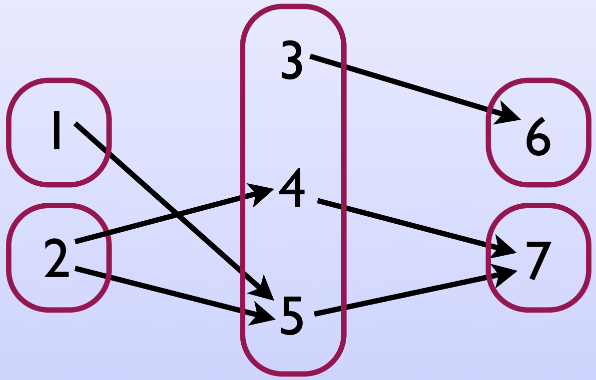
CEGAR

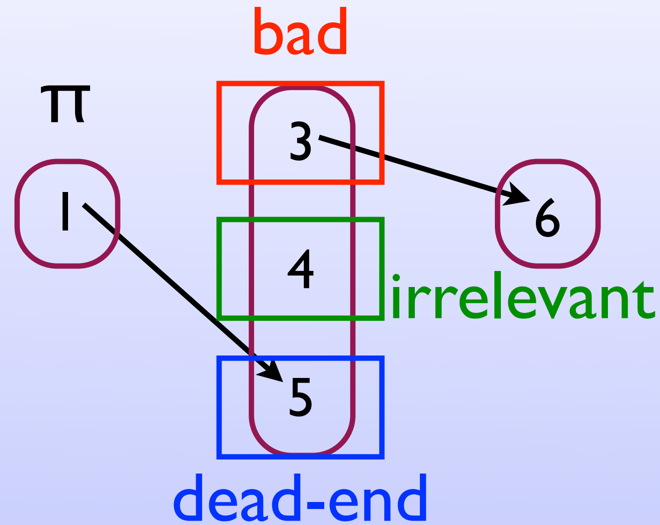




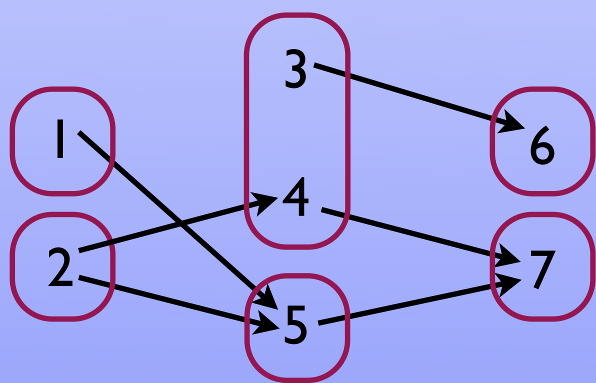
Finding the coarsest refinement is NP-hard
CEGAR heuristics: split into dead-end and bad U irrelevant

CEGAR

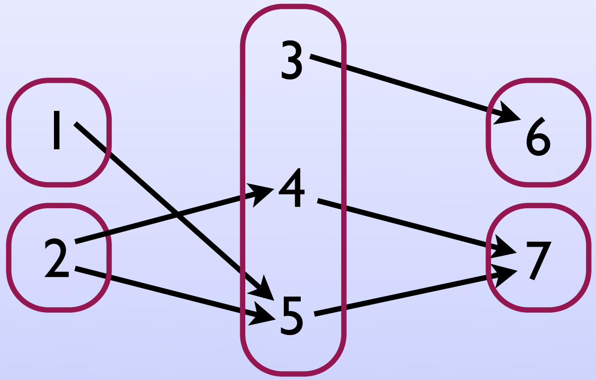


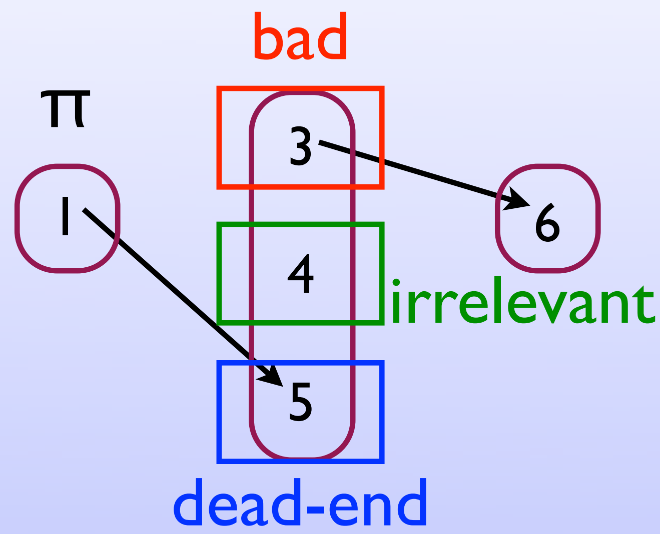


Finding the coarsest refinement is NP-hard
CEGAR heuristics: split into dead-end and bad U irrelevant

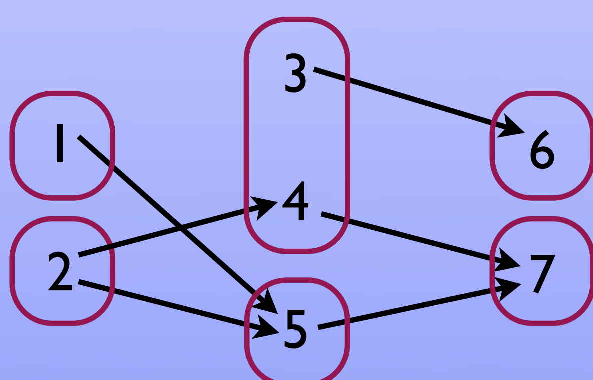


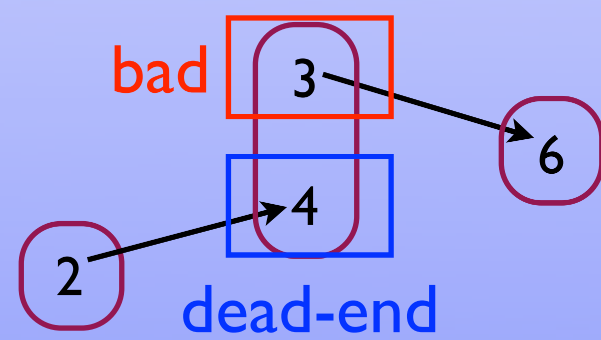
CEGAR



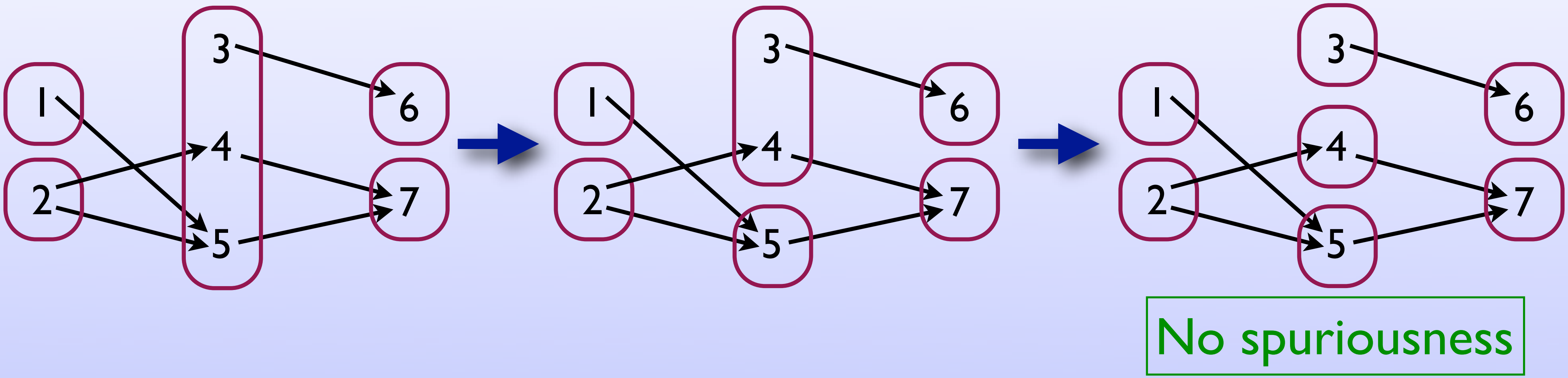


Finding the coarsest refinement is NP-hard
CEGAR heuristics: split into dead-end and bad U irrelevant

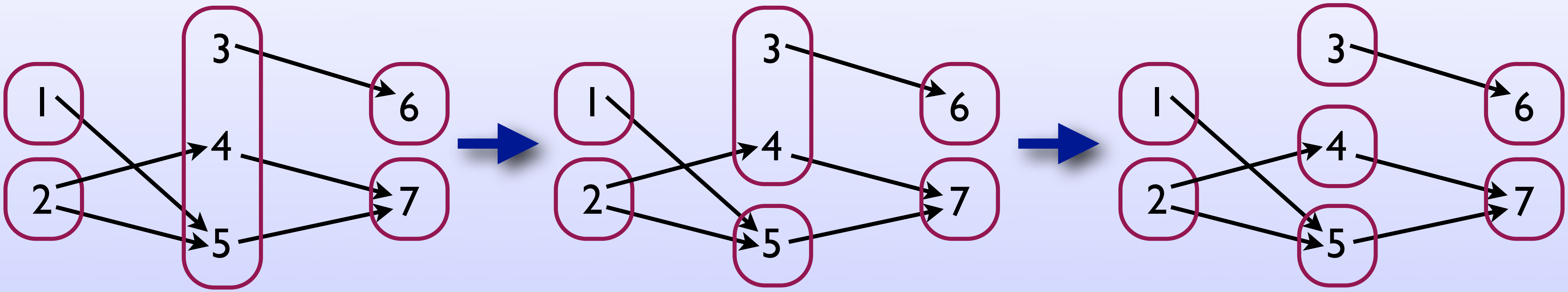




CEGAR



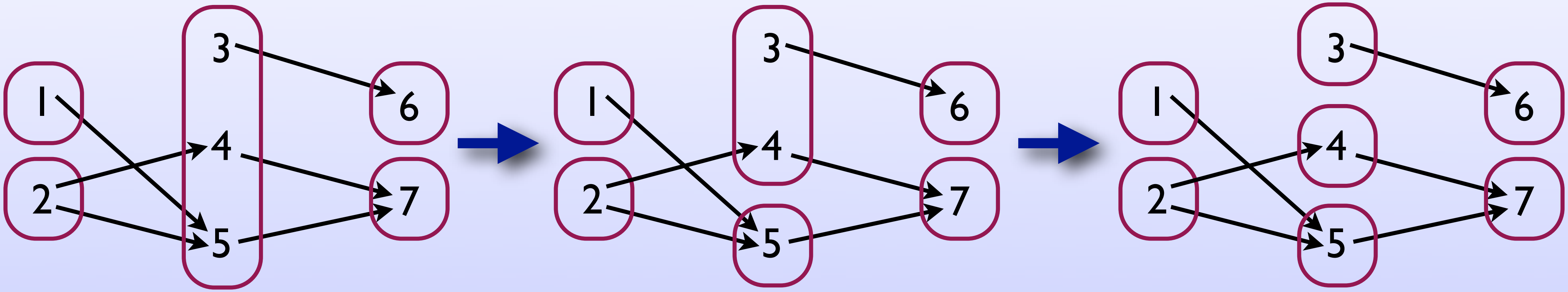
CEGAR



No spuriousness

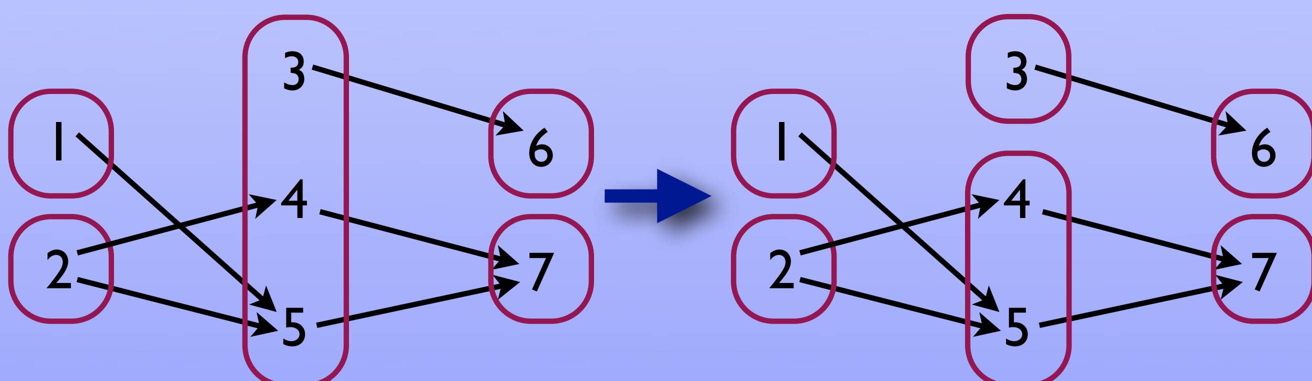
When irrelevant are joined with dead-end:

CEGAR



No spuriousness

When irrelevant are joined with dead-end:



No spuriousness

EGAS and CEGAR

CEGAR heuristics may lead to ineffective abstraction refinements

EGAS and CEGAR

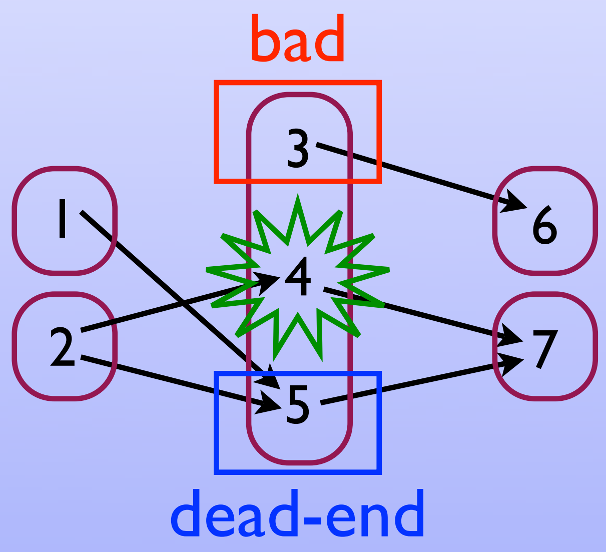
CEGAR heuristics may lead to ineffective abstraction refinements

EGAS suggests a sharper refinement heuristics

EGAS and CEGAR

CEGAR heuristics may lead to ineffective abstraction refinements

EGAS suggests a sharper refinement heuristics

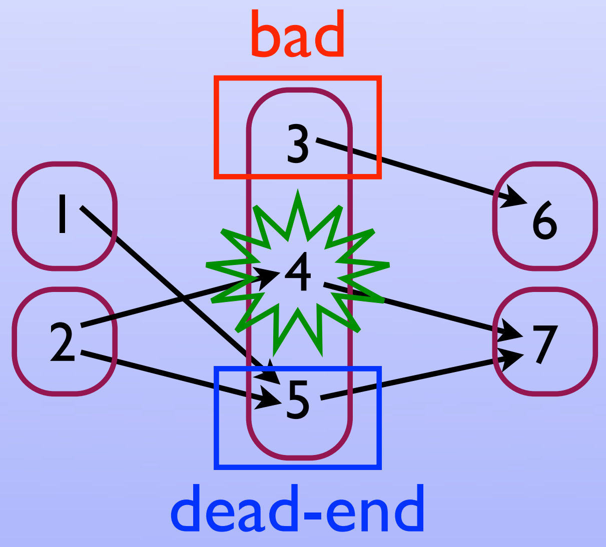


The state irrelevant 4 is dead-irrelevant
1) can be reached from a block that also reaches a dead-end
2) can reach a block that is also reached by a dead-end

EGAS and CEGAR

CEGAR heuristics may lead to ineffective abstraction refinements

EGAS suggests a sharper refinement heuristics



The state irrelevant 4 is dead-irrelevant
1) can be reached from a block that also reaches a dead-end
2) can reach a block that is also reached by a dead-end

Thus, by EGAS, merging dead-irrelevant states with dead-end states does not add spurious paths wrt keeping them separate

EGAS Refinement Heuristics

Dead-irrelevant states
1) can be reached from a block that also reaches a dead-end
2) can reach a block that is also reached by a dead-end

EGAS Refinement Heuristics

Dead-irrelevant states
1) can be reached from a block that also reaches a dead-end
2) can reach a block that is also reached by a dead-end

Bad-irrelevant states
1) can be reached from a block that also reaches a bad
2) can reach a block that is also reached by a bad

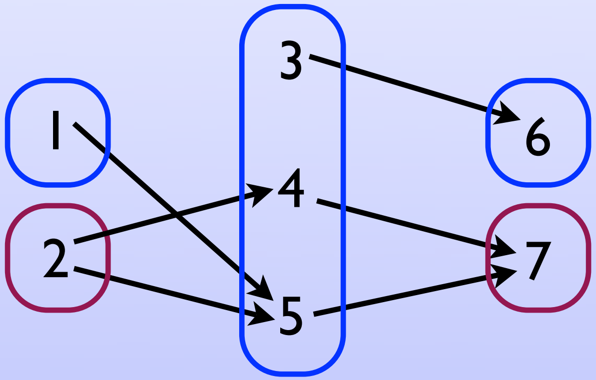
EGAS Refinement Heuristics

Dead-irrelevant states
1) can be reached from a block that also reaches a dead-end
2) can reach a block that is also reached by a dead-end

Bad-irrelevant states
1) can be reached from a block that also reaches a bad
2) can reach a block that is also reached by a bad

Fully-irrelevant states
1) neither bad- nor dead-irrelevant OR
2) both bad- and dead-irrelevant

EGAS Refinement Heuristics



EGAS Refinement Heuristics

Related Work

Related Work

- Core of an abstract domain [Giacobazzi et al.]
  - Given an abstract domain property P, this is the most concrete simplification of A that satisfies P

Related Work

- Core of an abstract domain [Giacobazzi et al.]
  - Given an abstract domain property P, this is the most concrete simplification of A that satisfies P
- Compressor of an abstract domain [Giacobazzi et al.]
  - Given a refinement Ref, this is the most abstract simplification of A such that: Ref(Compressor(A))=Ref(A)

Conclusions

Conclusions

- First step in studying abstraction simplifications in static analysis and model checking

Conclusions

- First step in studying abstraction simplifications in static analysis and model checking

- Future work

  - precise relationship between EGAS and CEGAR

Conclusions

- First step in studying abstraction simplifications in static analysis and model checking

- Future work

  - precise relationship between EGAS and CEGAR

  - integrating EGAS in CEGAR