Translation Validation of System Abstractions

Jan Olaf Blech Ina Schaefer Arnd Poetzsch-Heffter

University of Kaiserslautern

DASMOD
Dependable Adaptive Systems and Mathematical Modeling

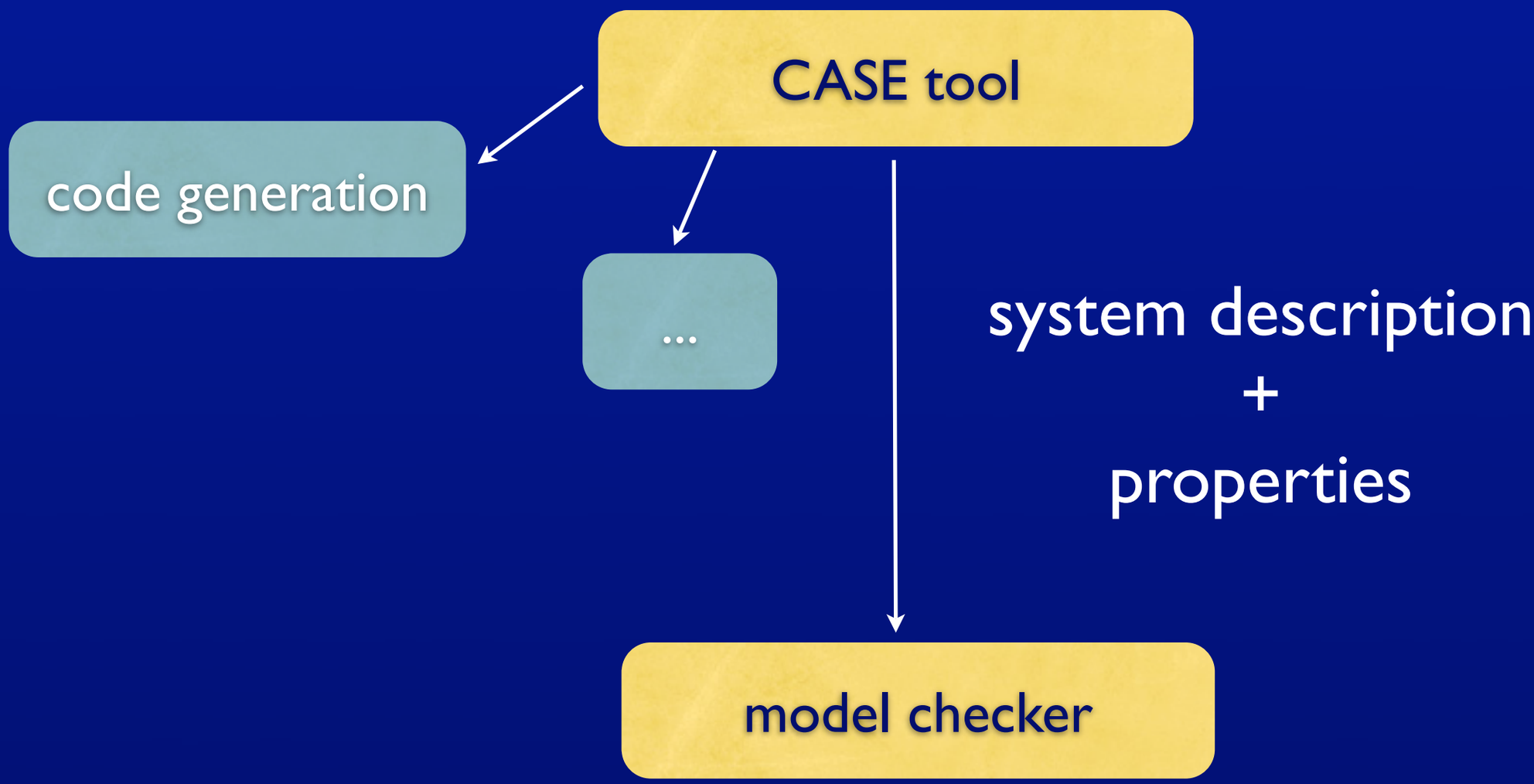
Motivation

CASE tool

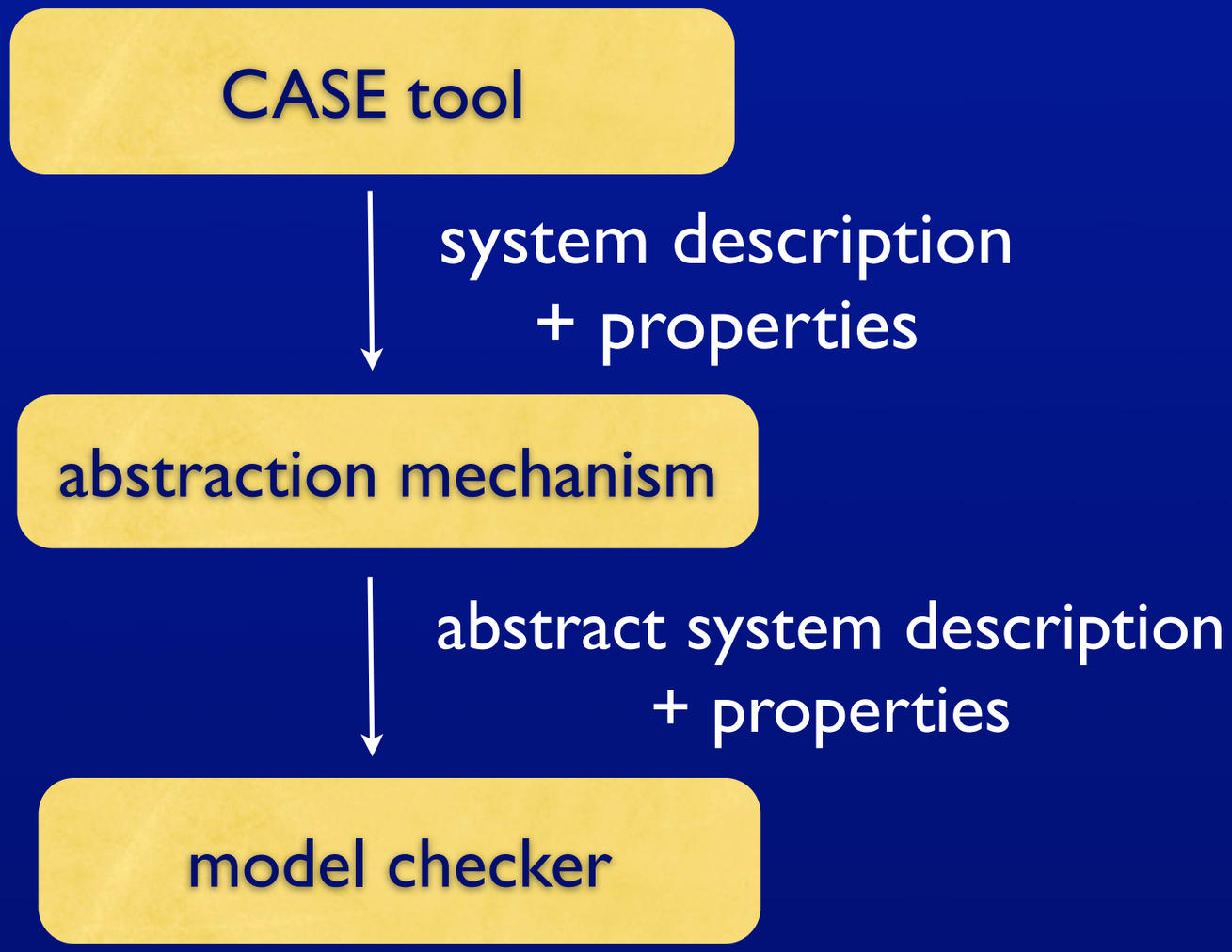
system description
+
properties

model checker

Motivation

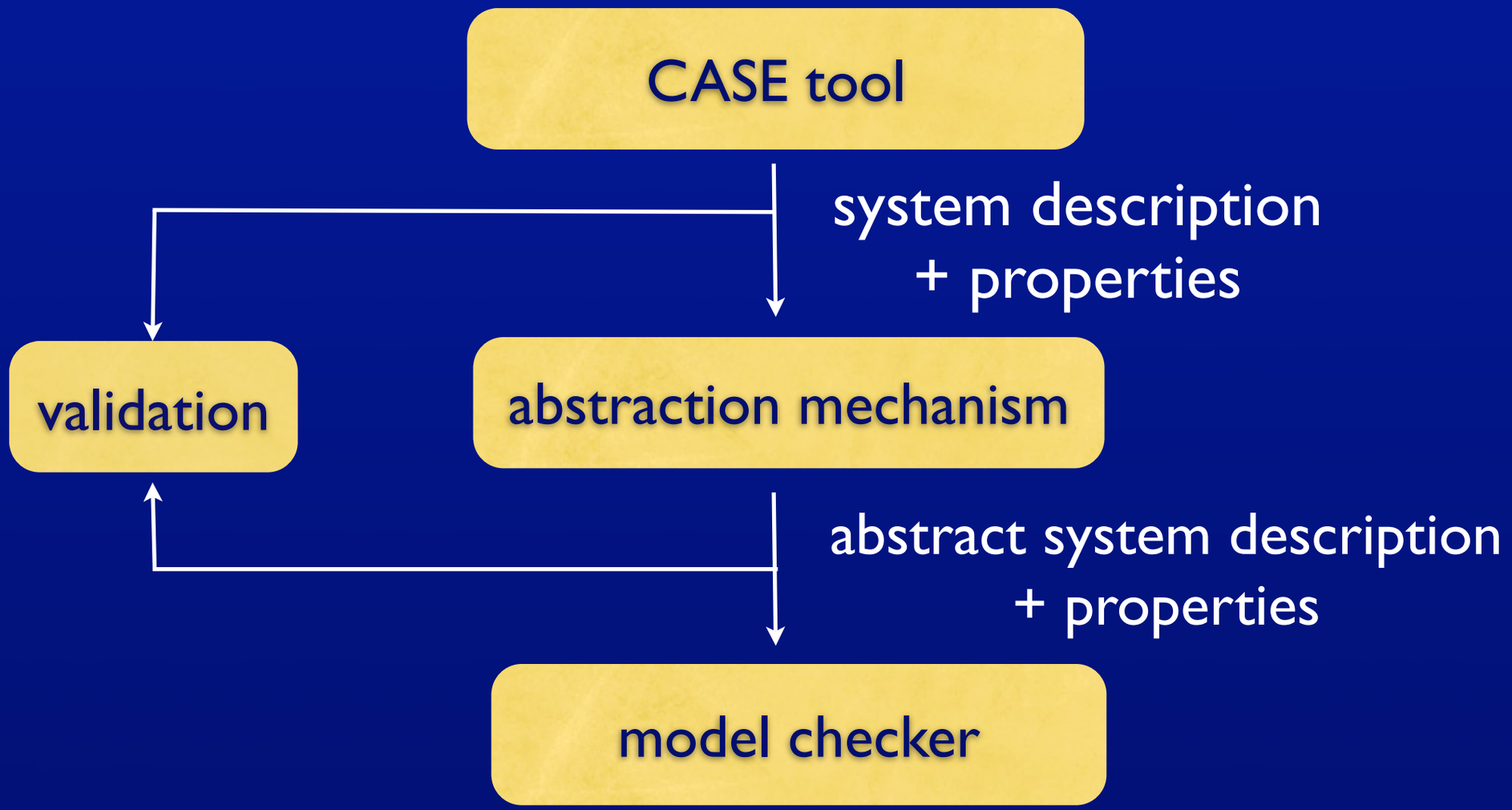

CASE tool
code generation
...
system description
+
properties
model checker

Motivation


CASE tool
system description
+ properties
abstraction mechanism
abstract system description
+ properties
model checker

Motivation

Translation Validation

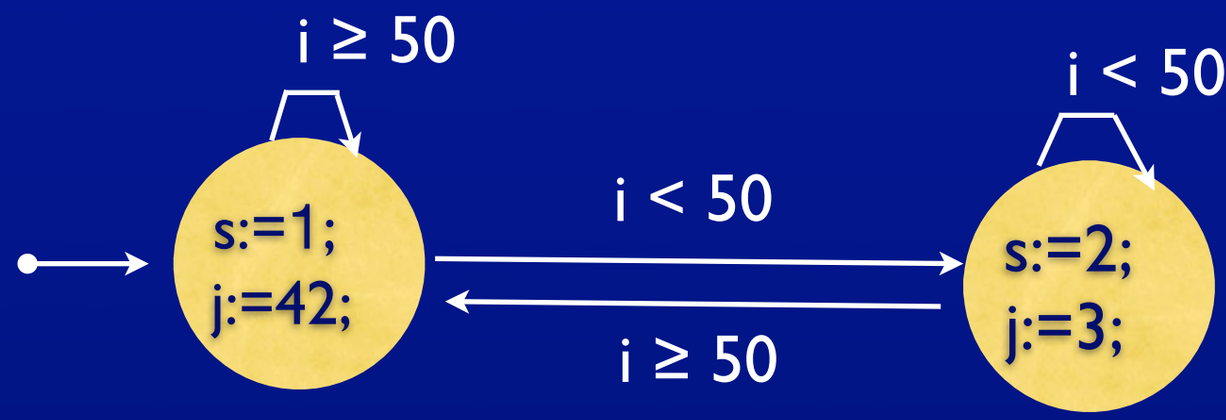


Overview

- Example
- Proving Abstractions Correct
- Translation Validating Abstractions
- Evaluation
- Conclusion & Future Work

Example

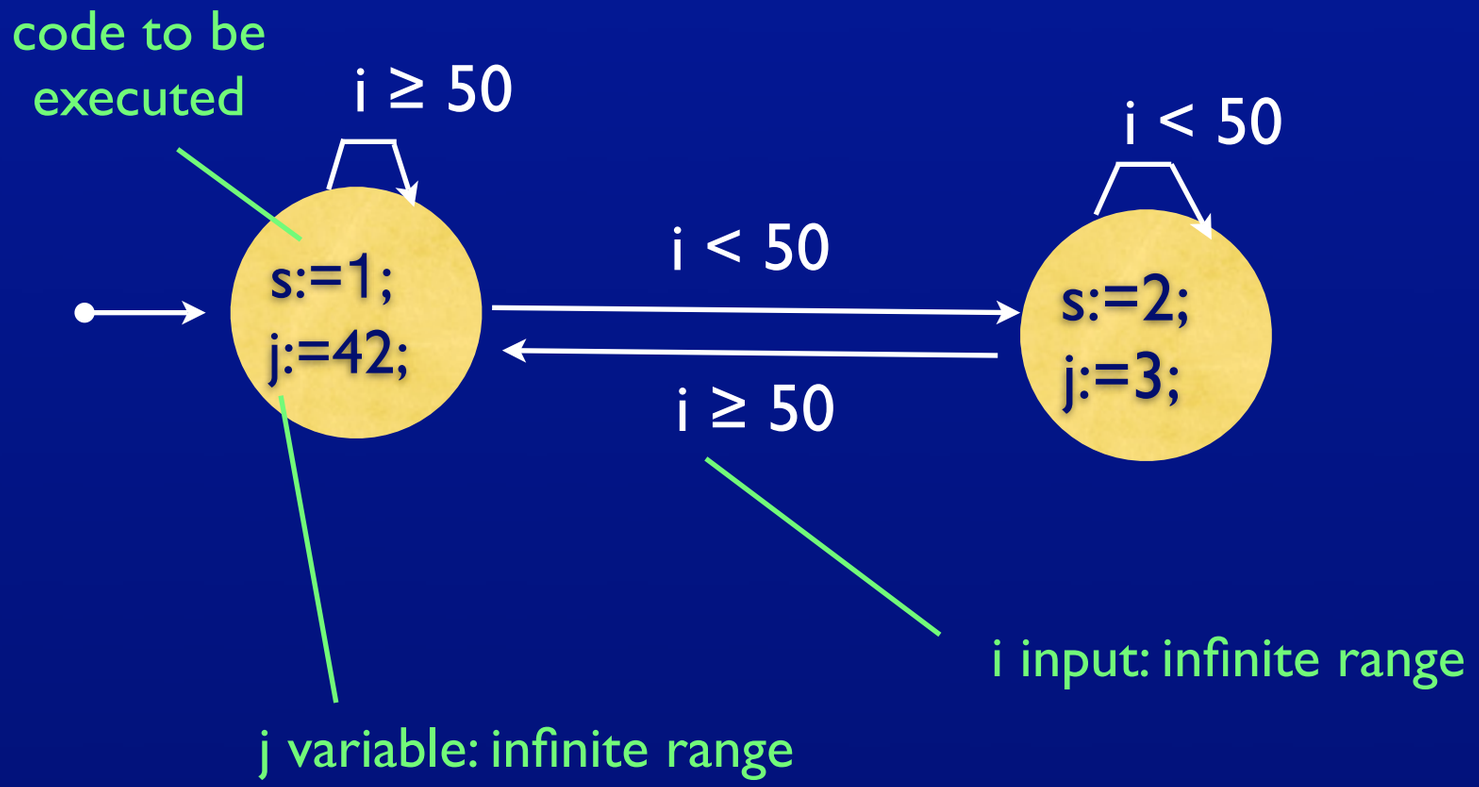
Original System



property: AG (i ≥ 50 ⇒ s = 1)

Example

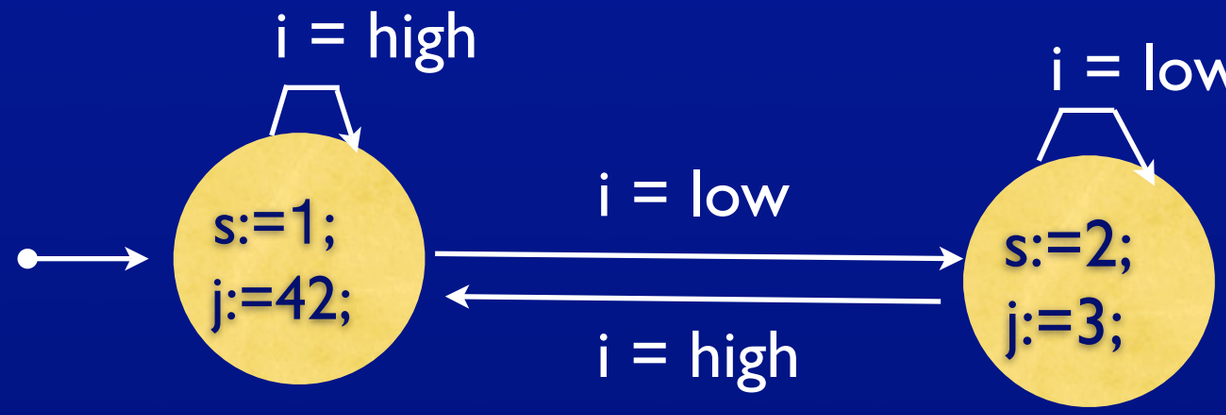
Original System



property: AG (i ≥ 50 ⇒ s = 1)

Example

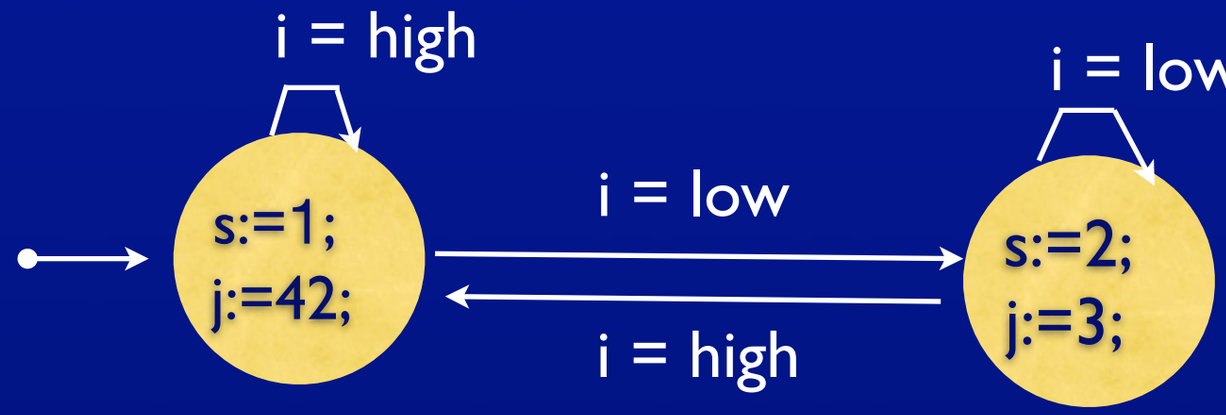
Input Domain Abstraction



property: AG (i = high ⇒ s = 1)

Example

Input Domain Abstraction

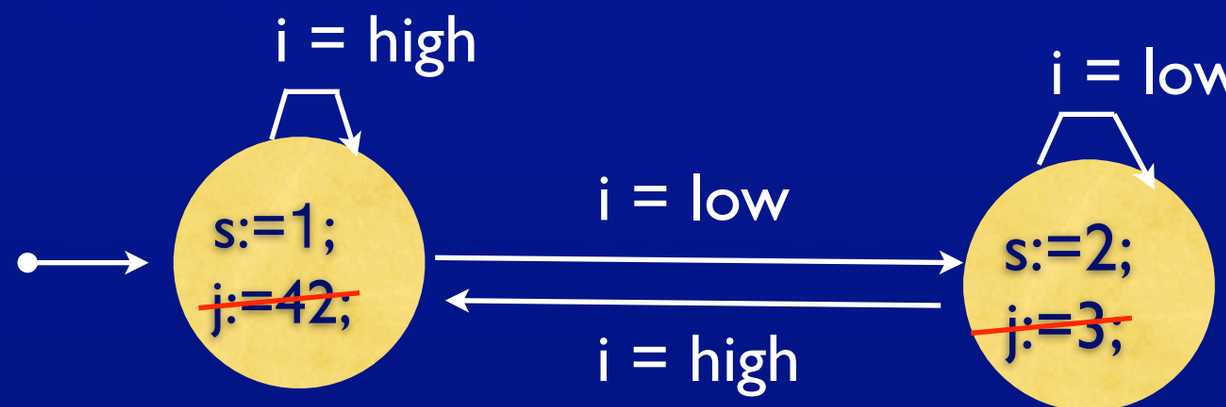


i = low : i < 50
i = high : i ≥ 50

property: AG (i = high ⇒ s = 1)

Example

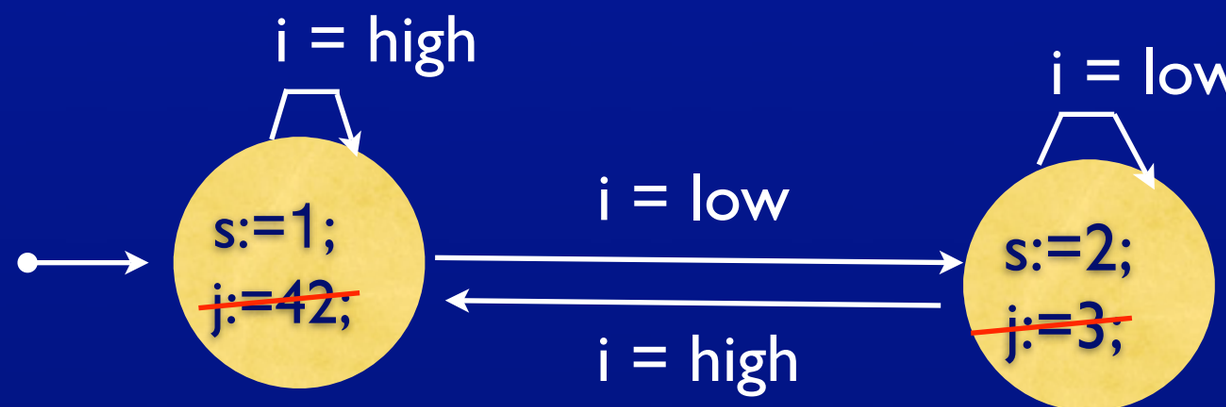
Omitting Variables



property: AG (i = high ⇒ s = 1)

Example

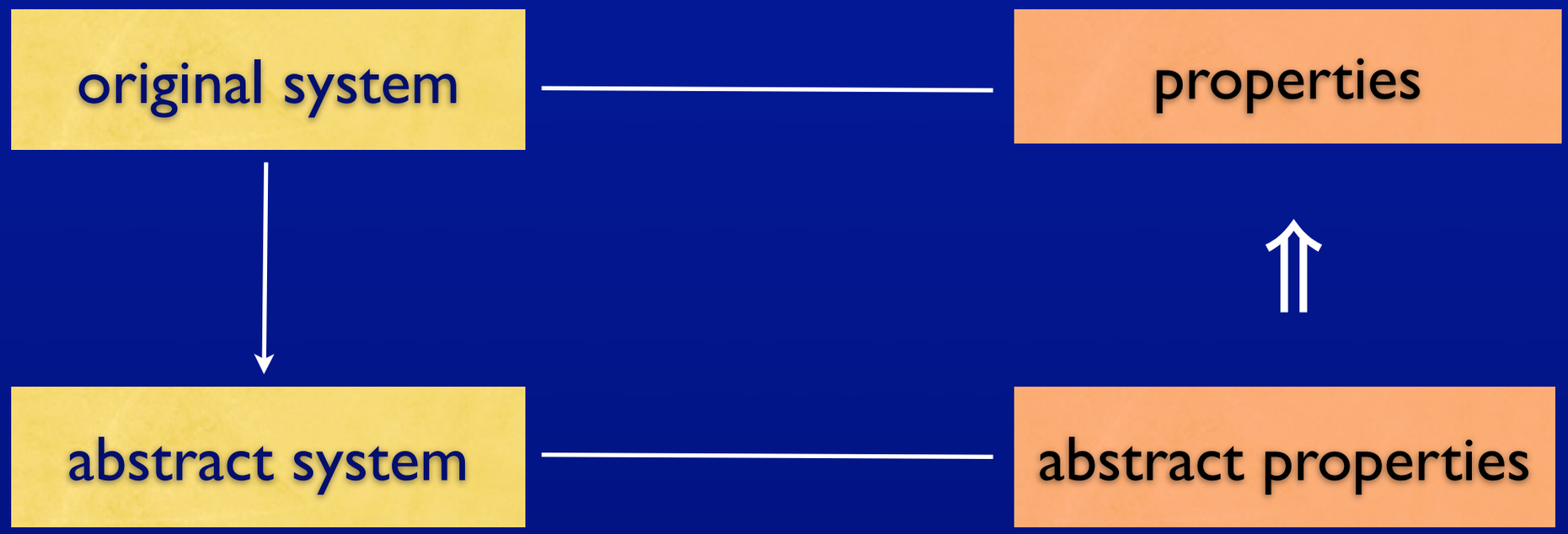
Omitting Variables



i = low : i < 50
i = high : i ≥ 50

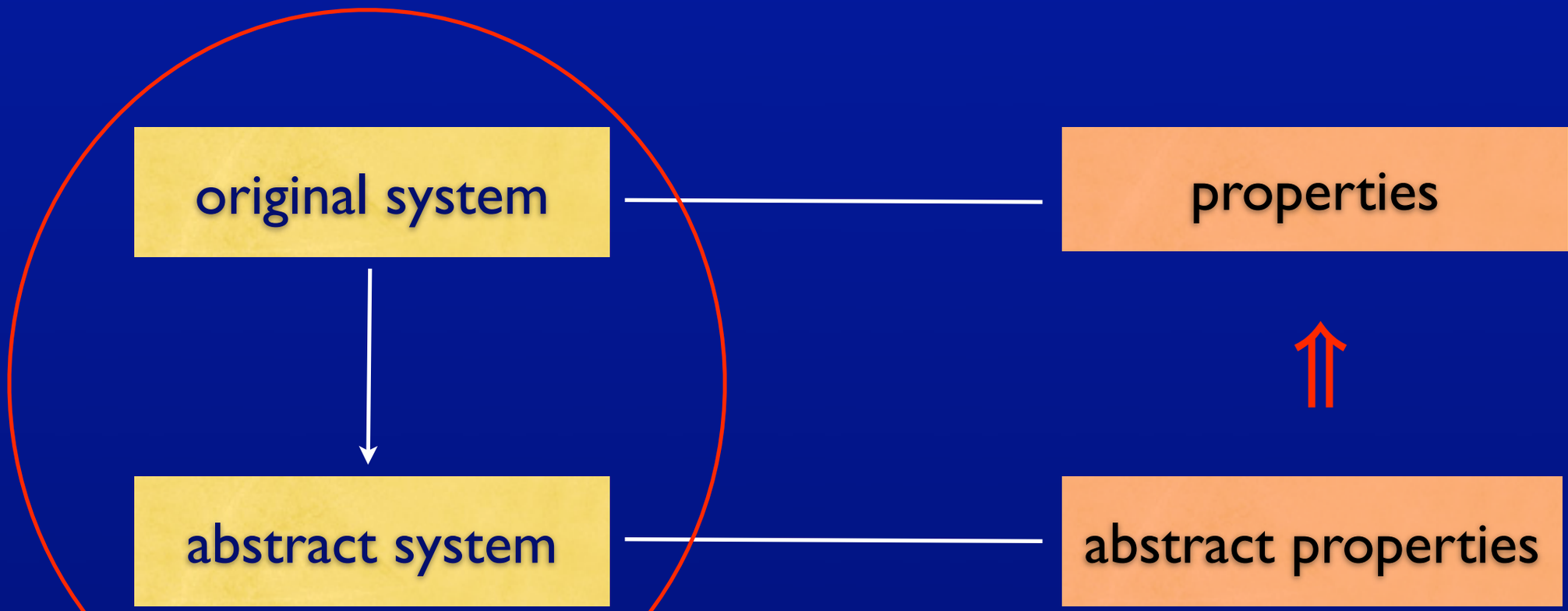
property: AG (i = high ⇒ s = 1)

Proving Abstractions Correct



- every property of the abstract system shall hold in the concrete as well
- properties are formalized in either CTL* or their fragments

Proving Abstractions Correct



simulation

⇒ preservation of ACTL* properties

[Clarke, Grumberg, Long 1994] [Loiseaux, Graf, Sifakis, Bouajjani, Bensalem 1995]

Translation Validating Abstractions

Proving the Simulation

- establish simulation relation

$R(s_0, s_0')$

$R(s,s') \Rightarrow R(next(s), next'(s'))$

Translation Validating Abstractions

Proving the Simulation

- establish simulation relation

$R(s_0, s_0')$ initial states

$R(s,s') \Rightarrow R(next(s), next'(s'))$

state transition functions

Translation Validating Abstractions

Proving the Simulation

- establish simulation relation

$R(s_0, s_0')$ — initial states

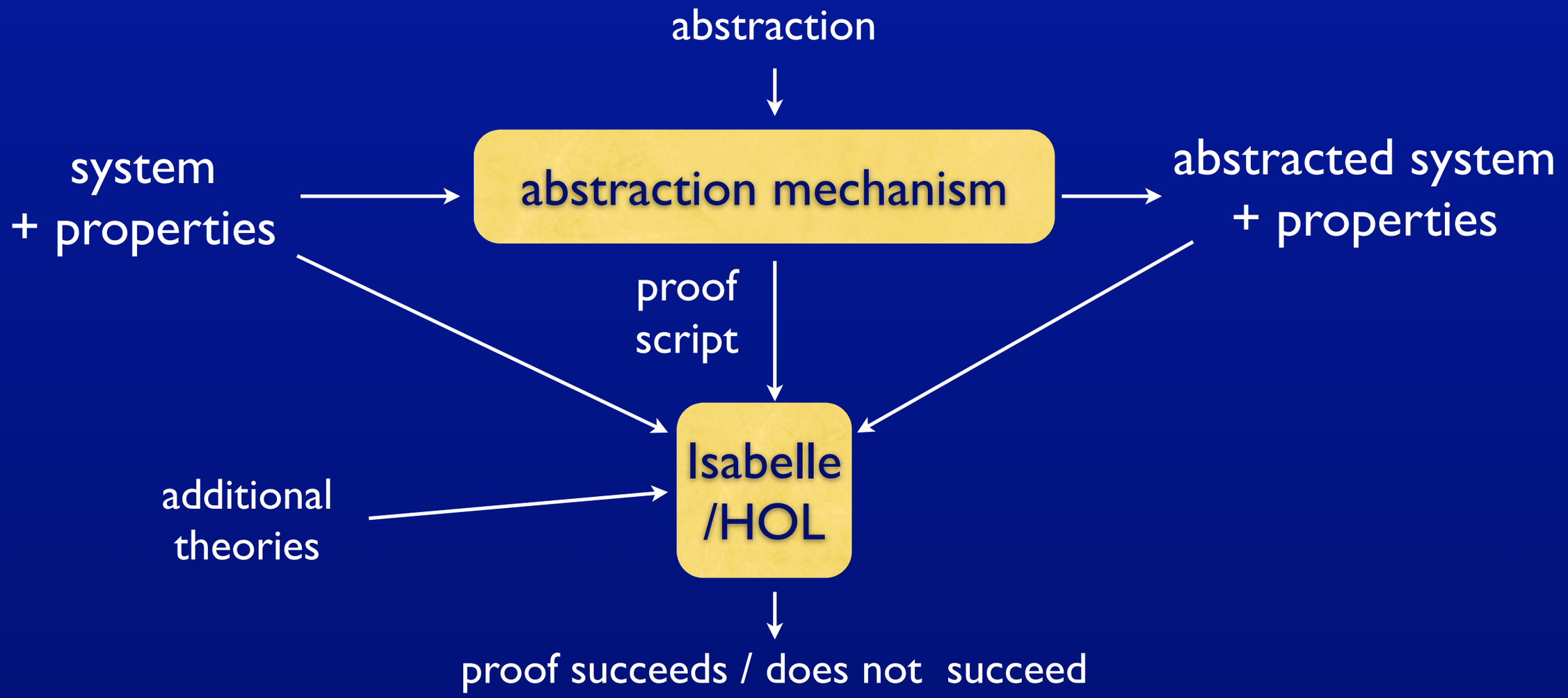
R ensures property preservation
e.g. i < 50 iff i = low

$$R(s,s') \Rightarrow R(next(s), next'(s'))$$

state transition functions

Translation Validating Abstractions

Our Setting



Translation Validating Abstractions

Formalizing the Description Language

- operational semantics of systems
- deep vs. shallow embedding
- state transition function as Isabelle functions
- states as Isabelle records/tuples

Evaluation

Current Status of Implementation

- system design with CASE tool
- implementation of Isabelle code and proof generator in Java
- Isabelle/HOL as validator
- connection to Averest model checking framework

Evaluation

Case Studies from Adaptive Embedded Systems

- sensor and light control scenarios
  - domain abstractions
  - dead-assignment abstractions
  - examples having up to 39 variables with infinite domains
  - model checking of sensor example only possible with abstractions

[Schaefer, Poetzsch-Heffter 2006] for description of sensor example

Conclusion & Future Work

- successful port of translation validation to abstraction of systems
- formal verification of abstractions using a theorem prover

- automatical verification of more abstractions

Thank you for your attention

Questions?