Most General Property-Preserving Updates

Davide Bresolin – University of Padova, Italy
Ivan Lanese – Focus Team, University of Bologna/INRIA, Italy

DB

LATA 2017, 9 March 2017

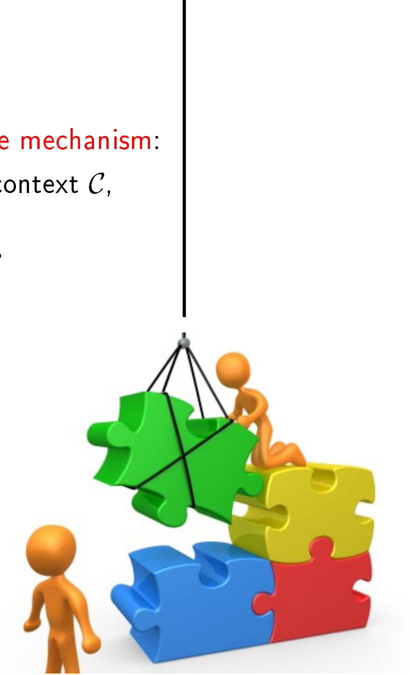
Motivation

- Modern applications are increasingly designed as distributed systems that cooperates to achieve a common goal.
- They must face dynamically changing requirements and unpredictable behaviour by the environment
- Applications and systems need to be updated, statically or dynamically:
  - changing business rules
  - changing environment conditions
  - bug fixes
  - specialize the application to user preferences
  - ...

Which Updates

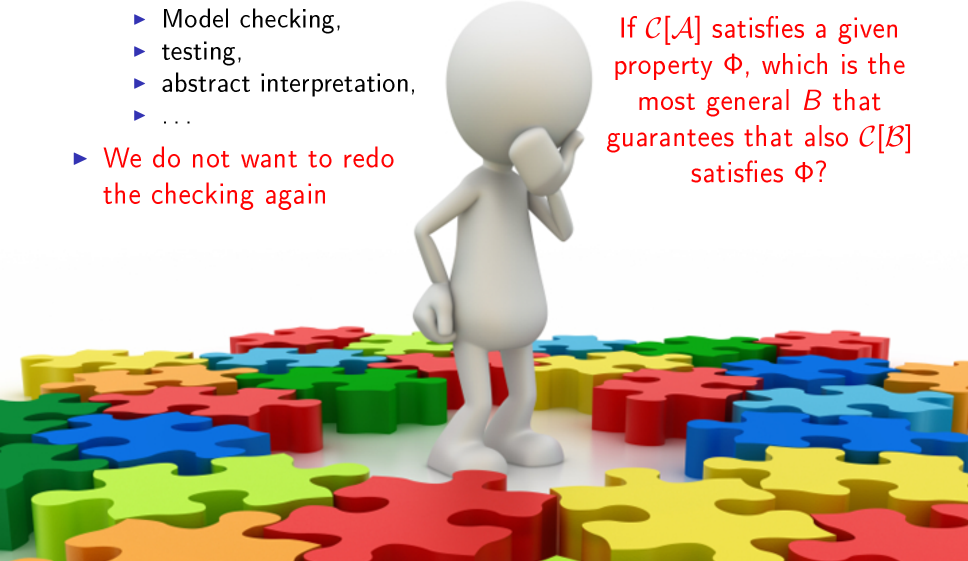
We consider a simple but general update mechanism:

- The application is composed by a context $\mathcal{C}$, and a component to be updated $\mathcal{A}$
- The component $\mathcal{A}$ is replaced by $\mathcal{B}$
- We go from $\mathcal{C}[\mathcal{A}]$ to $\mathcal{C}[\mathcal{B}]$



Property preservation

- Many approaches:
  - Model checking,
  - testing,
  - abstract interpretation,
  - ...
- We do not want to redo the checking again



If $\mathcal{C}[\mathcal{A}]$ satisfies a given property Φ, which is the most general B that guarantees that also $\mathcal{C}[\mathcal{B}]$ satisfies Φ?


One property
one context
All properties
one context
One property
all contexts
All properties
all contexts

Constraint Automata

We model contexts and components as constraint automata

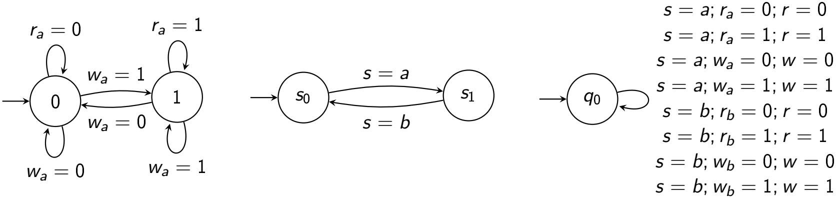
$$\mathcal{A} = \langle Q, N, q_0, \rightarrow \rangle$$

- Automata that communicates via (internal and external) nodes N
- Transitions are labelled with functions from N to $data \cup \{\perp\}$
- We consider embeddings:
  - The component communicates only with the context
- We consider both synchronous and asynchronous composition

Two register example

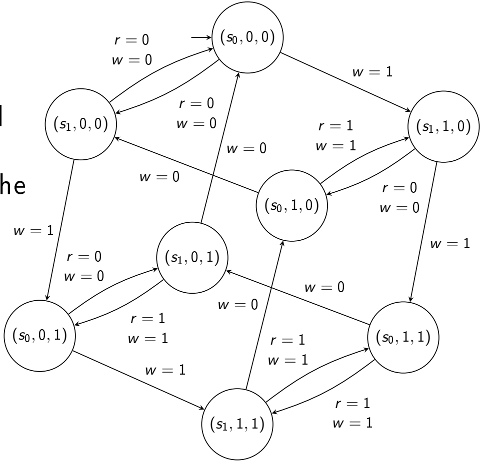
Consider a system composed by two 1 bit registers, A and B

- Registers can be read and written
- A scheduler decides which register can be accessed from outside
- A synchronizer that exposes to the outside world the nodes r (read) and w (write) to access the active register



The complete system

- We consider the Scheduler as the component to be updated
- The contex is then given by the Registers and Synchronizer
- The complete system is obtained by asynchronous composition



Properties

We adopt a general definition of **property**. . .

- for us, a property is a prefix-closed sets of traces . . .
- . . . that can be accepted by some Constraint Automaton
- we represent a property by the automaton Φ that accepts it
- $\mathcal{A}$ satisfies the property Φ iff $\mathscr{L}(\mathcal{A}) \subseteq \mathscr{L}(\Phi)$

. . . to exploit the link between **automata and logic:**

- Constraint Automata are as expressive as the safety linear μ-calculus
- our approach can cope with any safety property written in some temporal logic


One property
one context
All properties
one context
One property
all contexts
All properties
all contexts

One property, one context

Problem

Given a context $\mathcal{C}$, a component $\mathcal{A}$ and a property Φ, find a most general $\mathcal{B}$ such that if $\mathcal{C}[\mathcal{A}]$ satisfies Φ, then also $\mathcal{C}[\mathcal{B}]$ satisfies Φ.

- This is equivalent to finding the largest $\mathcal{B}$ (w.r.t. language containment) that solve the equation

$$\mathscr{L}(\mathcal{C}[\mathcal{B}]) \subseteq \mathscr{L}(\Phi)$$

- We know from automata theory that

$$\mathscr{L}(\mathcal{B}) = \overline{\mathscr{L}(\mathcal{C}[\overline{\Phi}])}$$

One property, one context

We cannot directly apply the equation:

- The solution may not be prefix-closed
  - we restrict to prefix-closed solutions
- Constraint Automata are not closed under complementation
  - we need to add final states and an acceptance condition
- We are dealing with infinite traces, where complementation is expensive
  - can we avoid Safra's construction?
  - what about determinization and subset construction?

A (somehow) simpler way to the solution

1. Φ can be easily complemented:
   - all states of Φ are final
   - complete the automaton and add a unique, non-final, sink state
   - determinize with the subset construction
   - complement by switching final and non-final states

$$\overline{\Phi} = \text{Switch}(\text{Subset}(\Phi))$$

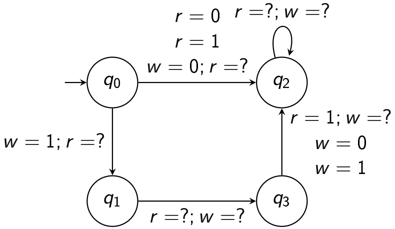
2. Then the maximal prefix closed $\mathcal{B}$ is

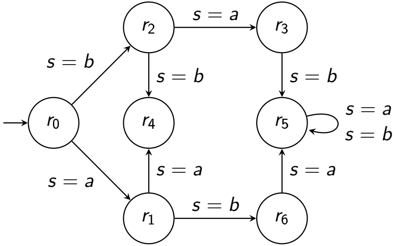
$$\mathcal{B} = \text{Prefix}(\text{Switch}(\text{Subset}(\mathcal{C}[\overline{\Phi}])))$$

   - this is a solution of the equation, but non necessarily the most general one
   - every other prefix-closed solution is contained in $\mathcal{B}$

One property, one context – example

P1: "if *w=1* is executed at the first step, then at the third step *r=0* cannot be executed"





- The solution accepts traces that ends before the third step, or longer traces that starts with either $s = a, s = b, s = a$ or $s = b, s = a, s = b$


One property
one context
All properties
one context
One property
all contexts
All properties
all contexts

All properties, one context

Problem

Given a context $\mathcal{C}$ and component $\mathcal{A}$, find a most general $\mathcal{B}$ such that for every property Φ, if $\mathcal{C}[\mathcal{A}]$ satisfies Φ, then also $\mathcal{C}[\mathcal{B}]$ satisfies Φ.

- This is equivalent to finding the largest $\mathcal{B}$ (w.r.t. language containment) that solve the equation

$$\mathscr{L}(\mathcal{C}[\mathcal{B}]) \subseteq \mathscr{L}(\mathcal{C}[\mathcal{A}])$$

- We know from automata theory that

$$\mathscr{L}(\mathcal{B}) = \overline{\mathscr{L}(\mathcal{C}[\overline{\mathcal{C}[\mathcal{A}]}])}$$

All properties, one context

Problem

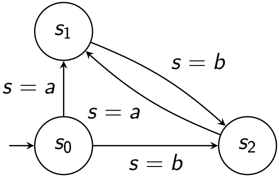
Given a context $\mathcal{C}$ and component $\mathcal{A}$, find a most general $\mathcal{B}$ such that for every property Φ, if $\mathcal{C}[\mathcal{A}]$ satisfies Φ, then also $\mathcal{C}[\mathcal{B}]$ satisfies Φ.

- By the same argument of the "one property, one context" case we can show that the maximal prefix closed $\mathcal{B}$ is

$$\mathcal{B} = \mathrm{Prefix}(\mathrm{Switch}(\mathrm{Subset}(\mathcal{C}[\overline{\mathcal{C}[\mathcal{A}]}])))$$

All properties, one context – example

We can apply the construction to obtain the most general scheduler that replaces the original one.



- The solution accepts only traces of the form
  $s = a, s = b, s = a, s = b, \dots$ or
  $s = b, s = a, s = b, s = a, \dots$

Complexity

- The same construction can be used for both the "one property, one context" and "all properties, one context" cases
- The problem is in 2-EXPTIME, since it requires a double complementation
  - even though the final result may be much smaller in practical cases
- Solving the "one property, one context" case is EXPSPACE-hard
  - Proved by reducing a suitable three-player game to it
  - The component and the property play against the context
  - Hardness of the "all properties, one context" case is open


One property
one context
All properties
one context
One property
all contexts
All properties
all contexts

One property, all contexts

Problem

Given a component $\mathcal{A}$ and a property Φ, find a most general $\mathcal{B}$ such that for all contexts $\mathcal{C}$, if $\mathcal{C}[\mathcal{A}]$ satisfied Φ then also $\mathcal{C}[\mathcal{B}]$ satisfies Φ.

- For asynchronous embedding, unless the formula is true or false, we need:

  $$\mathscr{L}(\mathcal{B}) \subseteq \mathscr{L}(\mathcal{A})$$

  and thus we have that $\mathcal{A}$ is the most general solution we are looking for

One property, all contexts

Problem

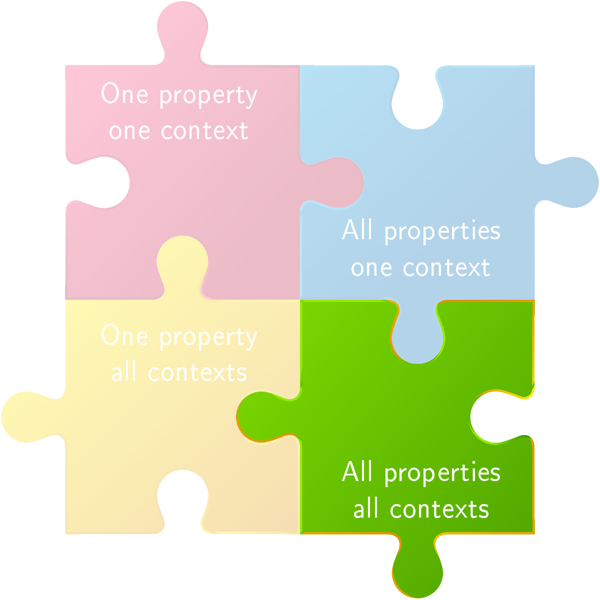
Given a component $\mathcal{A}$ and a property Φ, find a most general $\mathcal{B}$ such that for all contexts $\mathcal{C}$, if $\mathcal{C}[\mathcal{A}]$ satisfied Φ then also $\mathcal{C}[\mathcal{B}]$ satisfies Φ.

- For synchronous embedding we need:

$$\mathscr{L}(\mathcal{B}) \subseteq \mathscr{L}(\mathcal{A}) \cup \overline{\mathcal{R}(\Phi)}$$

  where $\mathcal{R}(\Phi)$ is the observation-point language of Φ that contains all the words of lengths n such that there exists zc of length n such that z is in Φ and zc is not in Φ.

- Once again, the most general solution $\mathrm{MGU}(\mathcal{A}, \Phi)$ can be built by a double complementation


One property
one context
All properties
one context
One property
all contexts
All properties
all contexts

All properties, all contexts

Problem
Given a component $\mathcal{A}$, find a most general $\mathcal{B}$ such that for all contexts $\mathcal{C}$ and for all properties Φ, if $\mathcal{C}[\mathcal{A}]$ satisfied Φ then also $\mathcal{C}[\mathcal{B}]$ satisfies Φ.

- Given two distinct $\mathcal{A}$ and $\mathcal{B}$, it is always possible to find a context and a property that distinguish them
- Hence, we need:

$$\mathscr{L}(\mathcal{B}) \subseteq \mathscr{L}(\mathcal{A})$$

  and thus we have that $\mathcal{A}$ is the most general solution we are looking for


One property one context
All properties one context
One property all contexts
All properties all contexts

Conclusions

Results:

- We studied under which conditions updates preserve properties
- We have show how to build a most general update
- We generalized to all properties and/or all contexts

Future Work:

- Consider the same problem in different settings
  - More expressive automata automata
  - More expressive properties
  - More efficient constructions
- What happens when multiple updates are considered?