**COMP424/524-06A**  
**Topics in Software Engineering**  
Part I – Model Checking Algorithms  
18. Incremental Verification  
Robi Malik

---

**Subsets and Intersection**

Strict synchronous composition is intersection of languages:
\[ \mathcal{L}(A_1 \parallel A_2 \parallel \cdots \parallel A_n) = \mathcal{L}(A_1) \cap \mathcal{L}(A_2) \cap \cdots \cap \mathcal{L}(A_n) \]

Language inclusion check tests for subset of languages:
\[ \mathcal{L}(A_1 \parallel A_2 \parallel \cdots \parallel A_n) \subseteq \mathcal{L}(P) \]

---

**Modelling Automata in VALID**

- Each automaton imposes new constraints.
- Behaviour is restricted by adding automata.

---

**Result about Language Inclusion**

**Proposition.**
Let \( A = A_1 \parallel A_2 \parallel \cdots \parallel A_n \).
If \( A_i \) satisfies a property \( P \),
then \( A \) satisfies \( P \).

**Proof.**
\[ \mathcal{L}(A) = \mathcal{L}(A_1) \cap \cdots \cap \mathcal{L}(A_n) \subseteq \mathcal{L}(A_i) \subseteq \mathcal{L}(P) \]

---

**Language Inclusion Check**

Language inclusion is a safety property.

- \( \text{start2} \) should not occur here.
- \( \text{start1} \) should not occur here.

---

**Applying the Result**

**Does small factory satisfy AltStart?**

If one of the automata, e.g., Buffer, satisfies AltStart, then we are done.
Checking a Subsystem
Does Buffer satisfy AltStart?

Buffer does not satisfy AltStart!
Counterexample: start1 start1.

Property Not Satisfied
Language inclusion check stops here.

Counterexample start1 break1 repair1 start1 accepted by all components.

What Now?
Use the counterexample ...

What About Controllability?

Checking an Enlarged Subsystem
Does Buffer || Machine1 satisfy AltStart?

Buffer || Machine1 does not satisfy AltStart!
New counterexample: start1 break1 repair1 start1.

Incremental Language Inclusion

What About Controllability?
Proposition.
Let $P = P_1 || P_2 || \ldots || P_n$ be a plant, and let $S = S_1 || S_2 || \ldots || S_m$ be a specification. If each $S_i$ is controllable with respect to a subsystem of the plant, e.g., $P_i' = P_{i1} || \ldots || P_{ik}$, then $S$ is controllable with respect to $P$. 
Example

```
Machine1       Machine2
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>WOR</td>
<td>WOR</td>
</tr>
<tr>
<td>WOR</td>
<td>COM</td>
</tr>
<tr>
<td>COM</td>
<td></td>
</tr>
</tbody>
</table>
```  

Plant  Specification

**Controllable?**

Incremental Verification

**An automatic abstraction procedure**

- Abstractions, i.e. subsystems, computed automatically.
- Counterexamples used to augment subsystems until the property is proven or shown to be not satisfied.

Small Factory is Controllable

```
Buffer       Repair
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>WOR</td>
<td>WOR</td>
</tr>
<tr>
<td>WOR</td>
<td>COM</td>
</tr>
<tr>
<td>COM</td>
<td></td>
</tr>
</tbody>
</table>
```  

is controllable with respect to and ...

```
Machine1      Machine2
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>WOR</td>
<td>WOR</td>
</tr>
<tr>
<td>COM</td>
<td></td>
</tr>
</tbody>
</table>
```  

is controllable with respect to

Therefore, the entire system is controllable.

A Problem

There may be several components not accepting a counterexample.

**Which ones to choose?**

- All of them?
- First one found?
- Most promising one?

▶ Heuristics!

The End of Part I