Supervisory Control Theory;

A classical AI approach to build reliable systems

Ashfaq Farooqui
ashfaq@ashfaqfarooqui.me

Department of Electrical Engineering,
Chalmers University of Technology,
Göteborg, Sweden

Outline

1. Introduction
2. Proofs, Verification and Synthesis
3. Supervisory Control
4. Building Reliable Systems
5. Applications in the real world
6. Problems
7. Conclusion

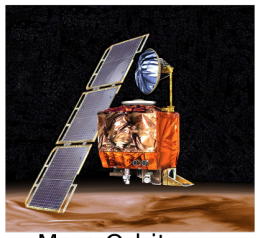
Topic

1. Introduction
2. Proofs, Verification and Synthesis
3. Supervisory Control
4. Building Reliable Systems
5. Applications in the real world
6. Problems
7. Conclusion

Past mistakes


Boing 737


Pacemakers


Mars Orbiter

Pentium 2


ARIANE-5

- Boeing 747 software bugs[1]
- ARAINE 5 guidance error[2]
- Mars orbiter metric system error[3]
- Intel pentium fdiv bug[4]
- Lack of specifications for pacemakers

[1] https://www.nytimes.com/interactive/2019/business/boeing-737-crashes.html

[2] https://www.bugsnag.com/blog/bug-day-ariane-5-disaster

[3] https://www.bugsnag.com/blog/bug-day-mars-climate-orbiter

[4] https://en.wikipedia.org/wiki/Pentium_FDIV_bug

Model-based design

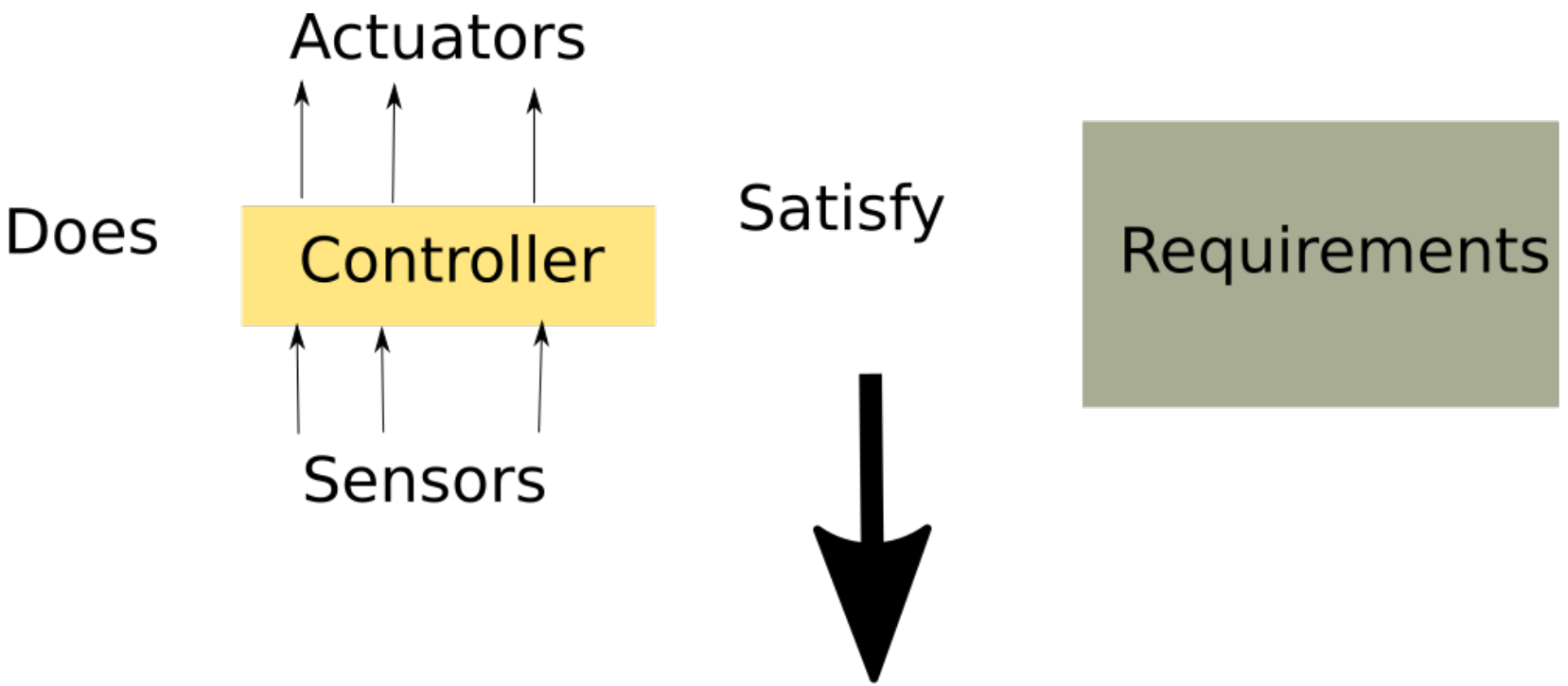
Model

Description of the behavior of the given system defined mathematically.

Specification

Mathematical description of the requirements pertaining to the model of the system.

Model-based design



Topic

1. Introduction
2. Proofs, Verification and Synthesis
3. Supervisory Control
4. Building Reliable Systems
5. Applications in the real world
6. Problems
7. Conclusion

Deductive Proofs [5]

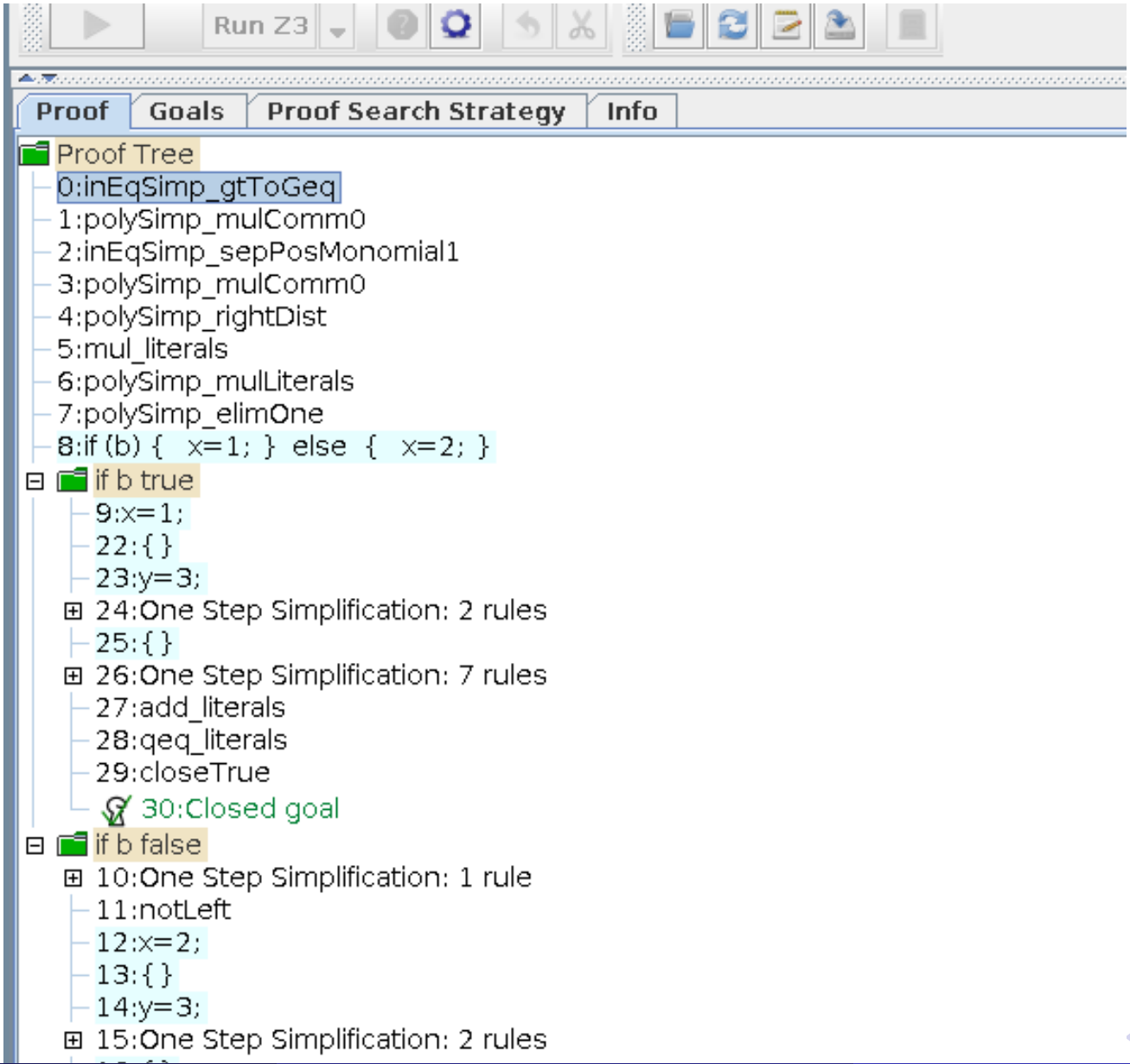
Given a program and a contract.
Goal: Does the program fulfill the contract?

```
\programVariables {
 int x;
 int y;
 boolean b;
}
```

```
\problem {
    \<{ if (b) { x = 1; } else { x = 2; }  y = 3; }\> y > x
}
```

[5] https://www.key-project.org/

Deductive Proofs



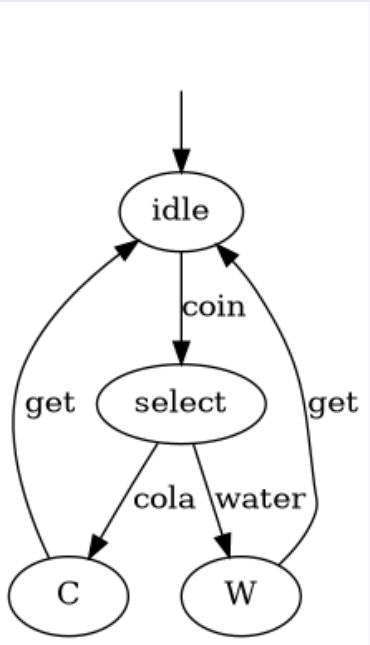
Deductive Proofs

- Contract based systems
- Example Etherum
- Develop and enforce legal contracts based on blockchain.

Verification

Given a model of a system and specifications of the system.
Goal check if all specifications are satisfied by the model.

Model

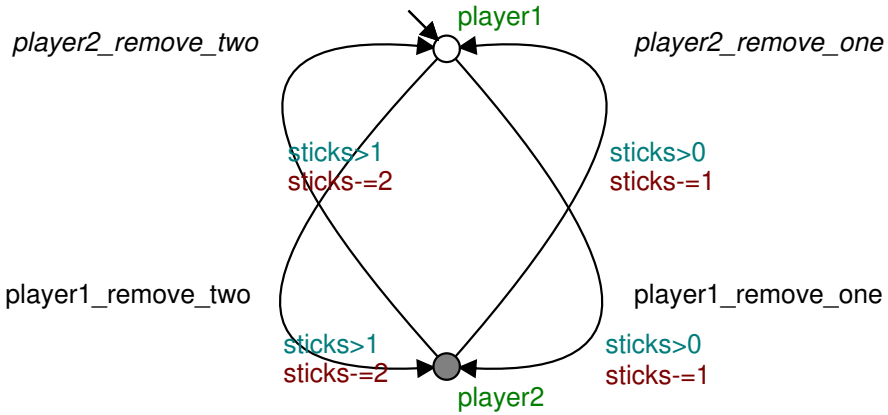


Specification

"The vending machine only delivers a drink after providing a coin"

Synthesis

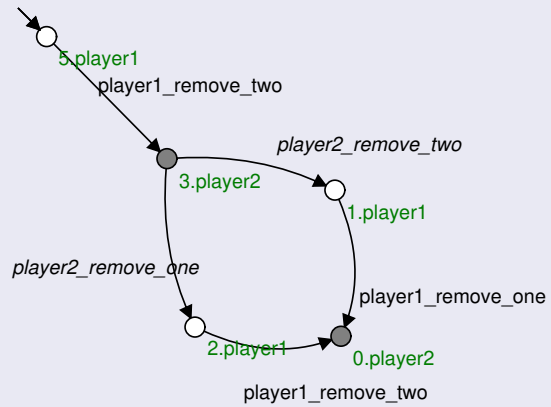
Given a model of a system and specifications of the system.
Goal: Generate a supervisor that restricts the behavior of the system to satisfy the all the specifications.



Specification

"Player one wins"

Supervisor



Topic

1. Introduction
2. Proofs, Verification and Synthesis
3. Supervisory Control
4. Building Reliable Systems
5. Applications in the real world
6. Problems
7. Conclusion

Introduction

- Developed by Peter Ramadge and Walter Wonham in the late 80's [6]
- Control of DES
- Has been confined within academia for 30 years now.

Automaton

$G=<Q,\Sigma,\delta,q_0>$

- Q is the set of *states*
- Σ is the *alphabet* containing the events, $\Sigma = \Sigma_c \cup \Sigma_u$
- δ:$Q \times \Sigma \rightarrow Q$ is the partial *transition function*
- $q_0 \in Q$ is the *initial state* of the system

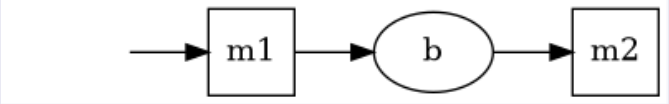
Language

The language $\mathcal{L}(G)$ is the set of all valid combinations of the alphabet in the automaton

[6] Ramadge, Peter J.; Wonham, Walter M. (January 1987). "Supervisory Control of a Class of Discrete Event Processes". SIAM Journal on Control and Optimization.

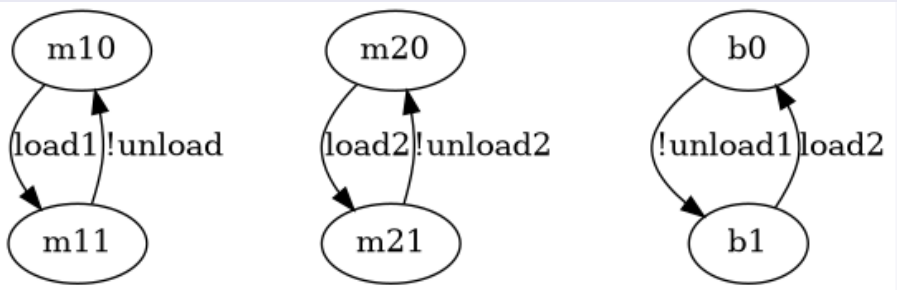
Machine Buffer Machine

Product flow



System Behavior

$\Sigma = \{load1, load2, !unload1, !unload2\}$



$\mathcal{L}(m1) = \{\epsilon, load1, load1.!unload1, load1.!unload1.load2, ...\}$

Properties

Controllability

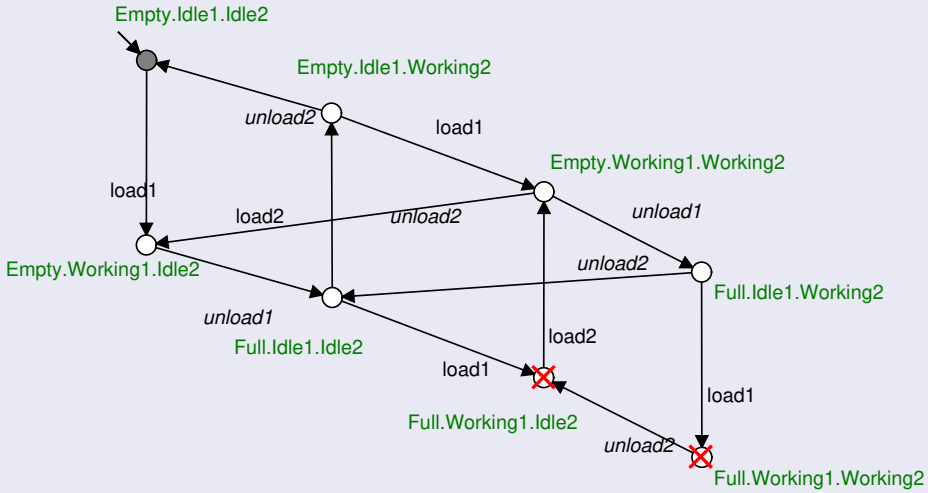
- Only some parts of the system can be controlled.
- Given a system, its specification, and the supervisor: the supervisor should always be able to control the desired outcome of the system to ensure specifications are met.

Non-blocking

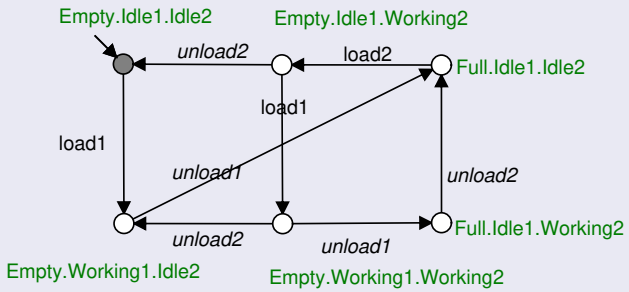
- At no point is the system stuck in one place with no possibility to reach its desired state.

Machine Buffer Machine

Synchronization



Supervisor



Topic

1. Introduction
2. Proofs, Verification and Synthesis
3. Supervisory Control
4. Building Reliable Systems
5. Applications in the real world
6. Problems
7. Conclusion

A general approach towards developing reliable systems:

1. Pick one or more techniques that are relevant for your application.
2. Develop the model of the intended behavior of the system.
3. Translate the requirements of the project into specifications.
4.
   - Verify and update the model till all specifications are satisfied.
   - Generate a supervisor for your system.
5. Tune and refine points 2,3 till the results are satisfactory.
6. Either implement the controller according the model or use automatic code generation if available.

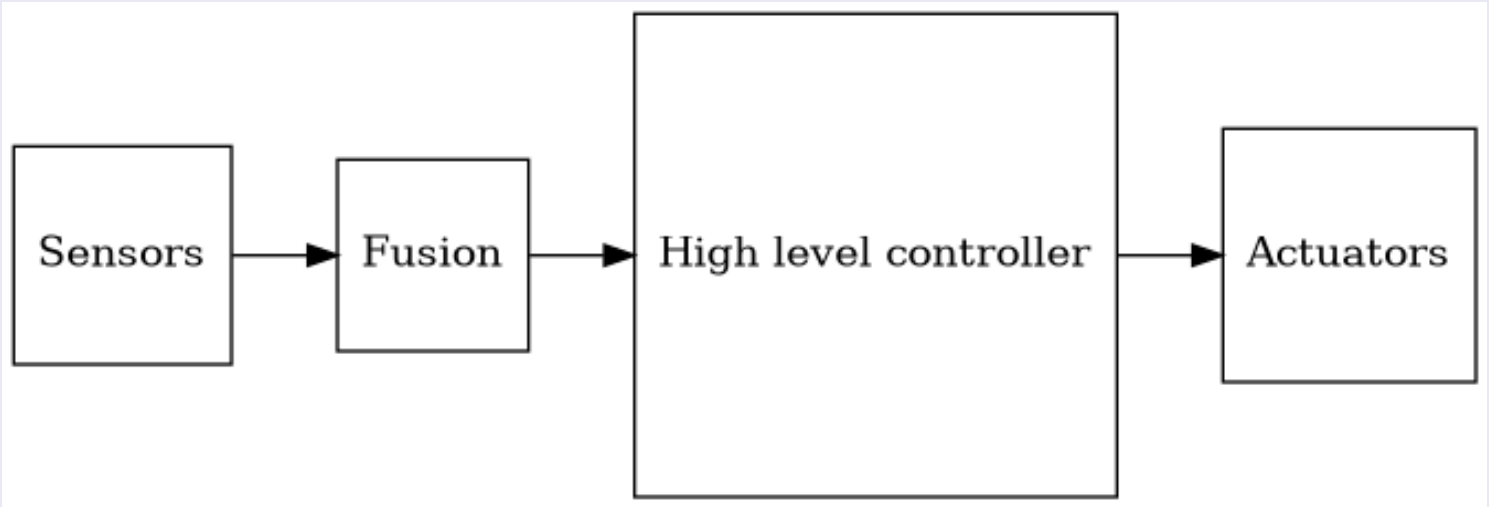
Topic

1. Introduction
2. Proofs, Verification and Synthesis
3. Supervisory Control
4. Building Reliable Systems
5. Applications in the real world
6. Problems
7. Conclusion

Verification of lane change algorithm[7]

- Keeps track of state of execution
- Performs indication and lane change
- Cyclic execution

System overview



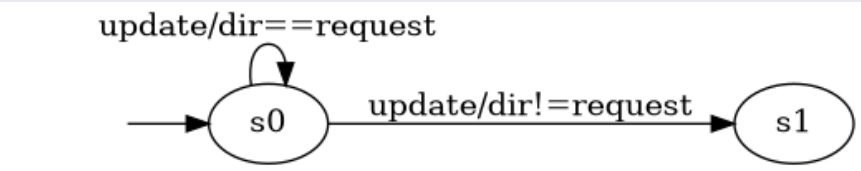
[7] A. Zita, S. Mohajerani and M. Fabian, "Application of formal verification to the lane change module of an autonomous vehicle," 2017 13th IEEE Conference on Automation Science and Engineering (CASE), Xi'an, 2017

Verification of lane change algorithm

Modeling

- Modeled the system according to the code
- 1.4 x 10^5 states

Specification



Verification

- ~3Million states
- ~1Million were unreachable states
- An issue was found where the system does not behave as specified.
- The specification defines that the requested direction to turn, must be the direction the vehicle turns to.

Topic

1. Introduction
2. Proofs, Verification and Synthesis
3. Supervisory Control
4. Building Reliable Systems
5. Applications in the real world
6. Problems
7. Conclusion

State space explosion

- Number of states grows exponentially.
- The problem is known to be NP-Hard

How to (and who) create(s) models?

- Defining models is an arduous task.
- Models are abstractions of reality.

Where do specifications come from?

- Unclear on how to create all specifications
- Possibility of human errors in translating Natural language to formal spec

Topic

1. Introduction
2. Proofs, Verification and Synthesis
3. Supervisory Control
4. Building Reliable Systems
5. Applications in the real world
6. Problems
7. Conclusion

- Automata are one way of modeling DES.
- They allow us to leverage on mathematical techniques to guarantee reliability of systems.
- Can be used to verify systems or generate a supervisor/controllers that satisfy given requirements.
- Suffer from state-space explosion problems.
- Lack of models and obtaining accurate models is a challenge.

Agv system

- 4 AGV's that transport material
- 4 interacting zones