Formal Methods vs Machine Learning

Feb 2020

Luigia Petre,

Åbo Akademi University, Turku, Finland

Some background

- MSc in Computer Science, University of Bucharest 1997
- PhD in Computer Science, Åbo Akademi University, 2005
- Senior lecturer at Åbo Akademi University
  - Adjunct professor
- Research in formal methods
  - Modeling based on math to analyze software systems
- Teaching in data science, since 2018
- CA17137 – gravitational waves with machine learning
  - Awesome physics (hard!)

Formal methods

- What is a formal method?
  - Set of techniques for analysing software-based systems
  - Has
    - Language with semantics
    - Methods of formulating + evaluating properties
    - Method/s of comparing different versions of the same system
- Examples
  - Z, B, Event-B, ASM, Alloy, VDM, TLA
  - CSP, CCS, pi-calculus, Ambient Calculus

Event-B: state-based formal method

- States and transitions
  - States
    - Variables
    - Constants
  - Transitions (Events)
    - Guards: necessary conditions
    - Actions: some variables change
- Certain abstraction level

Semantics

Kind	Assignment	Before-after Predicate
deterministic	$x := E(t, v)$	$x' = E(t, v) \ \land \ y' = y$
empty	skip	$v' = v$
non-deterministic	$x :\mid P(t, v, x')$	$P(t, v, x') \ \land \ y' = y$

- denote the relationship holding between the state variables of the machine just before (denoted by *v*) and after (denoted by *v'*) "applying" an assignment
- if *x* are variables of the machine, then *x'* are their values just after applying an assignment
- *y* denotes the set of variables drawn from *v* which are distinct from those in *x*

Formal Reasoning

- **Invariant**
  - Condition on the state variables that **must hold permanently**
  - We *prove* that, under the invariant in question and under the guard of each event, the invariant still holds after the variables have been modified according to the transition associated with that event
- **Reachability**
  - Condition that **does not hold permanently**
  - We *prove* that, an event whose guard is not necessarily true now will nevertheless certainly occur within a finite number of iterations

Invariant preservation

axioms invariant guard Before-after predicate

$$\boxed{P(s,c) \ \wedge \ I(s,c,v) \ \wedge \ G(s,c,v) \ \wedge \ R(s,c,v,v') \ \Rightarrow \ I(s,c,v') \qquad \text{INV}}$$

invariant

Where:
s sets, *c* constants, *v* variables, *v'* variables after action takes place

∀-quantified over all carrier sets, constants, and variables occurring free in the proof obligation

Refinement as spatial extension

- Reality is the same
- Our view of the refined reality is more accurate
- Previously invisible details of the reality are now revealed
- More powerful microscope reveals even more details

A refined model is spatially larger than its previous abstractions

Spatial extension has corresponding **temporal extension**

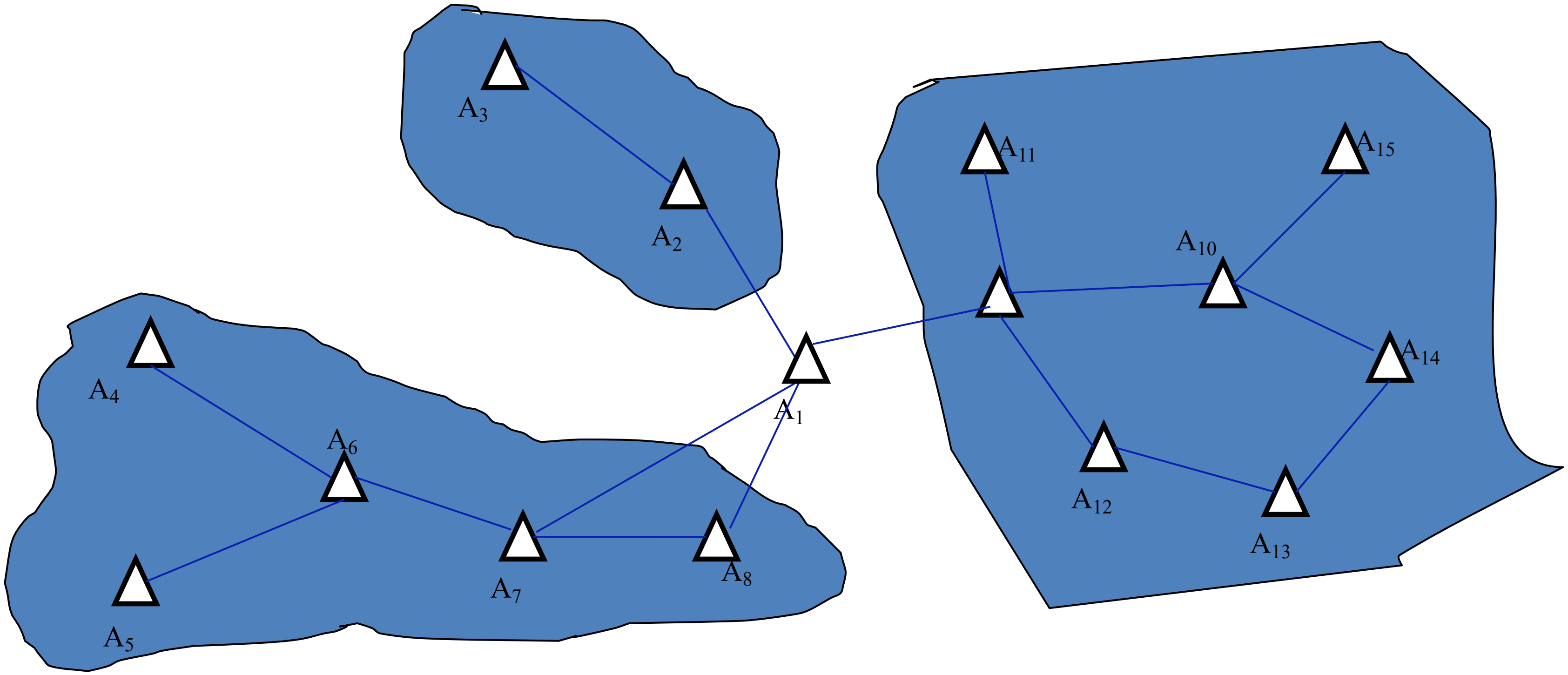
- The new variables can be modified by new transitions
  - Could not have been present in previous abstractions: the concerned variables did not exist in them
  - *New events* involve the new variables only
    - They refine some implicit events doing “nothing” in the abstraction
- Refinement will thus result in a discrete observation of reality, which is now performed using a *finer time granularity*.

Example 1

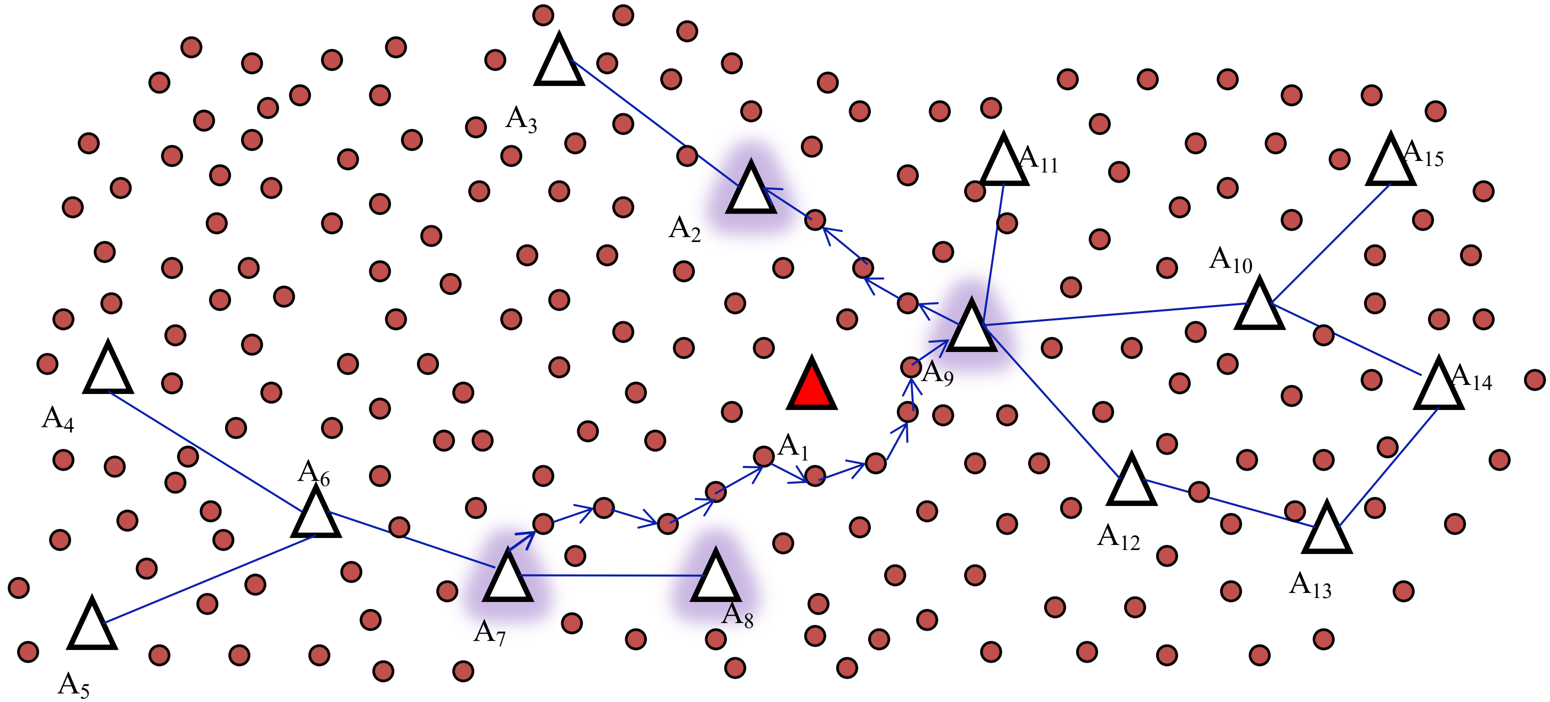
Wireless sensor-actor networks
Partitions and recovery

Main purpose of the algorithm:

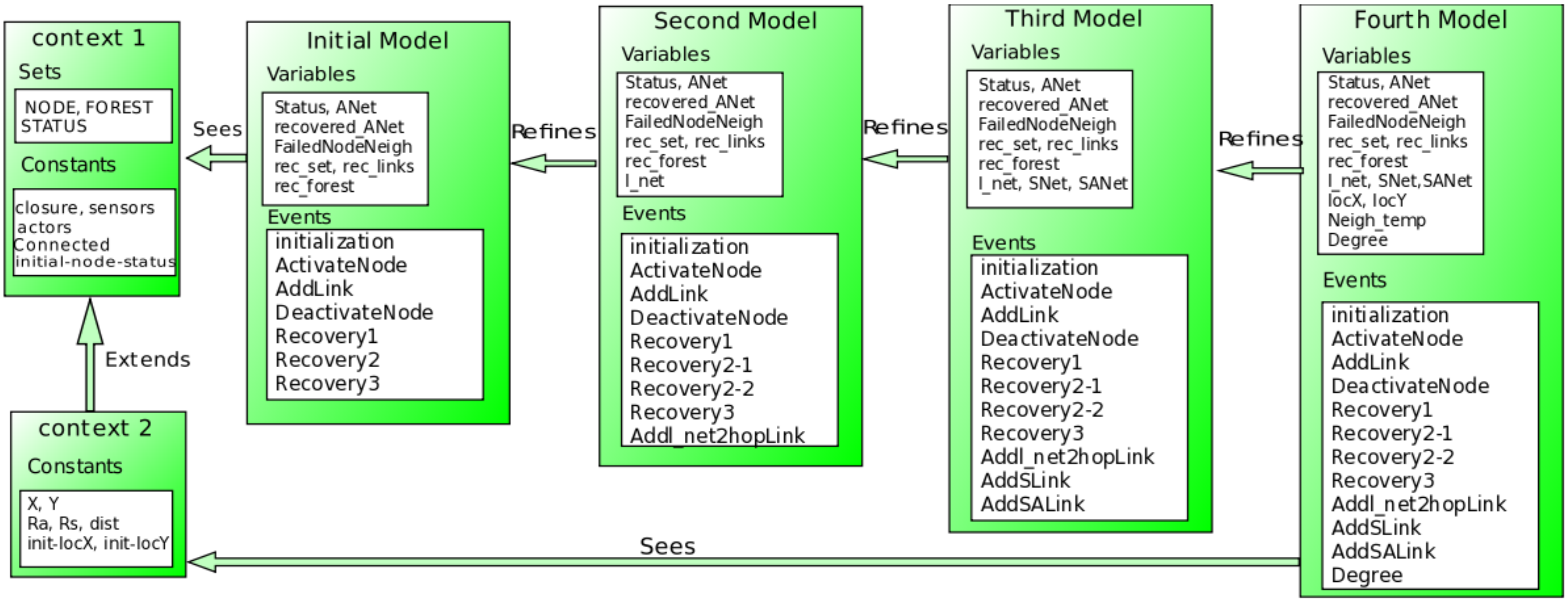
→ Re-establish connections among partitions formed by an actor failure



Result for Example 1



Refinement overview



We show that recovery is possible when the sensors are in a transitive closure relation.

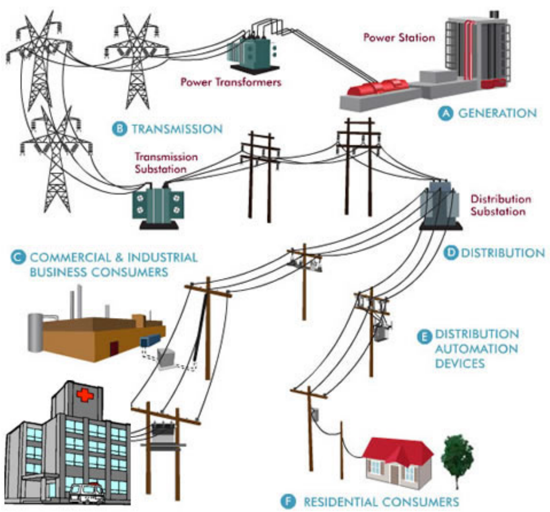
Åbo Akademi University

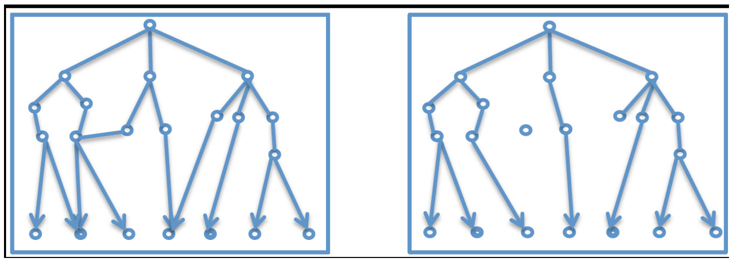
Example 2

Smart electrical grids modeling

Smart grid recovery from failures

- Configurations
    - Momentary: tree
    - All possibilities: graph
- Recovery
    - Reconfiguration
    - Reestablishment
- Priorities
    - Hospitals/street lighting





Åbo Akademi University

Refinement overview

Context C_0

Sets
- NODE
- MODE
- STATE

Constants
- consumer
- supplier
- pr1
- pr2
- recovery
- regular
- premium
- on
- off

Context C_1

Constants
- netrel
- cl
- tree
- root
- net
- initial_net_setting

M_0 sees C_0

VAR
- cons_st
- mode

EVENTS
- Initialisation
- random_failure
- recovery_pr1
- recovery_pr1_complete
- recovery_pr2
- recovery_pr2_complete

M_1 sees C_1

VAR
- nodes
- dyn_net
- failed_edges
- failed_paths
- failed_flag

EVENTS
- Initialisation
- edges_fail
- failed_paths_update
- failed_tree_update
- pr1_failure
- pr2_failure
- recovery_pr1
- recovery_pr1_complete
- recovery_pr2
- recovery_pr2_complete

VAR
- reconfigure
- reestablish

EVENTS
- Initialisation
- edges_fail
- failed_paths_update
- failed_tree_update
- pr1_failure
- pr2_failure
- recovery_pr1_reconfigure
- recovery_pr1_reestablish
- recovery_pr1_complete
- recovery_pr2_reconfigure
- recovery_pr2_reestablish
- recovery_pr2_complete

$$M_0 \sqsubseteq M_1 \sqsubseteq M_2$$

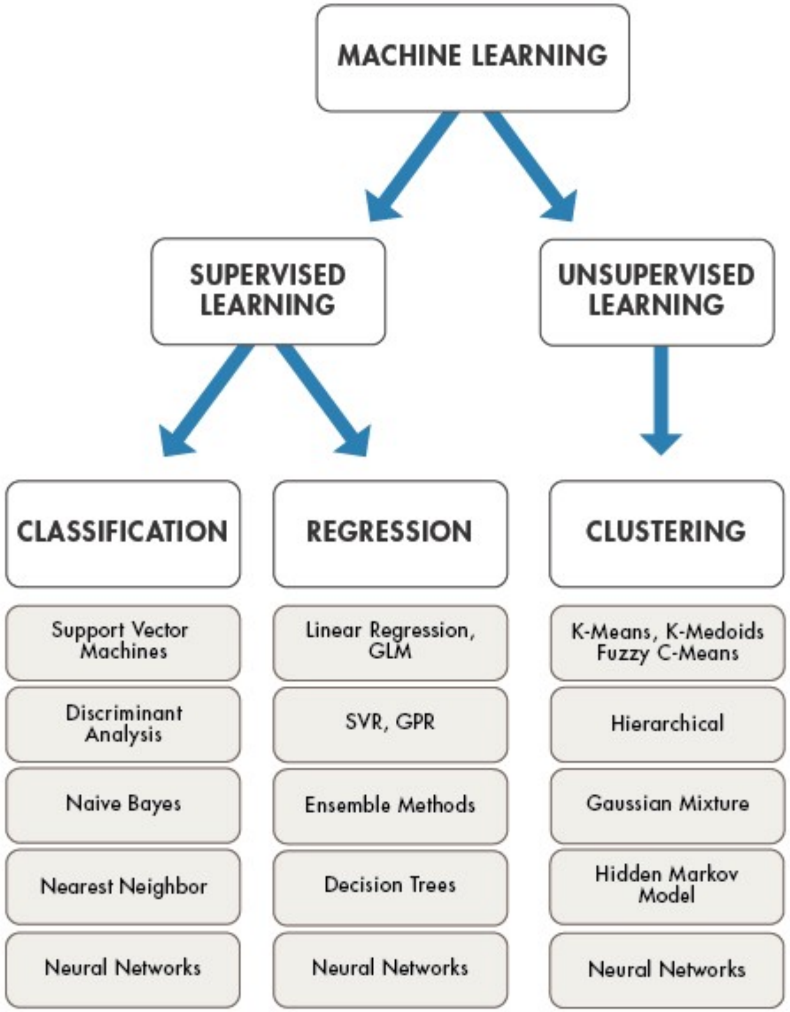
Summing up

- **The sciences do not try to explain, they hardly even try to interpret, they mainly make models.**

  John von Neumann

- **We aim at making *correct* models**
  - **With respect to the requirements**
  - **To analyze properties**
  - **What about machine learning?**

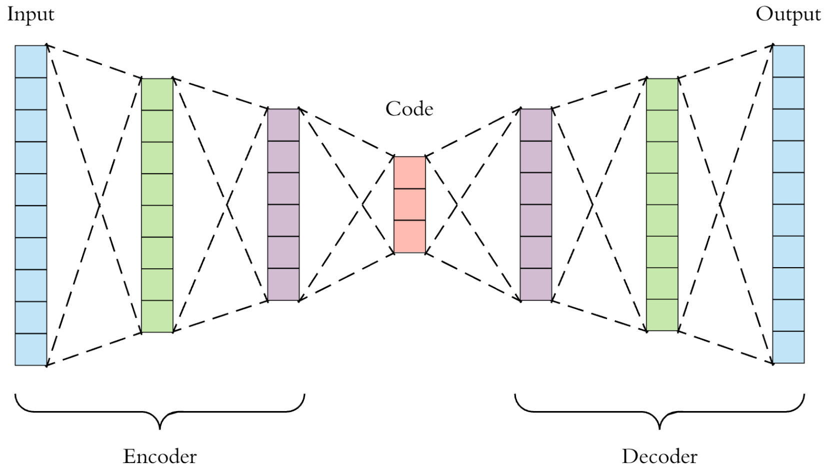
Machine Learning



Formal Methods vs Machine Learning

- Deductive vs inductive approach
- Max Planck Institute (Germany)
  - Can we learn invariants?
    - And then prove properties
  - Can we learn models?
    - And then invarants, and then prove
- Correctness of ML algorithms
  - Linear regression vs deep learning

What about CA13137?



- Autoencoder for GW
- GW – the anomaly?
- How about the latent space?
  - Can we learn something about the GW parameters from there?