FONDAZIONE
BRUNO KESSLER

Applications of SMT solvers

Alessandro Cimatti
Embedded System Unit
Fondazione Bruno Kessler
Trento, Italy
cimatti@fbk.eu

FONDAZIONE BRUNO KESSLER

Satisfiability Modulo Theories

- SMT solvers are finding their way in many different application domains
- Reasons for success?
  - allows to deal with richer representation
  - increase capacity by working above the boolean level

- Successful applications in various fields
  - verification of pipelined microprocessors
  - equivalence checking of Microcode
  - software verification
  - whitebox testing for security applications
  - design space exploration, configuration synthesis
  - discovery of combinatorial materials

Focus on three main areas

- SMT-based for verification of complex systems
  - See also tutorials at SAT/SMT'11, FMCAD'12, ICAPS'13
- SMT-based temporal planning
  - Scheduling with uncertainty
  - The role of quantification
- SMT-based reliability assessment
  - Analysis of redundancy architectures
  - The role of EUF and predicate abstraction

FONDAZIONE
BRUNO KESSLER

SMT-based verification of Hybrid Systems

Alessandro Cimatti
Embedded System Unit
Fondazione Bruno Kessler
Trento, Italy
cimatti@fbk.eu

Joint work with Sergio Mover and Stefano Tonetta

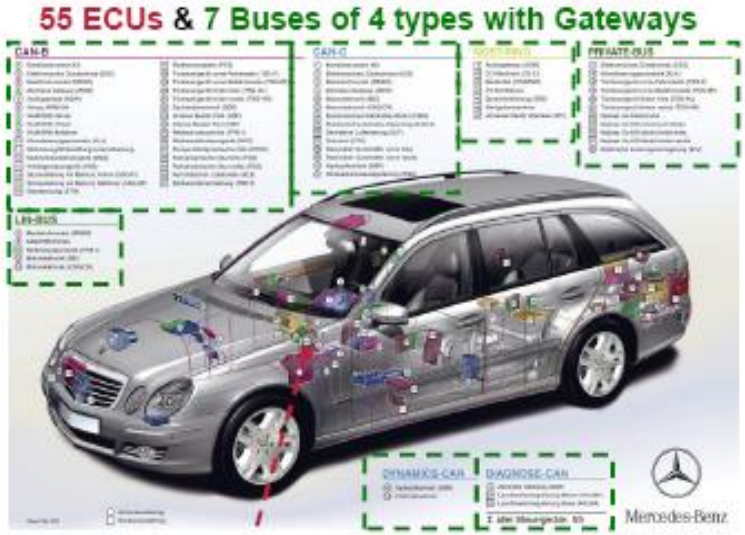
FONDAZIONE BRUNO KESSLER

Take away messages

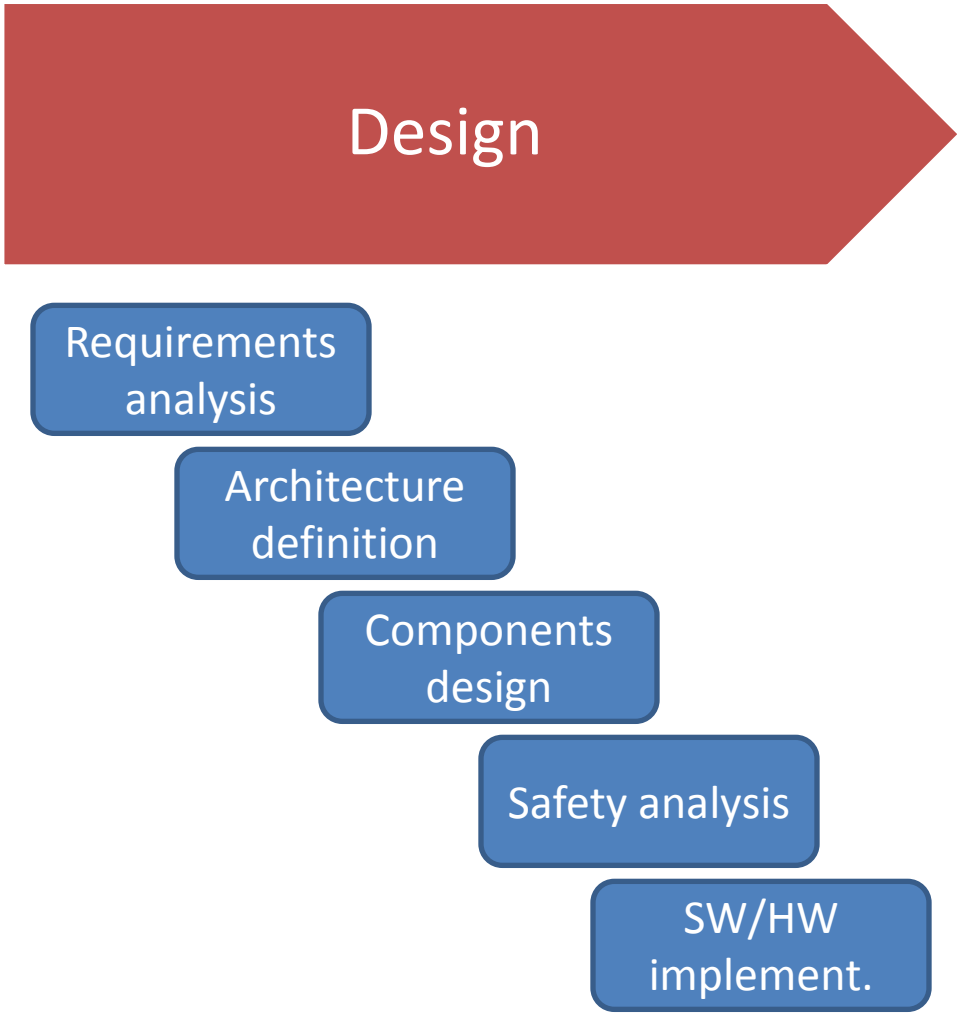
- The need for verification
  - Very complex systems
- Verification in a broader sense
  - Rigorous analysis of the behaviour of dynamic systems
- Hybrid automata
  - A uniform and comprehensive formal model
- Satisfiability Modulo Theories
  - Higher level symbolic modeling
  - Efficient engines: SAT + constraint solving
- SMT-based Verification
  - Many effective complementary algorithms

The Design Challenge

- Designing complex systems
  - Automotive
  - Railways
  - Aerospace
  - Industrial production
- Sources of complexity:
  - Hundreds of functions
  - Networked control
  - Real-time constraints
  - Complex execution model with mixture of real-time and event-based triggers
  - System composed of multiple heterogeneous subsystems
  - Critical Functions:
    - » ABS, drive-by-wire
    - » Operate switches, level crossings, lights
    - » Manage on-board power production
  - Conflicting objectives:
    - » Avoid crashes vs move trains



Life Cycle of Complex Systems

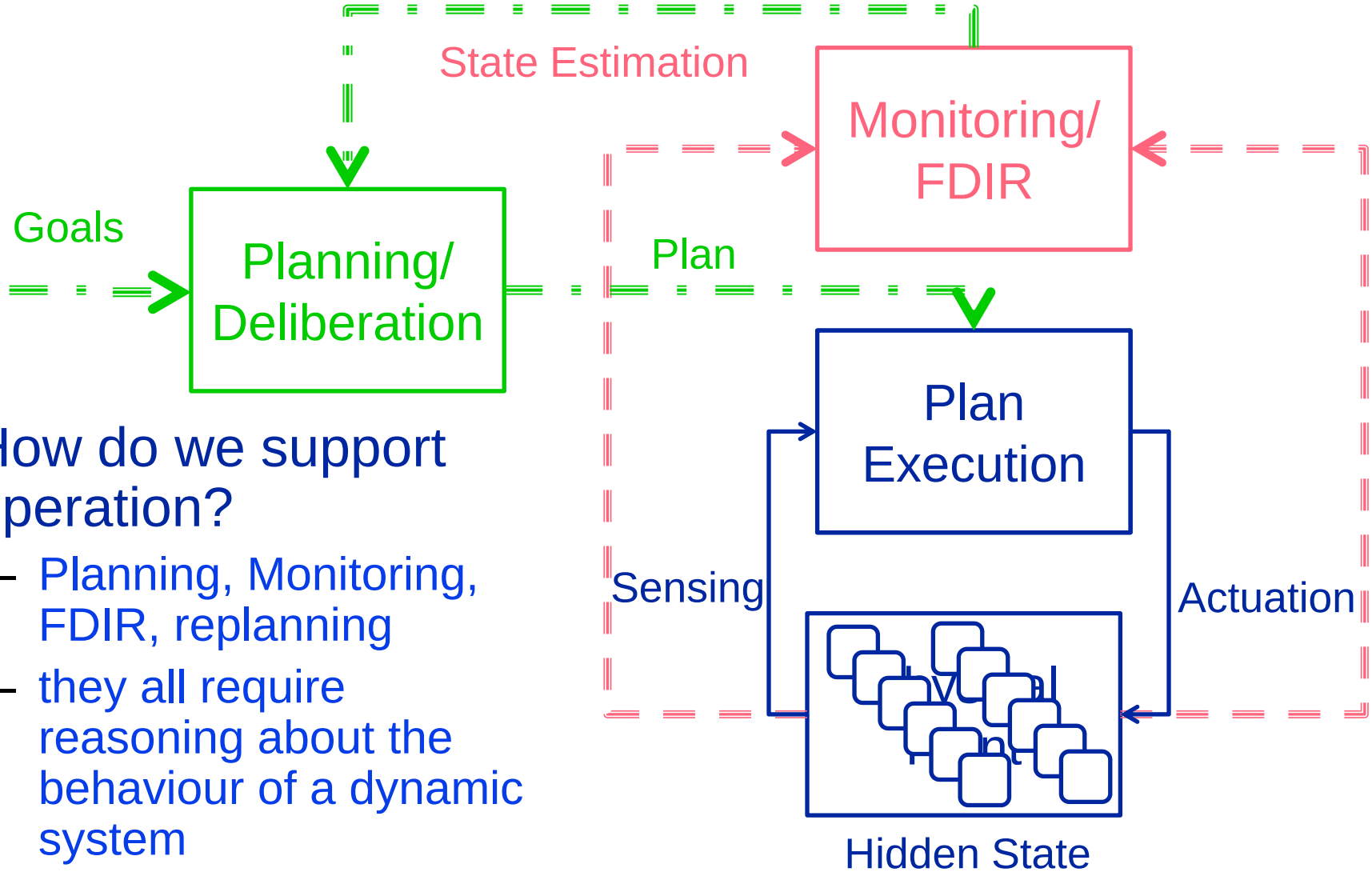


- How do we support the design?
- Requirements validation:
  - Are the requirements flawed?
- Functional correctness
  - Does the system satisfy the requirements?
- Safety assessment
  - Is the system able to deal with faults?

From design to operation…

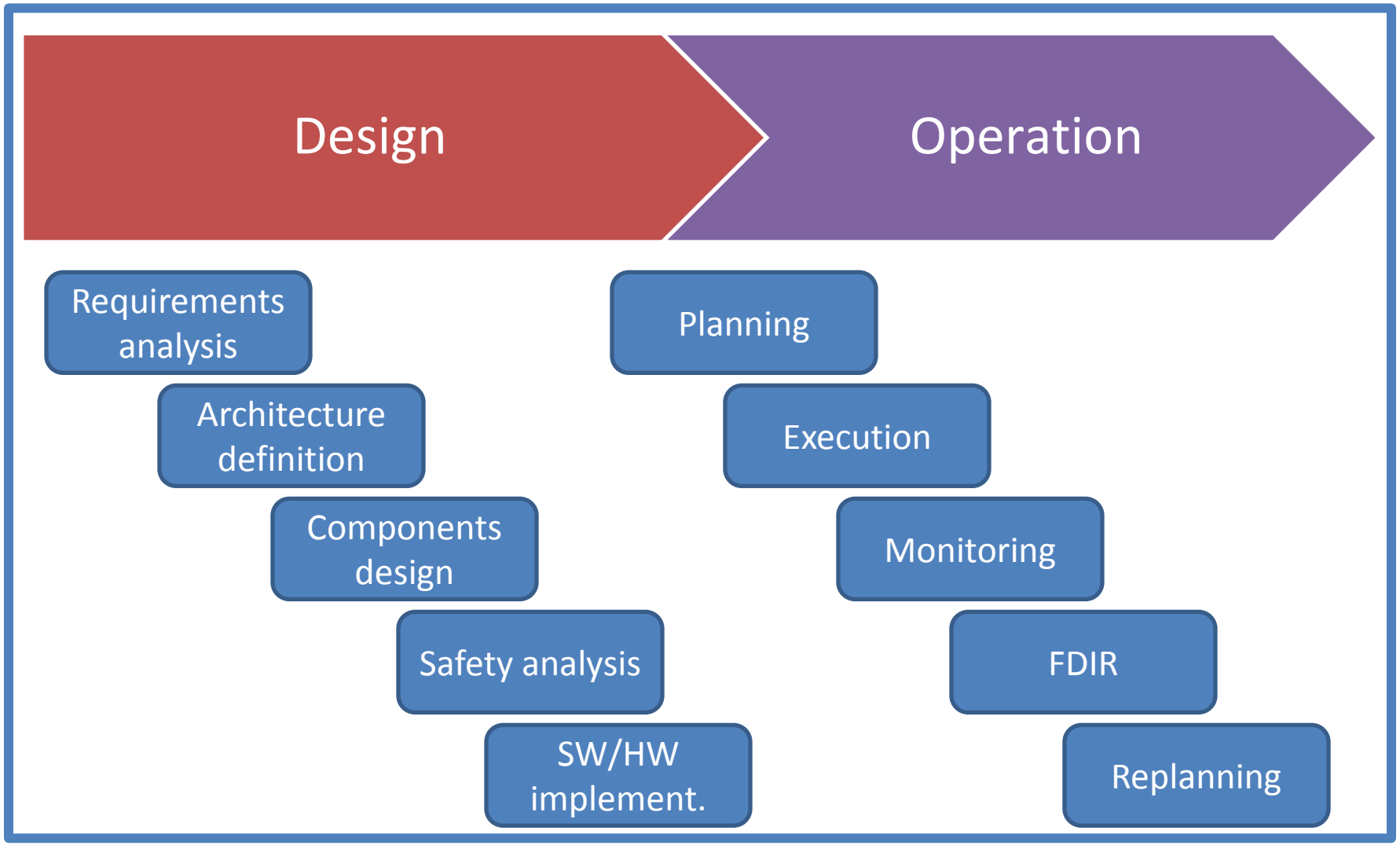
- Planning
  - plan how to achieve desired “firing” sequence
  - retrieve pipes from holds, pre-weld, send to firing line, final weld
- Execution Monitoring
  - welding may fail, activities can take more time than expected
  - plant may fail
- Fault Detection, Fault Identification/Isolation
  - is there a problem? where is it?
- Fault Recovery
  - put off-line problematic equipment
- Replanning
  - identify alternative course of actions, e.g. reroute pipes

Complex systems operation



- How do we support operation?
  - Planning, Monitoring, FDIR, replanning
  - they all require reasoning about the behaviour of a dynamic system

Life Cycle of Complex Systems

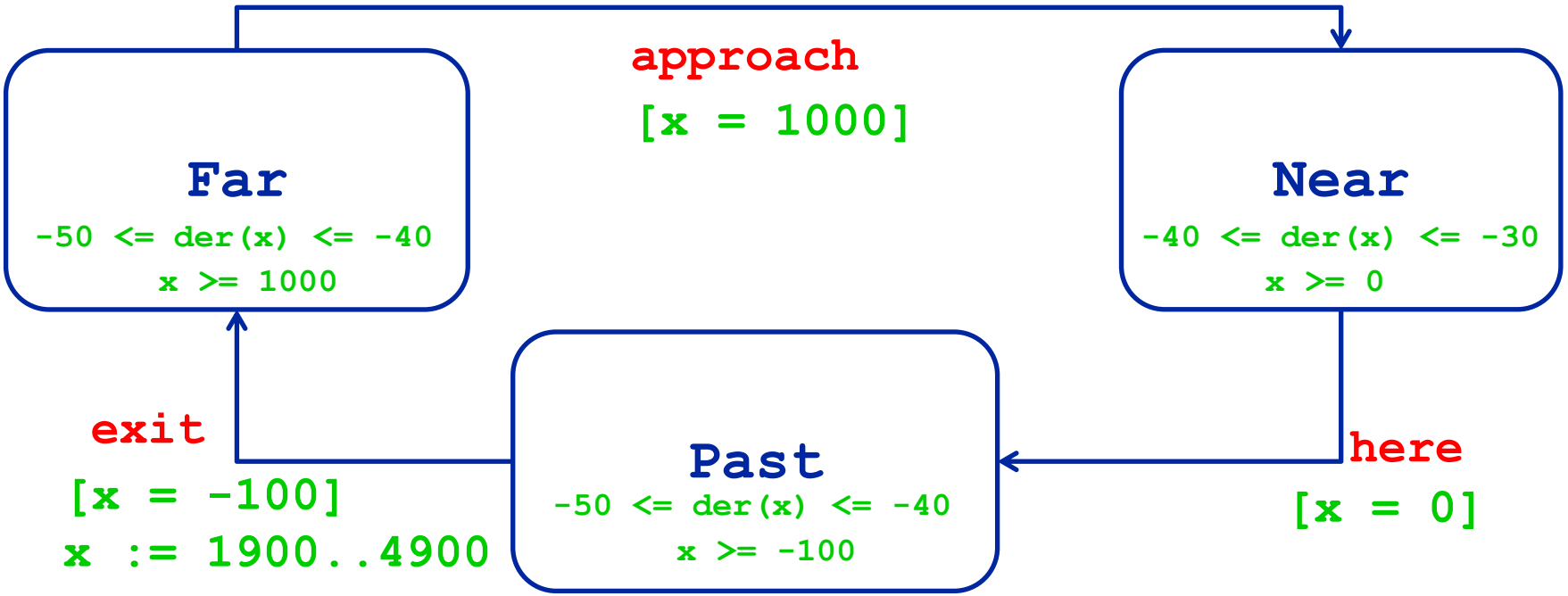


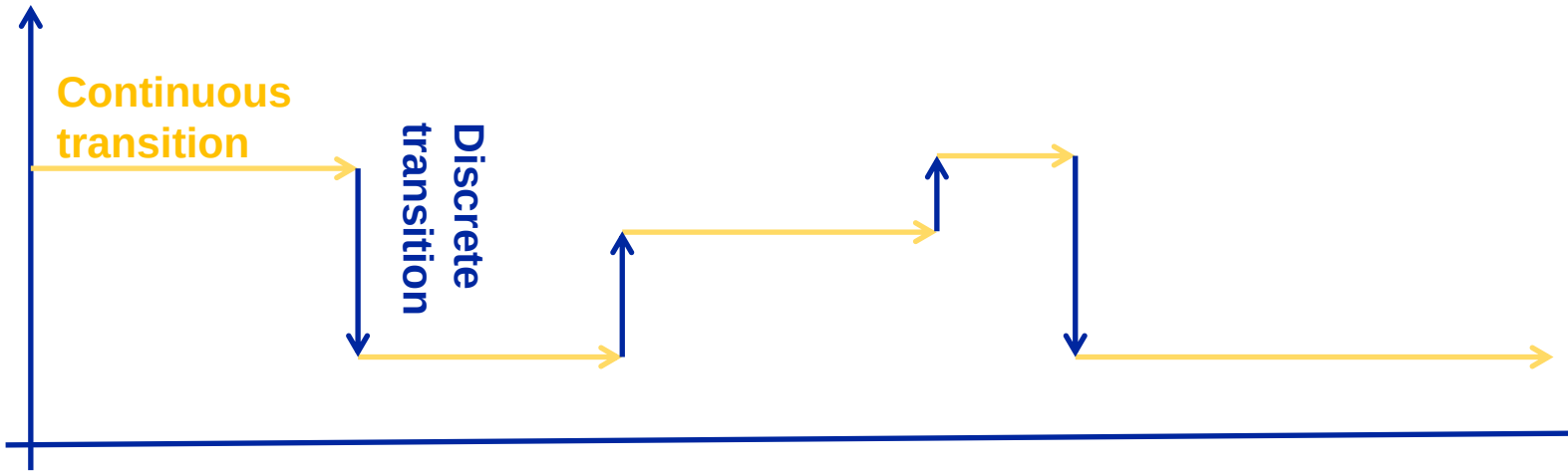
FONDAZIONE BRUNO KESSLER

The “formal” way

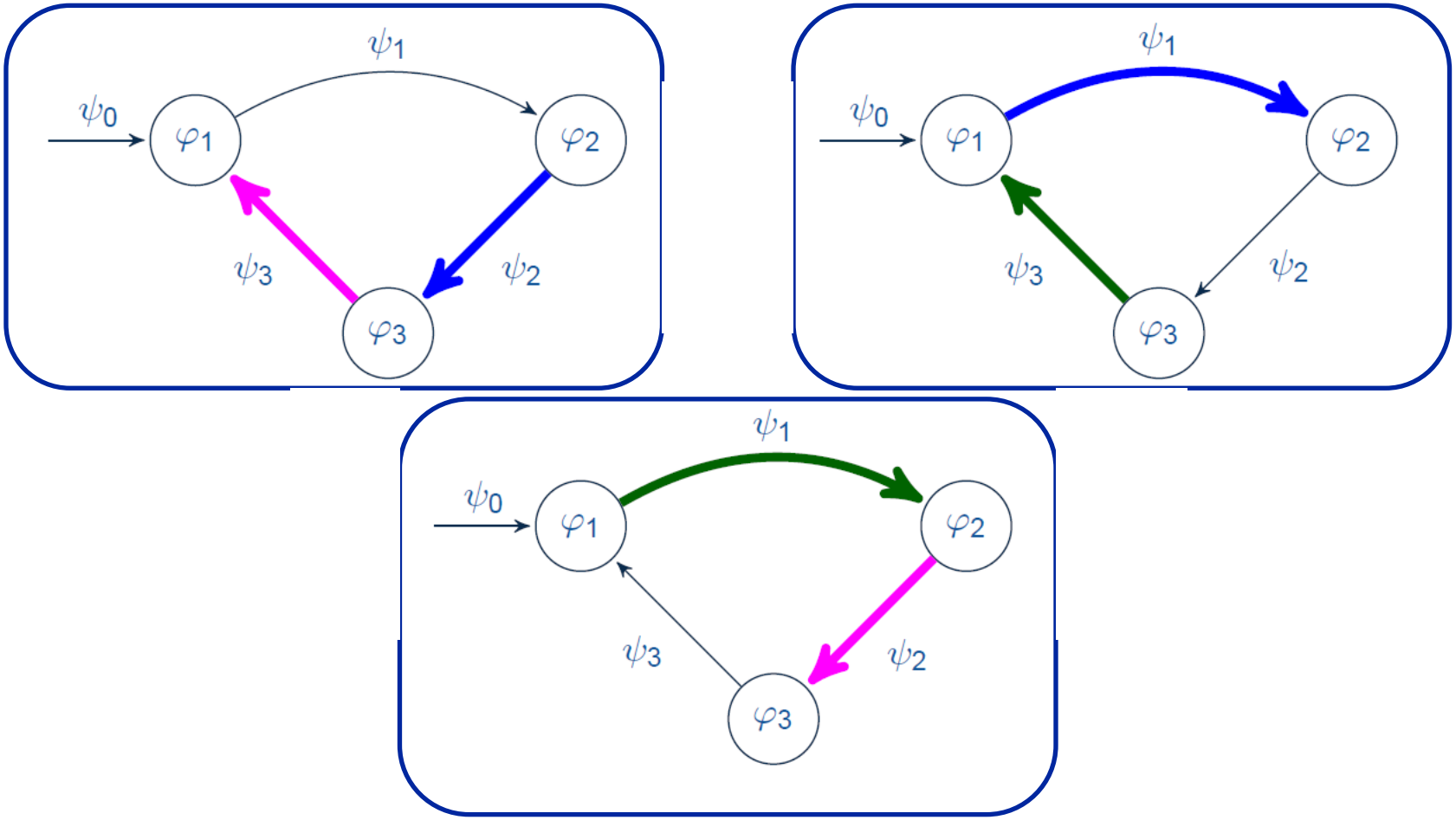
- The design-operation continuum
  - Both design and operation tasks require the analysis of the behaviour of dynamic systems over time
  - In fact, they often require the analysis of the *same* dynamic systems
  - the analysis must be “rigorous” (predictability, certification)
- We need a rich formalism
  - to represent the behaviour of complex systems
  - to provide the reasoning tasks required for design and for operation
- Representation challenges
  - Nondeterministic behaviours
  - Possible Faults
  - Operation in degraded modes
  - Limited Observability
  - Parallel actions/tasks
    - » Start actuations in different subsystems
  - Time
    - » Time taken by procedures
    - » e.g. moving, welding, checking, …
  - Resources
    - » Power consumption, space, bandwidth, memory, …

Hybrid automata





Networks of hybrid automata



Properties of hybrid automata

- Well founded, comprehensive and well studied
  - Clear definition of behaviors of model
  - Which states are reachable
- Temporal properties to express scenarios and requirements
  - *never two processes in critical region*
  - *always if req then within 5 sec response*

- Model checking
  - » Does the system satisfy the requirements?
- Temporal reasoning
  - » Strong/weak/dynamical controllability?
- Planning
  - » Find the inputs that will bring the system to required state

- The workhorse: satisfiability modulo theories

Satisfiability modulo theories

- Satisfiability of a first order formula …
  - where the atoms are interpreted modulo a background theory
- Theories of practical interest
  - Equality Uninterpreted Functions (EUF)
    - » $x = f(y), h(x) = g(y)$
  - Difference constraints (DL)
    - » $x – y \leq 3$
  - Linear Arithmetic
    - » $3x – 5y + 7z \leq 1$
    - » reals (LRA), integers (LIA)
  - Arrays (Ar)
    - » read(write(A, i, v), j)
  - Bit Vectors (BV)
  - Their combination

FONDAZIONE BRUNO KESSLER

Statisfiability Modulo Theories

- An extension of boolean SAT
- Some atoms have non-boolean (theory) content
  - » A1 : $x - y \leq 3$
  - » A2 : $y - z = 10$
  - » A3 : $x - z \geq 15$
- Theory interpretation for individual variables, constants, functions and predicates
  - » if $x = 0$, $y = 20$, $z = 10$
  - » then A1 = T, A2 = T, A3 = F
- Interpretations of atoms are constrained
  - » A1, A2 and A3 can not be all true at the same time

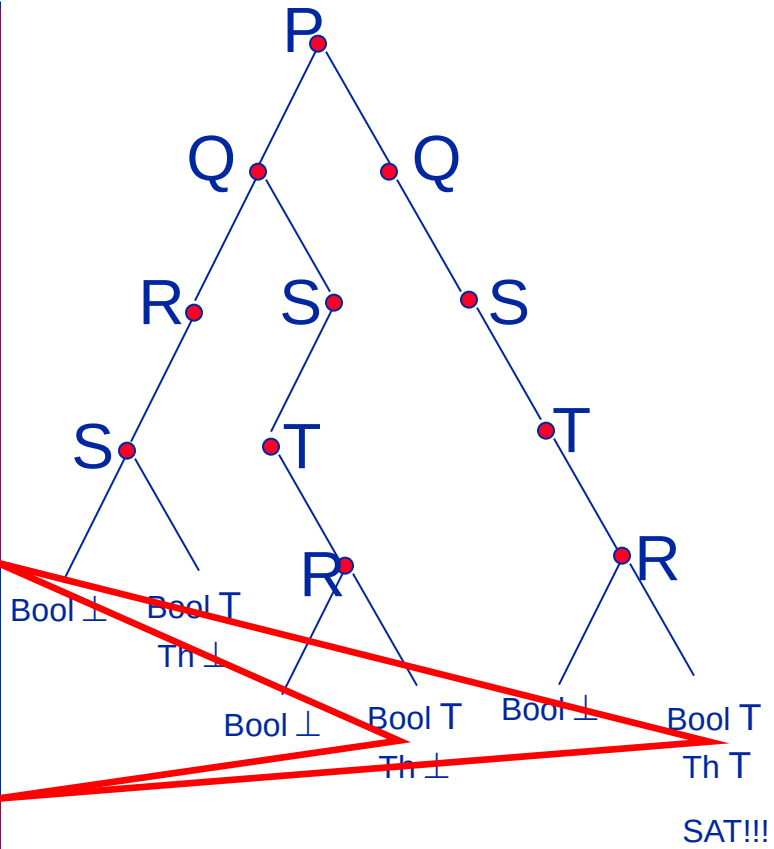
FONDAZIONE BRUNO KESSLER

SMT solvers

- Boolean reasoning + constraint solving
  - SAT solver for boolean reasoning
  - theory solvers to interpret numerical constraints

SMT search space

P	T	$x - y \leq 3$
P_1	F	
P_2	T	$y - z = 10$
Q	F	
R	T	$x - z \geq 15$
R_1	F	
S	F	$z - 2*w = 1$
S_1	T	



SMT solvers in practice

- In practice, the integration is very tight
  - SAT solver working as an enumerator
  - Theory solver follows the stack-based search
    - » Inconsistent partial assignments are pruned on the fly
    - » conflicts clauses learnt from theory reasoning
    - » used to drive search at the boolean level

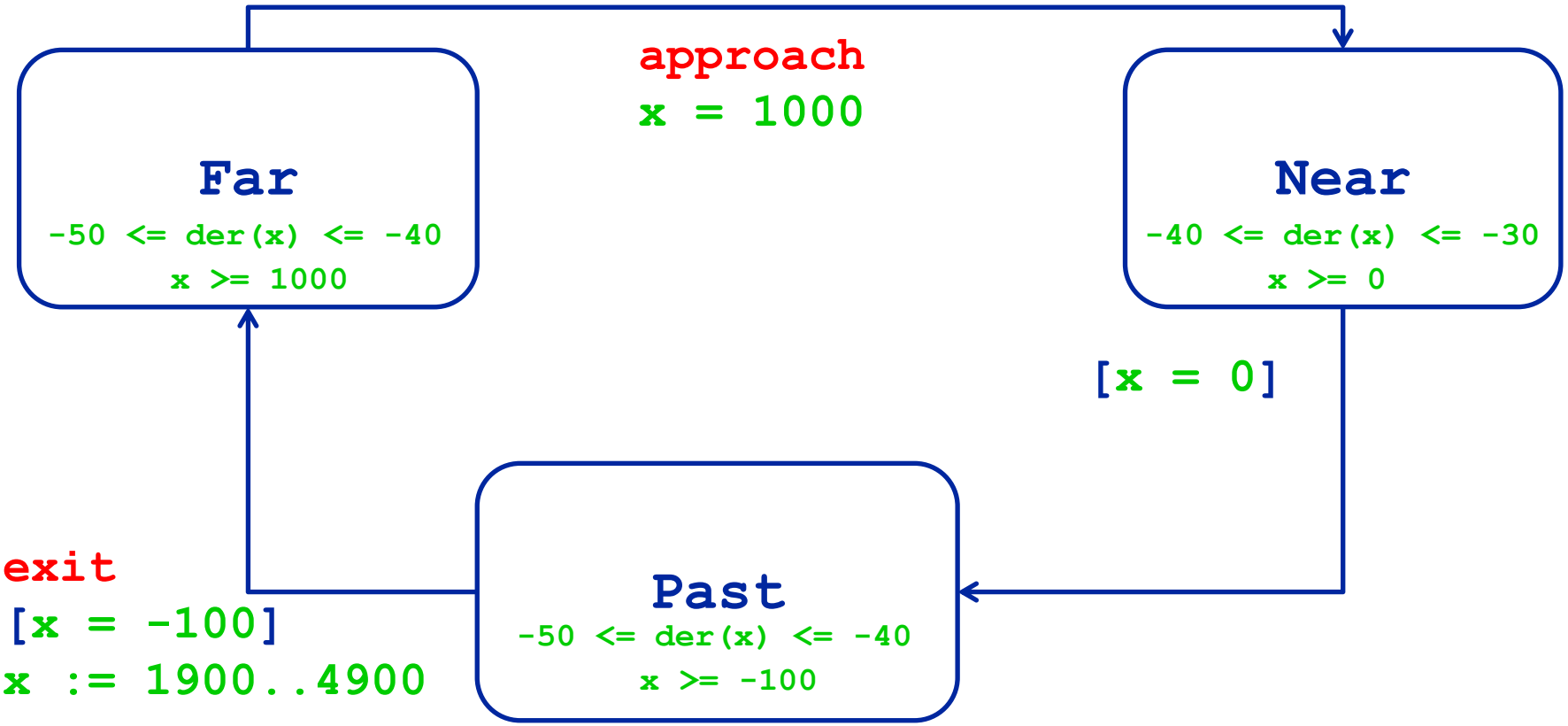
- Additional features
  - Model construction
  - Incremental interface
  - Unsatisfiable core
  - Proof production
  - Interpolation

- Satisfiability Modulo Theories: a sweet spot?
  - increase expressiveness
  - retain efficiency of boolean reasoning
- Trade off between expressiveness and reasoning
  - SAT solvers: boolean case, automated and very efficient
  - theorem provers: general FOL, limited automation

The SMT community

- Standard language and benchmarks
  - **http://www.smt-lib.org**
- Yearly competition
  - **http://www.smt-comp.org**
- Solvers
  - YICES, OpenSMT, MathSAT, Z3, CVC, …

From HA to SMT formulae




```
s = Past -> x >= -100
exit -> s = Past & x = -100
exit -> next(s) = Far
exit -> next(x) in 1900..4900
```

```
timed -> next(s) = s
timed & s = Past ->
   next(x) >= x - 50*delta &
   next(x) <= x - 40*delta
```

The SMT representation

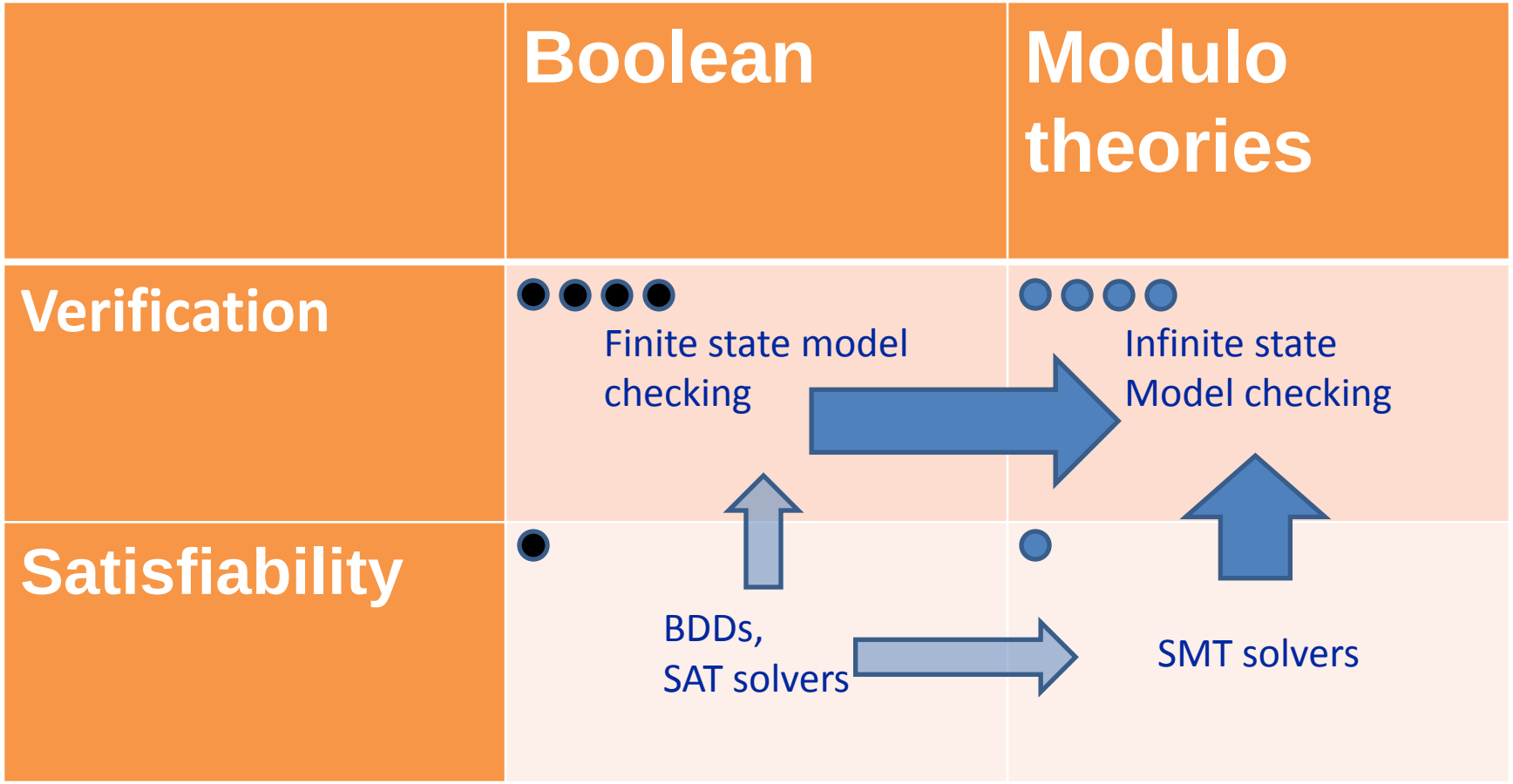
```
VAR s : { Past, Near, Far }
VAR x : real;
...
INIT x <= 5000
INIT s = Past
...
TRANS
s = Past -> x >= -100
exit -> s = Past
exit -> next(s) = Far
exit -> next(x) >= 1900
exit -> next(x) <= 4900
...
timed -> next(s) = s
timed -> next(x) >= x - 50*delta
timed -> next(x) <= x - 40*delta
```

Hybrid automata symbolically represented by SMT formulae!

I(X) initial states
R(X,X') transition relation
B(X) bad/target states

Satisfiability vs Verification

(or, combinational vs sequential)



SMT-based symbolic model checking

- Given representation as SMT formulae I(X), R(X,X'), B(X)
- Bounded model checking
  - State variables replicated K times
    - » X_0 , X_1, …, X_{k-1}, X_k
  - Look for bugs of increasing length
    - » $I(X_0) \wedge R(X_0, X_1) \wedge \ldots \wedge R(X_{k-1}, X_k) \wedge B(X_k)$
    - » bug if satisfiable
    - » increase k until …
- Many other verification techniques
  - K-induction
  - Interpolation
  - Abstraction/refinement
  - IC3
- Several enhancements for hybrid systems
  - use local time for each automaton
  - guided search for scenario-based analysis

Tools and applications

- The MathSAT SMT solver
  - **http://mathsat.fbk.eu**
- The NuSMV model checker
  - **http://nusmv.fbk.eu**
  - Forthcoming: a MathSAT-based extension of NuSMV

- Successfully applied in
  - OMC-ARE, COMPASS, AUTOGEF, FAME, FOREVER
    - » Support by the European Space Agency
  - Industrial technology transfer
    - » Avionics, railways, oil and gas

FONDAZIONE BRUNO KESSLER

Take away messages

- The need for verification
  - Very complex systems
- Verification in a broader sense
  - Rigorous analysis of the behaviour of dynamic systems
- Hybrid automata
  - A uniform and comprehensive formal model
- Satisfiability Modulo Theories
  - Higher level symbolic modeling
  - Efficient engines: SAT + constraint solving
- SMT-based Verification
  - Many effective complementary algorithms

FONDAZIONE
BRUNO KESSLER

Questions so far?