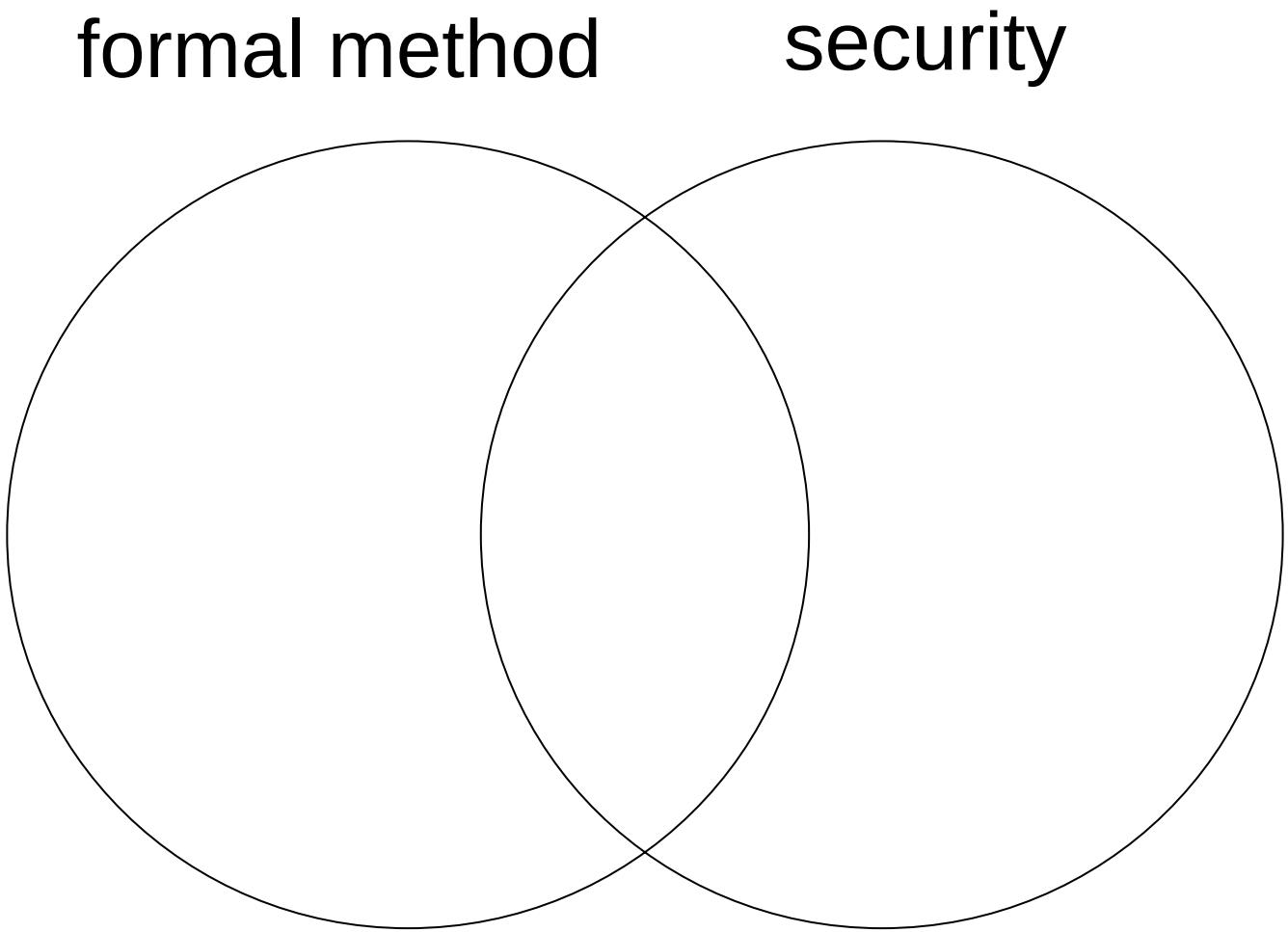
Overview of Formal Verification for Communication Protocols

通信プロトコルの形式的検証技術の概要

Tadashi ARARAGI (NTT CS labs)
櫟 粛之



verify your designed security protocol by computers
(generate desired security protocols by computers)

Verification of Systems

Target Systems:

- Hardware
- Software: Programs, Distributed Systems
- Security Protocols

Verification:

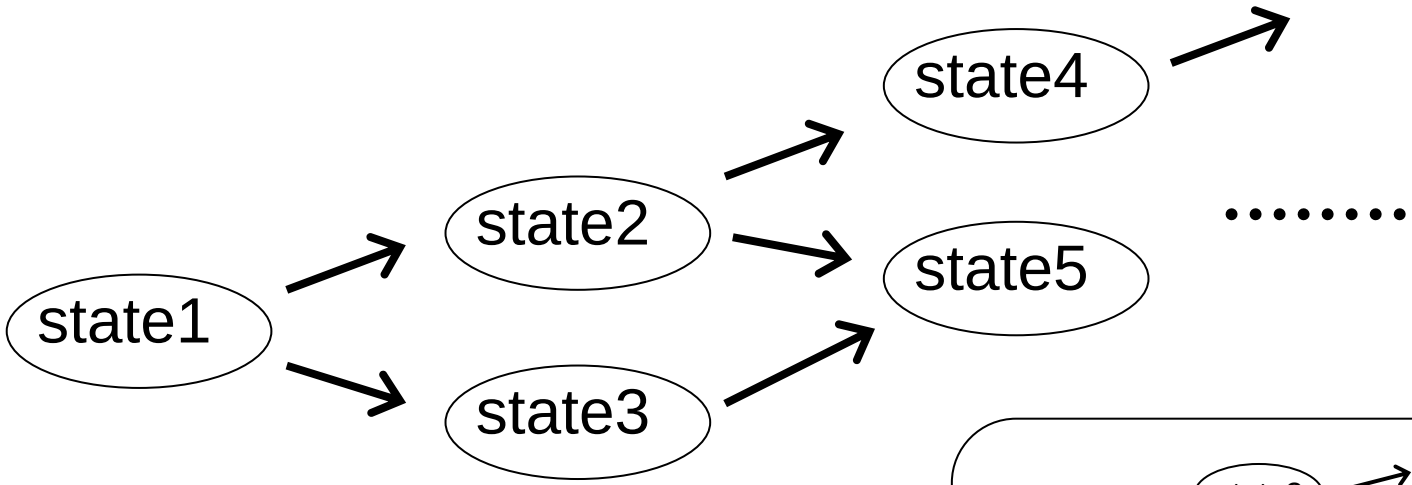
- Description of System's Behavior

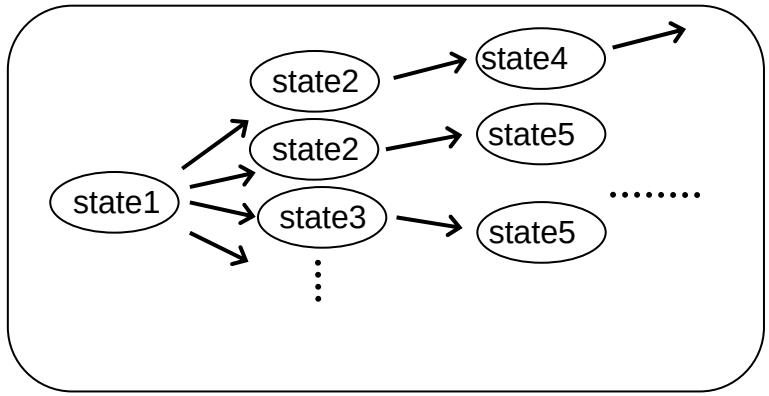
  （program: executable, protocol: abstract design)

- Requirement Specification

  (logic: □(rcv(m)->◇snd(m')), sim: abst. ideal behavior)

System Behavior





Branching:
- several inputs on the way
- timing of receiving messages (asynchronous systems)
- adversary's possible interaction

Formal Verification

Formal Description:
- Logic
- Process Algebra
- Automata

Verification Approach:
- Model Checking (automatic)
- Theorem Proving (proof assist)

Completeness & Soundness
- bounded

Research Issue

Infinity in verification:

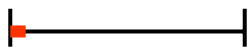
progress

(1) infinite length of executions

(2) infinite state of systems

(3) unbounded number of distributed systems

(4) unbounded number of sessions

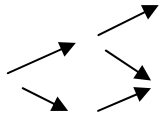


Model Checking

(1) infinite length of executions

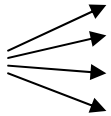
(1) temporal logic (finding loop)

- computational tree logic



|= AG(p->AFq)

- linier time logic



|= □(p->◇q)

(2) process algebra (partition refinement)

- pi-calculus, CSP

(bi-)simulation

Model Checking

(1) infinite length of executions

symbolic model checking

- logic:CTL, tool:SMV

  logical rep. of current sate and next state: R(c_x, n_X)

  $\exists$c_x. R(c_x, n_X) $\wedge \psi$(c_x): BDD, SAT solver

explict model checking

- logic:LTL, tool:SPIN

  search, trace inclusion :

  $M_{sys} \subset M_{spec}$?, $M_{sys \subset spec} = \phi$?

- partial order reduction
  remove redundant paths
- on the fly
  only related to spec

Abstraction

(2) infinite state of systems

- predicate/data abstraction

  if x > 3 then ... ---- x > 3 -> x_>_3

  if x = odd then ... ---- Nat -> {odd, even}

- interpolation

  refinement of a set of predicates

Parameterized Systems

(3) unbounded number of distributed systems

- counter abstraction

  #{P | P is in state s} : 0, 1, 1<

- invisible invariant

  $P_1 || \ldots || P_{N0} \models spec(safe) \Rightarrow P_1 || \ldots || P_N \models spec(safe)$

  tool: PAX-MONA (WS1S)

- regular expression

Dolve-Yao Model

(atomic1)

- P1, P2, P3, ... $\in M$: ID (<∞)

- R1, R2, R3, ... $\in R$: random-string (<∞) nonce、symmetric key
- K1, K2, K3, ... $\in K_{pub}$: public key (<∞) K^{-1}?、K_{adv} ?

（compound）

- encrypt : $\{m\}_K$ (m: msg, K: public key)
- pair : m1|m2 (m1, m2: msg)

==========

adversary's knowledge from S (closure)

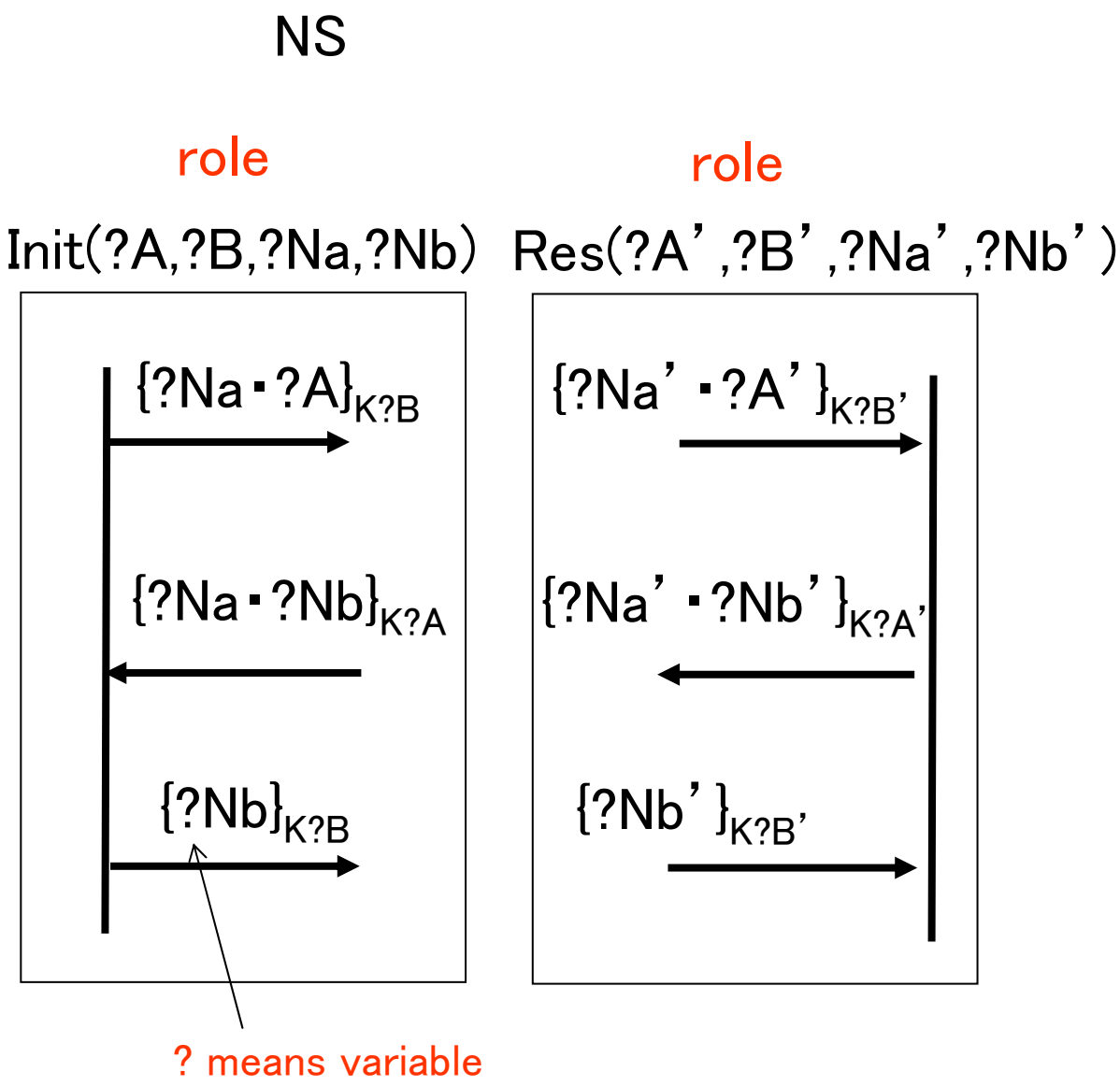
C[S] : the set of messages that S can compose from msg set S

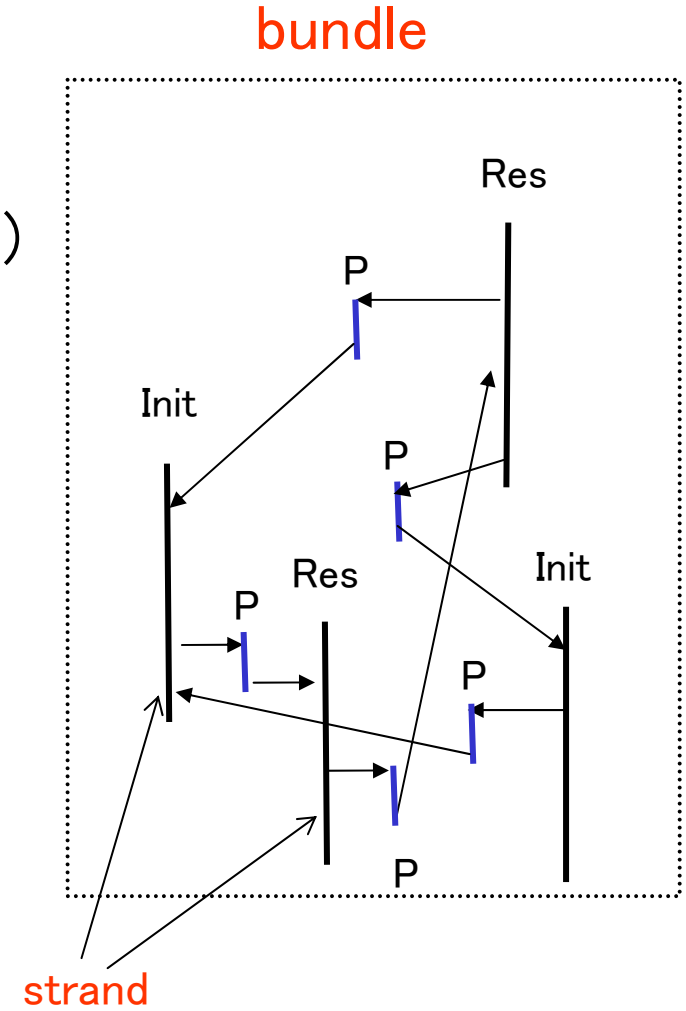
（adv's ability S |- m)

decrypt: K^{-1}, $\{m\}_K$ |- m

split: m1|m2 |- m1

Strand Space





Strand Space: Athena

requirement specification:

$\forall C.(s_1 \in C \wedge \ldots \wedge s_n \in C \Rightarrow s'_1 \in C \vee \ldots \vee s'_m \in C)$

(C: bundle, s1,..., s1,... strand)

mutual authentication

$\forall C.(Res(a, b, na, nb) \in C \Rightarrow Init(a, b, na, nb) \in C)$

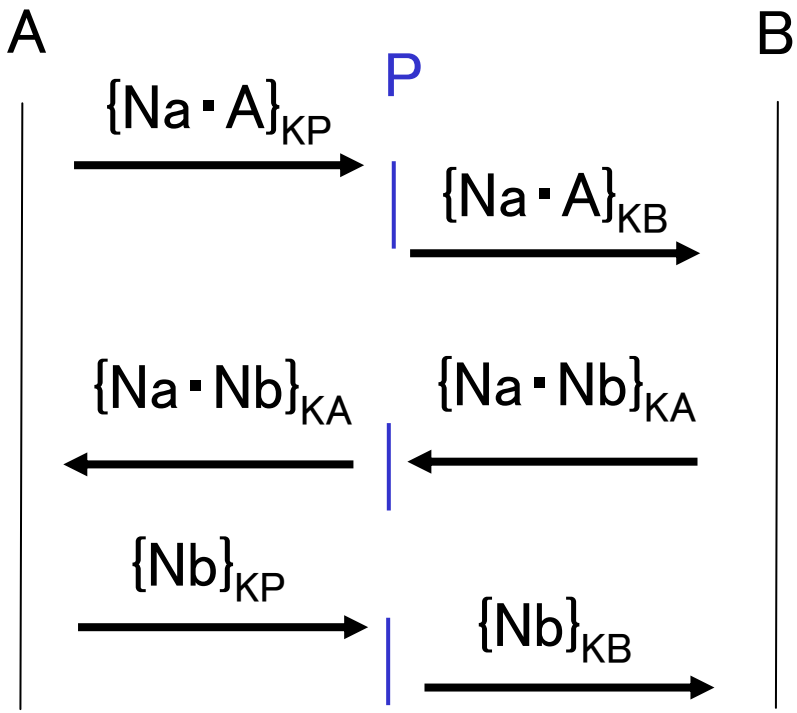
Verification :

from the roles of protocol and roles of penetrator P,
construct all possible bundles (relation between message send/receive and unification)
and check the requirement for each bundle.

Strand Space: Athena

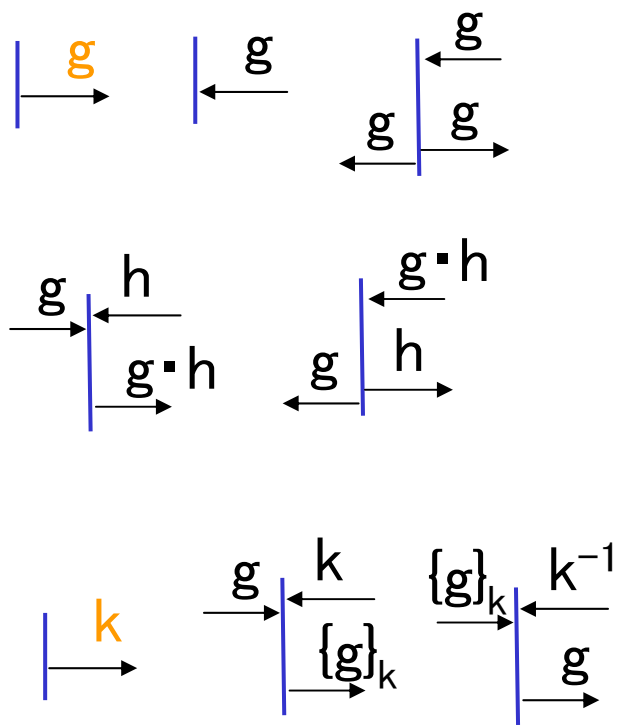
Penetrator's Attack

NS



B thinks P as A.

behavior (ability) of P



Others

- BAN-logic / knowledge logic (Halpern)
- Theorem Proving (HOL/Isabelle) assist
- ProVerif (logic programming) auto
- Casper (<- CSP+FDR-model checker)

Computational and Symbolic

Formalization:

polynomial time, probabilistic

passive, active, adaptive

Bellare-Rogaway P(succ of attack) < neg

Proof:

reduction to Hard problem primitives: one way func, DDH)

P(succ of attack) < P(succ of attack against hard problem)

sequence of games: A problem, Z known hard problem

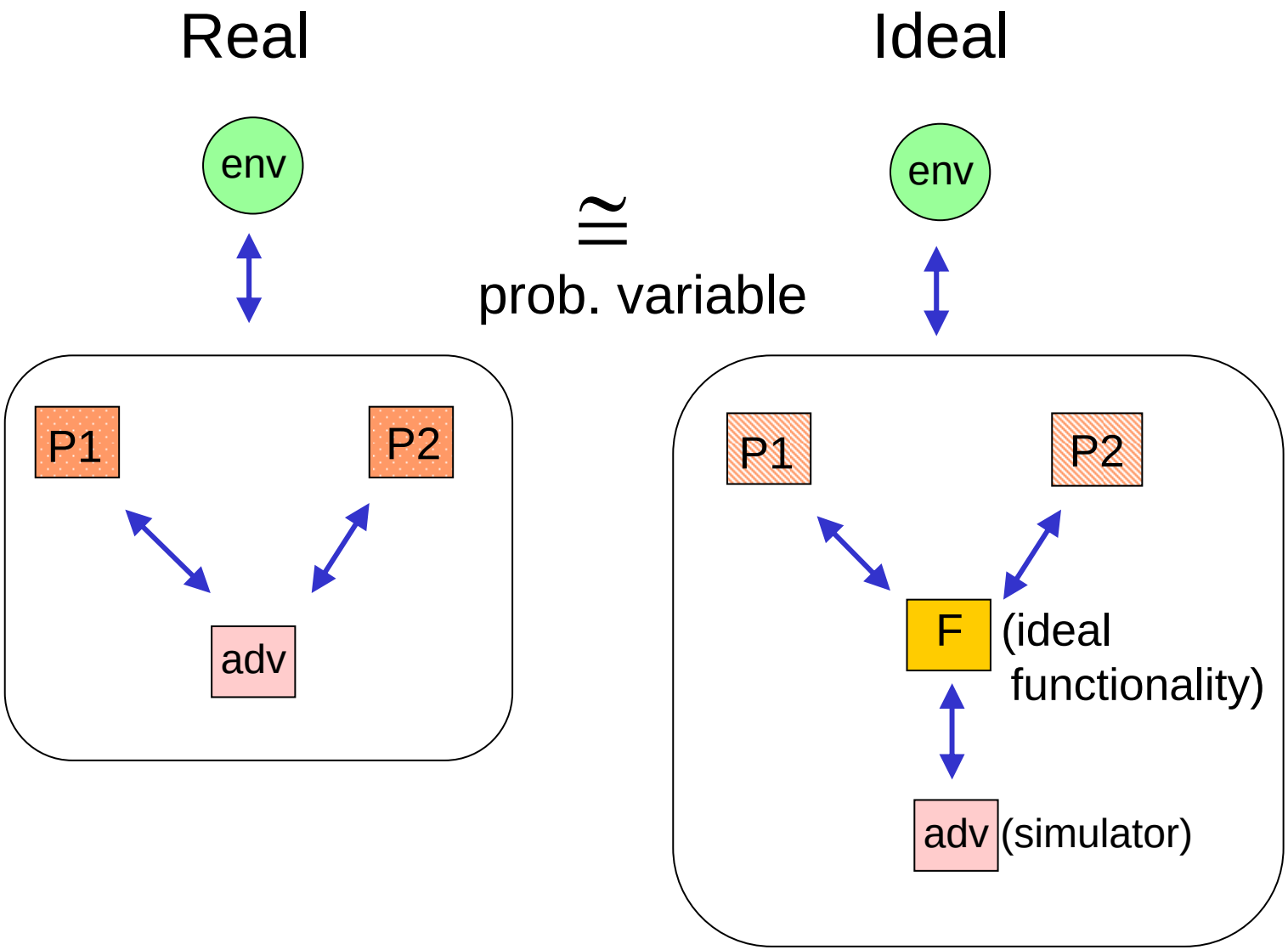
P(A)+neg < P(B), P(B)+neg<P(C) ...P(Y)+neg <P(Z)

Computational and Symbolic

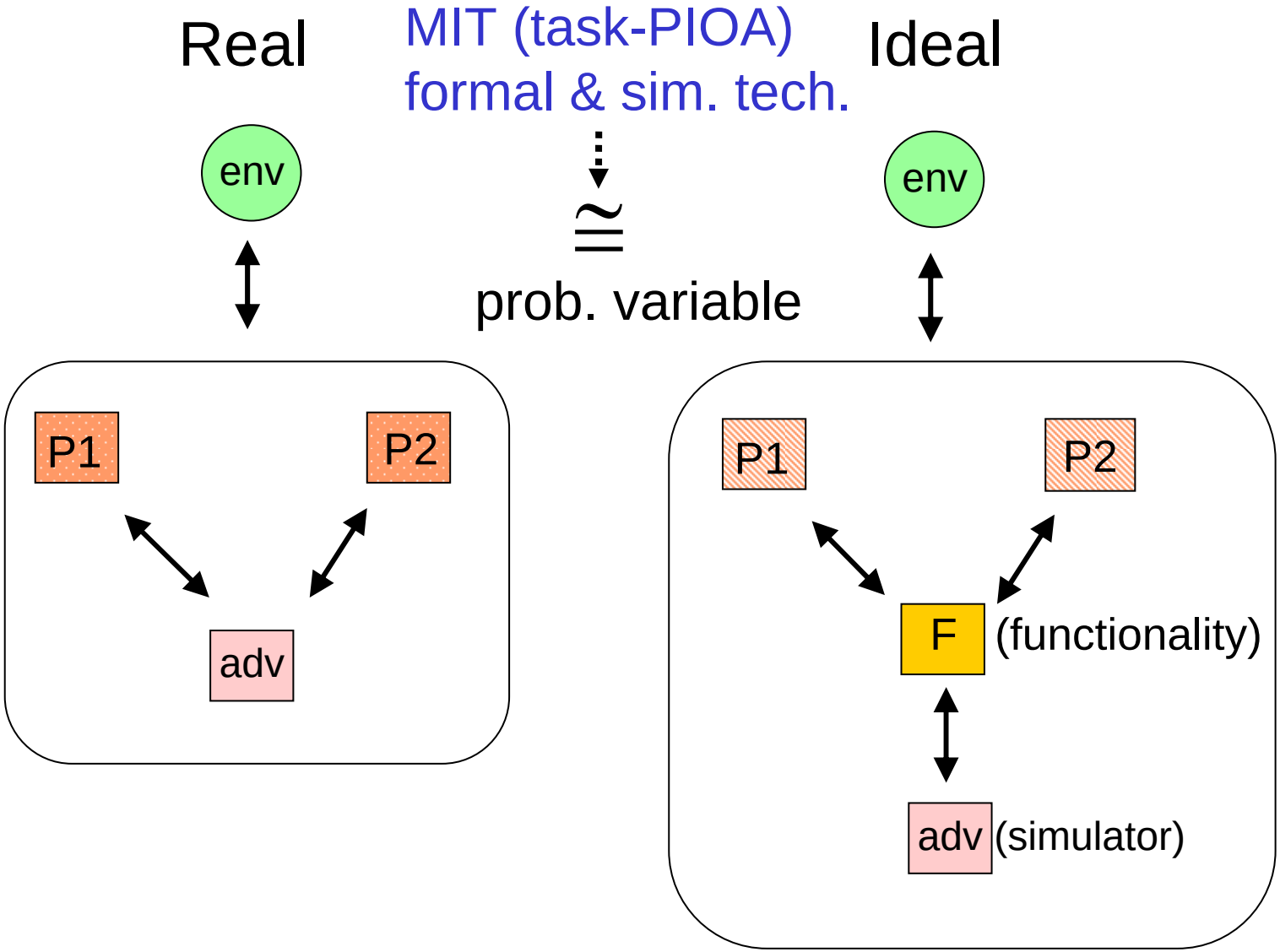
- UC (MIT): universal composability
- BPW (Zurich):
- Mitchell et al (Stanford)

- Abadi ...
- Pointcheval (ENS)

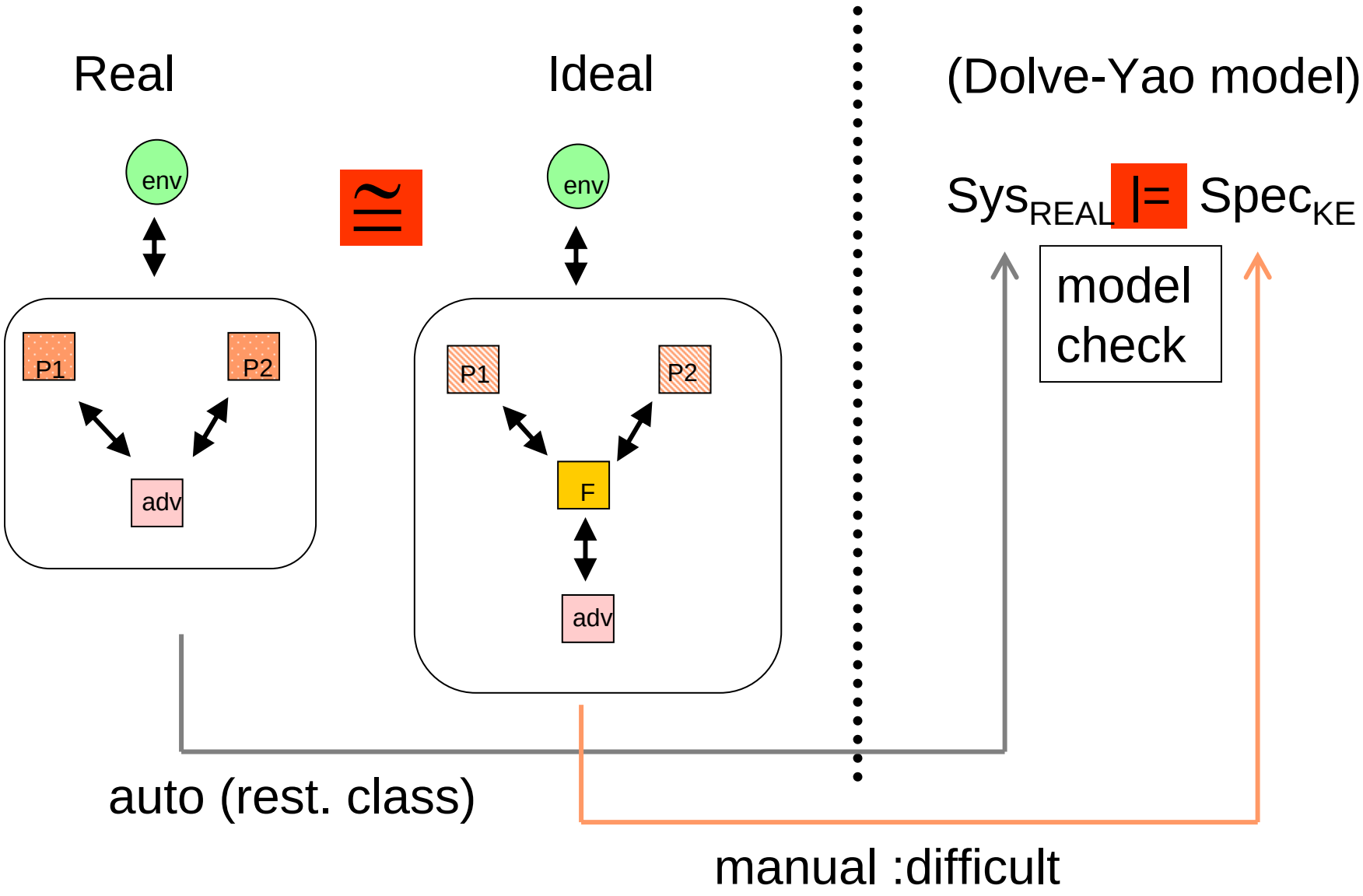
UC (Real & Ideal)



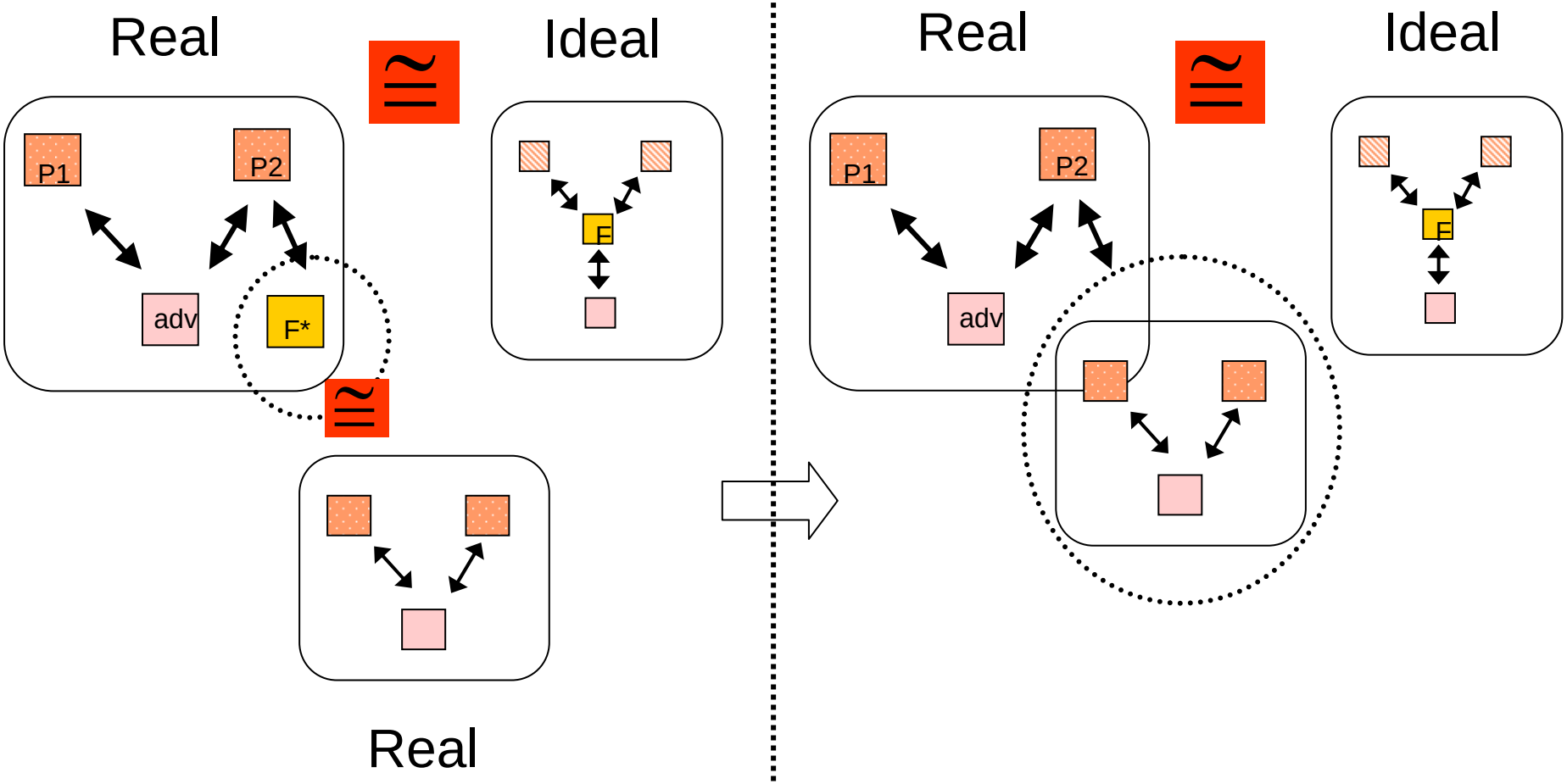
UC (Real & Ideal)



UC (Translation to Symbolic Model)



UC (Composability)



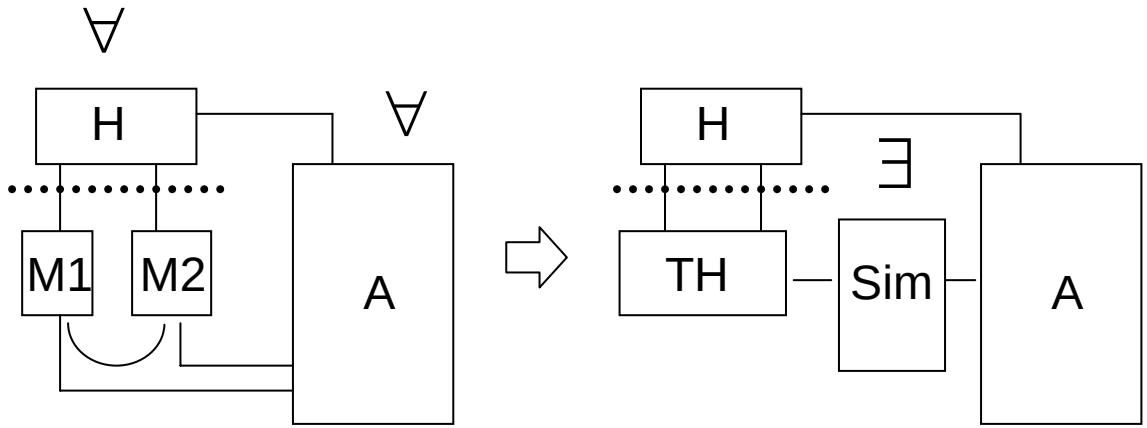
BPW (Backes-Pfitzmann-Waidner)

- real & ideal

- computational
  & symbolic

- (black box) simulation

- composition



(no specific model for M, A, TH)

Mitchell et al

protocol composition logic

- modal formula $\theta\,[P]_X\,\phi$:

  P : protocol (seq of action), θ , ϕ : formula, X: thread

  - action: Send(T,t), Receive(T,t), Encrypt(T,t,k), ...
  - formula: Possess(T,t), Indist(T,t) ...

- axiom & deduction rule
- semantics

ratio -> probability

[modal formula](T, D, ε) : a set of traces

T. a set of traces, D: distinguisher, ε : tolerance

Others

- Abadi : several topics
- Pointcheval (ENS): automatization of

  Shoup’s sequence of games