Quantitative Modal Transition Systems

Kim Guldstrand Larsen
Aalborg University, DENMARK

The Early Days – Edinburgh 83-85

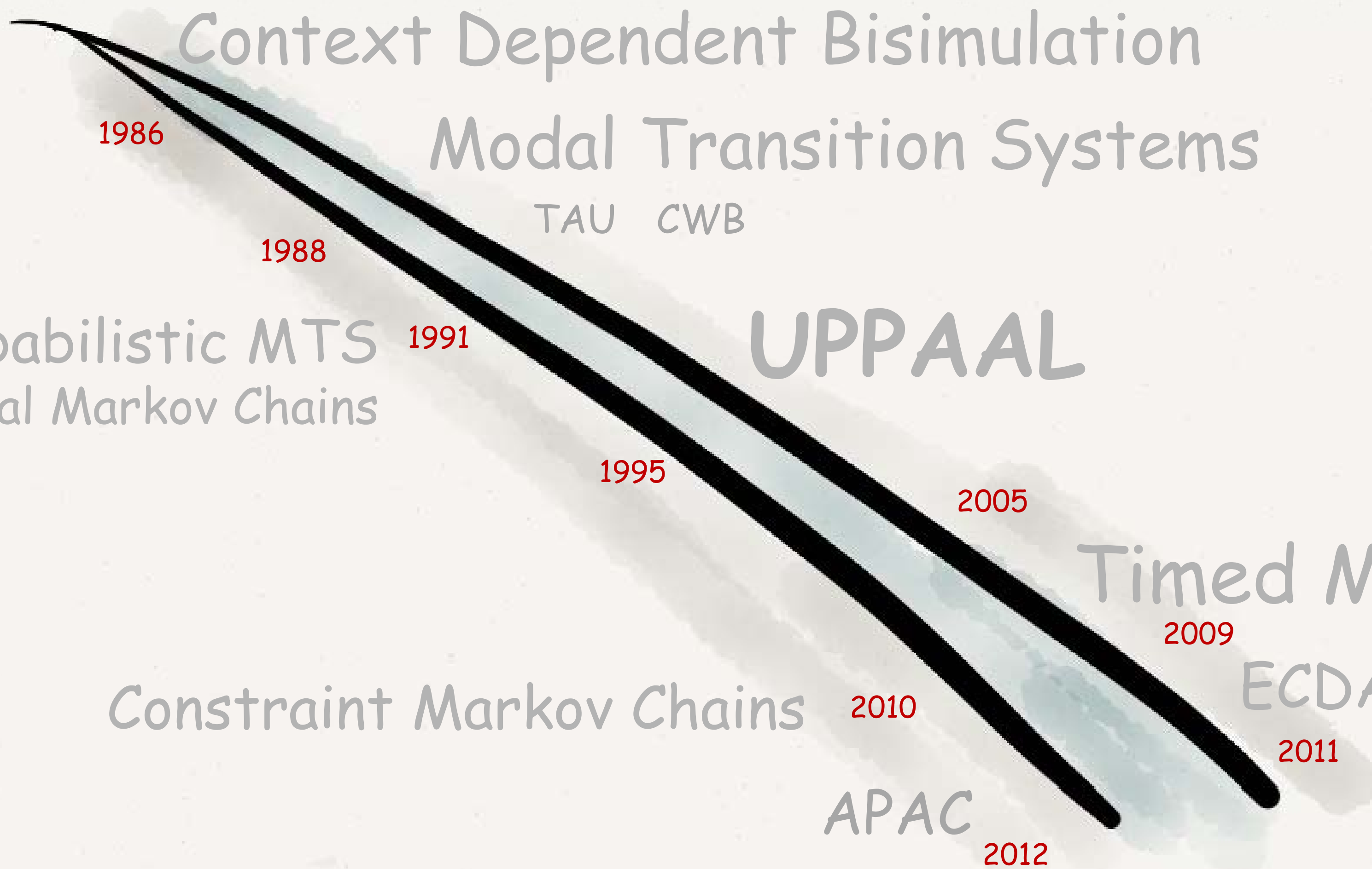
Original Aim

- Need for sound compositional specification formalisms supporting step-wise development and design of concurrent systems

- Components are specified in a formal way at a certain abstraction level.
- Specifications are gradually refined until a concrete system is produced.
- If the refinement steps preserve certain properties, the final system will as well.

- **STILL HIGHLY RELEVANT !**

Bisimulation



Bisimulation

[Park according to Milner]

- $R \subseteq Pr \times Pr$ is a (strong) bisimulation iff whenever $(P,Q) \in R$ then
  - i) whenever P-a-> P' then Q-a->Q'
    for some Q' with $(P',Q') \in R$
  - ii) whenever Q-a-> Q' then P-a->P'
    for some P' with $(P',Q') \in R$

- $P \sim Q$ iff $(P,Q) \in R$ for some bisimulation R

- $\sim$ is a congruence relation

Compositionality

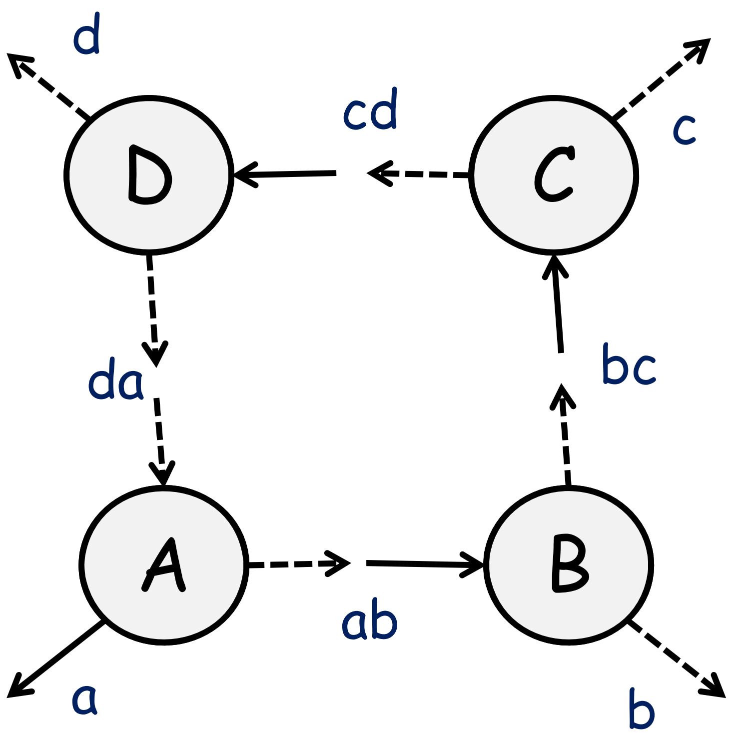
- Properties of a combined program should be obtained from properties of component!

- Correctness problem: $SYS \sim SPEC$

- Compositional Verification
  1. Decompose: $SYS = C[SYS_1, \ldots, SYS_n]$
  2. Verify: $SYS_i \sim SPEC_i$
  3. Combine: $SPEC \sim C[SPEC_1, \ldots, SPEC_n]$

- Problem: how to obtain simple subspecification?

A Simple Scheduler

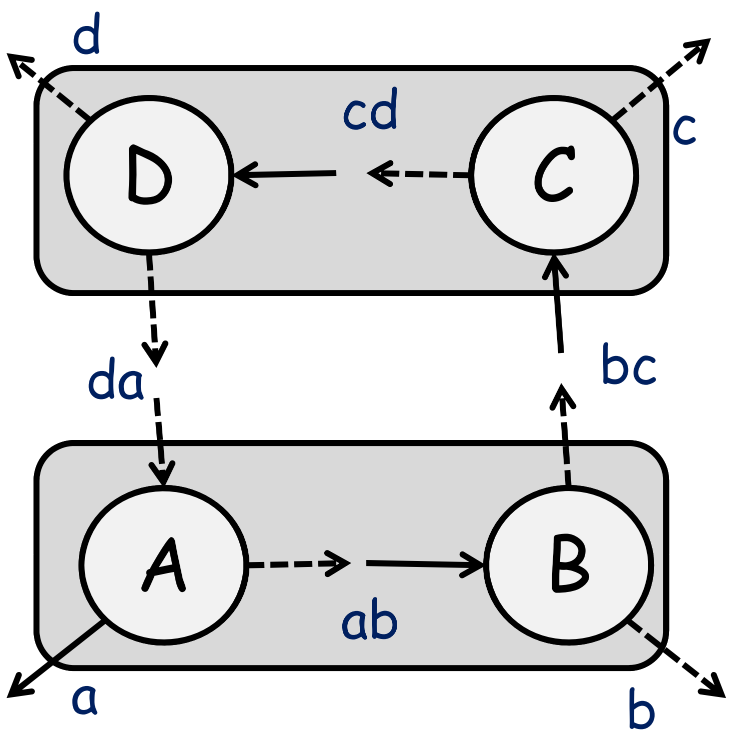


- A = a! ab! da? A
- B = ab? b! bc! B
- C = bc? c! cd! C
- D = cd? d! da! D

- SPEC =
  a! τ b! τ c! τ d! τ SPEC

- (A | B | C | D) ~ SPEC

Compositional Verification



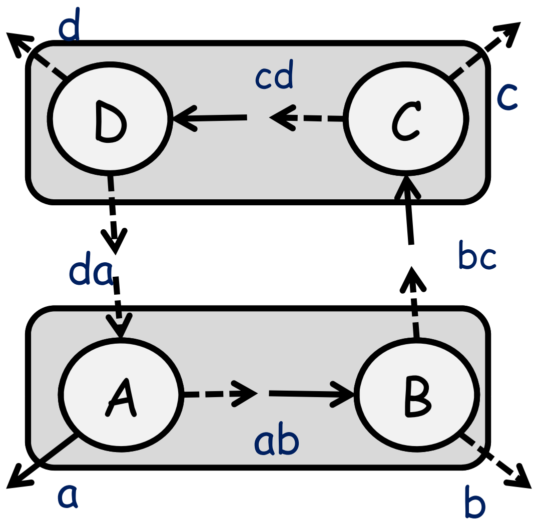
- $A = a!\ ab!\ da?\ A$
- $B = ab?\ b!\ bc!\ B$
- $C = bc?\ c!\ cd!\ C$
- $D = cd?\ d!\ da!\ D$
- SPEC =
  $a!\ \tau\ b!\ \tau\ c!\ \tau\ d!\ \tau\ .\ SPEC$

- $SYS_1 = D \mid C$
- $SYS_2 = A \mid B$

- $SPEC_1 = bc?\ c!\ \tau\ d!\ da!\ SPEC_1$
- $SPEC_2 = a!\ \tau\ b!\ bc!\ da?\ SPEC_2$

- However $SYS_i \not\approx SPEC_i$

Compositional Verification



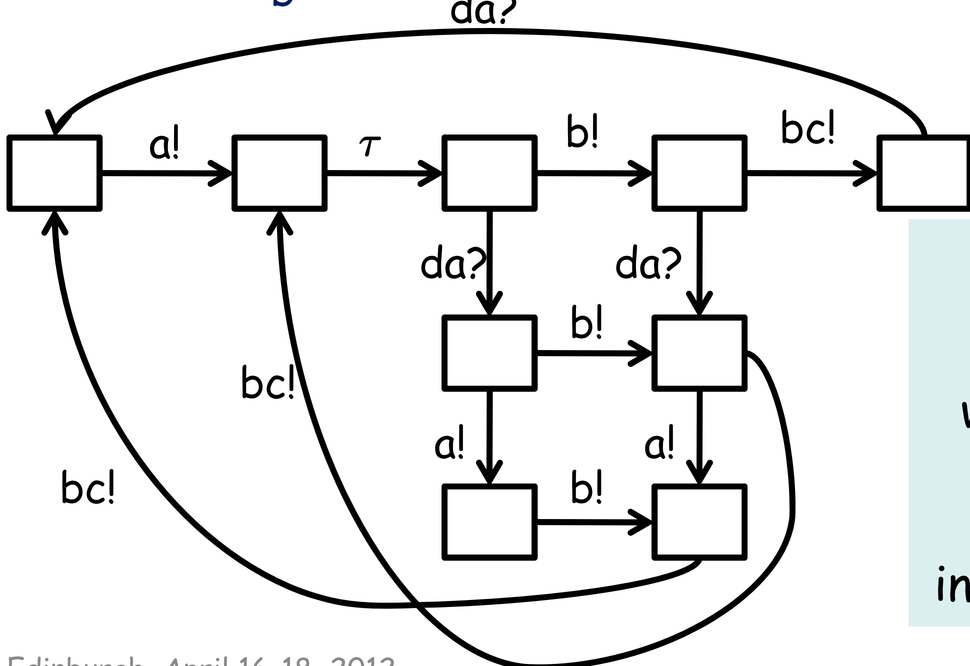
- $SYS_1 = D \mid C$
- $SYS_2 = A \mid B$

- $SPEC_1 = bc?\ c!\ \tau\ d!\ da!\ SPEC_1$
- $SPEC_2 = a!\ \tau\ b!\ bc!\ da?\ SPEC_2$

Clearly **$SYS_2 \not\sim SPEC_2$**

In fact no hope for a simple **$SPEC_2$**

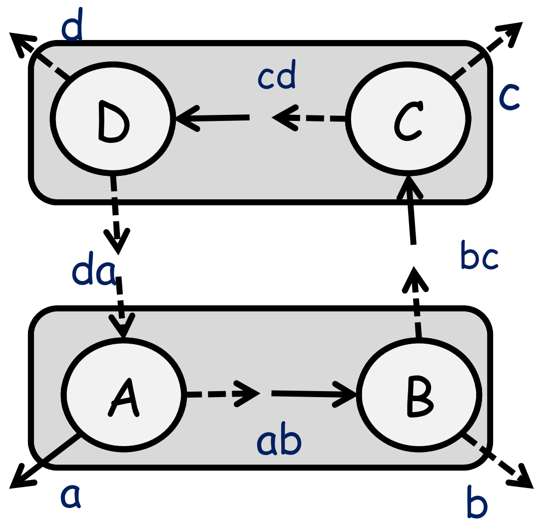
A | B



However **$SYS_2 \sim_{\mathcal{E}} SPEC_2$**

where $\mathcal{E}$ is an environment capturing behaviour relevant in the context **([] | C | D)**

Compositional Verification



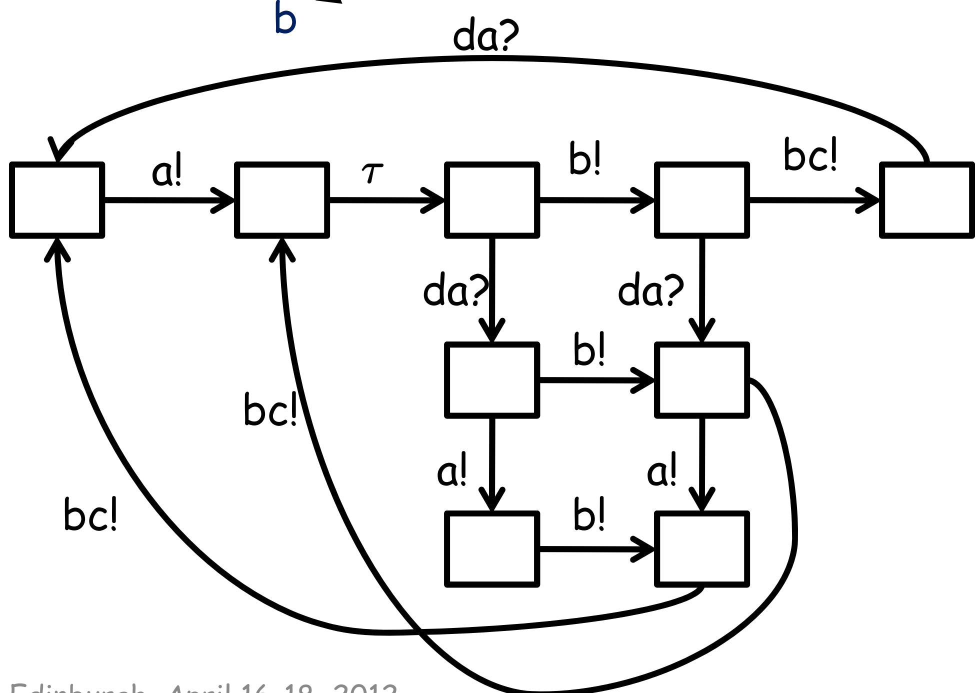
- SYS_1 = D | C
- SYS_2 = A | B

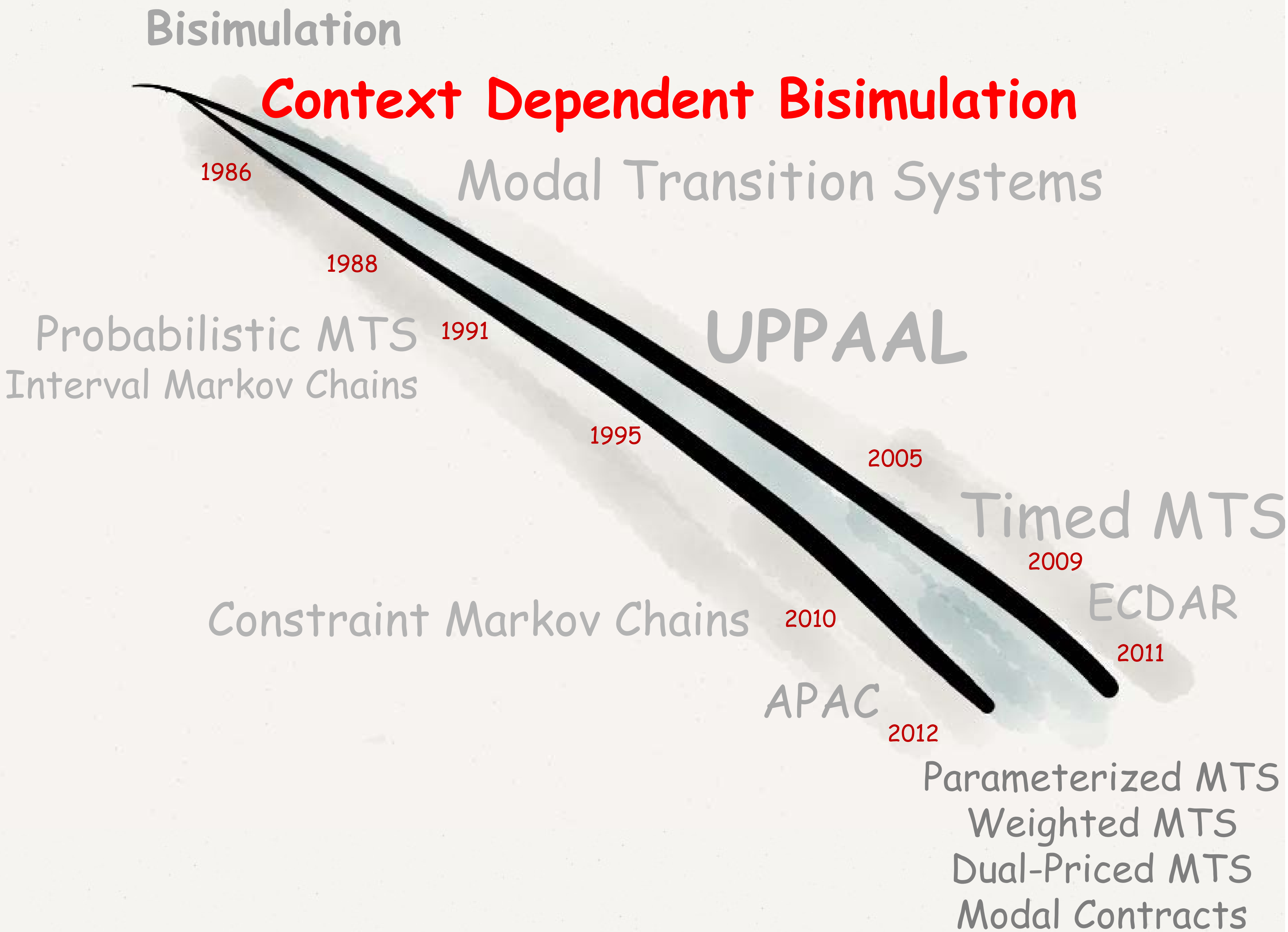
- $SPEC_1$ = bc? c! τ d! da! $SPEC_1$
- $SPEC_2$ = a! τ b! bc! da? $SPEC_2$

Clearly **$SYS_2 \not\approx SPEC_2$**

In fact no hope for a simple **$SPEC_2$**

A | B



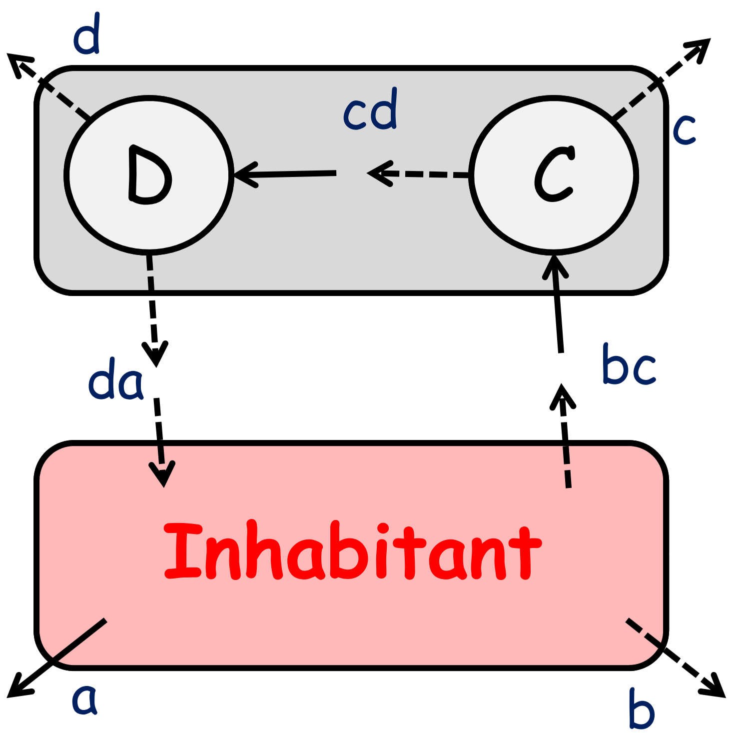

Bisimulation
Context Dependent Bisimulation
1986
Modal Transition Systems
1988
Probabilistic MTS
Interval Markov Chains
1991
UPPAAL
1995
2005
Timed MTS
2009
ECDAR
Constraint Markov Chains
2010
2011
APAC
2012
Parameterized MTS
Weighted MTS
Dual-Priced MTS
Modal Contracts

Environments

- **E** = (Env , Act, →) $P \sim_E Q$

- E –a-> E' :
  - E allows (can consume) the action a and become E'
- P -a-> P' :
  - P can produce the action a and become P'
- **Special Environments**
  - **O** : ¬ (O –a->) for all actions a.
    Thus we expect $P \sim_O Q$ for all P and Q
  - **U** : U –a-> U for any action a.
    Thus we expect $P \sim_U Q$ iff $P \sim Q$.

Environment

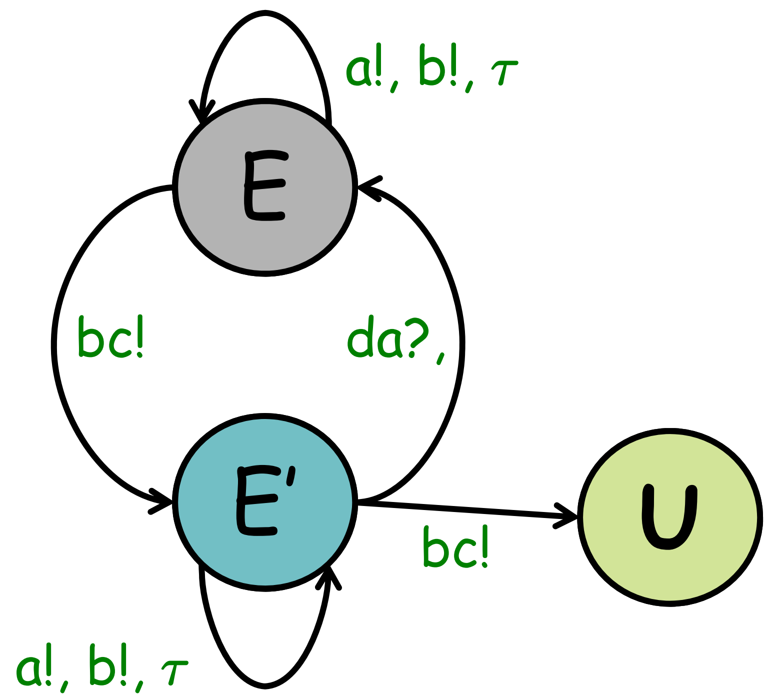


Environment should cover the behaviour allowed by the context

([] | C | D) ???

Only a!, b!, da?, bc!, τ

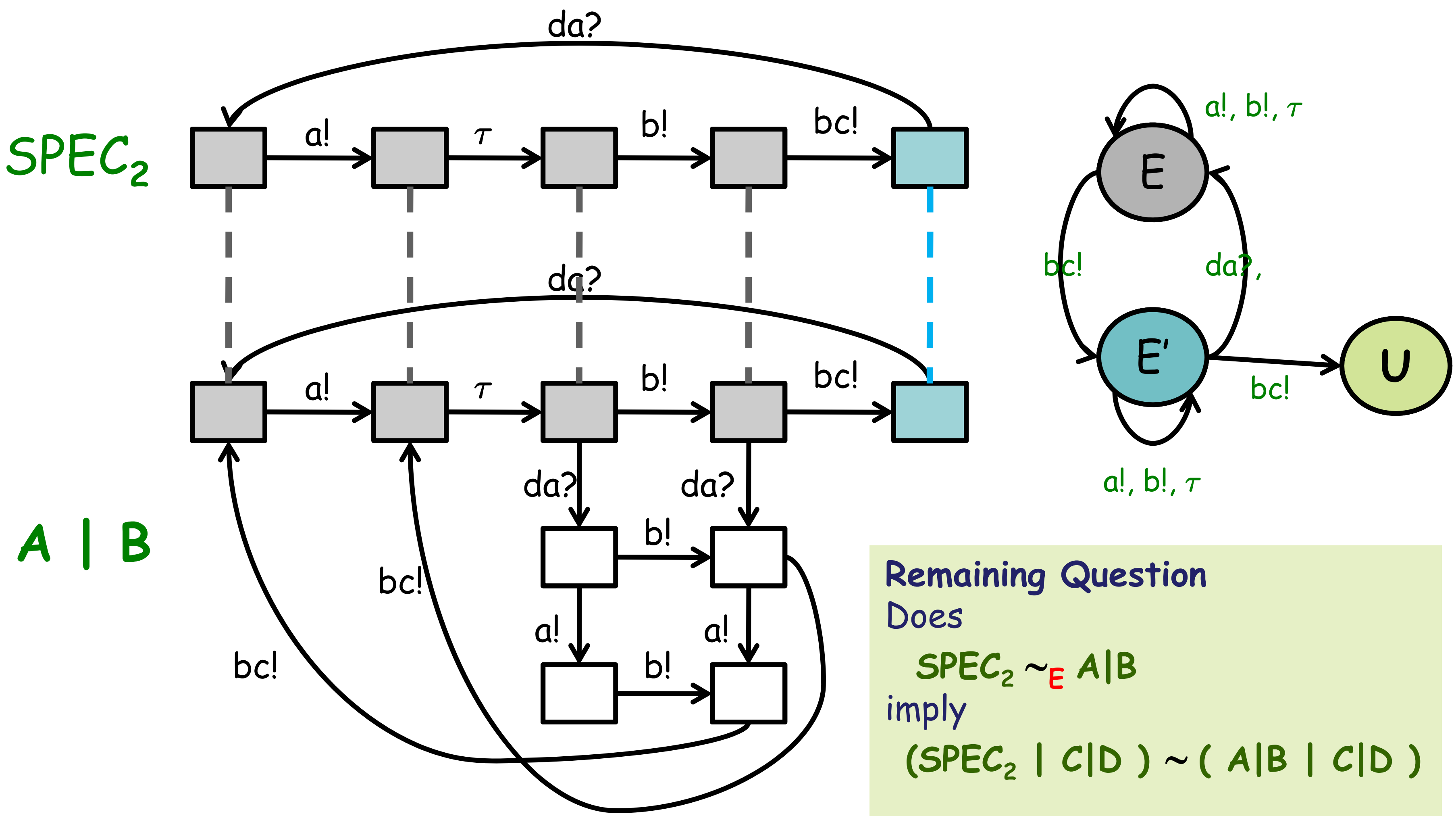
No restrictions on a!, b!, τ



Parameterized Bisimulation

- Let **E** =(Env , Act, →).
- An **E**-parameterized bisimulation is an Env-indexed family **R** = { R_E : $E \in$ Env } with $R_E \subseteq Pr \times Pr$, such that whenever whenever $(P,Q) \in R_E$ and E-a->E' then
  - i) whenever P-a->P' then Q-a->Q' for some Q' with $(P',Q') \in R_{E'}$
  - ii) whenever Q-a->Q' then P-a->P' for some P' with $(P',Q') \in R_{E'}$
- $P \sim_E Q$, whenever $(P,Q) \in R_E$ for some parameterized bisimulation **R**.

Compositional Verification – Revisited



Remaining Question

Does

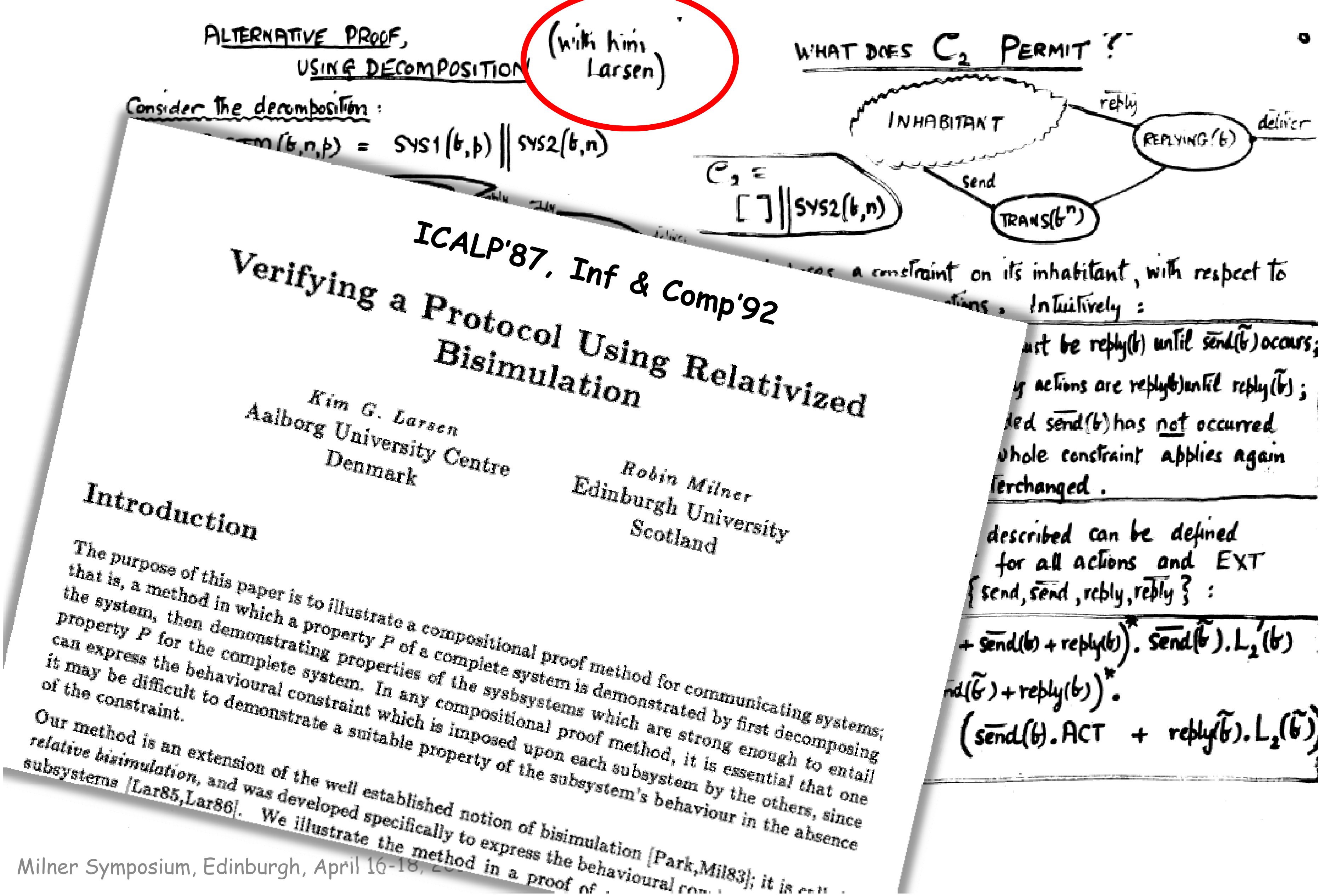
$SPEC_2 \sim_E A|B$

imply

$(SPEC_2 \mid C|D) \sim (A|B \mid C|D)$

Semantics of **contexts** as action transducer!

The Alternating Bit Protocol



Verifying a Protocol Using Relativized Bisimulation

Kim G. Larsen
Aalborg University Centre
Denmark

Robin Milner
Edinburgh University
Scotland

Introduction

The purpose of this paper is to illustrate a compositional proof method for communicating systems; that is, a method in which a property P of a complete system is demonstrated by first decomposing the system, then demonstrating properties of the sysbsystems which are strong enough to entail property P for the complete system. In any compositional proof method, it is essential that one can express the behavioural constraint which is imposed upon each subsystem by the others, since it may be difficult to demonstrate a suitable property of the subsystem's behaviour in the absence of the constraint.

Our method is an extension of the well established notion of bisimulation [Park,Mil83]; it is call... *relative bisimulation*, and was developed specifically to express the behavioural con... subsystems [Lar85,Lar86]. We illustrate the method in a proof of ...

ABP in the TAU Tool → CWB

```
acked0 :=:  in(ack0);acked0 + in(ack1);acked0 + acc;sending1.
acked1 :=:  in(ack1);acked1 + in(ack0);acked1 + acc;sending0.
sending0 :=:  out(send0);sending0 + in(ack1);sending0
           + in(ack0);acked0.
sending1 :=:  out(send1);sending1 + in(ack0);sending1
           + in(ack1);acked1.

transmitted0 :=:  in(transmit0);transmitted0 + in(transmit1);
                transmitted0 + del;replying0.
transmitted1 :=:  in(transmit1);transmitted1 + in(transmit0);
                transmitted1 + del;replying1.
replying0 :=:  out(reply0);replying0 + in(transmit0);
            replying0 + in(transmit1);transmitted1.
replying1 :=:  out(reply1);replying1 + in(transmit1);
            replying1 + in(transmit0);transmitted0.
```

```
protocol :=: (acked0 / trans_med / ack_med / replying0)
              \\[acc,del].
spec :=: acc;del;spec.
```

ABP in the TAU Tool → CWB

Tatsuya Hagino
Professor,
Faculty of
Environmental Information,
Keio University, Japan

```
bisim(P,Q,C) :- closure(P,Q,[[P,Q]],C).

closure(P,Q,B,D) :-
        matchl(P,Q,B,C), matchr(P,Q,C,D).

matchl(P,Q,B,C) :-
        derset(P,M), matchl+(P,Q,M,B,C).

matchr(P,Q,C,D) :-
        derset(Q,N), matchr+(P,Q,N,C,D).

matchl+(P,Q,[],B,B).
matchl+(P,Q,[[A,P']|M],B,D) :-
        der(Q,A,Q'), in([P',Q'],B), !,
        matchl+(P,Q,M,B,D).
matchl+(P,Q,[[A,P']|M],B,D) :-
        der(Q,A,Q'), closure(P',Q',[[P',Q']|B],C),
        matchl+(P,Q,M,C,D).

matchr+(P,Q,[],B,B).
matchr+(P,Q,[[A,Q']|N],B,D) :-
        der(P,A,P'), in([P',Q'],B), !,
        matchr+(P,Q,N,B,D).
matchr+(P,Q,[[A,Q']|N],B,D) :-
        der(P,A,P'), closure(P',Q',[[P',Q']|B],C),
        matchr+(P,Q,N,C,D).
```

(figure 6.2-5)

Bisimulation

Context Dependent Bisimulation

1986

Modal Transition Systems

TAU CWB

1988

Probabilistic MTS

Interval Markov Chains

1991

UPPAAL

1995

2005

Timed MTS

2009

ECDAR

Constraint Markov Chains

2010

2011

APAC

2012

Parameterized MTS

Weighted MTS

Dual-Priced MTS

Modal Contracts

Operations on Specifications

- **Structural Composition:**
  - Given S_1 and S_2 construct S_1 **par** S_2 such that
    $$| S_1 \textbf{ par } S_2 | = |S_1| \textbf{ par } |S_2|$$
  - $\leq$ should be precongruence wrt **par** to allow for compositional analysis !

- **Logical Conjunction:**
  - Given S_1 and S_2 construct $S_1 \wedge S_2$ such that
    $$|S_1 \wedge S_2| = |S_1| \cap |S_2|$$

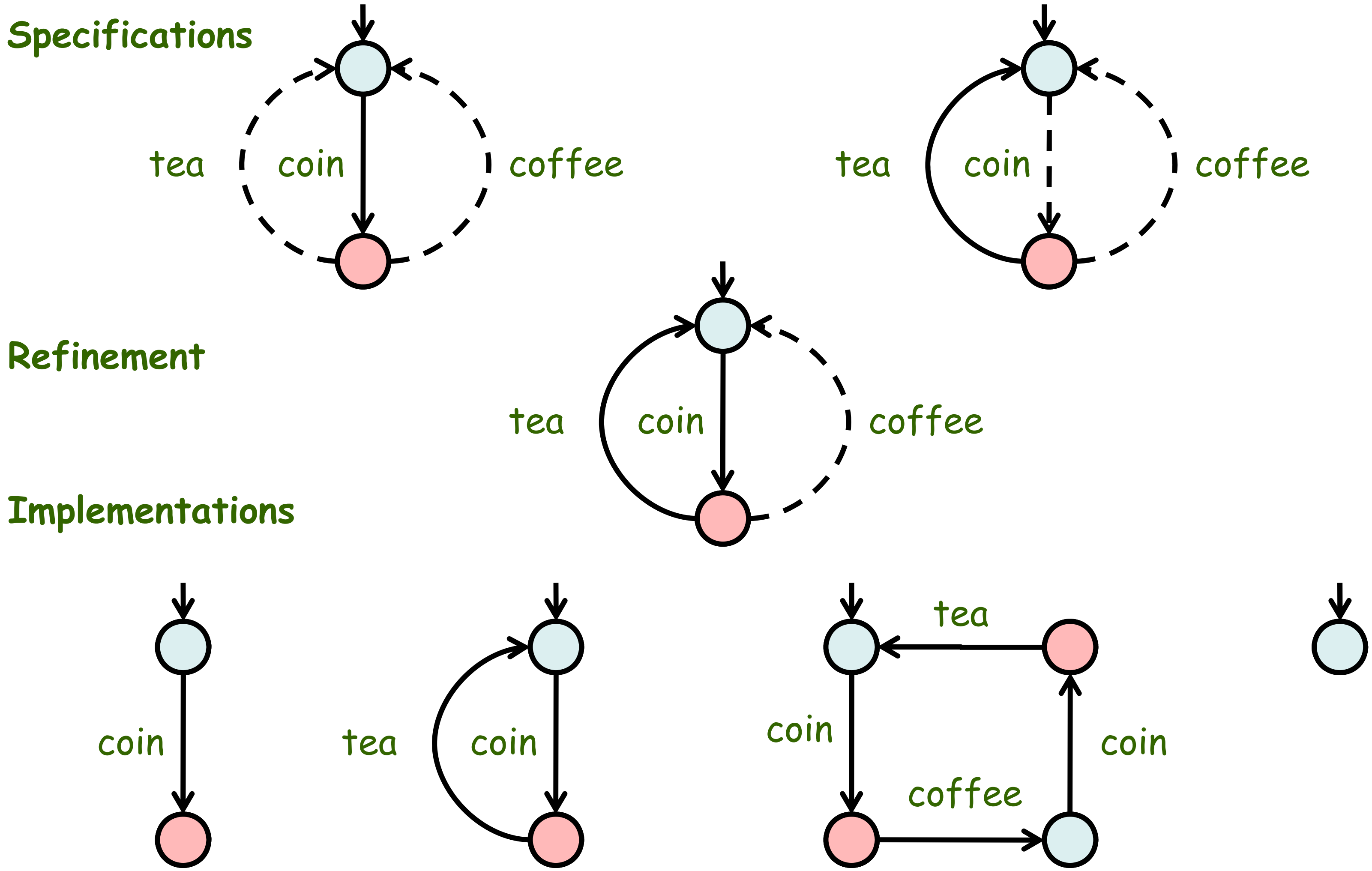
- **Quotienting:**
  - Given overall specification T and component specification S construct the quotient specification T\S such that
    $$S \textbf{ par } X \leq T \quad \text{iff} \quad X \leq T \backslash S$$

Modal Transition Systems

[L. & Thomsen 88
Boudol & L. 90]

- MTS is an **automata-based** specification formalism
- MTS allow to express that certain actions **may** or **must** happen in their implementation
- MTS supports **all** the required operations on specifications (conjunction, parallel composition, quotienting).
- **Applications** in component-based software development, interface theories, modal abstractions and program analysis.

Example – Tea-Coffee Machines



MTS Definition

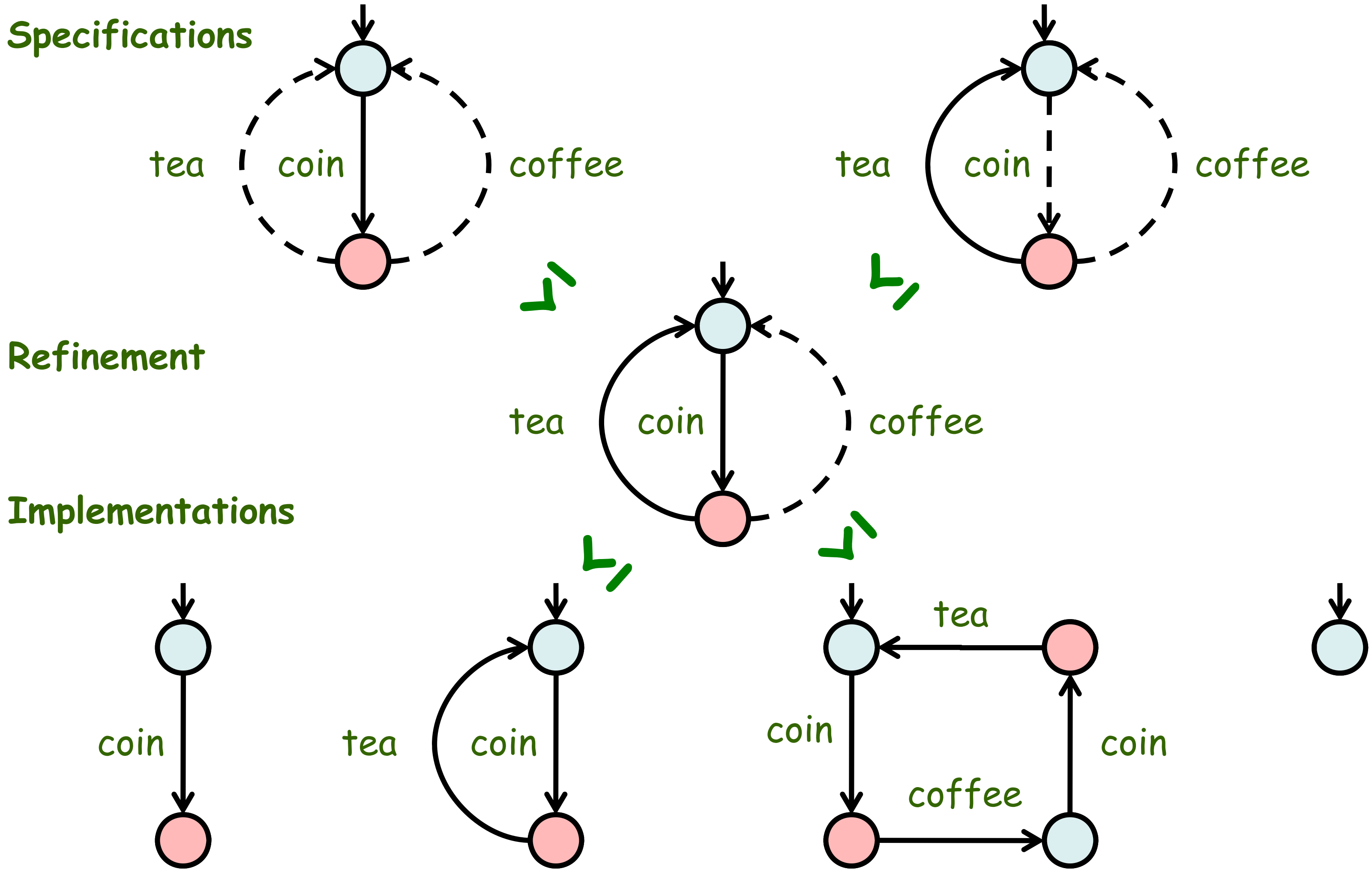
- An MTS is a triple $(P, \rightarrow_{\square}, \rightarrow_{\Diamond})$ where P is a set of states and $\rightarrow_{\square} \subseteq \rightarrow_{\Diamond} \subseteq P \times Act \times P$

If $\rightarrow_{\square} = \rightarrow_{\Diamond}$ then the MTS is an implementation.

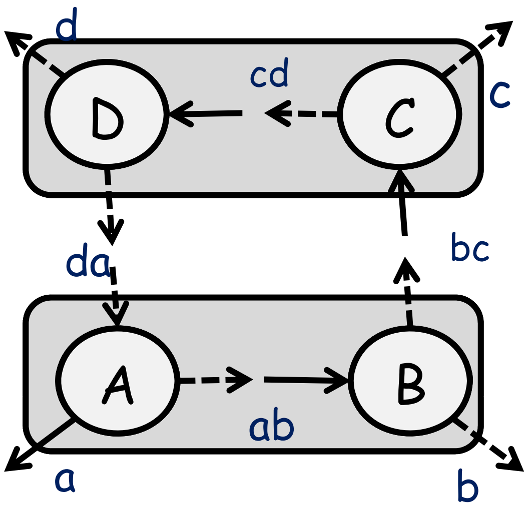
- $R \subseteq P \times P$ is a modal refinement iff whenever $(S,T) \in R$ then
  - i) whenever $S \xrightarrow{a}_{\Diamond} S'$ then $T \xrightarrow{a}_{\Diamond} T'$ for some T' with $(S',T') \in R$
  - ii) whenever $T \xrightarrow{a}_{\square} T'$ then $S \xrightarrow{a}_{\square} S'$ for some S' with $(S',T') \in R$

We write $S \leq_m T$ whenever $(S,T) \in R$ for some modal refinement R.

Example – Tea-Coffee Machines



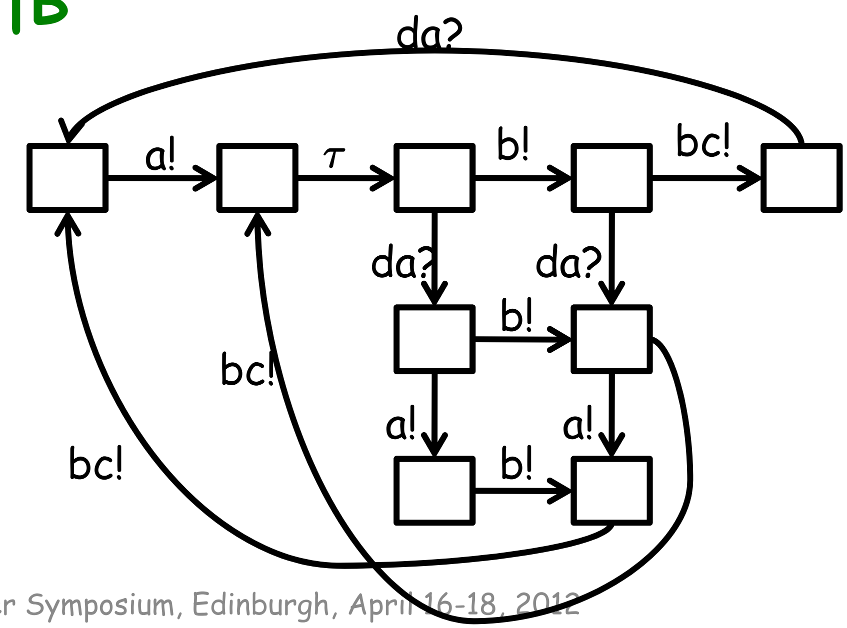
Compositional Verification – Rerevisited



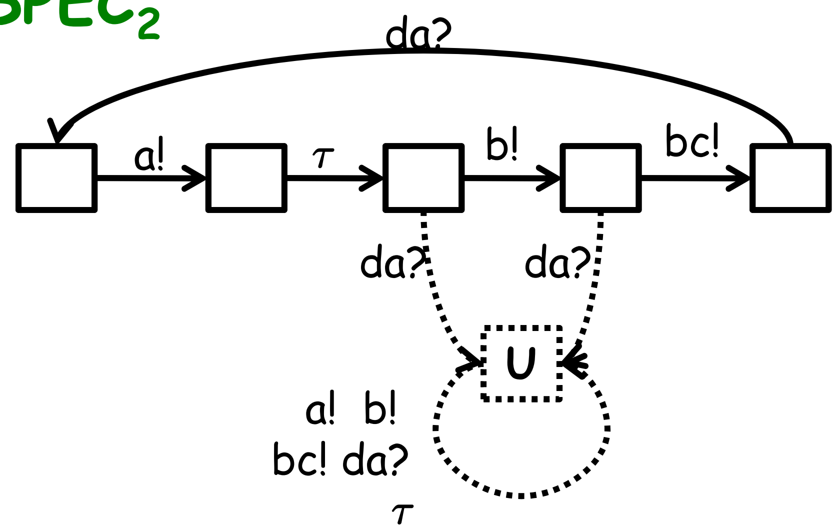
- A = a! ab! da? A
- B = ab? b! bc! B
- C = bc? c! cd! C
- D = cd? d! da! D

- SPEC =
  a! τ b! τ c! τ d! τ SPEC

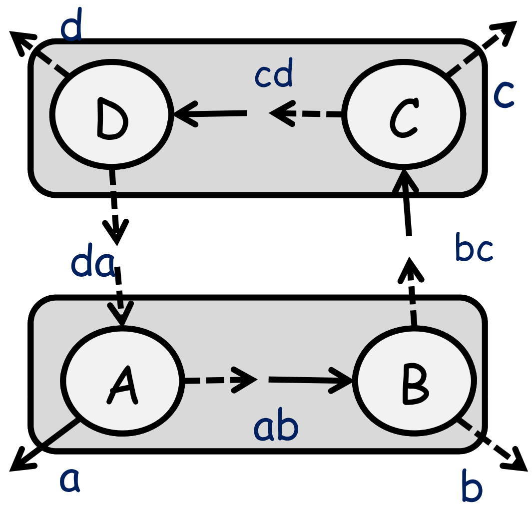
A|B



$SPEC_2$



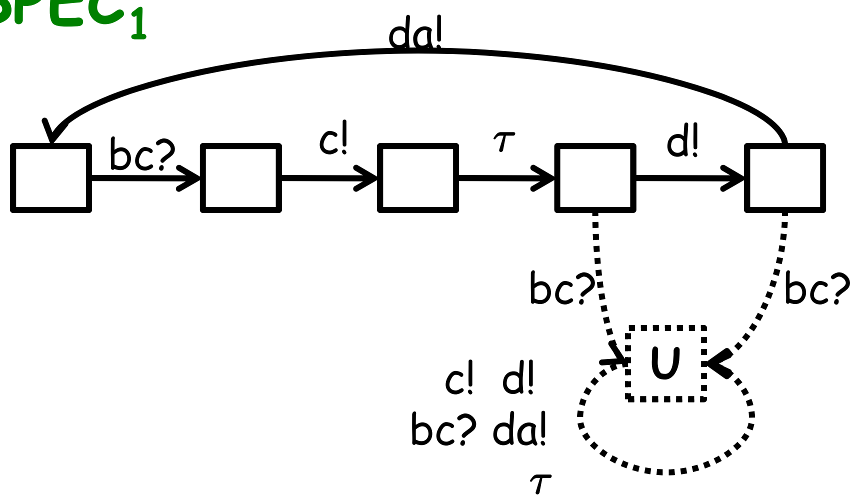
Compositional Verification – Rerevisited



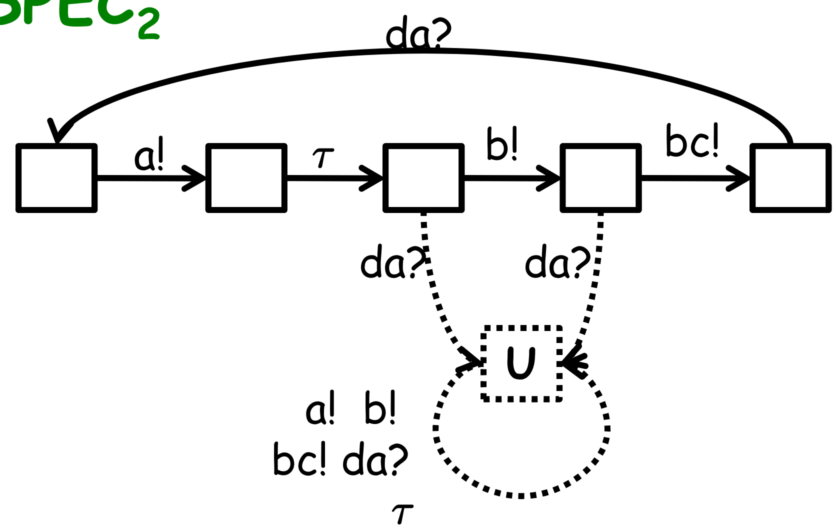
- SPEC = a! τ b! τ c! τ d! τ SPEC

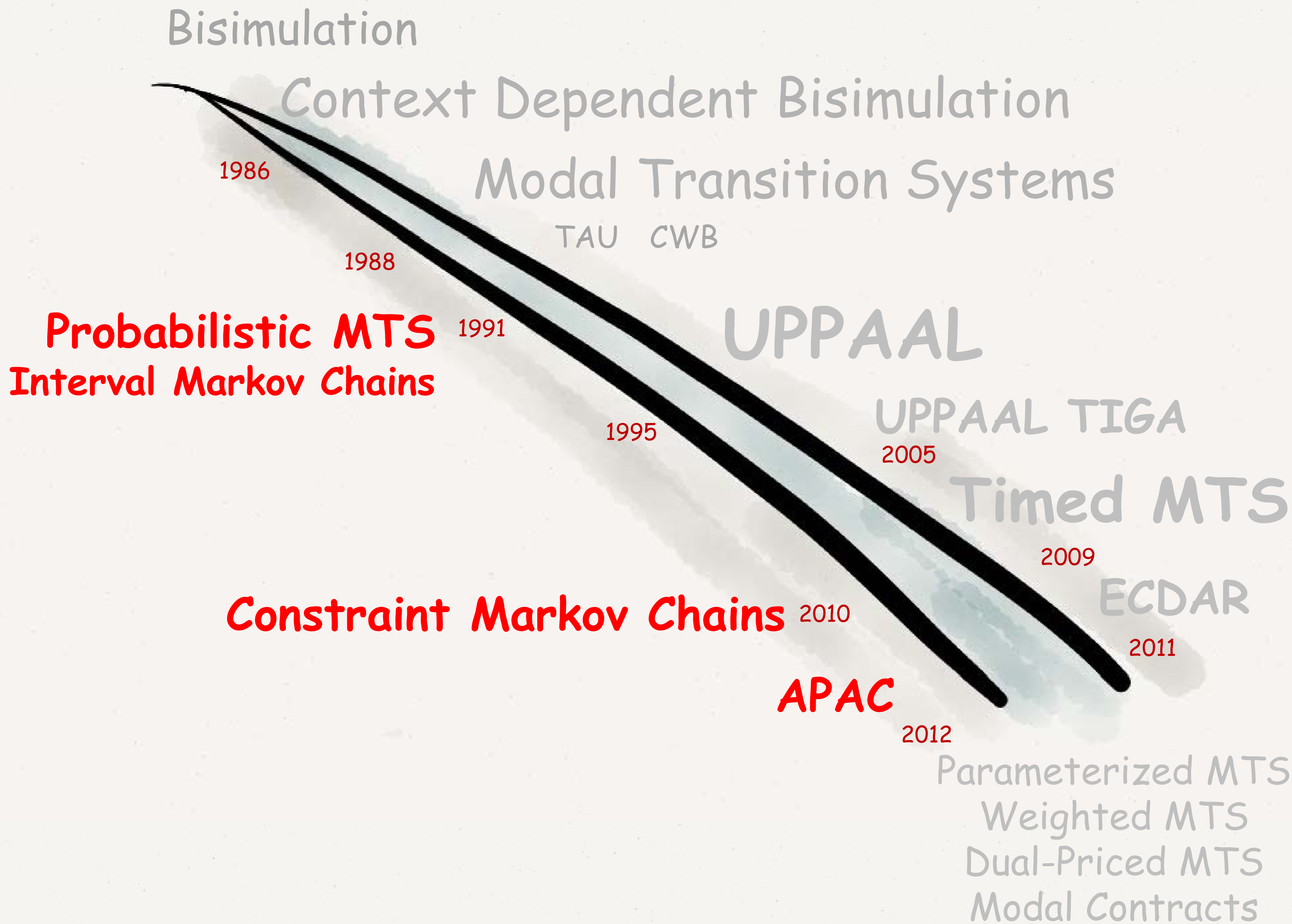
- ($SPEC_1$ || $SPEC_2$) $\leq_m$ SPEC
- C | D $\leq_m$ $SPEC_1$
- A | B $\leq_m$ $SPEC_2$
- Hence **(A | B | C | D) $\leq_m$ SPEC**

$SPEC_1$



$SPEC_2$




Bisimulation
Context Dependent Bisimulation
Modal Transition Systems
TAU CWB
1986
1988
Probabilistic MTS
Interval Markov Chains
1991
UPPAAL
1995
UPPAAL TIGA
2005
Timed MTS
2009
ECDAR
Constraint Markov Chains
2010
2011
APAC
2012
Parameterized MTS
Weighted MTS
Dual-Priced MTS
Modal Contracts

Probabilistic Process System

$\sim$ Markov Chain

$$(P, A, \Pi, V)$$

$V: P \to 2^A$ Valuation function

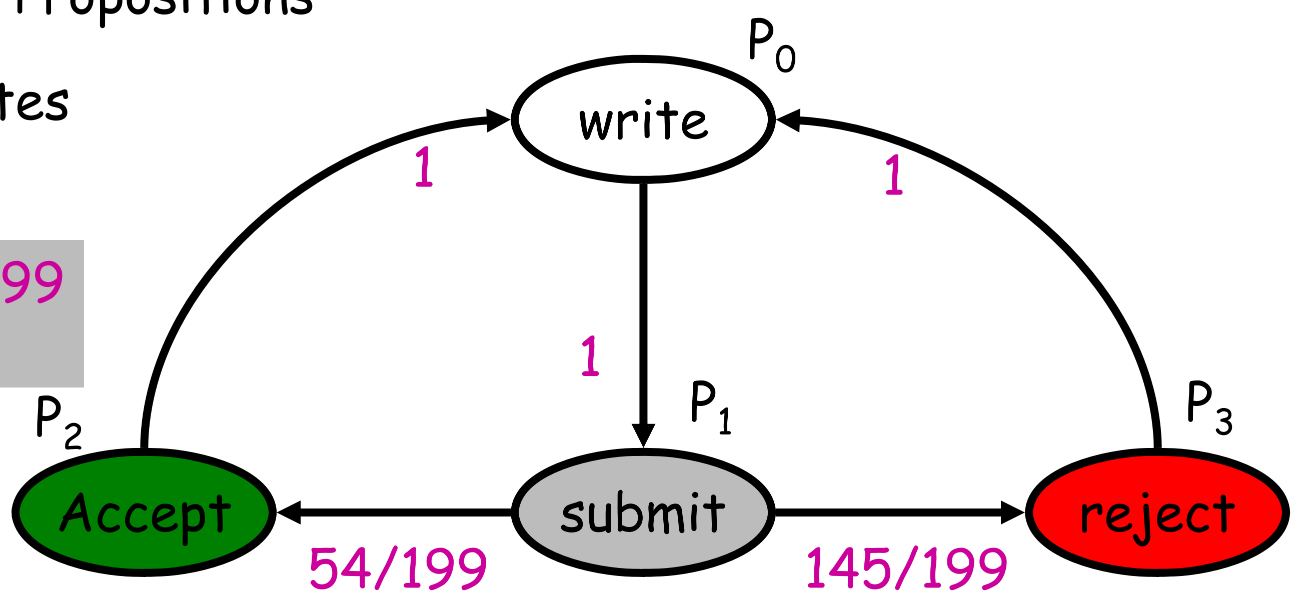
$\Pi : P \to (P \to [0,1])$ Transition probability function

Atomic Propositions

Processes / States

$\Pi\,(P_1)(P_2) = 54/199$

$\Pi\,(P_0)(P_2) = 0$



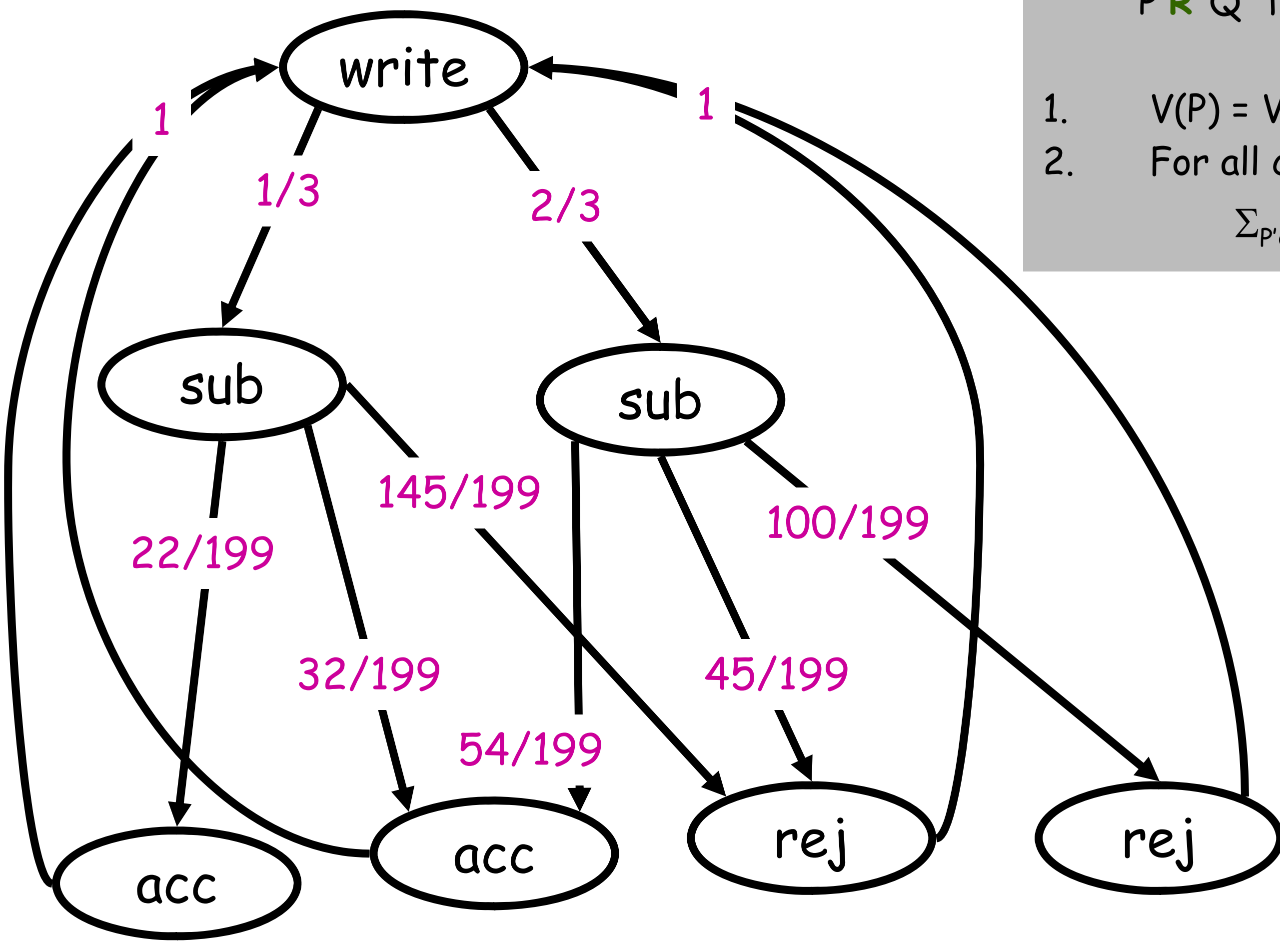
Probabilistic Bisimulation [L., Skou ' 89]

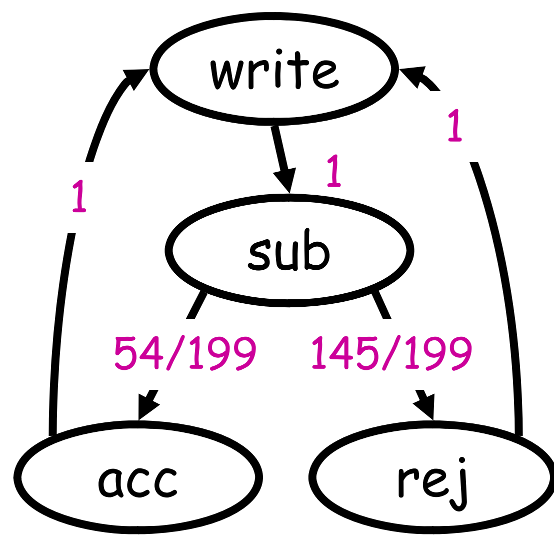
Definition

An equivalence relation **R** on process is a probabilistic bisimulation if whenever P **R** Q then

1. V(P) = V(Q)
2. For all classes $C \in P/R$

$$\Sigma_{P' \in C} \Pi(P)(P') = \Sigma_{P' \in C} \Pi(Q)(P')$$





Probabilistic Bisimulation

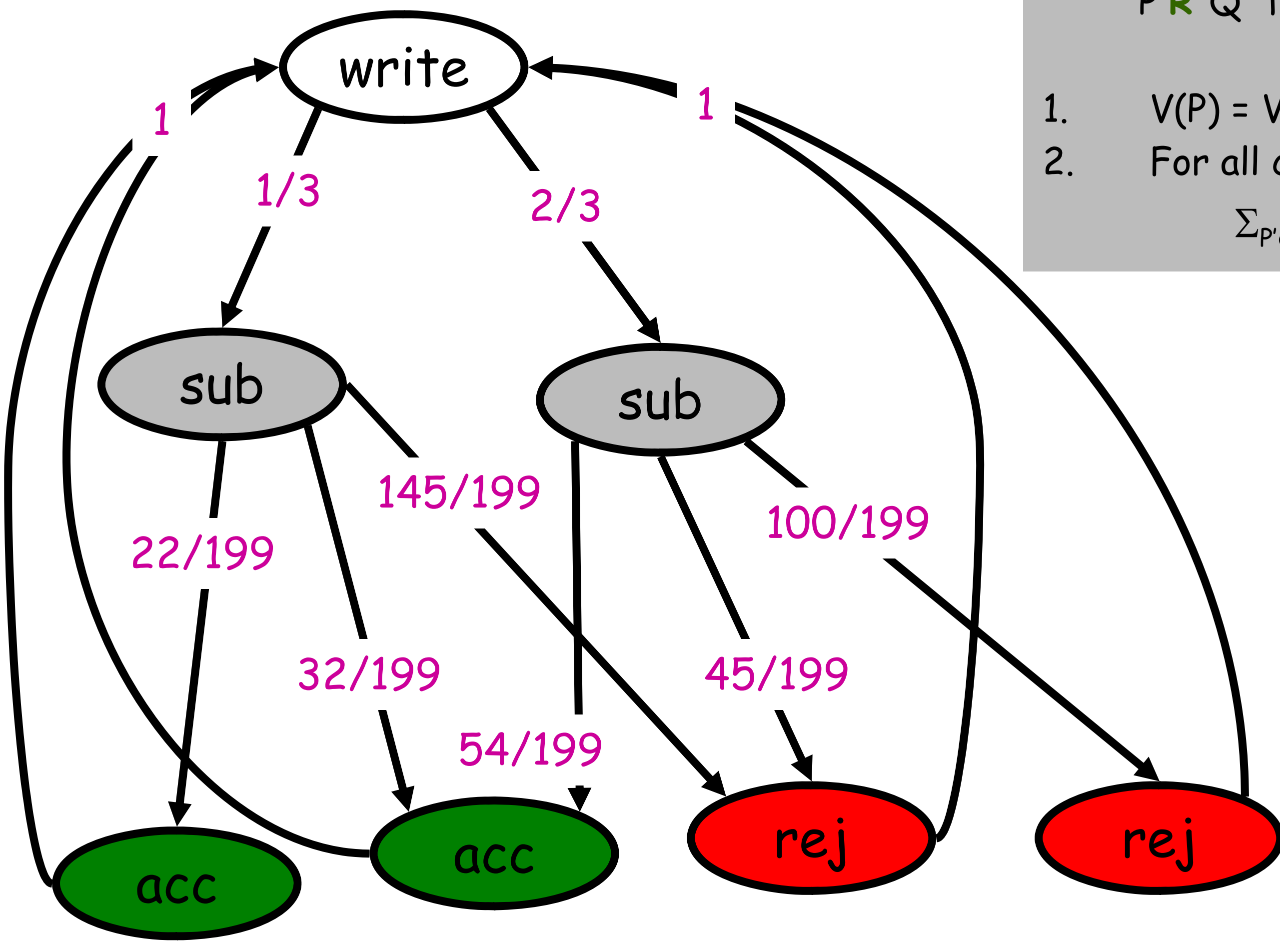
[L., Skou ' 89]

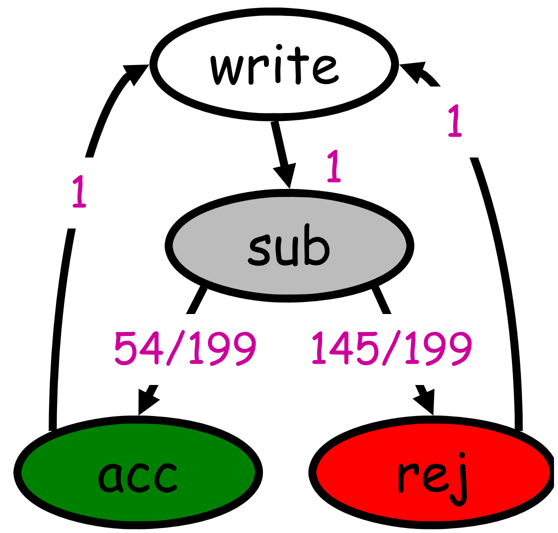
Definition

An equivalence relation **R** on process is a probabilistic bisimulation if whenever P **R** Q then

1. V(P) = V(Q)
2. For all classes $C \in P/R$

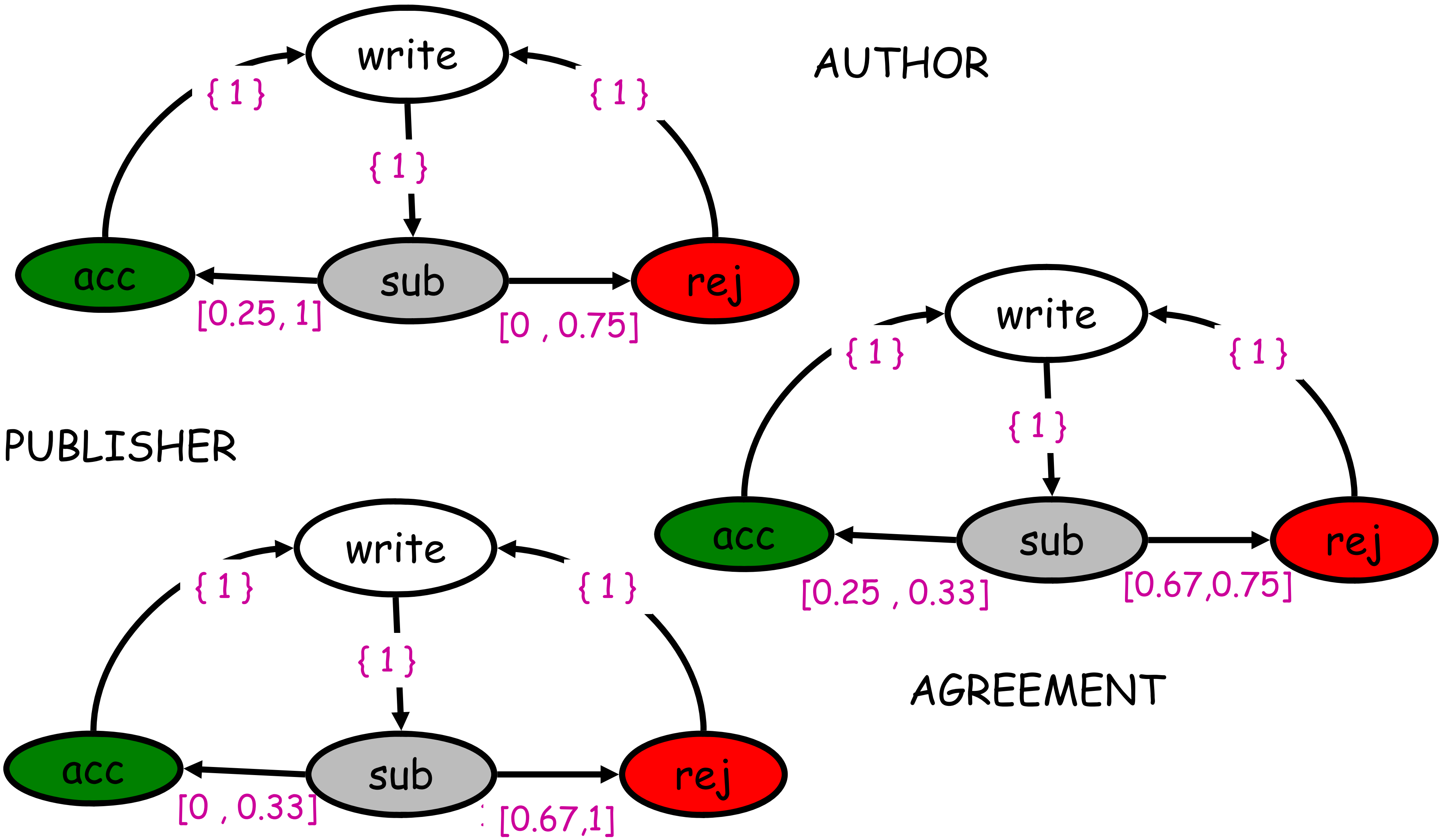
$$\Sigma_{P' \in C}\, \Pi(P)(P') = \Sigma_{P' \in C}\, \Pi(Q)(P')$$



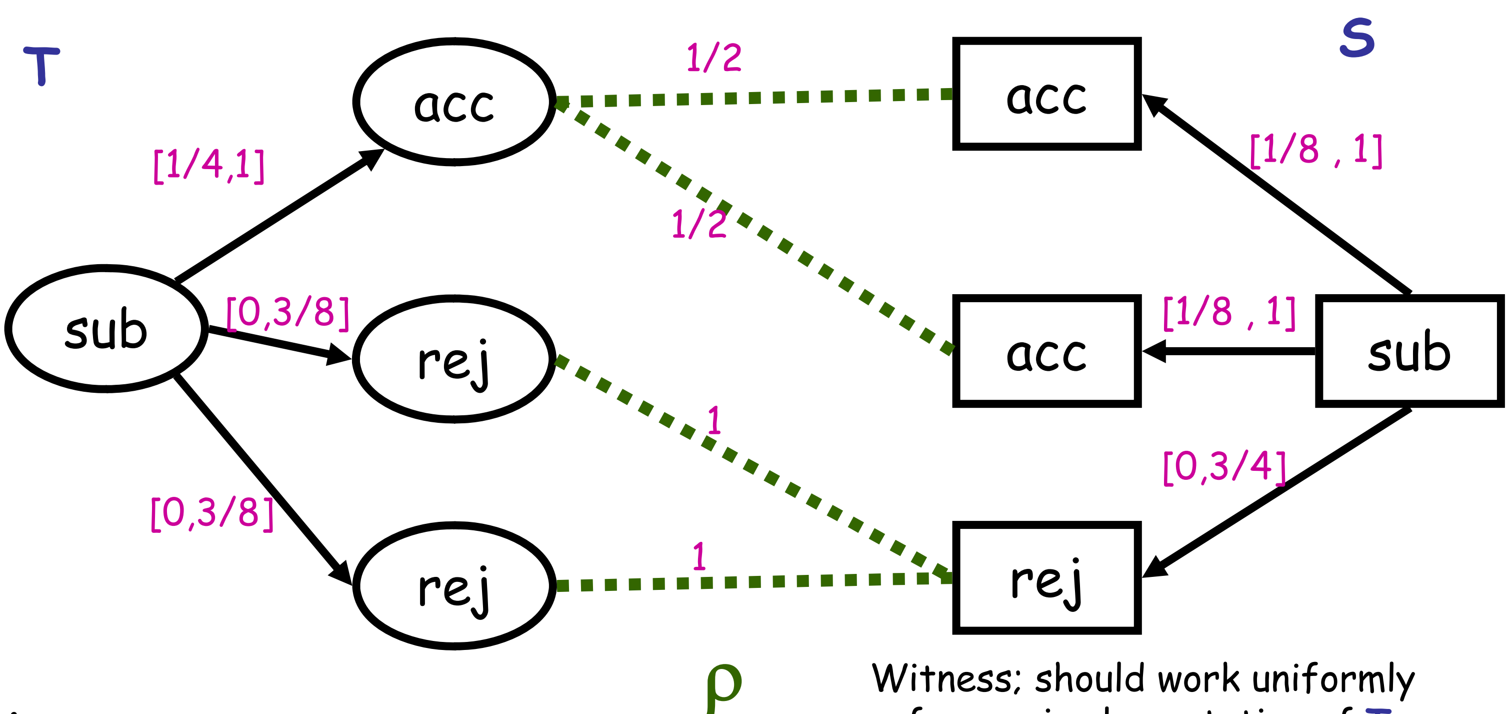


Probabilistic MTS

[Jonsson, L.' 91]



Probabilistic Refinement (Informally)

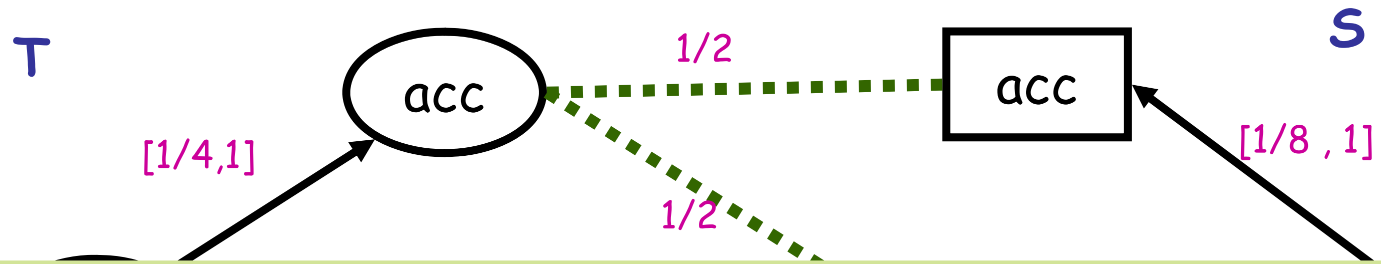


Witness; should work uniformly for any implementation of T

Show

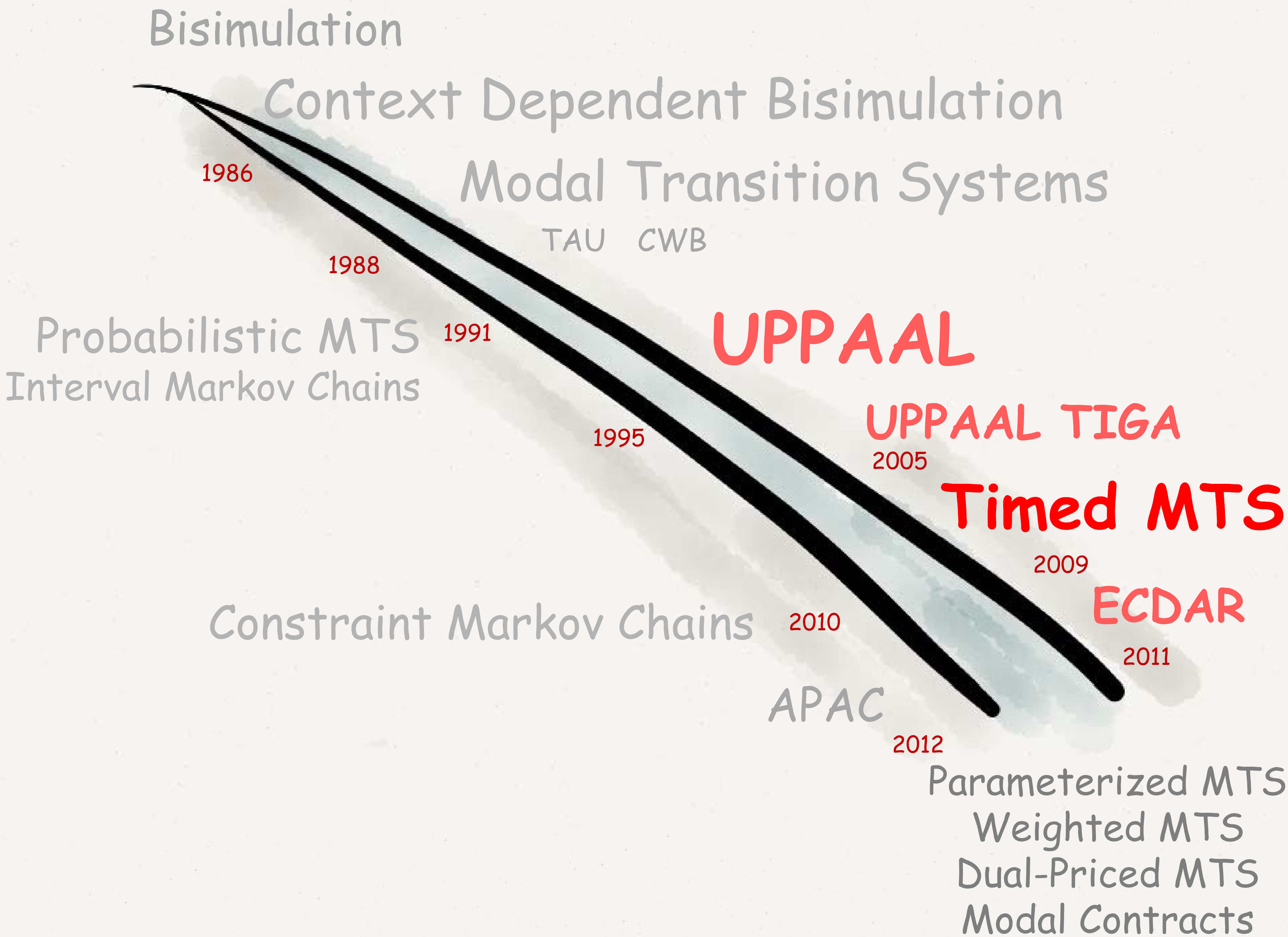
$$\left.\begin{array}{l} p_1 + p_2 + p_3 = 1 \\ p_1 \in [1/4,1] \\ p_2 \in [0,3/8] \\ p_3 \in [0,3/8] \end{array}\right\} \Rightarrow \left\{\begin{array}{l} \frac{1}{2} * p_1 \in [1/8,1] \\ \frac{1}{2} * p_1 \in [1/8,1] \\ 1*p_2 + 1*p_3 \in [0,3/4] \end{array}\right.$$

Probabilistic Refinement (Informally)

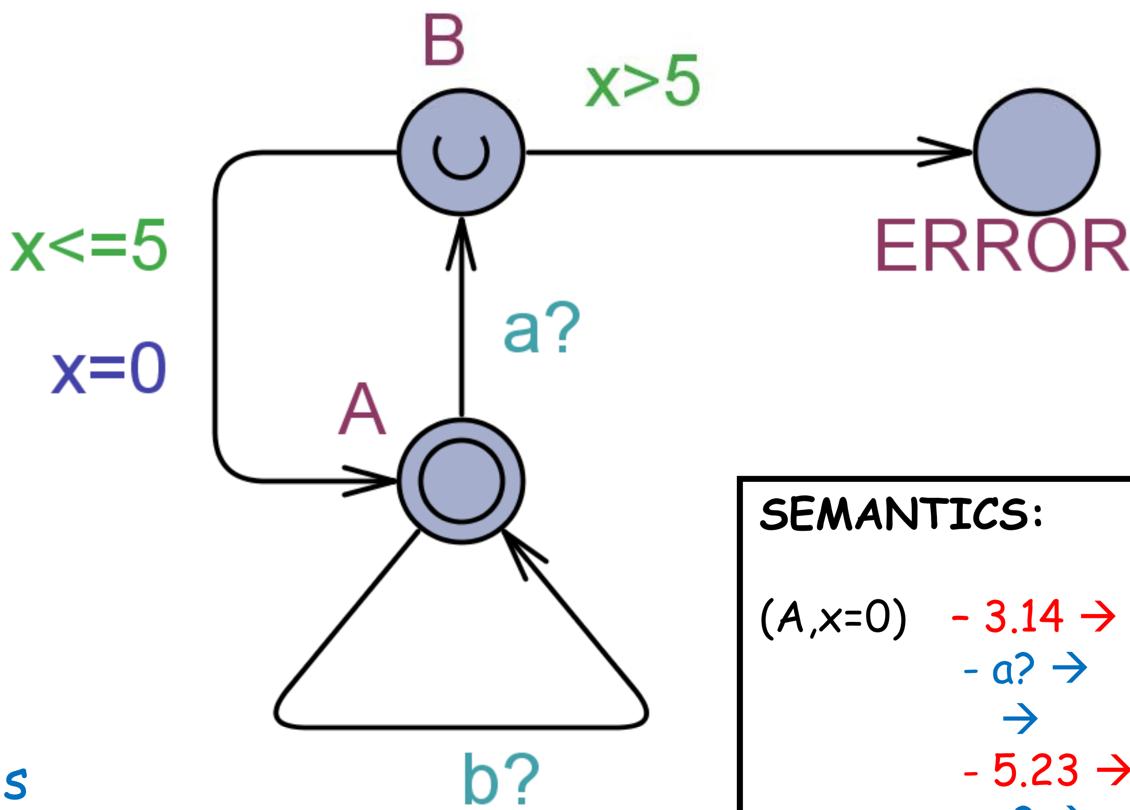


Constraint Markov Chains
to ensure closure under
conjunction and parallel compositions
[2010]

$$\left.\begin{array}{l} p_1 + p_2 + p_3 = 1 \\ p_1 \in [1/4,1] \\ p_2 \in [0,3/8] \\ p_3 \in [0,3/8] \end{array}\right\} \Rightarrow \left\{\begin{array}{l} \tfrac{1}{2} * p_1 \in [1/8,1] \\ \tfrac{1}{2} * p_1 \in [1/8,1] \\ 1{*}p_2 + 1{*}p_3 \in [0,3/4] \end{array}\right.$$


Bisimulation
Context Dependent Bisimulation
1986
Modal Transition Systems
TAU CWB
1988
Probabilistic MTS
Interval Markov Chains
1991
UPPAAL
1995
UPPAAL TIGA
2005
Timed MTS
2009
ECDAR
Constraint Markov Chains
2010
2011
APAC
2012
Parameterized MTS
Weighted MTS
Dual-Priced MTS
Modal Contracts

Timed Automata

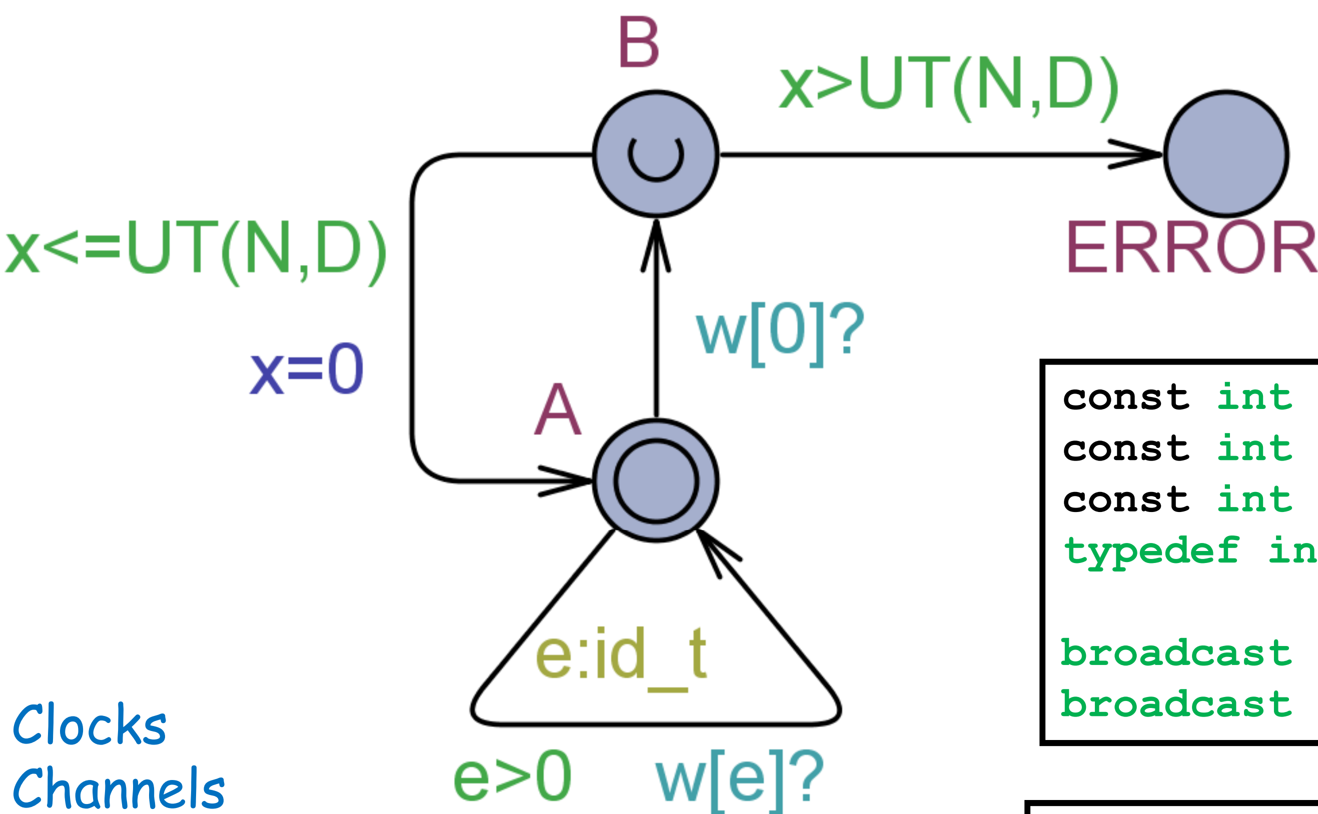


Clocks
Channels
Networks

SEMANTICS:

(A,x=0) - 3.14 → (A,x=3.14)
- a? → (B,x=3.14)
→ (A,x=0)
- 5.23 → (A,x=5.23)
- a? → (B,x=5.23)
→ (ERROR, x=5.23)

Extended Timed Automata

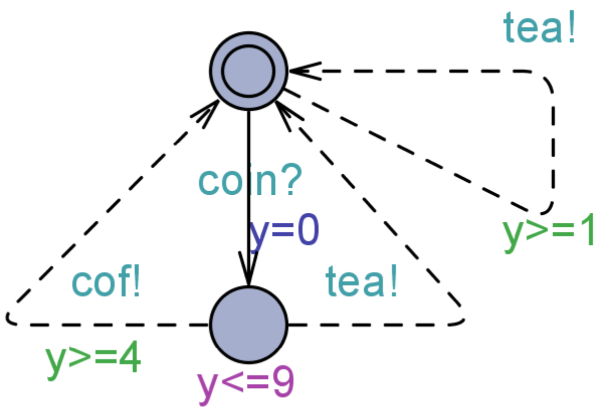


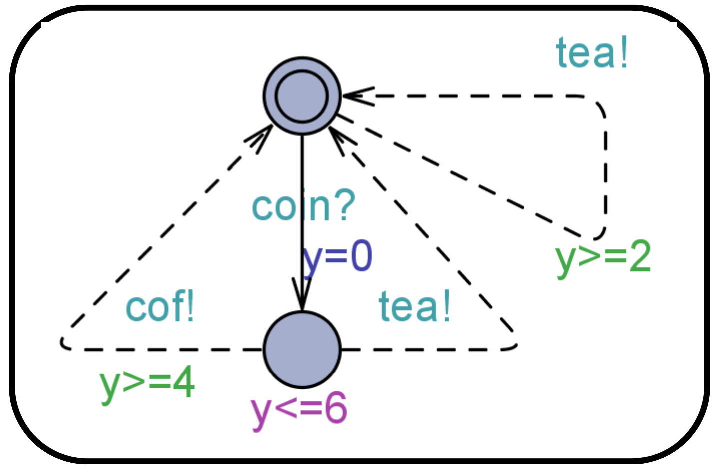

```
const int N = 10;
const int D = 30;
const int d = 4;
typedef int[0,N-1] id_t;

broadcast chan rec[N];
broadcast chan w[N];
```

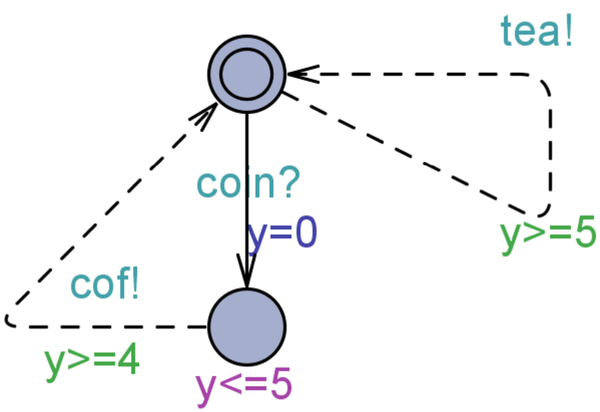
```
int UT (int X, int Y)
{
    return (X+1)*Y;
}
```

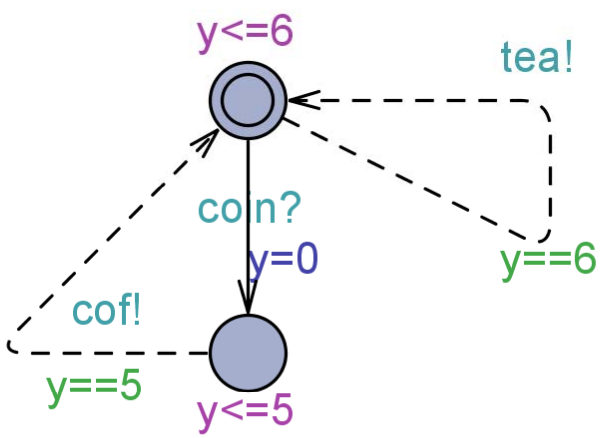
Clocks
Channels
Networks
Integer variables
Structure variables, clocks, channels
User defined types and functions

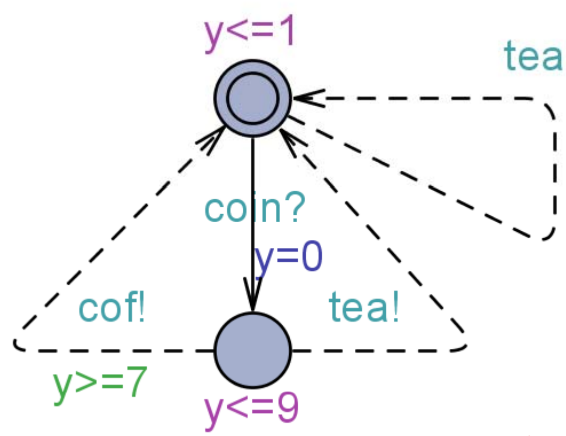
Timed MTS, Refinements & Implementations







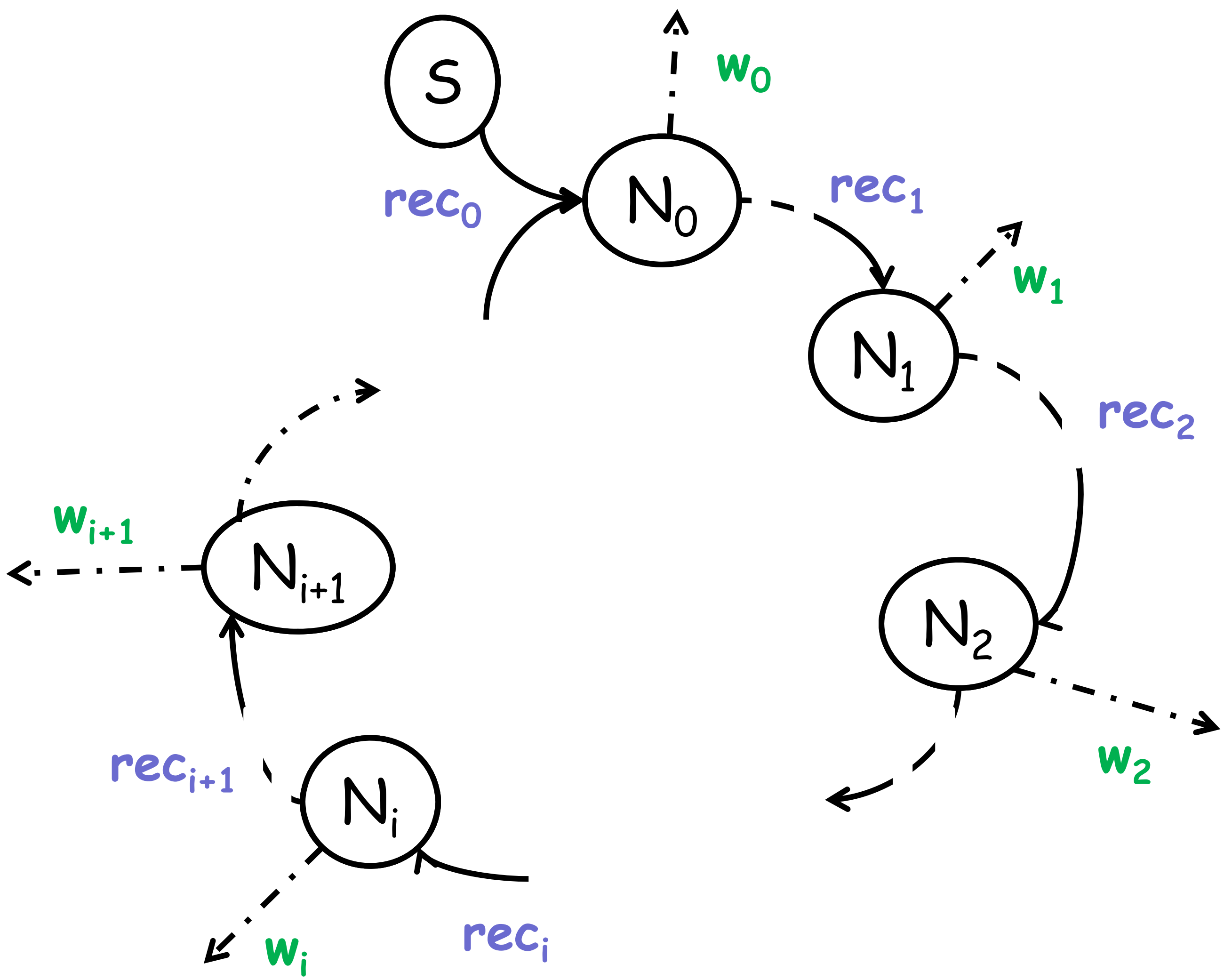




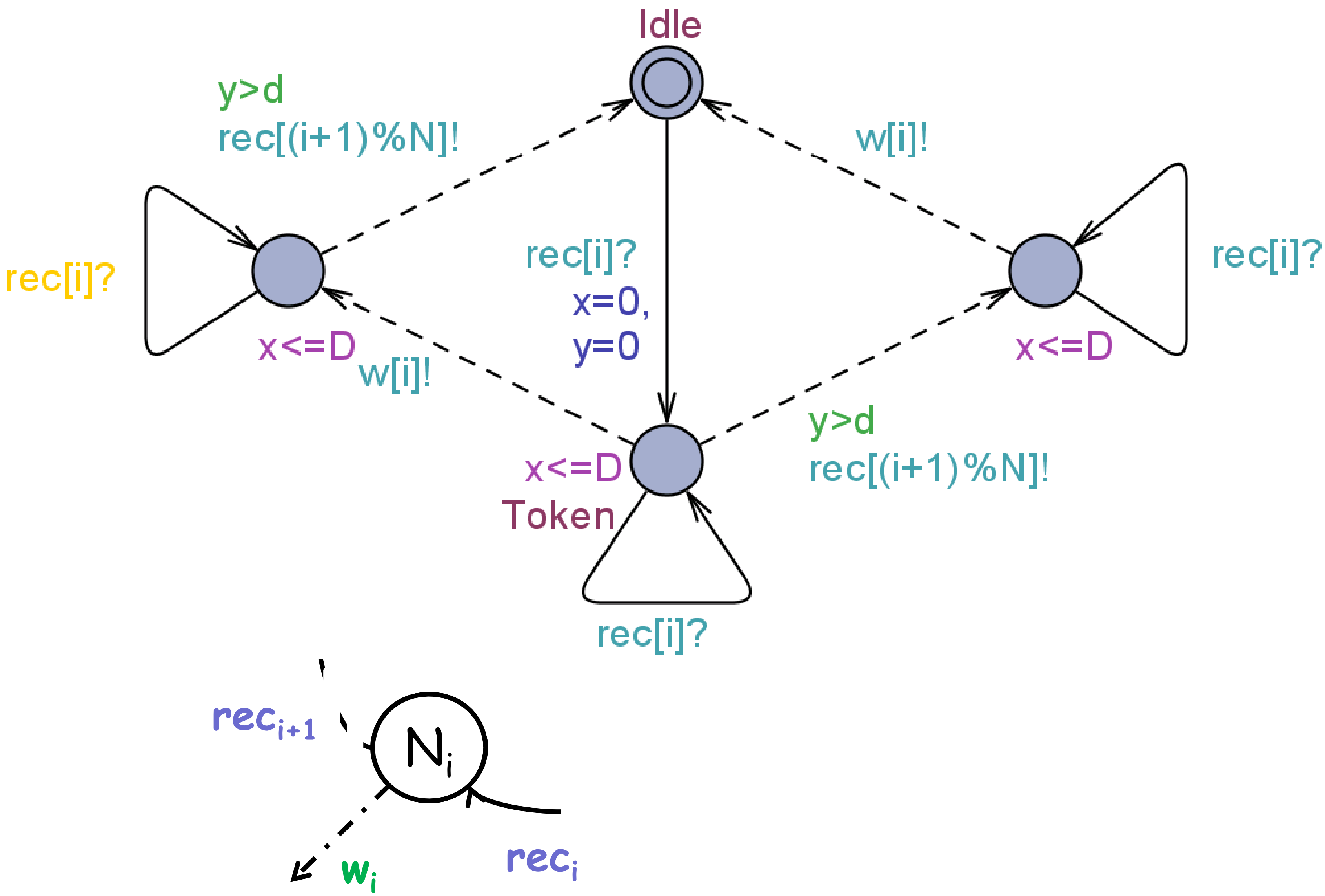
An Implementation

Inconsistent

Real-Time version of Milner's Scheduler



Real-Time version of Milner's Scheduler

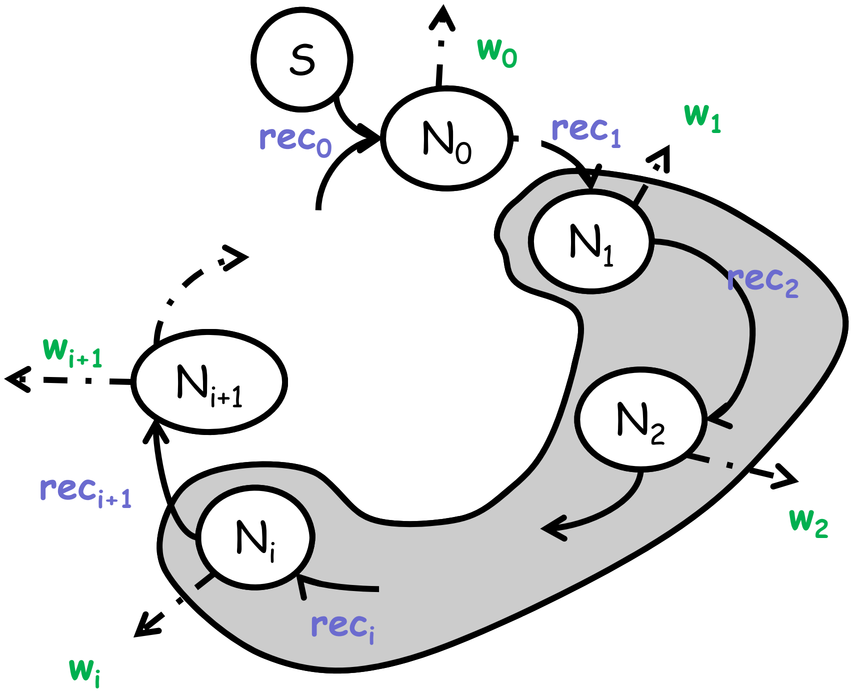


Simulation & Verification

A[] not Env.ERROR

A[] forall (i:id_t) forall (j:id_t)
(Node(i).Token and Node(j).Token imply i==j)

Milner's Scheduler Compositionaly



Find SS_i and verify:

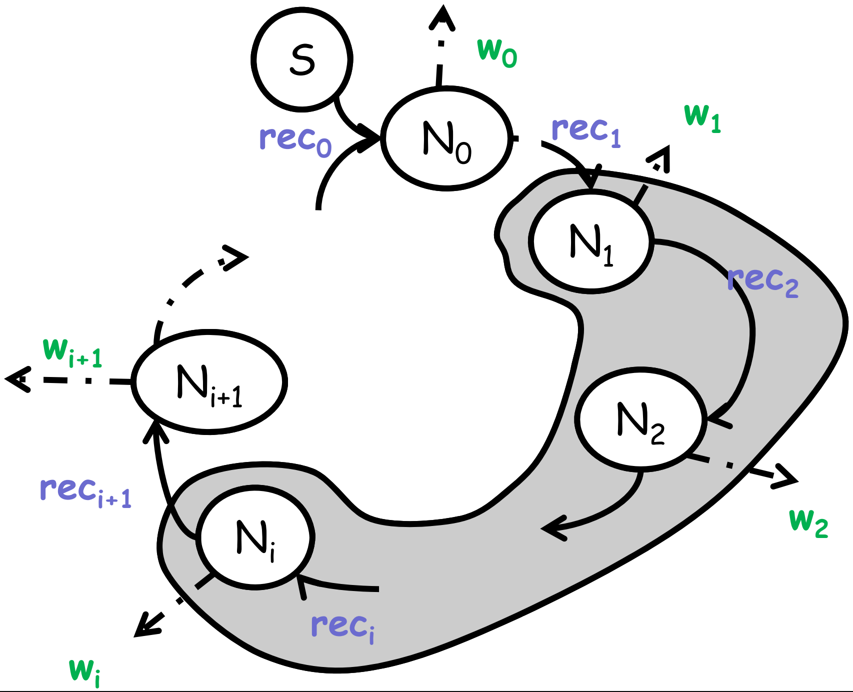
1. $N_1 \leq SS_1$
2. $SS_1 \mid N_2 \leq SS_2$
3. $SS_2 \mid N_3 \leq SS_3$

… …

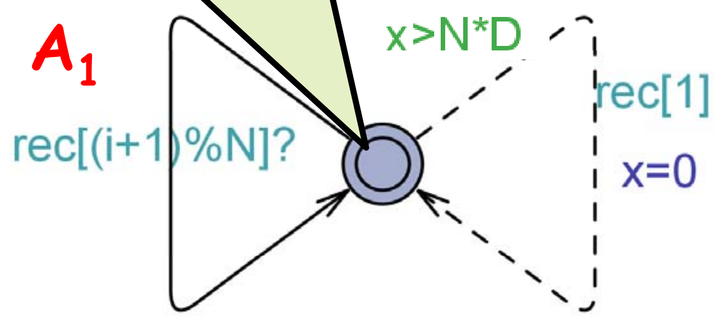
n. $SS_{n-1} \mid N_n \leq SS_n$

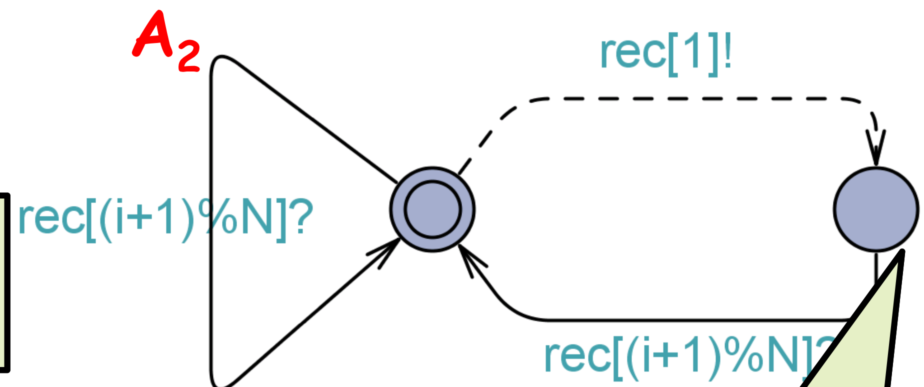
n+1. $SS_n \mid N_0 \leq SPEC$

Milner's Scheduler Compositionaly



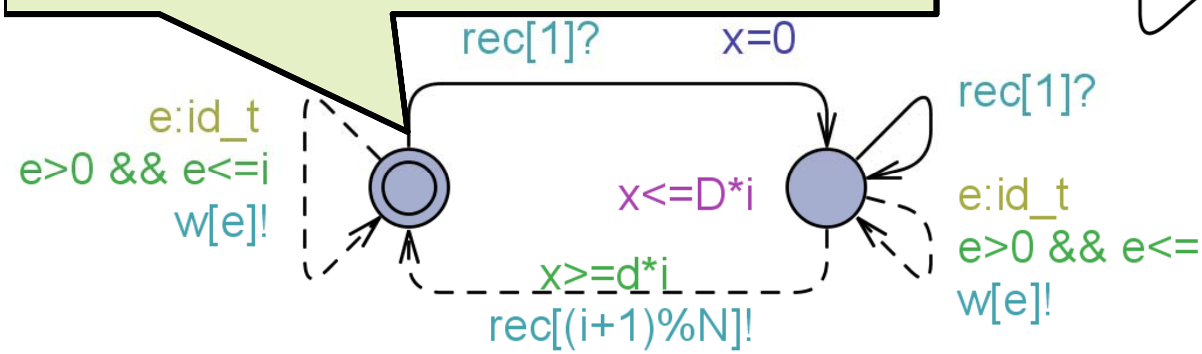
rec[1]! occurs with > N*D time sep.



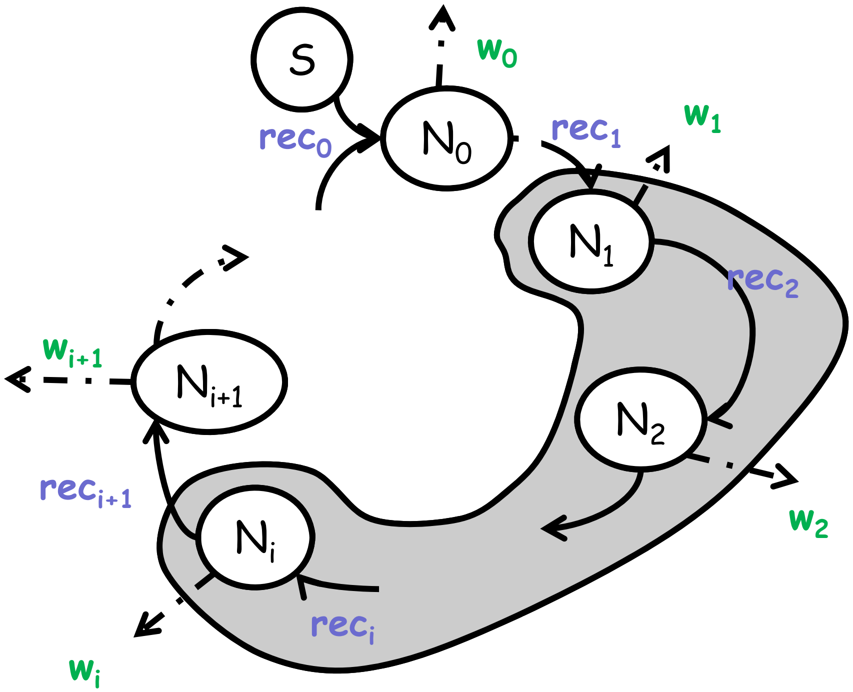


No new **rec[1]!** until rec[i+1]?

After **rec[1]?** then **rec[i+1]!** within [d*i,D*i]

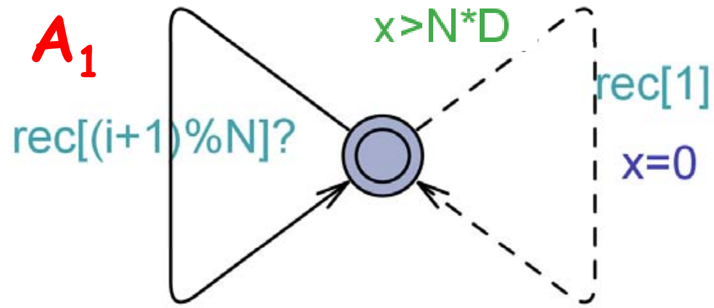


Milner's Scheduler Compositionaly

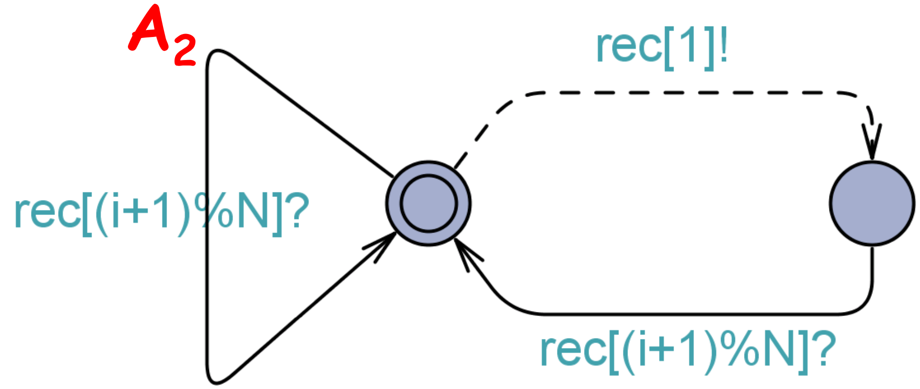


Take $SS_i = (A_1 \ \& \ A_2) \gg G$

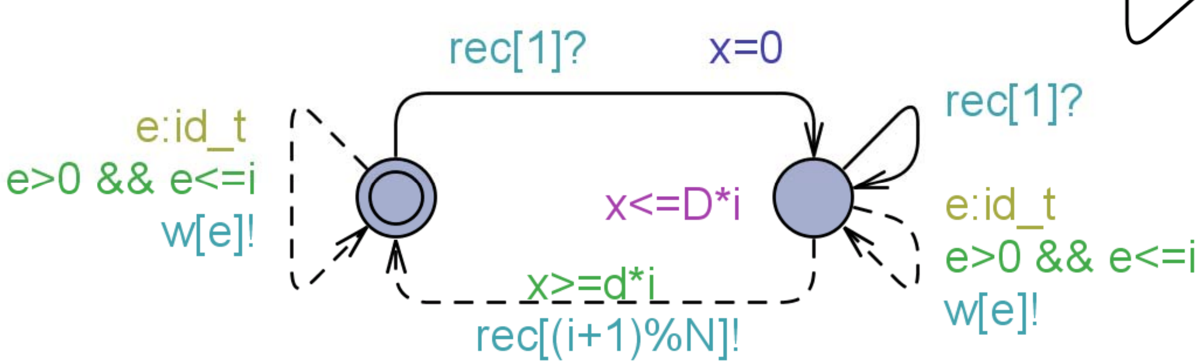
A_1



A_2

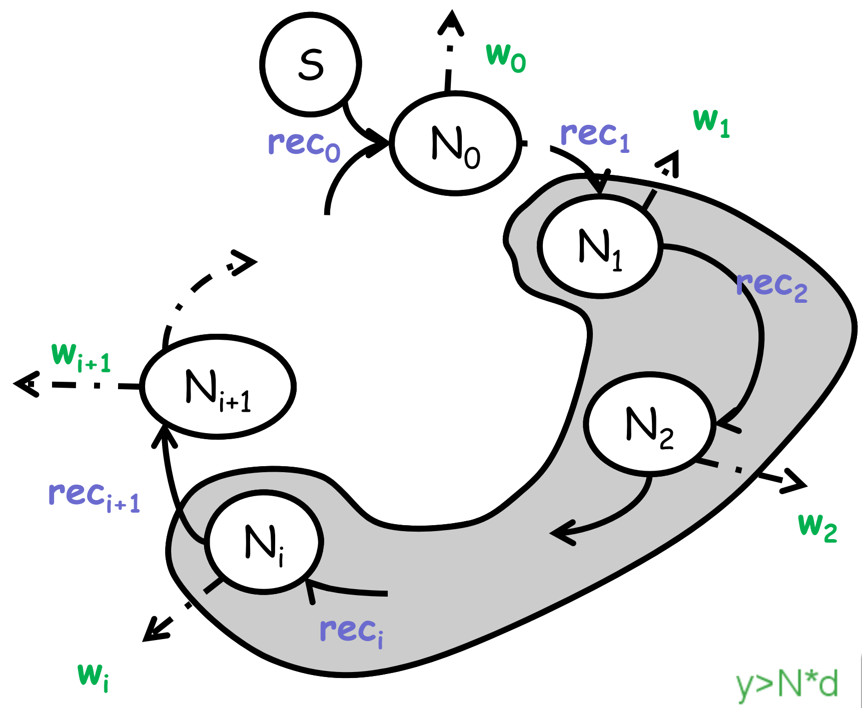


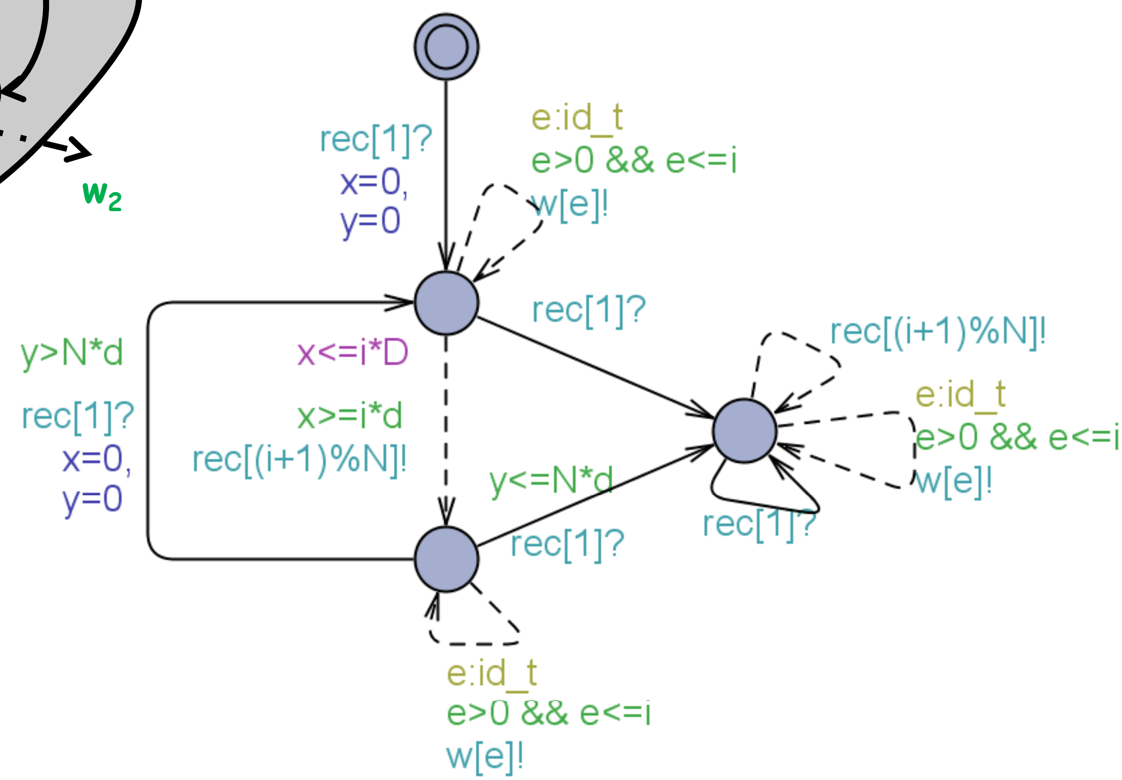
G



Milner's Scheduler Compositionaly

Take $SS_i = (A_1 \;\&\; A_2) \gg G$

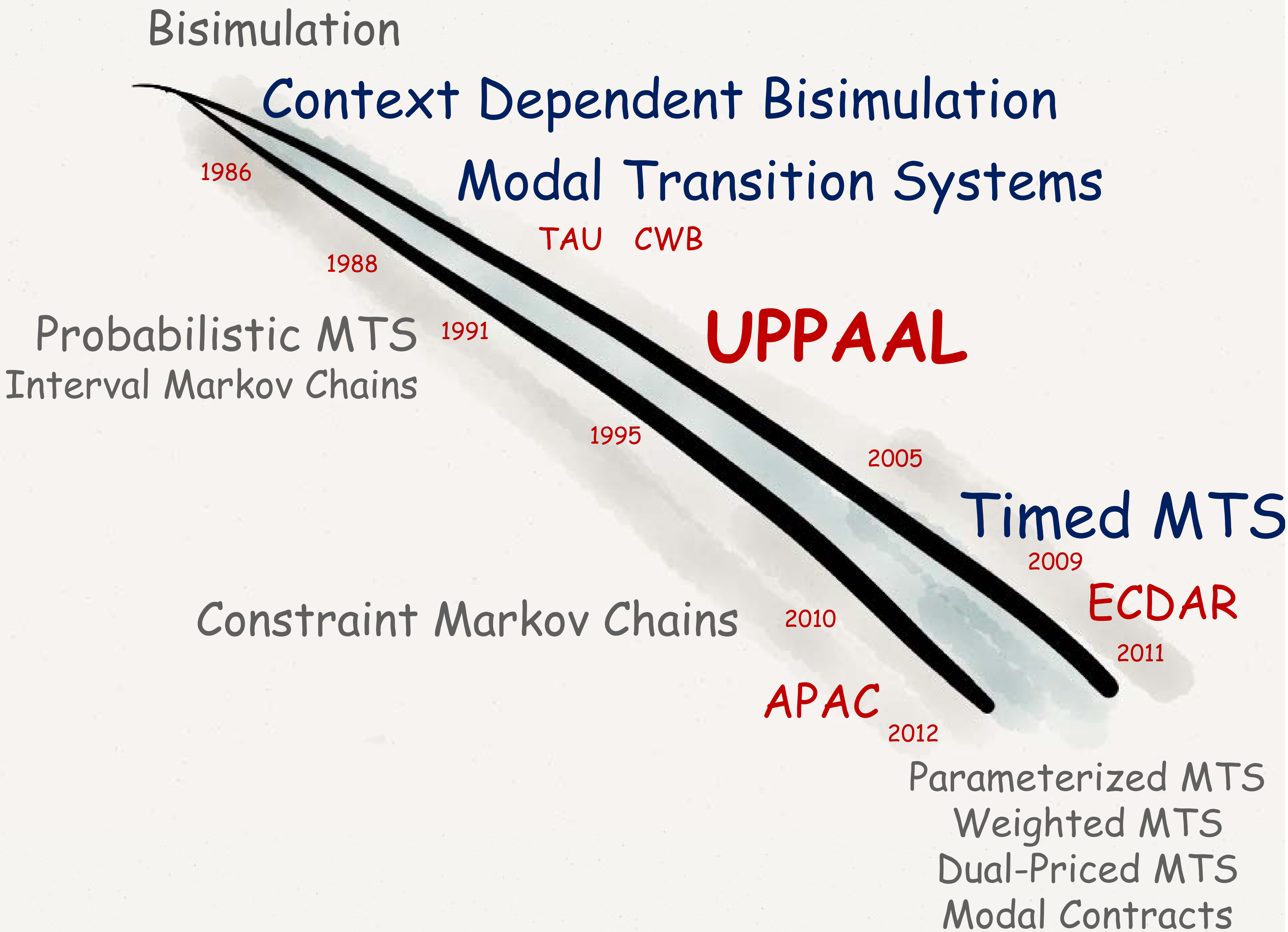




Experiments

D=30

	$d = 29$	20	10	9	8	6	4
$n = 5$	0.080	0.097	0.191	*0.169*	*0.172*	*0.151*	*0.205*
monolithic	0.034	0.034	0.073	1.191	1.189	64.933	> 600
$n = 6$	0.102	0.133	0.231	*0.228*	*0.238*	*0.238*	*0.294*
monolithic	0.040	0.043	0.095	6.786	6.791	> 600	> 600
$n = 8$	0.225	0.349	0.516	**0.515**	*0.540*	*0.600*	*0.582*
monolithic	0.076	0.076	0.230	88.542	88.642	> 600	> 600
$n = 12$	0.830	1.414	1.802	**1.895**	**1.831**	*2.079*	*2.181*
monolithic	0.220	0.223	0.843	> 600	> 600	> 600	> 600
$n = 20$	4.990	9.739	12.377	**11.923**	**12.041**	**12.438**	*12.764*
monolithic	1.038	1.030	4.523	> 600	> 600	> 600	> 600
$n = 30$	22.053	45.709	55.728	**55.345**	**55.112**	**54.702**	*56.164*
monolithic	3.791	3.778	17.652	> 600	> 600	> 600	om



Bisimulation
Context Dependent Bisimulation
Modal Transition Systems
1986
1988
TAU CWB
Probabilistic MTS
Interval Markov Chains
1991
UPPAAL
1995
2005
Timed MTS
2009
ECDAR
Constraint Markov Chains
2010
2011
APAC
2012
Parameterized MTS
Weighted MTS
Dual-Priced MTS
Modal Contracts

Bisimulation

Context Dependent Bisimulation

1986

Modal Transition Systems

TAU CWB

1988

Probabilistic M

Interval Markov Ch

MTS

AR

Constr

Parameterized MTS

Weighted MTS

Dual-Priced MTS

Modal Contracts

2012

Metrics