

Probabilistic Automata and Equivalences

Roberto Segala
University of Verona



Why Formal Analysis?



Why Formal Analysis?

- 1994: The pentium processor computes wrong divisions
 - INTEL forced to replace most processors
 - Economic damage of 450 million US Dollars
- 1995: The software MacInTax spreads the secrets of US tax payers
 - Error in the debug code distributed with MacInTax
 - Users can use it to access the server of Intuit
 - Everybody can read and modify any tax form



Why Formal Analysis?

- 1995: Problems in Denver Airport
 - The fully automated baggage system fails
 - Scheduled to open in 1993
 - The system loses or tears apart luggage
 - Considerable congestion
 - Considerable lack of design
 - In 2005 the system is still not working
 - The system is too complex
 - Extensive research activity is necessary



Why Formal Analysis?

- 1996: Vector Ariane 5 explodes during take-off
 - The control software assigns a 64 bit number to a 16 bit variable
 - The code was recycled from Ariane 4
 - Ariane 5 is fast and its lateral speed does not fit in 16 bits
 - Result: overflow - the system shuts down
 - The back up computer is started
 - ... but the software is the same
 - Result: again overflow - the system shuts down
 - Ariane, without guidance, self destroys
 - Damage: 1 billion Euros



Why Formal Analysis?

- 1982 Mutual exclusion solved with small shared variables
 - Rabin proposes a randomized distributed algorithm
 - The proof is semi-formal but credible
- 1990 Some problems appear
 - Nancy Lynch gives a lecture on Rabin's algorithm
 - Roberto Segala is the scribe and tries to formalize the proof
 - Problem in an informally obvious step
 - Two events are compared but they belong to different probability spaces
 - Nondeterminism is the cause of the problem
- 1991 An attack is found
- Later many other algorithms turned out to be bogus



Why Formal Analysis?

- 1978: Needham and Schroeder
 - Propose an authentication protocol
 - The correctness proof is semi-formal
- 1981: Problems with freshness
 - Replay attacks are possible
- 1995: An attack found
 - Parallel sessions may lead to attack
- Needham: you changed my definitions
- Later: many protocols have been attacked



Lessons that we can Learn

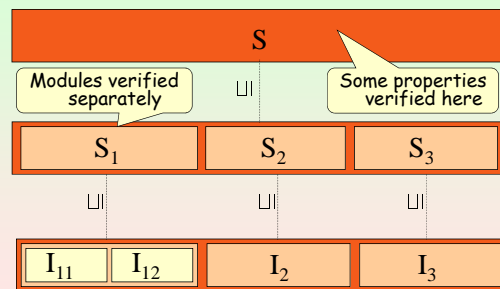
- Formal methods are useful (necessary)
 - Need to define what we want
 - Objectives should be clear and accepted
 - We should communicate with others
 - Need to prove properties rigorously
 - We may miss pieces otherwise
 - We need techniques
 - Need modular verification techniques
 - We want to reuse existing proofs
 - Need ways to automate the analysis
 - Large systems require considerable effort



Hierarchical and Compositional Approach



Hierarchical Compositional Verification



Implementation

- Typically some form of behavioral inclusion
 - Traces
 - Ordinary, complete, quiescent, fair
 - Failures
 - Traces followed by actions the system refuses to perform
 - Tests
 - Occurrence of some success event in appropriate contexts
- Nice properties
 - Transitive
 - Compositional
 - Affine with logical implication
 - ... when properties are sets of behaviors
- Hard to check
 - Usually Pspace-complete
 - But simulation relations help




Proving Implementation

- Behavioral inclusion
 - Behaviors are full computations
 - Possibly infinite length
 - Properties of complex objects
 - Global reasoning
 - Easy to end up with "proofs by intuition"
- Simulation relations
 - Sound for behavioral inclusion
 - Properties of single computational steps
 - Local reasoning
 - Easier to be rigorous



Why Nondeterminism with Probability?


13

 Probabilistic Automata and Equivalences
 Bertinoro, June 21, 2010 Roberto Segala - University of Verona

Why Nondeterminism with Probability? Distributed Algorithms

- Some problems are unsolvable
 - Consensus [FLP85]
- ... but are solvable with randomization
 - Probabilistic consensus [Ben83, AH90]
- Probability and nondeterminism coexist
 - Probability:
 - Processes flip coins
 - Nondeterminism:
 - Several processes in parallel
 - Do not care whether the coin is fair
- Quantitative analysis
 - What is the worst expected complexity?


14

 Probabilistic Automata and Equivalences
 Bertinoro, June 21, 2010 Roberto Segala - University of Verona

Why Nondeterminism with Probability? Stochastic Games

- Nondeterminism
 - Each player has several moves available
- Probability
 - Moves may involve coin flipping
- Quantitative analysis
 - What is the best probability to win the game?


15

 Probabilistic Automata and Equivalences
 Bertinoro, June 21, 2010 Roberto Segala - University of Verona

Why Nondeterminism with Probability? Security

- Nondeterminism
 - User behavior (adversary in Dolev-Yao)
 - Relative speeds of agents
 - Agent behavior (usually deterministic)
- Probability
 - Users and agents flip coins
 - Nonces, keys, random protocols
- Quantitative analysis
 - Probability of attack (negligible)


16

 Probabilistic Automata and Equivalences
 Bertinoro, June 21, 2010 Roberto Segala - University of Verona

Why Nondeterminism with Probability? Concurrency Theory


- Nondeterminism
 - Scheduling within parallel composition
 - Unknown behavior of the environment
 - Underspecification
- Probability
 - Environment may be stochastic
 - Processes may flip coins

17

 Probabilistic Automata and Equivalences
 Bertinoro, June 21, 2010 Roberto Segala - University of Verona

How Probability with Nondeterminism?

18

 Probabilistic Automata and Equivalences
 Bertinoro, June 21, 2010 Roberto Segala - University of Verona

The Main Idea

- Add probability to Concurrency Theory
 - Nondeterminism should remain
 - Should obtain a conservative extension
- Proposals to tackle the problem
 - Replace points with **measures**
 - Replace functions with **measurable functions**



Probability and Nondeterminism: How?

- Reactive, Generative Systems [LS89,GSST90]
 - Labeled transition systems
 - Add probabilities to the arcs
 - Process algebras
 - Replace + with probabilistic +
- Probabilistic Automata [Seg95]
 - Labeled transition systems
 - Replace target states with target measures in transitions
 - Process Algebras
 - Add a probabilistic + operator (named \oplus)



Automata

$$A = (Q, q_0, E, H, D)$$

- Transition relation
 $D \subseteq Q \times (E \cup H) \times Q$
- Internal (hidden) actions
- External actions: $E \cap H = \emptyset$
- Initial state: $q_0 \in Q$
- States



Probabilistic Automata

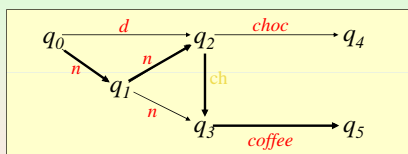
$$PA = (Q, q_0, E, H, D)$$

- Transition relation
 $D \subseteq Q \times (E \cup H) \times \text{Disc}(Q)$
- Internal (hidden) actions
- External actions: $E \cap H = \emptyset$
- Initial state: $q_0 \in Q$
- States



Example: Automata

$$A = (Q, q_0, E, H, D)$$

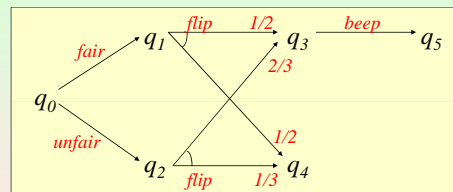


Execution: $q_0 \ n \ q_1 \ n \ q_2 \ ch \ q_3 \ coffee \ q_5$

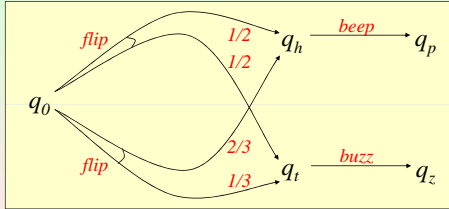
Trace: $n \ n \ coffee$



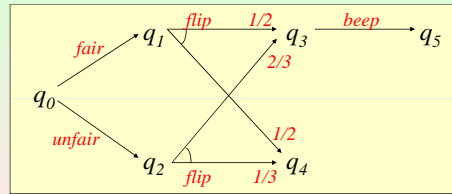
Example: Probabilistic Automata



Example: Probabilistic Automata

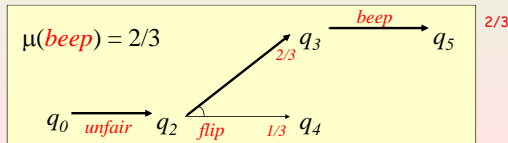
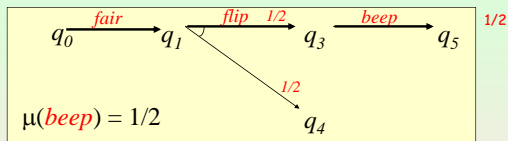


Example: Probabilistic Automata

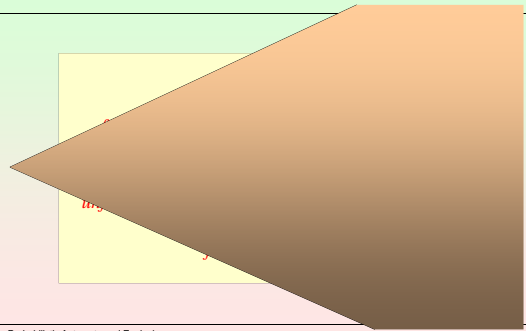


What is the probability of beeping?

Example: Probabilistic Executions



Example: Probabilistic Executions



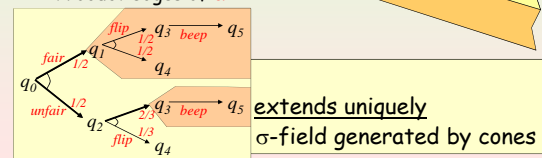
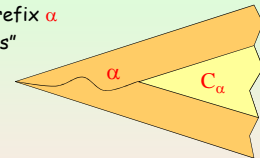
Measure Theory

- Sample set**
 - Set of objects Ω
- Sigma-field (σ -field)**
 - Subset F of 2^Ω satisfying
 - Inclusion of Ω
 - Closure under complement
 - Closure under countable union
 - Closure under countable intersection
- Measure on (Ω, F)**
 - Function μ from F to \mathbb{R}^{+0}
 - For each countable collection $\{X_i\}$ of pairwise disjoint sets of F , $\mu(\cup_i X_i) = \sum_i \mu(X_i)$
- (Sub-)probability measure**
 - Measure μ such that $\mu(\Omega) = 1$ ($\mu(\Omega) \leq 1$)
- Sigma-field generated by $C \subseteq 2^\Omega$**
 - Smallest σ -field that includes C

Why not $F = 2^\Omega$?
 Example: set of executions
 Flip a fair coin infinitely many times
 $\Omega = \{0,1\}^\omega$
 Study probabilities of sets of executions
 which sets can I measure?
 Theorem: there is no probability measure on 2^Ω such that $\mu(\omega) = 0$ for each $\omega \in \Omega$.

Cones and Measures

- Cone of α**
 - Set of executions with prefix α
 - Represent event " α occurs"
- Measure of a cone**
 - Product edges of α



Examples of Events

- Eventually action a occurs
 - Union of cones where action a occurs once
- Action a occurs at least n times
 - Union of cones where action a occurs n times
- Action a occurs at most n times
 - Complement of action a occurs at least $n+1$ times
- Action a occurs exactly n times
 - Intersection of previous two events
- Action a occurs infinitely many times
 - Intersection of action a occurs at least n times for all n
- Execution α occurs and nothing is scheduled after
 - Set consisting of α only
 - C_α intersected complement of cones that extend α



Schedulers - Probabilistic Executions

Scheduler

Function $\sigma : \text{exec}^*(A) \rightarrow \text{SubDisc}(D)$

if $\sigma(\alpha)((q, a, v)) > 0$ then $q = \text{lstate}(\alpha)$

Probabilistic execution generated by σ from state r

Measure

$$\mu_{\sigma,r}(C_s) = 0 \quad \text{if } r \neq s$$

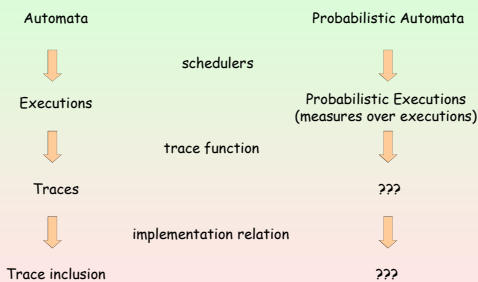
$$\mu_{\sigma,r}(C_r) = 1$$

$$\mu_{\sigma,r}(C_{\text{seq}}) = \mu_{\sigma,r}(C_\alpha) \cdot \left(\sum_{(s,a,v) \in D} \sigma(\alpha)((s,a,v)) \nu(q) \right)$$

$\mu_{\sigma,r}$



Summing Up

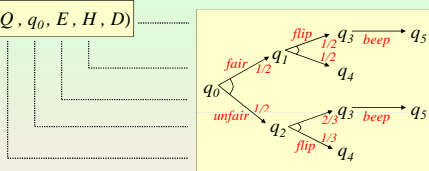


Related Models



Generative Probabilistic Automata

$GPA = (Q, q_0, E, H, D)$

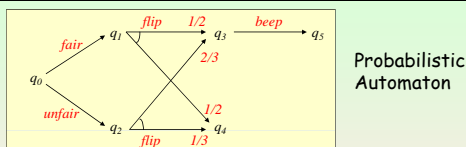


- Actions are chosen probabilistically within a transition
- It is possible to deadlock within a transition

A probabilistic execution "is" a generative Probabilistic Automaton

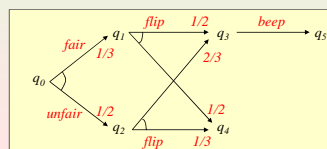


Ex. Generative Probabilistic Automata



Probabilistic Automaton

Generative Probabilistic Automaton



Reactive Systems [LS89,GSST90] (revised)

$RA = (Q, q_0, E, H, D)$

- Transition relation: $D \subseteq Q \times (E \cup H) \times (0,1] \times Q$ Disc(Q)
- Internal (hidden) actions
- External actions: $E \cap H = \emptyset$
- Initial state: $q_0 \in Q$
- States

• For each s and each a $\sum \{p \mid \exists_i (s, a, p, i) \in D\} \in \{0,1\}$

This is a Deterministic Probabilistic Automaton

Probabilistic Automata and Equivalences
 Bertinoro, June 21, 2010 Roberto Segala - University of Verona 37

Example: Reactive Systems

Reactive

Non reactive

Probabilistic Automata and Equivalences
 Bertinoro, June 21, 2010 Roberto Segala - University of Verona 38

Generative Systems (revised) [GSST90]

$GA = (Q, q_0, E, H, D)$

- Transition relation: $D \subseteq Q \times (E \cup H) \times (0,1] \times Q$ SubDisc((E ∪ H) × Q)
- Internal (hidden) actions
- External actions: $E \cap H = \emptyset$
- Initial state: $q_0 \in Q$
- States

• For each s $\sum \{p \mid \exists_{i,a} (s, a, p, i) \in D\} \leq 1$

This is a special Generative Probabilistic Automaton (at most one transition from each state)

Probabilistic Automata and Equivalences
 Bertinoro, June 21, 2010 Roberto Segala - University of Verona 39

Reactive Systems [LS89,GSST90]

$RA = (Q, q_0, E, H, D)$

- Transition relation: $D \subseteq Q \times (E \cup H) \times (0,1] \times Q$?
- Internal (hidden) actions
- External actions: $E \cap H = \emptyset$
- Initial state: $q_0 \in Q$
- States

$1/2a.F + 1/2a.F + 1b.G$

- If $(s, a, p, i, t) \in D$ and $(s, b, q, i, r) \in D$, then $a=b, p=q, t=r$
- For each s and each a $\sum \{p \mid \exists_{i,t} (s, a, p, i, t) \in D\} \in \{0,1\}$

Probabilistic Automata and Equivalences
 Bertinoro, June 21, 2010 Roberto Segala - University of Verona 40

Reactive Systems [LS89,GSST90]

Probabilistic Automata and Equivalences
 Bertinoro, June 21, 2010 Roberto Segala - University of Verona 41

Some Considerations

- According to [GSST90]
 - Generative is more detailed than reactive
 - Reactive retrieved from generative by abstraction
 - Renormalize probabilities on actions

abstraction

This is fine with deterministic systems

Probabilistic Automata and Equivalences
 Bertinoro, June 21, 2010 Roberto Segala - University of Verona 42

Some Considerations

The idea of [GSST90] does not work with nondeterminism

resolution of nondeterminism

abstraction

? ↑

Probabilistic Automata and Equivalences
 Bertinoro, June 21, 2010 Roberto Segala - University of Verona 43

Some Considerations

The idea of [GSST90] does not work with nondeterminism

resolution of nondeterminism

abstraction

? ↑

Probabilistic Automata and Equivalences
 Bertinoro, June 21, 2010 Roberto Segala - University of Verona 44

Markov Decision Processes [Bel57]

$MDP = (Q, q_0, A, p)$

- Transition probabilities: $p : Q \times Q \times Act \rightarrow [0,1]$
- Available actions: $A : Q \rightarrow 2^{Act}$
- Initial state: $q_0 \in Q$
- States

- A associates a set of available actions with each state
- For each state s and each action $a \in A(s)$
 - $0 \leq p_{s,a} \leq 1$ for each state s
 - $\sum_{a \in A(s)} p_{s,a} = 1$

This is a Reactive System or a Deterministic Probabilistic Automaton

Probabilistic Automata and Equivalences
 Bertinoro, June 21, 2010 Roberto Segala - University of Verona 45

Labeled Concurrent Markov Chains [HJ89 from Var85] – Strictly Alternating

[PL500]

$LCMC = (N, P, q_0, E, H, Dn, Dp)$

- Transition relation: $Dn \subseteq N \times (E \cup H) \times (P \cup N)$
- $Dp : P \rightarrow Disc(N) \subseteq P \times \{ \vec{v} \} \times Disc(N)$
- Internal (hidden) actions
- External actions: $E \cap H = \emptyset$
- Initial state: $q_0 \in N$
- Probabilistic states
- Nondeterministic states

$Dp \subseteq P \times N$
 $p : P \times N \rightarrow [0,1]$
 $\forall_{s \in P} \sum_q p(s,q) = 1$

Probabilistic Automata and Equivalences
 Bertinoro, June 21, 2010 Roberto Segala - University of Verona 46

Other Models

- Rabin's Probabilistic Automata
 - Introduced in the context of language theory
 - Extended by our Probabilistic Automata
- Unlabeled systems [Var85,BA95,BK98]
 - Can be Probabilistic Automata with a single invisible action
 - Labels may be associated with states
 - The theory does not change
- Markov Chains
 - Unlabeled systems that enable one transition from each state
- Probabilistic Input/Output Automata
 - Add Input/Output distinction on actions
 - Useful to handle composition of generative PAs

Probabilistic Automata and Equivalences
 Bertinoro, June 21, 2010 Roberto Segala - University of Verona 47

How about Process Algebras?

Probabilistic Automata and Equivalences
 Bertinoro, June 21, 2010 Roberto Segala - University of Verona 48

Probabilistic Process Algebra [BS01,PS05] - (convenience of alternation)

$$E ::= 0 \mid E+E \mid \alpha.P \mid X \mid \text{rec } X.E$$

$$P ::= \Delta(E) \mid P \oplus_p P$$

<p style="text-align: center;">Alternating prefix</p> $\frac{-}{\alpha.P \xrightarrow{\alpha} P}$	<p style="text-align: center;">Probabilistic processes</p> $\frac{-}{\Delta(E) \xrightarrow{1} E}$								
<p style="text-align: center;">Probabilistic processes</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="border: 1px solid black; padding: 5px;">$P_1 \xrightarrow{q} E$</td> <td style="border: 1px solid black; padding: 5px;">$P_2 \xrightarrow{r} E$</td> <td style="border: 1px solid black; padding: 5px;">$P_1 \xrightarrow{q} E$</td> <td style="border: 1px solid black; padding: 5px;">$P_2 \xrightarrow{r} E$</td> </tr> <tr> <td colspan="2" style="border: 1px solid black; padding: 5px;">$P_1 \oplus_p P_2 \xrightarrow{pq} E$</td> <td colspan="2" style="border: 1px solid black; padding: 5px;">$P_1 \oplus_p P_2 \xrightarrow{pq+(1-p)r} E$</td> </tr> </table>		$P_1 \xrightarrow{q} E$	$P_2 \xrightarrow{r} E$	$P_1 \xrightarrow{q} E$	$P_2 \xrightarrow{r} E$	$P_1 \oplus_p P_2 \xrightarrow{pq} E$		$P_1 \oplus_p P_2 \xrightarrow{pq+(1-p)r} E$	
$P_1 \xrightarrow{q} E$	$P_2 \xrightarrow{r} E$	$P_1 \xrightarrow{q} E$	$P_2 \xrightarrow{r} E$						
$P_1 \oplus_p P_2 \xrightarrow{pq} E$		$P_1 \oplus_p P_2 \xrightarrow{pq+(1-p)r} E$							

Probabilistic Automata and Equivalences
Bertinoro, June 21, 2010
Roberto Segala - University of Verona
49

Convex Combination of Measures

- Let μ_1 and μ_2 be probability measures
- Let p_1 and p_2 be reals in $[0,1]$ such that $p_1+p_2=1$
- Define a new measure $\mu = p_1\mu_1+p_2\mu_2$ as follows
 - $\forall X, \mu(X) = p_1\mu_1(X)+p_2\mu_2(X)$
- Theorem: μ is a probability measure
- Same result extends to countable summation

Probabilistic Automata and Equivalences
Bertinoro, June 21, 2010
Roberto Segala - University of Verona
50

Probabilistic Process Algebra [BS01,PS05] - (convenience of alternation)

$$E ::= 0 \mid E+E \mid \alpha.P \mid X \mid \text{rec } X.E$$

$$P ::= \Delta(E) \mid P \oplus_p P$$

<p style="text-align: center;">Alternating prefix</p> $\frac{-}{\alpha.P \xrightarrow{\alpha} \delta(P)}$	<p style="text-align: center;">Non-alternating prefix</p> $\frac{P \vdash \mu}{\alpha.P \xrightarrow{\alpha} \mu}$	<p style="text-align: center;">Probabilistic processes</p> $\frac{P \vdash \mu}{P \xrightarrow{\tau} \mu}$						
<p style="text-align: center;">Measures associated with probabilistic expressions</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="border: 1px solid black; padding: 5px;">$\Delta(E) \vdash \delta(E)$</td> <td style="border: 1px solid black; padding: 5px;">$P_1 \vdash \mu_1$</td> <td style="border: 1px solid black; padding: 5px;">$P_2 \vdash \mu_2$</td> </tr> <tr> <td colspan="3" style="border: 1px solid black; padding: 5px;">$P_1 \oplus_p P_2 \vdash p\mu_1+(1-p)\mu_2$</td> </tr> </table>			$\Delta(E) \vdash \delta(E)$	$P_1 \vdash \mu_1$	$P_2 \vdash \mu_2$	$P_1 \oplus_p P_2 \vdash p\mu_1+(1-p)\mu_2$		
$\Delta(E) \vdash \delta(E)$	$P_1 \vdash \mu_1$	$P_2 \vdash \mu_2$						
$P_1 \oplus_p P_2 \vdash p\mu_1+(1-p)\mu_2$								

Probabilistic Automata and Equivalences
Bertinoro, June 21, 2010
Roberto Segala - University of Verona
51

Example and Considerations

$$\alpha.(\Delta(E) \oplus_{1/2} \Delta(F)) + \alpha.(\Delta(E) \oplus_{2/3} \Delta(F))$$

non-alternating

transformation

 \longleftrightarrow

split/merge transitions

This is a Probabilistic Automaton

alternating

Probabilistic Automata and Equivalences
Bertinoro, June 21, 2010
Roberto Segala - University of Verona
52

Parallel Composition

Probabilistic Automata and Equivalences
Bertinoro, June 21, 2010
Roberto Segala - University of Verona
53

Composition of Probabilistic Automata

$A_1 = (Q_1, q_1, E_1, H_1, D_1)$

↓

$A_2 = (Q_2, q_2, E_2, H_2, D_2)$

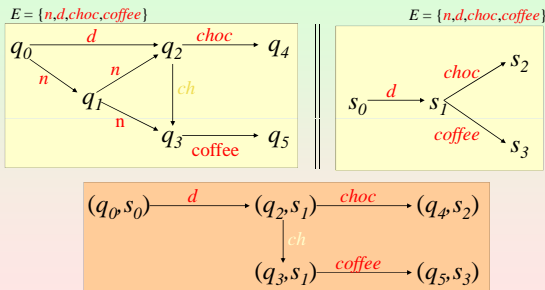
$A_1 \parallel A_2 = (Q_1 \times Q_2, (q_1, q_2), E_1 \cup E_2, H_1 \cup H_2, D)$

$$D = \left\{ (q, a, (s_1, s_2)) \mid \begin{array}{l} \text{if } a \in E_i \cup H_i \text{ then } (\pi_i(q), a, s_i) \in D_i \\ \text{if } a \notin E_i \cup H_i \text{ then } s_i = \pi_i(q) \end{array} \quad i \in \{1, 2\} \right\}$$

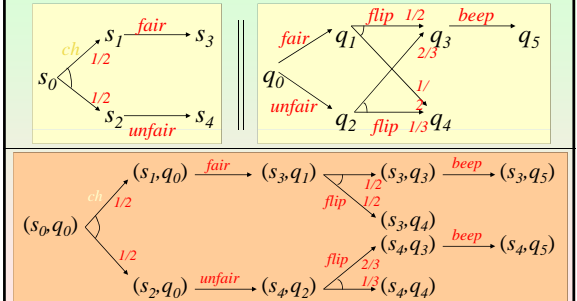
$$D = \left\{ (q, a, \mu_1 \times \mu_2) \mid \begin{array}{l} \text{if } a \in E_i \cup H_i \text{ then } (\pi_i(q), a, \mu_i) \in D_i \\ \text{if } a \notin E_i \cup H_i \text{ then } \mu_i = \delta(\pi_i(q)) \end{array} \quad i \in \{1, 2\} \right\}$$

Probabilistic Automata and Equivalences
Bertinoro, June 21, 2010
Roberto Segala - University of Verona
54

Example: Composition of Automata



Ex. Composition of Probabilistic Automata



Projections

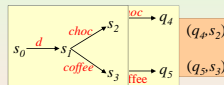
Let α be an execution of $A_1 \parallel A_2$

$$\alpha = (q_0, s_0) \xrightarrow{d} (q_2, s_1) \xrightarrow{ch} (q_3, s_1) \xrightarrow{coffee} (q_5, s_3)$$

What are the contributions of A_1 and A_2 ?

$$\pi_1(\alpha) \equiv q_0 \xrightarrow{d} q_2 \xrightarrow{ch} q_3 \xrightarrow{coffee} q_5$$

$$\pi_2(\alpha) \equiv s_0 \xrightarrow{d} s_1 \xrightarrow{coffee} s_3$$



Theorem

$$\alpha \in \text{execs}(A_1 \parallel A_2) \text{ iff } \forall i \in \{1, 2\} \pi_i(\alpha) \in \text{execs}(A_i)$$

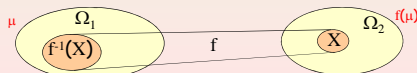
Measure Theory: Image Measure

• **Measurable function** from $(\Omega_1, \mathcal{F}_1)$ to $(\Omega_2, \mathcal{F}_2)$

- Function f from Ω_1 to Ω_2
- For each element X of \mathcal{F}_2 , $f^{-1}(X) \in \mathcal{F}_1$

• **Image measure** $f(\mu)$

$$f(\mu)(X) = \mu(f^{-1}(X))$$



Projections

The projection function is measurable

$\pi(\mu)$: image measure under π of μ

Theorem

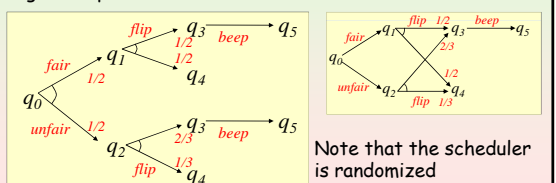
If μ is a probabilistic execution of $A_1 \parallel A_2$

then

$\pi_i(\mu)$ is a probabilistic execution of A_i

Example: Projection

Projection onto right component



Note that the scheduler is randomized

Use of Projections

- Let $M = MP \parallel CF$
- Suppose that MP satisfies Φ provided that the environment (CF) satisfies Ψ
- Suppose that CF satisfies Ψ with probability p
- Can I conclude that M satisfies Φ with probability p ?

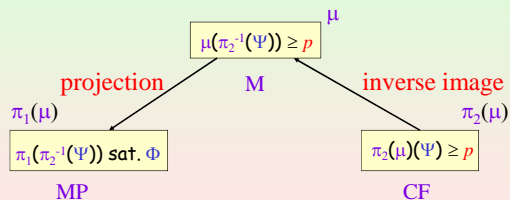
$$\frac{MP \models \Psi \Rightarrow \Phi \quad CF \models [\Psi]_{\geq p}}{M \models [\Phi]_{\geq p}}$$

- This example is taken from a real case study [PLS01]
 - Randomized consensus protocol of Aspnes and Herlihy [AH90]
 - MP is a complex non randomized protocol
 - CF is a relatively simple randomized coin flipper



Formal Argument

Let μ be a probabilistic execution of M .



Composition for Generative PAs

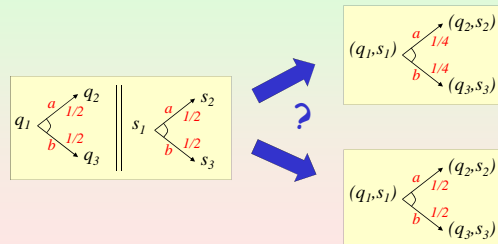
How to synchronize two generative transitions?

- SCCS**
 - Easy. Each automaton chooses independently
- CSP, CCS**
 - Difficult to handle nondeterminism between independent actions
 - There are some proposals [AHK99, PPO5]
- I/O automata**
 - OK if only output transitions are generative [WSS94, Seg95]



Composition for Generative PAs (Problems)

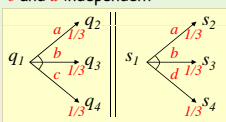
When and how should transitions synchronize?



Composition for Generative PAs (Problems)

When and how should transitions synchronize?

a and b in common
 c and d independent



Nondeterminism

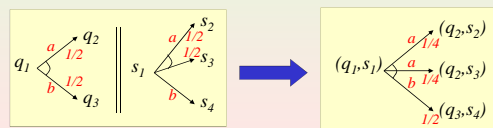
Pr	Lft	Rht	Effect
1/9	a	a	a
1/9	a	b	removed or δ
1/9	a	d	?
1/9	b	a	removed or δ
1/9	b	b	b
1/9	b	d	?
1/9	c	a	?
1/9	c	b	?
1/9	c	d	???



Composition for Generative PAs (A solution)

Introduce Input/Output Distinction (PIOAs)

- Reactive on Input
- Generative on output
- Impose input enabling
- Input transitions synchronize as before
- Output transitions synchronize with appropriate input transitions



Probabilistic I/O Automata (revised) [Wu, Smolka, Stark 94]

$PIOA = (Q, q_0, E, H, D)$

- Transition relation $D \subseteq Q \times (E \cup H) \times (0,1] \times Q$
- Internal (hidden) actions
- External actions: $E \cap H = \emptyset$
 E partitioned into I, O
- Initial state: $q_0 \in Q$
- States

- For each s and each input a $\sum \{p \mid \exists (s, a, p, t) \in D\} \in \{1\}$
- For each s $\sum \{p \mid \exists_{t, a \in O \cup H} (s, a, p, t) \in D\} = 1$

Deterministic PIOAs with at most 1 generative transition from each state

Probabilistic Automata and Equivalences
Bertinoro, June 21, 2010 Roberto Segala - University of Verona 67

Composition of PIOAs ([WSS94] definition)

- Problem**
 - How to choose between the generative transitions of the two components?
- Solution**
 - Assign a weight to each component
 - Use relative weights to choose the process that moves
- Looks a lot like Stochastic Process Algebras
 - Actions occur with an exponentially distributed delay
 - Race conditions between processes are resolved by the delays
 - It is a generalization of assigning weights to processes
 - The weights are the rates of the actions

Probabilistic Automata and Equivalences
Bertinoro, June 21, 2010 Roberto Segala - University of Verona 68

Language Inclusion

Probabilistic Automata and Equivalences
Bertinoro, June 21, 2010 Roberto Segala - University of Verona 69

Summing Up

Automata		Probabilistic Automata
↓	schedulers	↓
Executions		Probabilistic Executions (measures over executions)
↓	trace function	↓
Traces		Trace distributions (measures over traces)
↓	implementation relation	↓
Trace inclusion		Trace distribution inclusion

Probabilistic Automata and Equivalences
Bertinoro, June 21, 2010 Roberto Segala - University of Verona 70

Trace Distributions

The trace function is measurable

Trace distribution of μ

$tdist(\mu)$: image measure under trace of μ

Trace distribution inclusion preorder

$A_1 \leq_{TD} A_2$ iff $tdists(A_1) \subseteq tdists(A_2)$

Probabilistic Automata and Equivalences
Bertinoro, June 21, 2010 Roberto Segala - University of Verona 71

Trace Distribution Inclusion is not Compositional

$q_0 \xrightarrow{a} q_1 \xrightarrow{b} q_3$
 $q_0 \xrightarrow{a} q_2 \xrightarrow{c} q_4$

$s_0 \xrightarrow{a} s_1 \xrightarrow{b} s_2$
 $s_0 \xrightarrow{a} s_1 \xrightarrow{c} s_3$

$c_0 \xrightarrow{d} c_1 \xrightarrow{e} c_3$
 $c_0 \xrightarrow{d} c_2 \xrightarrow{f} c_4$

$(s_0, c_0) \xrightarrow{a} (s_1, c_0) \xrightarrow{d} (s_1, c_1) \xrightarrow{e} (s_1, c_3) \xrightarrow{b} (s_2, c_3)$
 $(s_0, c_0) \xrightarrow{a} (s_1, c_0) \xrightarrow{d} (s_1, c_2) \xrightarrow{f} (s_1, c_4) \xrightarrow{c} (s_3, c_4)$

Probabilistic Automata and Equivalences
Bertinoro, June 21, 2010 Roberto Segala - University of Verona 72

How to Get Compositionality

- Restrict the power of composition
 - Probabilistic reactive modules [AHJ01]
 - Switched probabilistic I/O automata [CLSV04]
- Trace Distribution Precongruence
 - Coarsest precongruence included in preorder
 - That is: close under all contexts
 - Alternative characterizations
 - Principal context [Seg95]
 - Testing [Seg96]
 - Forward simulations [LSV03]



... yet, Proving Language Inclusion is Difficult

- Language inclusion is a global property
 - Need to see the whole result of resolving nondeterminism
- We seek local proof techniques
 - Local arguments are easier
- We use simulation relations



Bisimulations



Bisimulation Relations

We have the following objectives

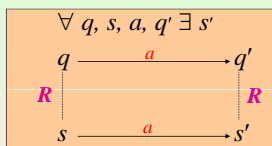
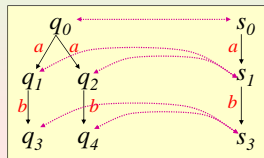
- Same definitional style as for automata
 - Where are the key differences?
 - Keep definitions simple
- Uniform treatment
 - The literature is not uniform
 - This causes a lot of confusion
 - How can we see everything from a single point of view?



Strong Bisimulation on Automata

Strong bisimulation between A_1 and A_2

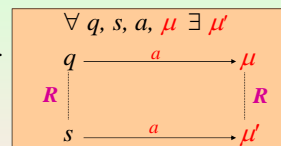
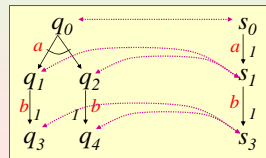
Relation $R \subseteq Q \times Q$,
 $Q = Q_1 \uplus Q_2$, such that



Strong Bisimulation on Probabilistic Automata

Strong bisimulation between A_1 and A_2

Relation $R \subseteq Q \times Q$,
 $Q = Q_1 \uplus Q_2$, such that



$$\mu R \mu' \quad [\text{LS89}]$$

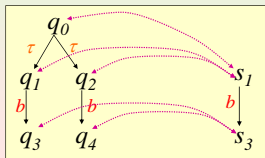
$$\Leftrightarrow \forall C \in \mathcal{Q}/R. \mu(C) = \mu'(C)$$



Weak Bisimulation on Automata

Weak bisimulation between A_1 and A_2

Relation $R \subseteq Q \times Q$,
 $Q = Q_1 \uplus Q_2$, such that



$$\forall q, s, a, q' \exists s'$$

$$q \xrightarrow{a} q' \quad R \quad s \xrightarrow{a} s'$$

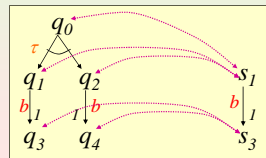
$$s \xrightarrow{a} s' \iff \exists \alpha: \text{trace}(\alpha)=a, \text{fstate}(\alpha)=s, \text{lstate}(\alpha)=s'$$



Weak bisimulation on Probabilistic Automata

Weak bisimulation between A_1 and A_2

Relation $R \subseteq Q \times Q$,
 $Q = Q_1 \uplus Q_2$, such that



$$\forall q, s, a, \mu \exists \mu'$$

$$q \xrightarrow{a} \mu \quad R \quad s \xrightarrow{a} \mu'$$

$$\mu R \mu' \quad [\text{LS89}]$$

$$\forall C \in \mathcal{Q}/R. \mu(C) = \mu'(C)$$



Weak Transition

$$q \xrightarrow{a} \rho$$

There is a probabilistic execution μ such that

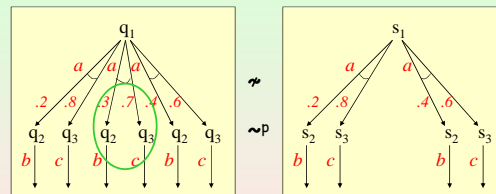
- $\mu(\text{exec}^*) = 1$ (it is finite)
- $\text{trace}(\mu) = \delta(a)$ (its trace is a)
- $\text{fstate}(\mu) = \delta(q)$ (it starts from q)
- $\text{lstate}(\mu) = \rho$ (it leads to ρ)

$$q \xrightarrow{a} s \iff \exists \alpha: \text{trace}(\alpha)=a, \text{fstate}(\alpha)=q, \text{lstate}(\alpha)=s$$



Probabilistic Bisimulations

- These two Probabilistic Automata are not bisimilar

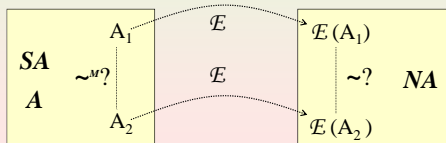


- Yet they satisfy the same formulas of a logic PCTL
 - The logic observes probability bounds on reachability properties
- Bisimilar if we match transitions with convex combinations of transitions



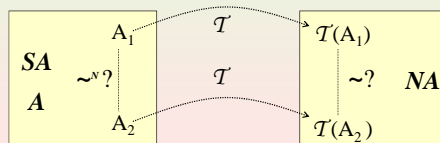
Bisimulation on Alternating Models Mixed Type - Embeddings

- Define a relation on all states
 - So we mix probabilistic and nondeterministic states
- Embed into NA model
 - Embeddings preserve all states
- Check bisimilarity on images in NA

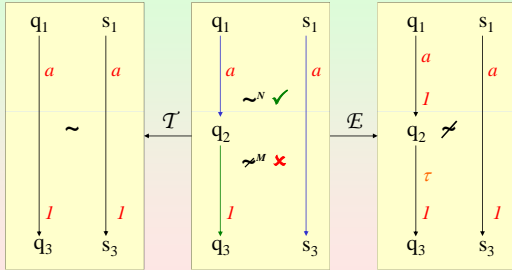


Bisimulation on Alternating Models Nondeterministic Type - Transformations

- Define a relation on nondeterministic states
- Transform into NA model
 - transformations preserve nondeterministic states
- Check bisimilarity on images in NA



Bisimulation on Alternating Models Example



Bisimulation on Alternating Models Literature

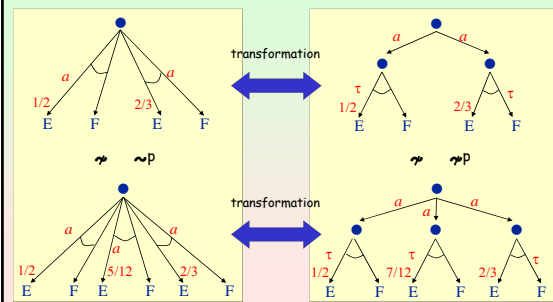
In literature there are also

- Strong bisimulation of Hansson on SA LCMCs
 - Relates only nondeterministic states
- Strong bisimulation of Philippou on A LCMCs
 - Relates all states
 - Probabilistic states are a technicality
- Weak bisimulation of Philippou on A LCMCs
 - Relates all states
 - Probabilistic states are meaningful
 - Uses conditional probabilities on self loop

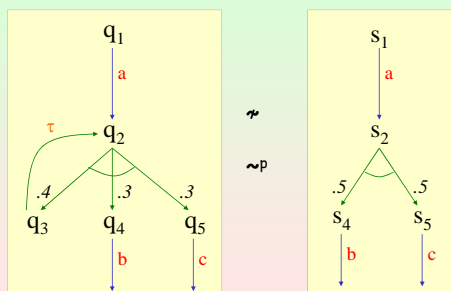
Bisimulation on Alternating Models Connections to Literature

	RA	SA	A
Strong \sim	$\sim^{pM} \sim^M$	$\sim^N \sim^{pM} \sim^M$	\sim^N
Weak \approx			\sim^{pM}

Bisimulation on Alternating Models Examples



Example: Weak on Alternating



Alternating vs. non-Alternating

Theorem

R is a bisimulation on alternating model iff

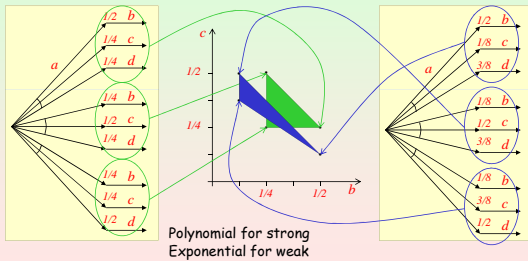
For each s, t , each a , and each equivalence class C

$$\max\{\mu(C), s \xrightarrow{a} \mu\} = \max\{\mu(C), t \xrightarrow{a} \mu\}$$

- Same result for weak bisimulations
- Consequence: efficient decision procedures

Alternating vs. non-Alternating

Previous result does not hold in the non-alternating model



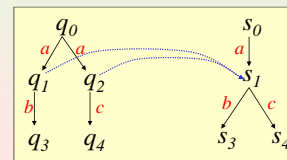
Alternating vs. non-Alternating

- Alternating
 - Efficient decision procedures
 - Maximum probabilities
- Non-Alternating
 - Strong bisimulations
 - Efficient decision procedures
 - Comparison of convex reachability sets
 - More complex than maximum probabilities
 - Weak bisimulations
 - Exponential complexity
 - Extremal points of reachability sets can be exponential

Simulations

Forward Simulations (Automata)

Forward simulation from A_1 to A_2 ($A_1 \leq_F A_2$)
Relation $R \subseteq Q_1 \times Q_2$ such that

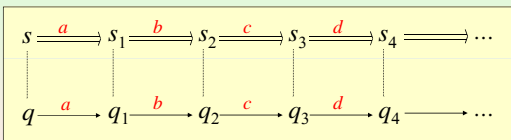


$$\forall q, s, a, q' \exists s'$$

$$\begin{array}{ccc} s & \xrightarrow{a} & s' \\ R \downarrow & & \downarrow R \\ q & \xrightarrow{a} & q' \end{array}$$

Simulation Implies Trace Inclusion

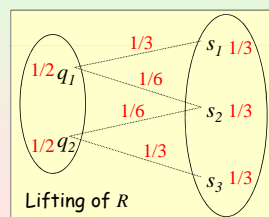
- The step condition can be applied repeatedly



- Thus existence of simulation implies trace inclusion
 - Even more it implies a close correspondence between executions

Forward Simulations

Forward simulation from A_1 to A_2 ($A_1 \leq_F A_2$)
Relation $R \subseteq Q_1 \times Q_2$ such that

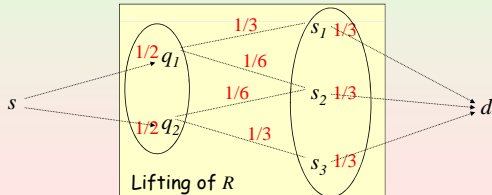


$$\forall q, s, a, \mu' \exists \sigma'$$

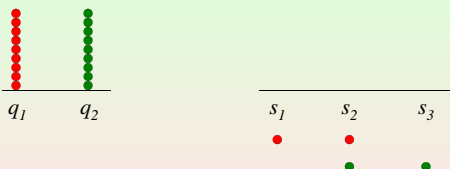
$$\begin{array}{ccc} s & \xrightarrow{a} & \sigma' \\ R \downarrow & & \downarrow R \\ q & \xrightarrow{a} & \mu' \end{array}$$

Considerations about Lifting

- It is the solution of a maximum flow problem
- Alternative characterization
 - $\mu_1 \preceq \mu_2$ iff for each upward closed set X
 - $\mu_1(X) \leq \mu_2(X)$



Lifting and Transfer of Masses



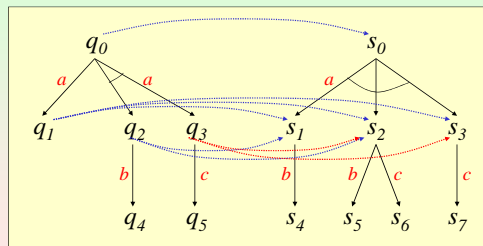
Lifting and joint Measures

$\mu_1 \preceq \mu_2$ iff there exists a probability measure w on $Q_1 \times Q_2$ such that

- support(w) $\subseteq R$
 - That is, $w(s_1, s_2) > 0$ implies $s_1 R s_2$
- $w(Q_1, Q_2) = \mu_1(s_1)$
 - That is, the left marginal is μ_1
- $w(Q_1, Q_2) = \mu_2(s_2)$
 - That is, the right marginal is μ_2

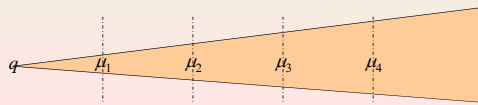
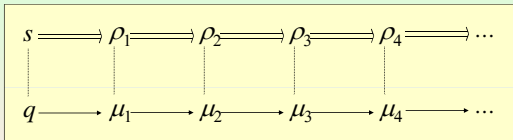


Example: Simulations

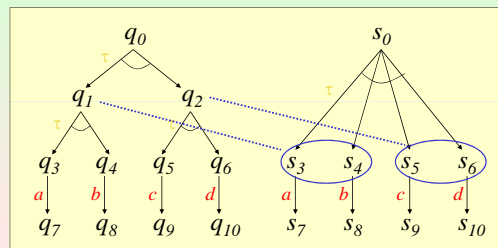


Simulation Implies Trace Inclusion

- The step condition can be applied repeatedly

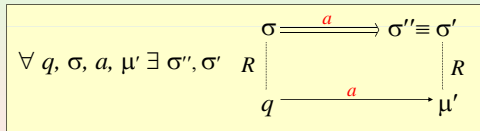


Example: Failure of Weak Forward Simulations



Characterization: Probabilistic Forward Simulations

Forward simulation from A_1 to A_2 ($A_1 \leq_{PF} A_2$)
Relation $R \subseteq Q_1 \times Disc(Q_2)$ such that



Theorem [LSV02] $A_1 \leq_{PF} A_2$ iff $A_1 \leq_{TDC} A_2$



Summing up ... we have seen

- Why formal analysis
- Why Probability and Nondeterminism
- Probabilistic Automata
 - Definition
 - Replace points with measures
 - Replace functions with measurable functions
 - Related Models
- Compositionality
- Language inclusion (equivalence)
- Bisimulations
 - The world is simpler than it seems to be
- Simulations
 - Sound for language inclusion
 - ... and also complete



A Note about Formal Analysis

- Formal methods are too heavy to use
 - Is it reasonable to apply them all the times?
 - Is it reasonable to use them all the times?
 - Is it reasonable to know them?
 - Are automatic tools everything we need?
- Rarely we can be absolutely rigorous
 - We rather limit the places where to use intuition
 - Formal methods give a lot of sanity checks
 - It is useful to be aware of the formal meaning of what we say
 - It is useful to have theoretical results
 - Some doubts can be eliminated quickly
 - Some bugs may be discovered in a few seconds



Thank You



Case Study:

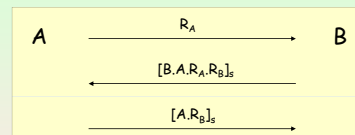
Agent Authentication

Bellare Rogaway 93

Segala, Turrini



Bellare and Rogaway MAP1 Protocol



- Nonces are generated randomly
- The key s is the secret for a Message Authentication Code
 - Specifically, MAC based on pseudo-random functions



Nonces

- Number ONCE
 - Typically drawn randomly
- Claim
 - For each constant c and polynomial p
 - There exists k such that for each $k \geq k$
 - If $n_1, n_2, \dots, n_{p(k)}$ are random nonces from $\{0,1\}^k$
 - Then $\Pr[\exists_{i \neq j} n_i = n_j] < k^{-c}$

Message Authentication Code

- Triple (G, A, V)
 - G on input l^k generates $s \in \{0,1\}^k$
 - For each s and each a
 - $\Pr[V(s, a, A(s, a))=1]=1$
- Forger
 - On input l^k obtains MAC of strings of its choice
 - Outputs a pair (a, b)
 - Successful if $V(s, a, b)=1$ and a different from previous queries
- Secure MAC
 - Every feasible forger succeeds with negligible probability

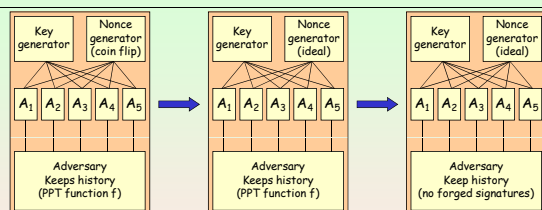
MAP1: Matching Conversations

- Matching conversation between A and B
 - Every message from A to B delivered unchanged
 - Possibly last message lost
 - Response from B returned to A
 - Every message received by A generated by B
 - Messages generated by B delivered to A
 - Possibly last message lost
- Correctness condition
 - Matching conversation implies acceptance
 - Negligible probability of acceptance without matching conversation

MAP1: Correctness Proof

- Let A be a PPT machine that interacts with the agents
- Show that A induces "no-match" with negligible probability
 - Argue that repeated nonces occur with negligible probability
 - Argue that A is an attack against a message authentication code
- Features
 - Relies on underlying pseudo-random functions
 - Proves correctness assuming truly random functions
 - Builds a distinguisher for PRFs if an attack exists
- Criticism
 - The arguments are semi-formal and not immediate
 - Three different concepts intermixed
 - Nonces
 - Message authentication codes
 - Matching conversations

MAP1: Hierarchical Analysis



- Agents indexed by X, Y, \dagger
- Need to find suitable simulations
 - Step conditions lead to local arguments
 - Yet transitions cannot be matched exactly

Nonce Generators

- State
 - $value_{X,Y,t}$ initially \perp
 - $FreshNonces$ initially $\{0,1\}^k$
- Transitions
 - Input $NonceRequest_{X,Y,t}$
 - Effect
 - Let $v \in_R \{0,1\}^k$
 - $value_{X,Y,t} = v$
 - $FreshNonces = FreshNonces - \{v\}$
 - Output $NonceResponse_{X,Y,t}(n)$
 - Precondition
 - $n = value_{X,Y,t}$
 - Effect
 - $value_{X,Y,t} = \perp$

Adversary

- Keeps a variable *history*
 - Holds all previous messages
- Real adversary
 - Runs a cycle where
 - Computes the next message to send using a PPT function f
 - Sends the message
 - Waits for the answer if expected
- Ideal adversary
 - Highly nondeterministic
 - Stores all input
 - Sends messages that do not contain forged authentications

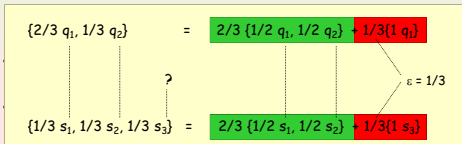
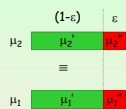
Problems with Simulations

- Problem
 - Consider a transition of the real nonce generator
 - With some probability there is a repeated nonce
 - The ideal nonce generator does not repeat nonces
 - Thus, we cannot match the step
- Solution
 - Match transitions up to some error

Approximate Simulations [ST07]

Change equivalence on measures

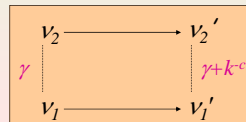
- $\mu_1 \equiv_\epsilon \mu_2$ iff
 - $\mu_1 = (1-\epsilon)\mu_1' + \epsilon\mu_1''$
 - $\mu_2 = (1-\epsilon)\mu_2' + \epsilon\mu_2''$
 - $\mu_1' \equiv \mu_2'$



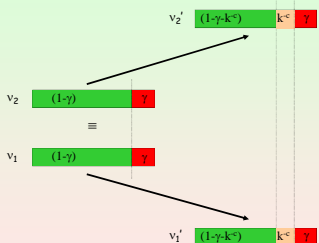
Approximate Simulations

$\{A_i\} \{R_i\} \{B_i\}$

- For each constant c and polynomial p
- There exists k such that for each $k \geq k$
- Whenever
 - v_1 reached within $p(k)$ steps in A_k
 - $v_1 \xrightarrow{L(R_i, \gamma)} v_2$
 - $v_1 \rightarrow v_1'$
- There exists v_2' such that
 - $v_2 \rightarrow v_2'$
 - $v_1' \xrightarrow{L(R_i, \gamma+k^c)} v_2'$

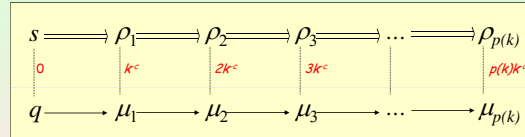


Approximate Simulations Step Condition



Simulation Implies Behavioral Inclusion

- The step condition can be applied repeatedly



- Observation
 - $p(k)k^c$ can be smaller than any k^c by choosing $c = c' + \text{degree}(p)$

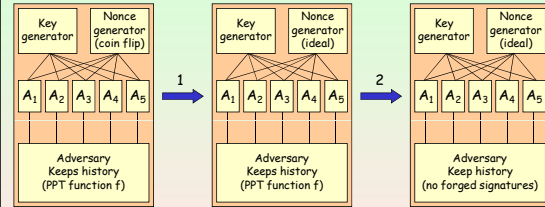
Execution Correspondence under Approximated Simulations

If $\{A_i\} \{R_i\} \{B_i\}$ then

- For each constant c and polynomial p
- There exists k such that for each $k \geq k$
- For each scheduler σ_1
 - v_1 reached within $p(k)$ steps in A_k with σ_1
- There exists σ_2 such that
 - v_2 reached within $p(k)$ steps in B_k with σ_2
 - $v_1 \xrightarrow{L(R_k, p(k)k^{-c})} v_2$
- Observation
 - $p(k)k^{-c}$ can be smaller than any k^{-c} by choosing $c = c' + \text{degree}(p)$



Example: Approximate Simulations Bellare-Rogaway MAP1 Protocol



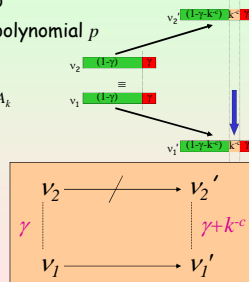
- Negation of the step condition
 - 1: Two random nonces are equal with high probability
 - 2: Function f defines a forger for a signature scheme



Negation of Step Condition

$\{A_i\} \{R_i\} \{B_i\}$

- There exists constant c and polynomial p
- For each k there exists $k \geq k$
- There exists
 - v_1 reached within $p(k)$ steps in A_k
 - $v_1 \xrightarrow{L(R_k, \gamma)} v_2$
 - $v_1 \rightarrow v_1'$
- There is no v_2' such that
 - $v_2 \rightarrow v_2'$
 - $v_1' \xrightarrow{L(R_k, \gamma+k^{-c})} v_2'$
- ~~Signature~~ **Signature** not forged in v_1'
 - Probability at least k^{-c}

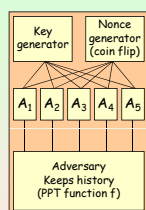


Nonces

- Number ONCE
 - Typically drawn randomly
- Claim
 - For each constant c and polynomial p
 - There exists k such that for each $k \geq k$
 - If $n_1, n_2, \dots, n_{p(k)}$ are random nonces from $\{0, 1\}^k$
 - Then $\Pr[\exists_{i \neq j} n_i = n_j] < k^{-c}$



Problems with Nondeterminism MAP1 Protocol [BR93]



- Authentication protocol
 - Symmetric key signature schema
 - Computational Dolev-Yao
 - Adversary queries agents
- Potential problems
 - Let s be the shared key
 - Adversary queries k agents
 - Agent i replies if i th bit of s is 1
 - The adversary knows the shared key
- Solution
 - One query at a time
 - Wait for the answer (agents as oracles)



More About Approximated Simulations



Conditional Automata

- Let A be a probabilistic automaton
- Let B be a set of bad states
- Let $G = Q - B$ be a set of good states

- Let $A|G$ be the same as A except that
 - $D_{A|G} = \{(q,a,\mu) \mid (q,a,\mu) \in D_A \text{ and } \mu(G) > 0\}$

Theorem

id_Q is a polynomially accurate simulation from A to $A|G$ iff B is negligible

id_Q is a polynomially accurate simulation from $A|G$ to A iff B is negligible

Probabilistic Automata and Equivalences
Bertinoro, June 21, 2010 Roberto Segala - University of Verona 127

A Property of Approximated Lifting

Given a relation R from Q_1 to Q_2

Then $\mu_1 \perp L(R, \varepsilon) \mu_2$ iff there exists

$w: Q_1 \times Q_2 \rightarrow [0,1]$

- w supported on R
- $w(Q_1, Q_2) = 1 - \varepsilon$
- $w(s, Q_2) \leq \mu_1(a)$
- $w(Q_1, s) \leq \mu_2(a)$

Probabilistic Automata and Equivalences
Bertinoro, June 21, 2010 Roberto Segala - University of Verona 128

Approximated Correspondence

This means that ...

Probabilistic Automata and Equivalences
Bertinoro, June 21, 2010 Roberto Segala - University of Verona 129

Transitivity

Claim. $\mu \perp L(R, \varepsilon) \rho$ and $\rho \perp L(R', \tau) \eta$ imply $\mu \perp L(RR', \varepsilon + \tau) \eta$

Probabilistic Automata and Equivalences
Bertinoro, June 21, 2010 Roberto Segala - University of Verona 130

Are approximated simulations transitive?

- We do not know
 - ... but the result of the previous slide suffices

Probabilistic Automata and Equivalences
Bertinoro, June 21, 2010 Roberto Segala - University of Verona 131

Are Approximated Simulations Compositional?

No. Need a more refined relation.

$s \perp S(R, \varepsilon) q$ iff

$\forall q, s, a, \mu' \exists \sigma'$

$s \xrightarrow{a} \sigma'$

$R \varepsilon$

$q \xrightarrow{a} \mu'$

Step condition

For each c there exists k

For each $k > k$, each μ_1, μ_2, γ, w

If $\mu_1 \perp L(R_{k,\gamma}) \mu_2$ via w

then

$\Sigma \{w(q_1, q_2) : q_1 \text{ not } \perp S(R_k, k^c) q_2\} < k^c$

Conditional automata continue to work

Probabilistic Automata and Equivalences
Bertinoro, June 21, 2010 Roberto Segala - University of Verona 132

How About Weak Relations?

- Only one constraint to add
 - Length of matching steps bounded
 - By a constant
 - By a polynomial on length of history



A Note about Formal Analysis

- Formal methods are too heavy to use
 - Is it reasonable to apply them all the times?
 - Is it reasonable to use them all the times?
 - Is it reasonable to know them?
 - Are automatic tools everything we need?
- Rarely we can be absolutely rigorous
 - We rather limit the places where to use intuition
 - Formal methods give a lot of sanity checks
 - It is useful to be aware of the formal meaning of what we say
 - It is useful to have theoretical results
 - Some doubts can be eliminated quickly
 - Some bugs may be discovered in a few seconds



Thank You

