MPRI C.2.3 - Concurrency

Probabilistic models and applications
Lecture 1

Kostas Chatzikokolakis

Dec 13, 2012

Outline of the lectures

- The need for randomization
- Probabilistic automata
- Probabilistic bisimulation
- Probabilistic asynchronous pi-calculus
- Encoding of the pi-calculus into the asynchronous fragment
- Introduction to probabilistic model checking and PRISM
- Verification of anonymity protocols: Dining Cryptographers, Crowds

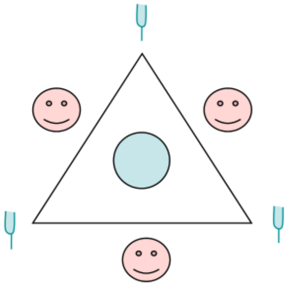
Outline of the lectures

- Dec 13
- Dec 20
- Jan 10
- Jan 17
- Jan 24

Motivation

- Expressiveness: some problems can only be solved through randomization
  - Dining Philosophers
  - Leader election
  - Consensus
  - Anonymity
- Modeling: describe complex phenomena for which we only have an estimation
  - Message loss
  - Failures
  - User behaviour

The dining philosophers problem



- Each philosopher needs two forks
- Each fork is shared by 2 philosophers
- Each philosopher can access one fork at a time

The dining philosophers problem

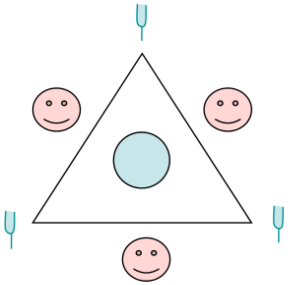
- Goal: an algorithm that guarantees progress:
  - Some philosopher will eventually eat (assuming someone is hungry)
  - No deadlocks or livelocks
- with the following constraints:
  - fully distributed: no central control or memory
  - works for all (fair) schedulers (deciding the order of execution)
  - symmetry: the philosophers run the same code, the initial state is the same

No solution exists satisfying the constraints

Proof (sketch) [Lehmann and Rabin, '81]

- Construct an infinite computation without progress
- Let P_1 be the first philosopher who makes a move
- When P_1 is ready to make a move, the scheduler stops him and runs P_2
- Since the state is symmetric, P_2 will decide to make a symmetric move
- Schedule $P_3, \ldots, P_n$
- Make all moves, the system goes to a new symmetric state
- Eating means that some philosopher will have 2 forks, while some other will have zero. This is impossible without breaking the symmetry

The dining philosophers problem



Solutions violating the constraints:

- centralized control
- no symmetry

Randomized algorithm of Lehmann and Rabin

```
1. Think
2. randomly choose fork in {left,right}   %commit
3. if taken(fork) then goto 3
4.                     else take(fork)
5. if taken(other(fork))  then {release(fork); goto 2}
6.                             else take(other(fork))
7. eat
8. release(other(fork))
9. release(fork)
10. goto 1
```

Randomized algorithm of Lehmann and Rabin

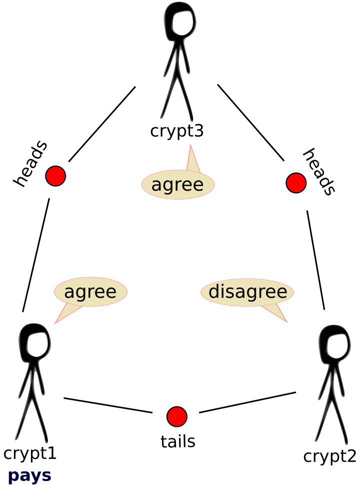
- Assuming a fair scheduler, the randomized algorithm satisfies progress with probability 1
- Repeated random choices break the symmetry with prob. 1
- Infinite runs without progress are still possible but have probability 0

Exercises

- Exercise 1: is it possible to have an algorithm that does not depend on scheduler fairness?
- Exercise 2: Give a solution of the dining philosophers problem (satisfying all constraints) in the π-calculus. Hint: use mixed choice

The dining cryptographers protocol

- Goal: find whether a cryptographer pays without revealing who
- Coins are fair and only visible to adjacent cryptographers
- Announce agree/disagree, the payer says the opposite
- A cryptographer is the payer ⇔ the number of disagrees is odd

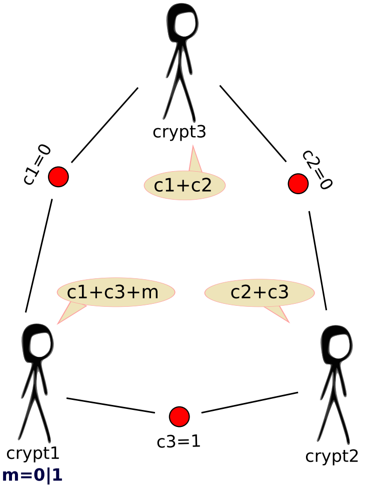


The dining cryptographers protocol

Sending messages:

- Payer: wants to send a message m
- Each user outputs the sum of his coins
- The sender also adds m
- The sum (mod 2) of all announcements is

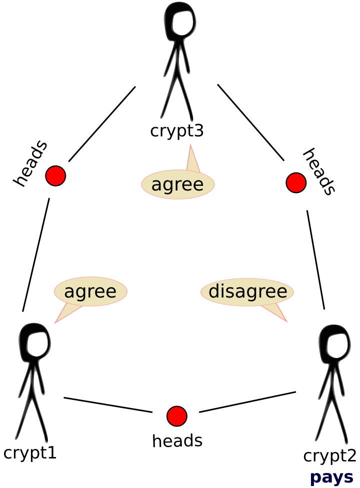
  $$(c_1{+}c_2) + (c_2{+}c_3) + (c_1{+}c_3{+}m) = m$$



Anonymity of the DC protocol (intuition)

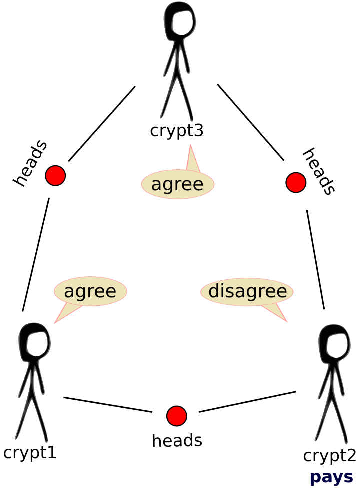
1. Attacker is an outside observer

- Assume that crypt2 is the payer
- This is impossible given the previous coins
- BUT: there is a coin outcome that makes the same announcement valid!
- The attacker cannot distinguish the 2 cases
- The two coin outcomes have the same probability



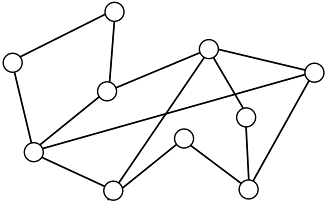
Anonymity of the DC protocol (intuition)

2. Attacker is cryptographer 3
   - Now the attacker knows the 2 coins
   - But he is still confused
   - The coin that makes the announcement valid under crypt2 is not visible to crypt3



Generalized protocol

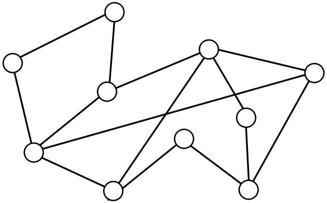
- Any number of users, arbitrary connection graph
- Vertices are cryptographers
- An edge is a coin shared between two cryptographers
- Each cryptographer announces the sum of its adjacent coins, the sender adds m
- sum of all announcements $= m$ (each coin is added twice)



Anonymity of the generalized protocol

1. Attacker is an outside observer

- For any graph we can find a coin outcome that makes any announcement valid under any payer
- Strong anonymity is guaranteed



Anonymity of the generalized protocol

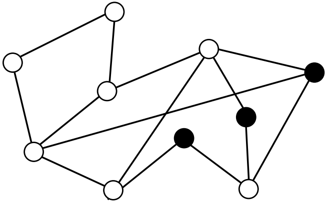
2. Some cryptographers are corrupted

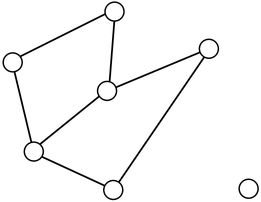
- A cryptographer might be "surrounded" by the attacker

- Anonymity cannot hold

- We remove the corrupted vertices and their edges

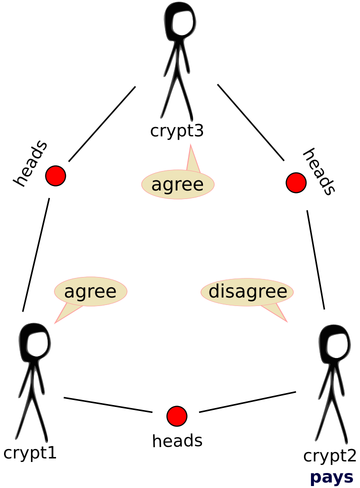
- Strong anonymity holds inside each connected component





DC: biased coins

- Is anonymity satisfied?
- Extreme case: totally biased coins
- From the coins we can find who said the opposite (total loss of anonymity)
- Less extreme: $pb(heads) = 99\%$
- It is still more probable that *crypt*2 is the payer
- Are all cases the same?
  what if $pb(heads) = 51\%$



Probabilistic automata

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

Probabilistic automata

$A = (S, q, A, D)$

- S: set of states (countable)
- $q \in S$: initial state (or distribution on states)
- A: set of actions
- $D \subseteq S \times A \times Disc(S)$: transition relation
- we write $s \xrightarrow{a} \mu$ for $(s, a, \mu) \in D$

Probabilistic automata

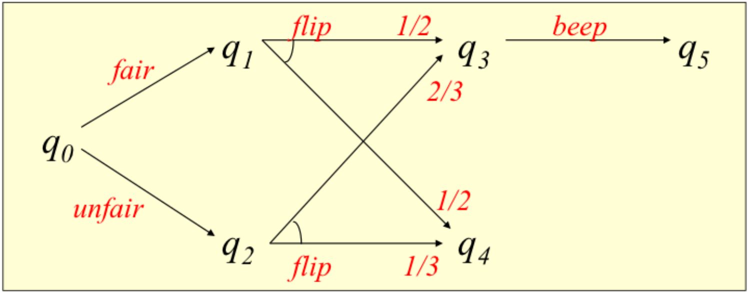
- Special case: Markov Decision Processes

  $D : (S \times A) \rightarrow Disc(S)$

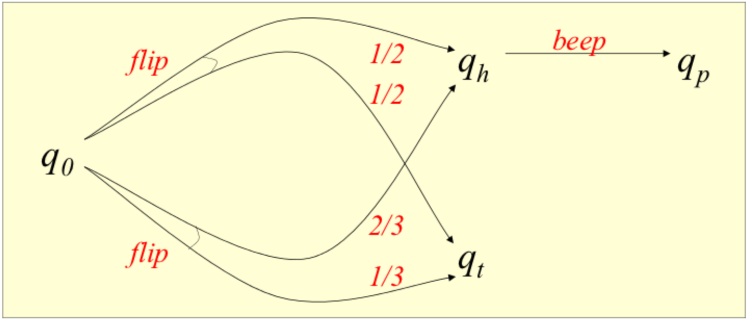
- More abstract model: general probabilistic automata

  $D \subseteq S \times Disc(A \times S)$

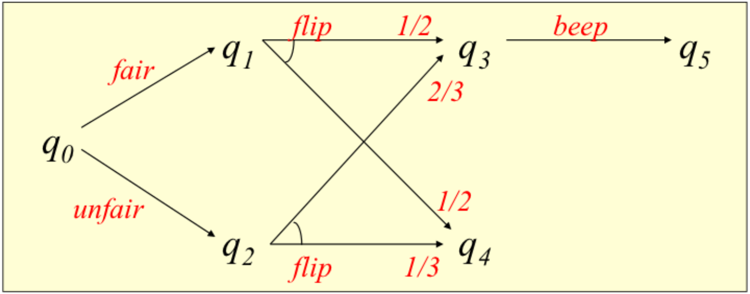
Example



Example

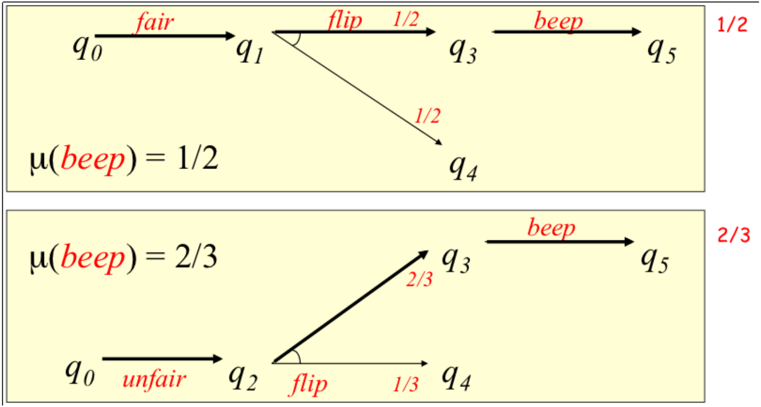


Example



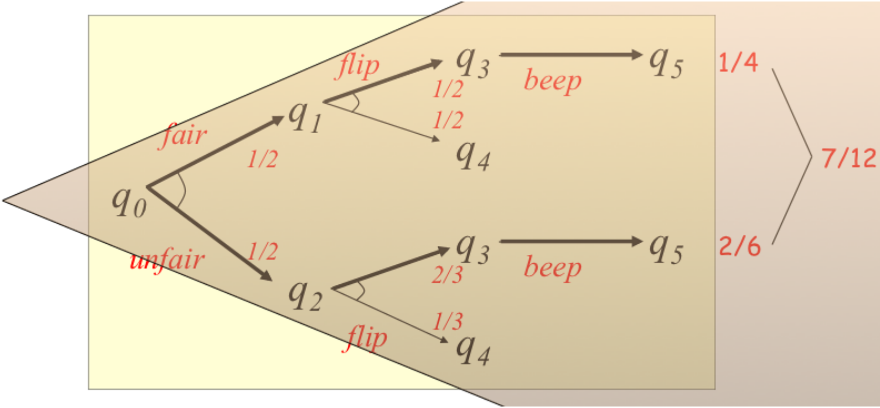
What is the probability of beeping?

Example



q_0
fair
q_1
flip
1/2
q_3
beep
q_5
1/2
1/2
q_4
μ(beep) = 1/2
beep
μ(beep) = 2/3
q_3
q_5
2/3
2/3
q_0
unfair
q_2
flip
1/3
q_4

Example



q0
fair
1/2
q1
flip
1/2
q3
beep
q5
1/4
1/2
q4
7/12
unfair
1/2
q2
2/3
q3
beep
q5
2/6
1/3
flip
q4

Measure theory

- Ω: sample set
- Sigma-algebra
  - $F \subseteq 2^{\Omega}$
  - $\Omega \in F$
  - Closed under complement
  - Closed under countable unions/intersections
- Probability measure on (Ω, F)
  - $\mu : F \to [0, 1]$
  - $\mu(\cup_I X_i) = \sum_I \mu(X_i)$
    for each countable collection $\{X_i\}_I$ of mutually disjoint sets
- Sigma algebra generated by some $F^* \subseteq 2^{\Omega}$
  e.g. Borel algebra

Measure theory

Example: infinite coin tosses

- $\Omega = \{h, t\}^{\infty}$
  set of all infinite sequences of h, t

- What sigma-algebra can allow to define a probability measure on this set?

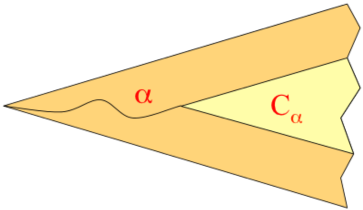
- We want to be able to measure events such as "the first toss is h"

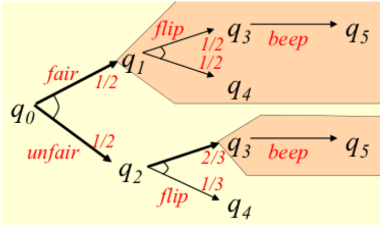
Cones

Cone C_α

- set of executions with prefix α
- represents the fact that "α occurs"
- F generated by the set of all cones

Measure of a cone: product of edges of α





Events expressible by cones

- 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 the 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
  - C_α intersected with the complement of all cones extending α