Distributed Probabilistic Systems

Madhavan Mukund

Chennai Mathematical Institute
http://www.cmi.ac.in/~madhavan

Joint work with
Javier Esparza, R Jagadish Chandra Bose, Sumit Kumar Jha,
Ratul Saha and P S Thiagarajan

ACTS 2017, CMI, 30 January 2017

Overview

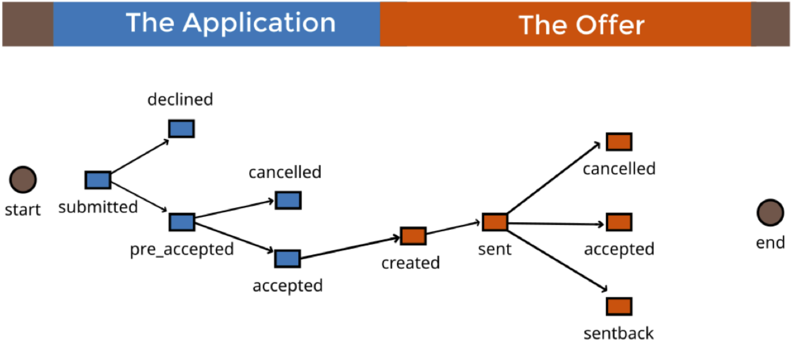
- Probability is a useful way to model uncertainty
- Rich theory of probabilistic systems
  - Markov chains, Markov Decision Processes (MDPs)
- Quantitative analysis
  - Fixed point computations, graph theoretic analysis
  - Statistical methods
- Add time, costs?
- Distributed probabilistic models?
  - State explosion due to parallel components
  - Factorize global probabilities via local transitions
  - Synchronizations through actions: MDPs unavoidable

Resource constrained processes

- A process is a collection of tasks
  - Assembling a car, approving a loan application
- Tasks have logical, temporal dependencies
  - Some tasks may be independent of each other
- Tasks are allocated to resources
  - Items of machinery, desk staff
  - Heterogenous resources — the slow immigration counter
- Cases: Multiple instances of a task
  - Can process in parallel, but contention for resources
  - Arrival pattern

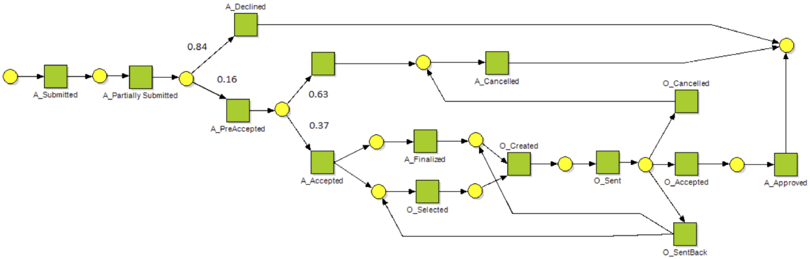
An individual case

Loan application

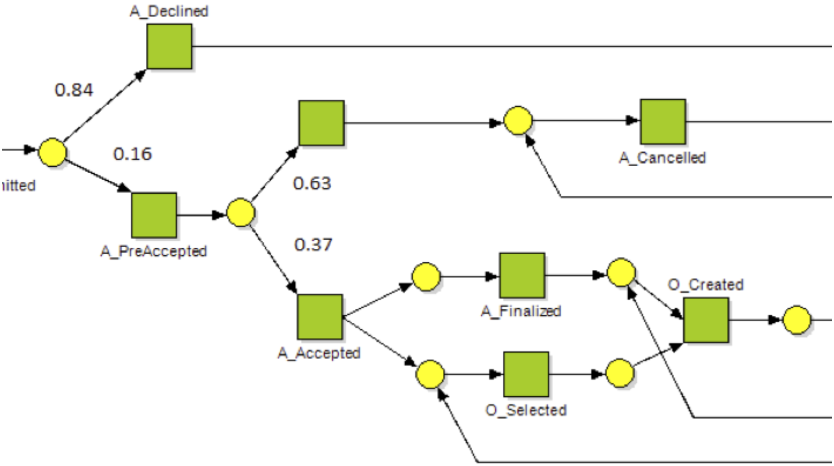


The full story

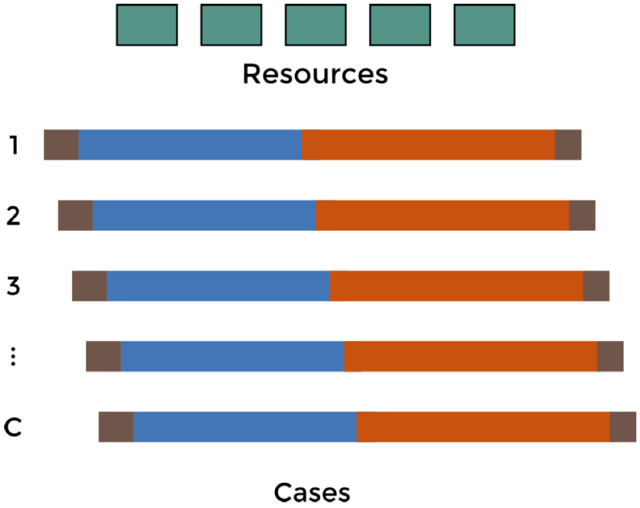
- Causality and concurrency — like a Petri net
- Derive probabilities from past history



The full story . . .



Resource constrained cases



Towards a formal model

- Tasks and resources are agents
- Agents interact
  - Task-task causal dependency
  - Allocation of task to a resource
- Each interaction can have a duration and a cost

Typical question

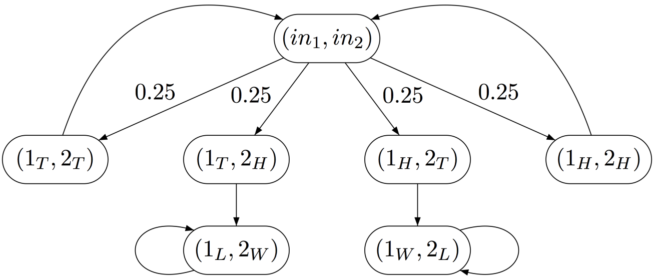
- C cases arrive at λ cases per second.
- Do at least $x\%$ complete within time t, with probability at least p?

Probabilistic asynchronous automata

- Local components $\{1, 2, \ldots, n\}$, with local states S_i
- For $u = \{i, j, k, \ldots\}$, $S_u = S_i \times S_j \times S_k \times \cdots$
- Set of distributed actions A
- Each action a involves subset of agents: $loc(a) \subseteq \{1, 2, \ldots, n\}$
- Transition relations: $\Delta_a \subseteq S_{loc(a)} \times S_{loc(a)}$
- With each a event $e = (u, v)$, associate a cost $\chi(e)$ and a delay $\delta(e)$
  - For simplicity, delay is a fixed quantity
- Assign a probability distribution across all a-events $(u, v_1), (u, v_2), \ldots, (u, v_k)$ from same source state u

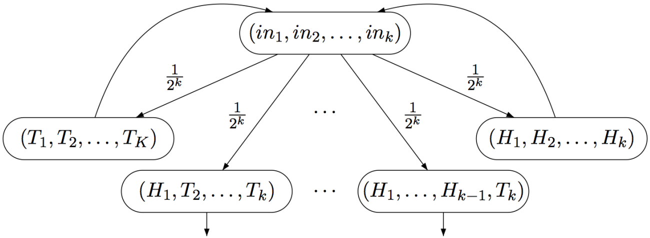
Succinctness advantage

- Two players each toss a fair coin
- If the outcome is the same, they toss again
- If the outcomes are different, the one who tosses Heads wins



Succinctness advantage . . .

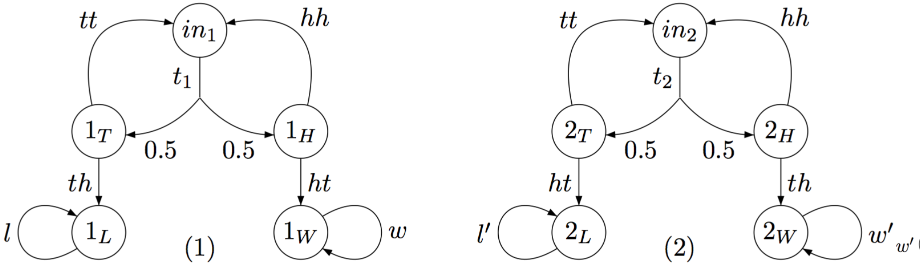
- What if there were k players?



- k parallel probabilistic moves generate 2^k global moves

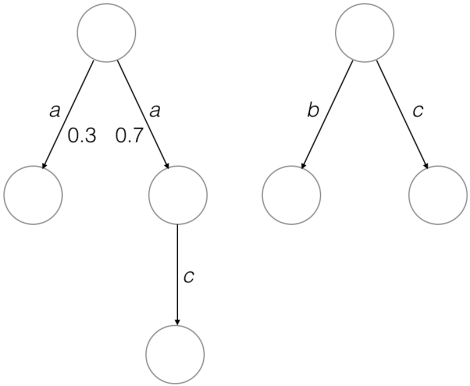
Distributed model for coin toss

- Decompose into local components
- Coin tosses are local actions, deciding a winner is synchronized action



Resolving non-determinism

What is the probability of observing *ab*?



Distributed Markov Chains

- Structural restriction on state spaces, transitions
- Agent i in local state s_i always interacts with a fixed set of partners
- Previous example violates this
- Each run is a Mazurkiewicz trace
- Fix a canonical maximal step interleaving (Foata normal form)
- Each finite trace has a probability derived from underlying events
- Combine to form a Markov chain
- Though restricted, can model distributed protocols like leader election

Distributed Probabilistic Systems

- Alternatively, work with schedulers
- Traditional MDP analysis analyzes best-case or worst-case behaviour across all possible schedulers
- In applications such as business processes, schedulers are typically simple
  - Round-robin
  - Priority based
  - ...
- Fix such a scheduling strategy and analyze

Defining schedulers

- At each global state u, some set of actions $en(u)$ is enabled
- A subset of actions is schedulable if the participating agents are pairwise disjoint
- Without delays on events, can define a global scheduler and execute maximal steps
- With delays, steps end at different time points
- Scheduler should make decision at each relevant time point respecting concurrency

Snapshots

- A snapshot (s, U, X) is a global state with information about events in progress
  - s is a global state
  - U is a set of actions currently in progrews
  - X has an entry (a, e, t) for each $a \in U$, where
    - e is the event probabilistically chosen for a
    - t is the time left for e to complete—recall that e has associated delay $\delta(e)$
- Events in X can be sorted by finishing time
- Choose the subset Y that will finish earliest, say in time t'
- Update (s, U, X) accordingly
  - Reduce time for all unfinished events in X by t'

Schedulers and snapshots

- Scheduler has to choose a subset of $en(s)$ at each snapshot (s, U, X)
- Choice should respect concurrency
  - State is updated only when an event completes
  - Actions in progress, U, must continue to be scheduled
- Demand that scheduler chooses a subset of $en(s)$ that includes all of U

Claim

Under such a scheduler, a distributed probabilistic system describes a Markov Chain

Analysis

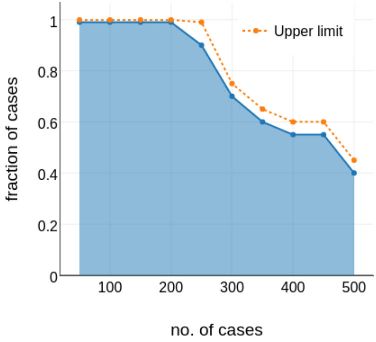
Typical question

- C cases arrive at λ cases per second.
- Do at least $x\%$ complete within time t, with probability at least p?

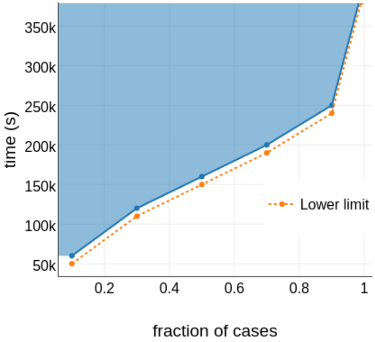
- Statistical model checking
  - Simulate system and check fraction of runs that meet the requirement
- Statistical probabilistic ratio test (SPRT) determines number of simulations required to validate property within a desired confidence bound

Experiments

The loan processing example



Fixed time bound



Fixed number of cases

Extensions

- Stochastic delays
- Analysis based on cost and time
- Structural reduction rules (a la negotiations)
- More sophisticated analysis of schedulers
- ...

References

- Distributed Markov Chains
  R Saha, J Esparza, S K Jha, M Mukund and P S Thiagarajan
  Proc. VMCAI 2015, Springer LNCS 8931 (2015) 117–134.

- Time-bounded Statistical Analysis of Resource-constrained Business Processes with Distributed Probabilistic Systems
  R Saha, M Mukund and R P J C Bose
  Proc. SETTA 2016, Springer LNCS 9984 (2016) 297–314