Quantitative Model Checking of Resource-constrained Business Processes

Ratul Saha
National University of Singapore

with, *Madhavan Mukund* and *R P J C Bose*

The Problem

- Assume a loan providing business
- Thousands of cases come in a day
- Such systems involve probability, nondeterminism and concurrency

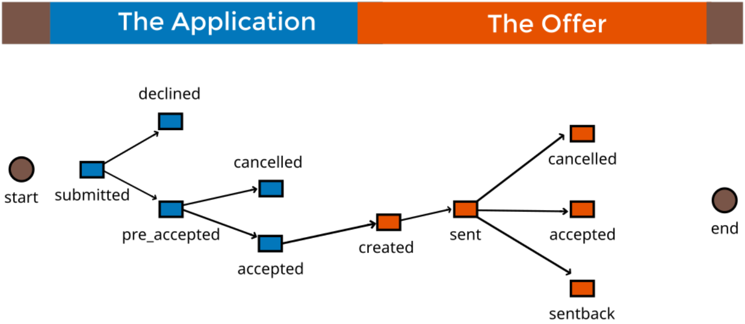
Primary question:
will each case eventually be resolved in time?

The Problem

A complex question:
do most cases get resolved in time?

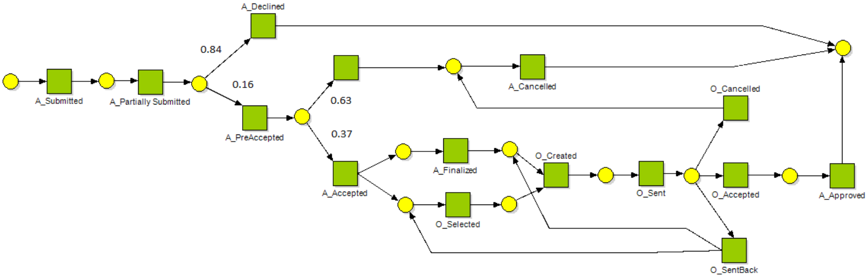
→ When C cases arrive
at a rate of λ per second,
with probability $\geq p$,
at least $x\%$ are completed within time t.

An individual case

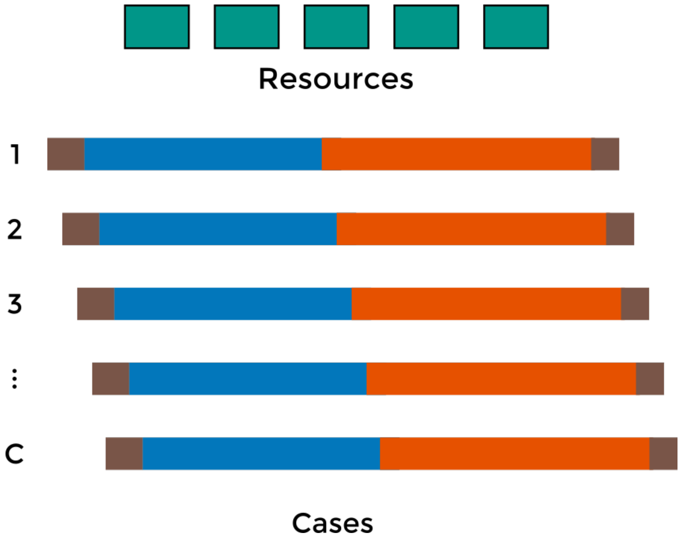


(some edges are omitted)

The Real-life Case



Resource-constrained Process



Resource-constrained Processes

An Individual case

- → Start state, finite set of tasks and final state
- → Tasks are combined in sequence or parallel
- → Probabilistic branches

Finite resources

- → Resource allocation strategies are simple
- → Often priority-based (platinum/gold/silver)

State-of-the-art Approach

→ Simulate an arbitrary number of times

→ Claim that the probability $\geq p$ if p fraction satisfies the property

→ No sample size or error bound analysis

“simulation does not provide any proof”
– *Van der Aalst* on using simulation for analyzing processes

Our Approach

→ Let’s use quantitative model checking

→ Rigorous modeling formalism

→ Statistical model checking for simulation

First Step: The Model

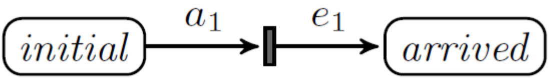
→ model tasks and resources as agents

→ agents interact among each other

→ fixed-duration real-time and cost

→ resource-contention brings nondeterminism

The Model



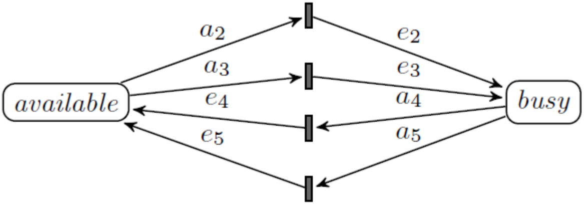
(i) the starter.

a: action

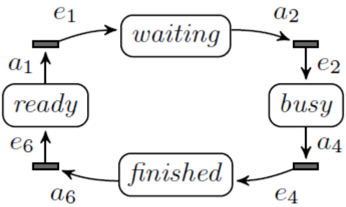
e: event

p_e: probability, δ_e: duration

The Model



(ii) the resource



(iii) task `A_Submitted`

The Model

- → r resources, C cases each with k_T tasks
- → modeled with $r + C \times (k_T + 2)$ agents
- → Global state: product of local state of agents
- → A set of actions are enabled at a global state
- → Some of them are schedulable

Defining Schedulers

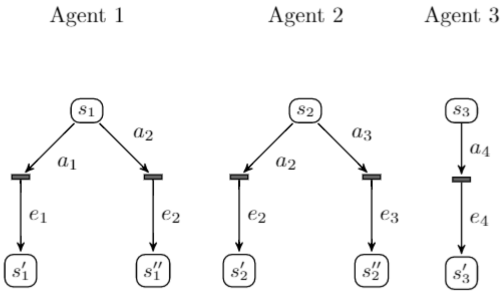
→ Given a global state, multiple set of schedulable actions

→ Need a scheduler to resolve it

→ Not trivial due to concurrency

→ Scheduler has to “obey” concurrency

Defining Schedulers



enabled: a_1, a_2, a_3, a_4
schedulable: $\{a_1, a_3, a_4\}$, $\{a_2, a_4\}$...

Snapshots

A snapshot includes

→ a global state

→ a list of partially executed events

→ with their time to completion from the current time point

Schedulers are defined in the snapshot space

Schedulers

→ The scheduler respects the partially executed events, by design

→ That is the only restriction

→ Finite memory schedulers

→ Given a scheduler, the transition system is a countably-infinite Markov chain

Defining Transition

- Given a snapshot
  - take the scheduled actions,
  - set of events are chosen probabilistically
  - those with minimum duration will complete first
  - that defines the transition and the next snapshot
  - The probability is the product of chosen event probabilities

Simulation with Guarantees

→ Use traditional statistical model checking

→ Sample size analysis

→ Error of simulation is bounded upfront

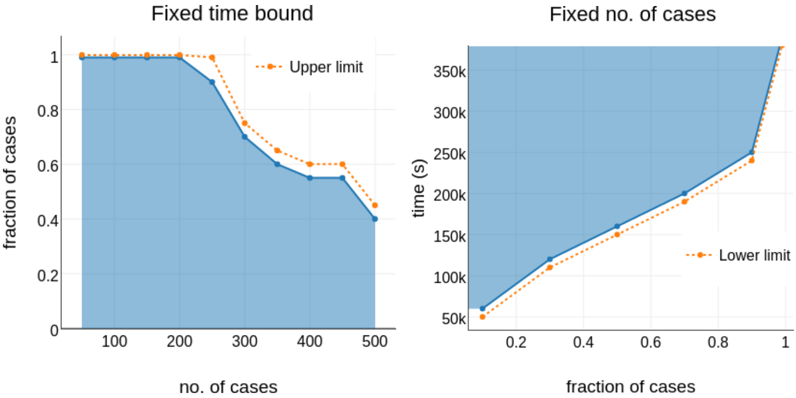
Experiments

We are interested in:

When C cases arrive at 1 case per 10s,
with probability ≥ 0.99,
x fraction are completed
within t seconds.

Let's play with x, C, and t.

Experiments



(left) time bound is fixed, (right) total number of cases is fixed

Future Directions

→ Integrate with process mining

→ Compare schedulers and suggest improvements

→ Not much ‘business’ about this technique – apply to generic processes

Thank you!

Questions?