

07Lab

Stochastic modelling: uncertainty, probability and frequencie

Mirko Viroli

`mirko.viroli@unibo.it`

C.D.L. Magistrale in Ingegneria e Scienze Informatiche
ALMA MATER STUDIORUM—Università di Bologna, Cesena

a.a. 2023/2024

One slide sum-up on Stochastic Modelling

Concepts

- evolve our reference meta-meta-model of transition systems in two directions
- DTMC: adding probabilities to “quantify non-determinism”, still in a discrete-time setting
 - ▶ this will later be extended to deal with rewards, hence Reinforcement Learning
- CTMC: adding markovian rates to “quantify continuous and probabilistic passage of time”
 - ▶ this will later be extended to deal with event-based simulations

CTMC analysis

- a CTMC can be analysed by probabilistically drawing possible traces: that’s actually a simulation
- will later introduce the concept of probabilistic approximate model-checking

CTMC-based meta-models

- adding rates to PN gives SPN (rates can linearly depend on number of tokens in incoming places)
- SPN are interesting: they model chemistry, bio-chemistry, and possibly also distributed networks

Starting point and goals

References

- 07-repo: from virtuale (<https://github.com/mviroli/asmd23-public-models>)

General goals

- get acquainted with CTMCs, their simulation, and meta-models
- play with SPN
- reason about probability, time, and testing

Tasks

SIMULATOR

Take the communication channel CTMC example in `StochasticChannelSimulation`. Compute the average time at which communication is done—across n runs. Compute the relative amount of time (0% to 100%) that the system is in fail state until communication is done—across n runs. Extract an API for nicely performing similar checks.

GURU

Check the SPN module, that incorporates the ability of CTMC modelling on top of Petri Nets, leading to Stochastic Petri Nets. Code and simulate Stochastic Readers & Writers shown in previous lesson. Try to study how key parameters/rate influence average time the system is in read or write state.

CHEMIST

SPNs can be used to simulate dynamics of chemical reactions. Experiment with it. E.g.: search the “Brussellator” chemical reaction on wikipedia: it oscillates! Try to reproduce it.

RANDOM-UNIT-TESTER

How do we unit-test with randomness? And how we test at all with randomness? Think about this in general. Try to create a repeatable unit test for `Statistics` as in `utils.StochasticSpec`.

PROBABILITY-LLM

We know that LLMs/ChatGPT can arguably help in write/improve/complete/implement/reverse-engineer standard ProgLangs. But is it of help in taking into account probability? Seemingly, it shorty fails. But with the proper prompt, it might say something reasnable.