FMSE, Lecture 8:

Stochastic Process Algebra

Rom Langerak

- Markov processes
- Interactive Markov Processes (IMC)
- weak and strong bisimulation laws

Random variables, distributions

Continuous random variable D:

- may assume all values in some real interval. We assume a time interval $[0, \infty)$.
- has a probability density function f.

Examples: arrival time, departure time, service time, time before breakdown, ...

Interpretation: for a small time interval Δ,

$$P(t \leq D \leq t + \Delta) = \Delta \cdot f(t)$$

so the cumulative distribution:

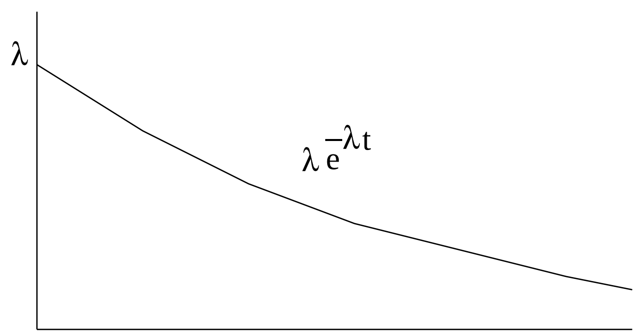
$$F(a) = P(D \leq a) = \int_0^a f(t)dt$$

Many different distribution functions exist in the stochastic literature.

Exponential distribution

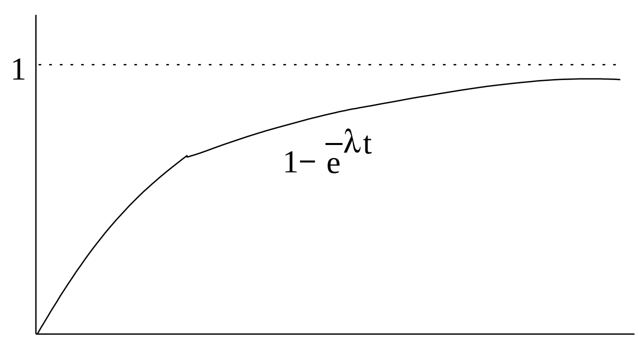
A stochastical variable D has exponential distribution if the density function is

$$f(t) = \lambda e^{-\lambda t}$$



Cumulative distribution:

$$F(a) = \int_0^a \lambda e^{-\lambda t} = -e^{-\lambda t}|_0^a = 1 - e^{-\lambda a}$$



Note that $P(D > a) = 1 - F(a) = e^{-\lambda a}$

Memoryless property

Expected time:

$$\int_0^\infty t \cdot f(t)dt = \int_0^\infty t\lambda e^{-\lambda t} = 1/\lambda$$

so $1/\lambda$ is the mean time per event, so the *rate* = number of events per time unit = λ

The exponential distribution is the only *memoryless* distribution, i.e. the probability to wait a certain amount of time is independent of how long has been waited already:

$$P(D > t + d \,|D > t) = P(D > d)$$

Examples: random autobus service, waiting for someone in a phone booth.

Exponential distribution often assumed because of its nice mathematical properties.

Markov chains

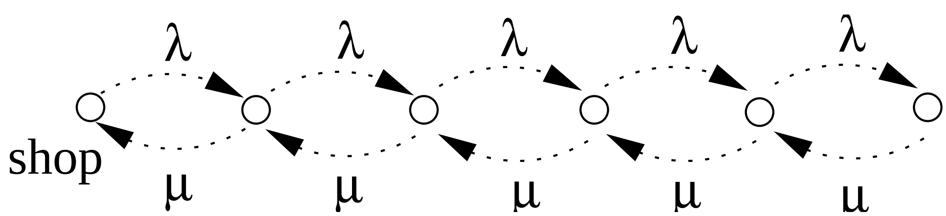
Continuous Time Markov Chain (CTMC):

- states
- transitions, each labelled with a rate

Each rate λ characterizes an exponential distribution, i.e. $P(D > t) = e^{-\lambda t}$ where D is the time at which the transition is taken.

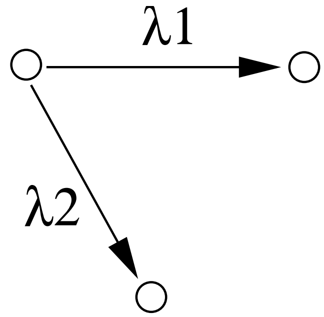
Example: a shop

- on average 1 customer per 5 minutes (so arrival rate $\lambda = 1/5$)
- average service time 3 minutes (so service rate $\mu = 1/3$)
- maximum number of 5 customers in the shop



Race conditions

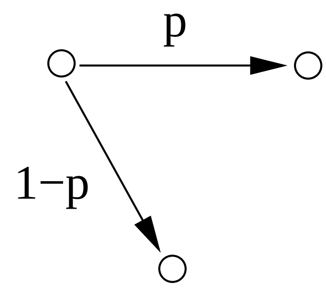
Two outgoing transitions with rate $\lambda 1$ and $\lambda 2$:



Chance that a transition is taken after time t:
$P(D1 > t \wedge D2 > t) = e^{-\lambda 1 t} e^{-\lambda 2 t} = e^{-(\lambda 1 + \lambda 2) t}$
so total outgoing rate: $\lambda 1 + \lambda 2$.

Chance of taking the $\lambda 1$ transition:
$P(D1 < D2) = \lambda 1 / (\lambda 1 + \lambda 2)$, and similarly
$P(D2 < D1) = \lambda 2 / (\lambda 1 + \lambda 2)$

So in this way a probabilistic choice can be modelled:



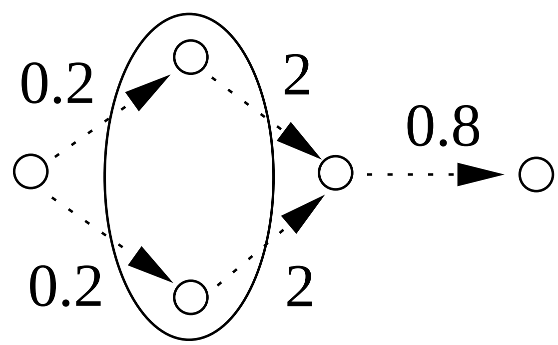
with $p = \lambda 1 / (\lambda 1 + \lambda 2)$

Bisimulation for CTMC's

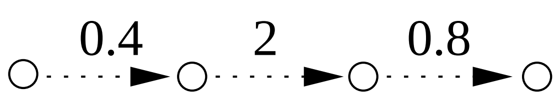
A bisimulation relation can be defined for CTMC's:

similar to transition systems BUT: in addition, the cumulative rate from a state to a set of states has to be taken into account.

Example:



bisimilar to



In the stochastic literature this is covered by the concept of *lumpability*.

Performance modelling

- CTMC well investigated model for performance analysis
- widely used in practice
- efficient numerical algorithms

But:

- performance modelling an art, depending on experience
- very complex if there are many components
- problem: compatibility of functional and performance model

Therefore:
find a compositional language (process algebra!) that combines functional specification with performance information.

Interactive Markov Chains

Basic operators of IMC:

inaction	0
action prefix	$a.E$
delay prefix	$(\lambda).E$
choice	$E + F$
process instantiation	X
process definition	$[X := E]$

Example of an IMC expression:

$$a.(\lambda).(\mu).a.(\mu).0$$

The nature of choice

What does

$$a.P + (\lambda).Q$$

mean?

Maximal Progress assumption:

an action happens as soon as it is enabled - but here we do not know if a is enabled (may depend on environment)

But:

τ is always enabled, so we have the axiom:

$$(\lambda).E + \tau.F = \tau.F$$

(so the τ always happens, while the delay is discarded)

Strong bisimulation laws

For process algebra:

$$E + F = F + E$$

$$(E + F) + G = E + (F + G)$$

$$E + E = E$$

$$E + 0 = E$$

For IMC:

$$E + F = F + E$$

$$(E + F) + G = E + (F + G)$$

$$E + 0 = E$$

$$a.E + a.E = a.E$$

$$(\lambda).E + (\mu).E = (\lambda + \mu).E$$

$$(\lambda).E + \tau.F = \tau.F$$

recognize the race condition, and the maximal progress assumption

Operational semantics

Two types of transitions:

- action transitions: $\xrightarrow{a}$
- delay transitions: $\overset{\lambda}{\dashrightarrow}$

Axiom:

$$a.E \xrightarrow{a} E$$

Rules:

$$\frac{E \xrightarrow{a} E'}{E + F \xrightarrow{a} E'}$$

$$\frac{F \xrightarrow{a} F'}{E + F \xrightarrow{a} F'}$$

Operational semantics (2)

Axiom:

$$(\lambda).E \xrightarrow{\lambda} E$$

Rules:

$$\frac{E \xrightarrow{\lambda} E',\ F \not\xrightarrow{\tau}}{E + F \xrightarrow{\lambda} E'}$$

$$\frac{F \xrightarrow{\lambda} F',\ E \not\xrightarrow{\tau}}{E + F \xrightarrow{\lambda} F'}$$

The conditions $F \not\xrightarrow{\tau}$ resp. $E \not\xrightarrow{\tau}$ are necessary because of the maximal progress assumption.

Operational semantics (3)

Process definition and instantiation:

Rules:

$$\frac{E\{[X := E]/X\} \xrightarrow{a} E'}{[X := E] \xrightarrow{a} E'}$$

$$\frac{E\{[X := E]/X\} \stackrel{\lambda}{\dashrightarrow} E'}{[X := E] \stackrel{\lambda}{\dashrightarrow} E'}$$

Example derivation:

$$(\lambda).[X := a.X] \stackrel{\lambda}{\dashrightarrow} [X := a.X]$$

Now

$$a.X\{[X := a.X]/X\} = a.[X := a.X]$$

so

$$\frac{a.[X := a.X] \xrightarrow{a} [X := a.X]}{[X := a.X] \xrightarrow{a} [X := a.X]}$$

so $(\lambda).[X := a.X] \stackrel{\lambda}{\dashrightarrow} \xrightarrow{a} \xrightarrow{a} \xrightarrow{a} \ldots$

Weak bisimulation laws

Process algebra: laws for strong bisimulation, plus:

$$a.\tau.E = a.E$$

$$E + \tau.E = \tau.E$$

$$a.(E + \tau.F) + a.F = a.(E + \tau.F)$$

IMC: IMC laws for strong bisimulation, plus:

$$a.\tau.E = a.E$$

$$(\lambda).\tau.E = (\lambda).E$$

$$E + \tau.E = \tau.E$$

$$a.(E + \tau.F) + a.F = a.(E + \tau.F)$$

Weak bisimulation (2)

- note that $E + \tau.E = \tau.E$ needs no special cases for delays and actions
- note that $(\lambda).(E + \tau.F) + (\lambda).F \neq (\lambda).(E + \tau.F)$ as lefthand side has outgoing rate 2λ, whereas righthand side has rate λ

Strong and weak bisimulation can defined in a similar way as for process algebra

(but like bisimulation for CTMC's, cumulative outgoing rates have to be taken into consideration, see paper)

Parallel composition

$P||a_1 \dots a_n||Q$ (similar to LOTOS).

Rules:

$$\frac{P \xrightarrow{a} P' \quad a \notin \{a_1 \dots a_n\}}{P||a_1 \dots a_n||Q \xrightarrow{a} P'||a_1 \dots a_n||Q}$$

$$\frac{Q \xrightarrow{a} Q' \quad a \notin \{a_1 \dots a_n\}}{P||a_1 \dots a_n||Q \xrightarrow{a} P||a_1 \dots a_n||Q'}$$

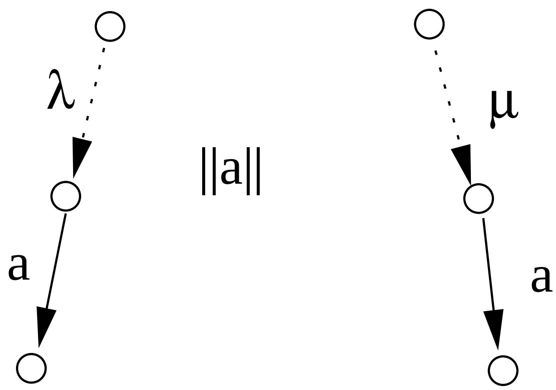
$$\frac{P \xrightarrow{a} P' \quad Q \xrightarrow{a} Q' \quad a \in \{a_1 \dots a_n\}}{P||a_1 \dots a_n||Q \xrightarrow{a} P'||a_1 \dots a_n||Q'}$$

$$\frac{P \overset{\lambda}{\dashrightarrow} P' \quad Q \not\xrightarrow{\tau}}{P||a_1 \dots a_n||Q \overset{\lambda}{\dashrightarrow} P'||a_1 \dots a_n||Q}$$

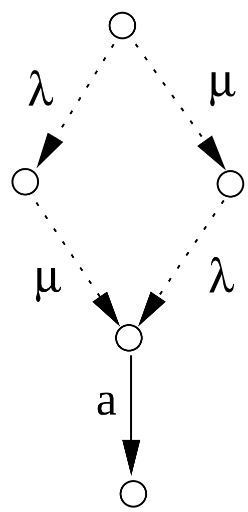
$$\frac{Q \overset{\lambda}{\dashrightarrow} Q' \quad P \not\xrightarrow{\tau}}{P||a_1 \dots a_n||Q \overset{\lambda}{\dashrightarrow} P||a_1 \dots a_n||Q'}$$

(negative conditions: maximal progress)

Interleaving of delay actions



results in



This is correct, because

- first the earliest transition happens
- then the second transition happens, but because of the memoryless property, its delay starts after the first transition

Abstraction

$$\textbf{hide } a_1 \ldots a_n \textbf{ in } P$$

(called *hiding* in LOTOS)

Rules:

$$\frac{P \xrightarrow{a} P' \quad a \notin \{a_1 \ldots a_n\}}{\textbf{hide } a_1 \ldots a_n \textbf{ in } P \xrightarrow{a} \textbf{hide } a_1 \ldots a_n \textbf{ in } P'}$$

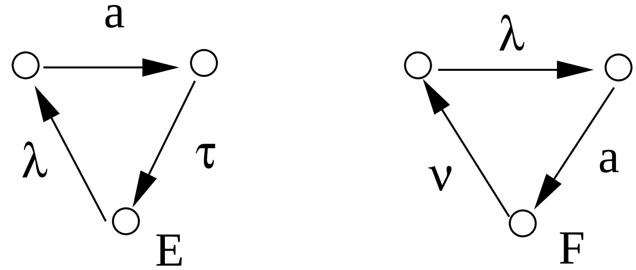
$$\frac{P \xrightarrow{a} P' \quad a \in \{a_1 \ldots a_n\}}{\textbf{hide } a_1 \ldots a_n \textbf{ in } P \xrightarrow{\tau} \textbf{hide } a_1 \ldots a_n \textbf{ in } P'}$$

$$\frac{P \overset{\lambda}{-\!\!\!\longrightarrow} P' \quad \textbf{hide } a_1 \ldots a_n \textbf{ in } P \not\xrightarrow{\tau}}{\textbf{hide } a_1 \ldots a_n \textbf{ in } P \overset{\lambda}{-\!\!\!\longrightarrow} \textbf{hide } a_1 \ldots a_n \textbf{ in } P'}$$

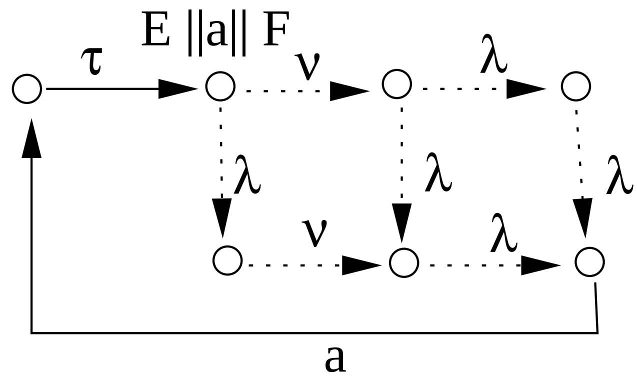
(negative condition: maximal progress)

Example: obtaining a CTMC

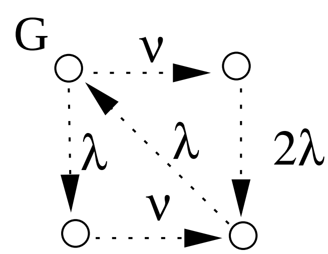
The following two processes are synchronized:



The result:



If a is hidden, this is weak bisimulation equivalent with



So hiding all actions and applying weak bisimulation may lead to a CTMC (possible problem: nondeterminism)

Example: shop

We specify the shop in IMC (using some obvious syntactic extensions):

hide $enter, serve$ **in**
$Customer \ ||enter|| \ (Shop(0) \ ||serve|| \ Clerk)$

$Customer := (\lambda).enter.Customer$

$$Shop(i) := [i < 5]-> enter.Shop(i+1)$$
$$[i > 0]-> serve.Shop(i-1)$$

$Clerk := serve.(\mu).Clerk$

This specification is weak bisimulation equivalent to the shop on slide 5.

Conclusions

- Compositional process algebraic framework for Interactive Markov Chains
- integration of functional specification and performance analysis
- resolving nondeterminism, hiding all actions and performing weak bisimulation leads to a CTMC that can be analysed in the usual way
- weak bisimulation may lead to an enormous decrease of the size of a CTMC
- other approaches: PEPA, TIPP, EMPA. Combine delays with actions; problem: what is the delay of a synchronised action?