# Model checking stochastic properties

Istvan Majzik
majzik@mit.bme.hu

**Budapest University of Technology and Economics**
**Dept. of Measurement and Information Systems**

# Motivation: Service quality properties

- Properties beyond state reachability
  - QoS: Quality of Service
  - SLA: Service Level Agreement
- Examples for complex QoS properties:
  - It happens with probability less than 0.2 that the recovery after an error needs more than 15 time units
  - It happens with probability less than 0.5 that after start in 85 time units the service level decreases below Minimum
  - Its probability is more than 0.7 that reaching the service level Minimum it is possible to deliver service level Premium in 5 time units
- Characteristics of QoS properties:
  - Probabilities of states / scenarios (e.g., service levels, recovery)
  - Time to reach states / execute scenarios (e.g., repair)

# Extensions of “classic” temporal logics

## Stochastic logics:

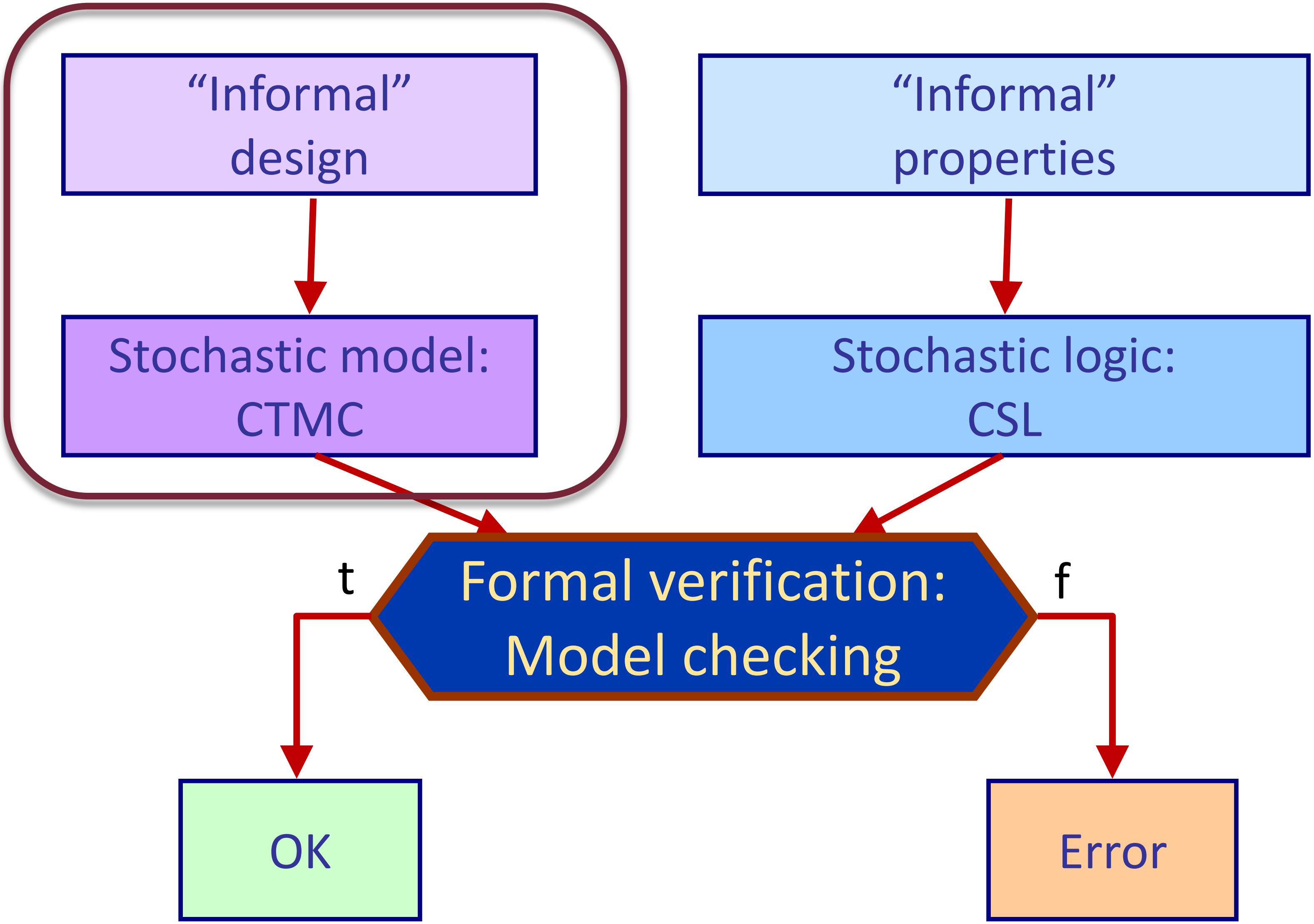
- Probability and timing related requirements:
  - E.g.: if the current state is Error then the probability
    that this condition holds after 2 time units as well, is less than 0.3
- Extension of CTL:
  - Interpreted over Continuous-time Markov chains (not a Kripke structure)
  - Probability criteria for state reachability (steady state), path execution
  - Timing criteria (time intervals) for operators X and U

## Real-time logics:

- Requirements of real-time systems
  - The logic can reference clock variables
  - Handling of time intervals

# Modeling stochastic processes

# Formal verification of stochastic properties



# Stochastic models (overview)

- Used to model performance and dependability
  - Stochastic Petri-nets
  - Stochastic process algebra
  - Stochastic activity networks

Assigning timing (with exponential distributions) to the activities

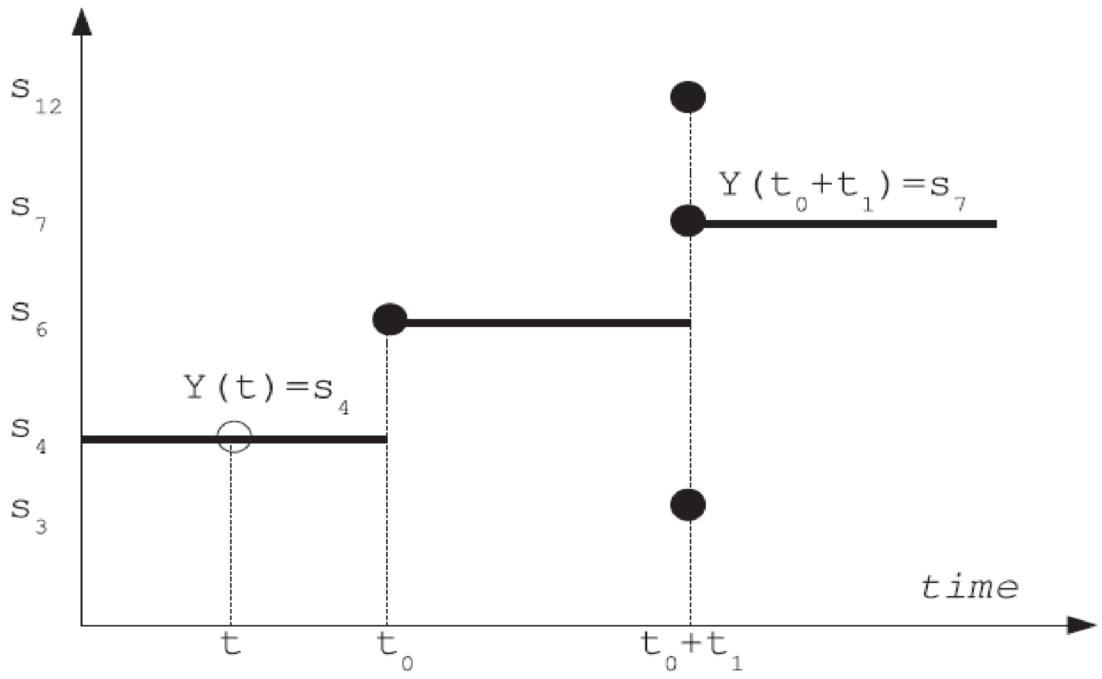
- Underlying lower-level mathematical formalism: Continuous time Markov chains
  - Steady state analysis
  - Transient analysis

Continuous time
Discrete states
Transition rates

- Solution techniques
  - Analytical („symbolic formulas”)
  - Numerical („iterations”)
  - Simulation based („collecting data”)

# Stochastic processes

- Stochastic process:
  - Mathematical model of a system or phenomena that changes in time in a random manner – characterized by a set of random variables
  - A stochastic process is characterized by its possible trajectories
  - IT systems: Typically, holding times of states are represented by random variables



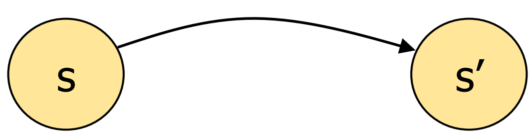
# Markov processes

- Markov process with $S(t)$ state is a stochastic process such that

  $$P\{S(t)=s \mid S(t_n)=s_n, S(t_{n-1})=s_{n-1}, \ldots, S(t_0)=s_0\} \;=\; \mathbf{P\{S(t)=s \mid S(t_n)=s_n\}}$$

  for all $t > t_n > t_{n-1} > \ldots > t_0$

  - I.e., the conditional probability distribution of future states (conditional on both past and present states) depends only on the present state
  - "Memoryless property" of the stochastic process
- Markov processes with discrete states: Markov chains
  - Behaviour can be given by the holding times of discrete states
  - Holding times of states are characterized by random variables of negative exponential distributions
    - This is the only distribution that satisfies the Markov property
    - In each time point, the distribution of the remaining time in the given state is statistically independent from the time the process has spent in that state




$$P\{\text{holding s for t}\} = e^{-\lambda t}$$

# Continuous Time Markov chains

- CTMC: Continuous Time Markov Chain
  - Continuous time, discrete state space
- Notations and properties
  - Discrete states: $s_0, s_1, \ldots, s_n$, state of the CTMC is $S(t)$
  - Probability of a transition: $Q_{ij}(t_{n-1},t_n) = P\{S(t_n)=s_j \mid S(t_{n-1})=s_i\}$
  - In case of time homogenous process: $Q_{ij}(t,t+\Delta t)= Q_{ij}(\Delta t)$
    - The transition probability does not depend on time
  - Transition rates:

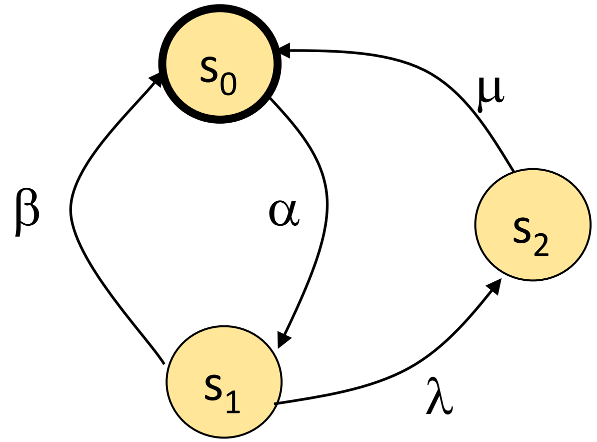
$$R_{ij}(t) = \lim_{\Delta t \to 0} \frac{1}{\Delta t} Q_{ij}(\Delta t)$$

  - Rate of leaving a state:

$$E(s) = \sum_{s' \in S} R_{s,s'}$$

# Model: Continuous Time Markov Chain

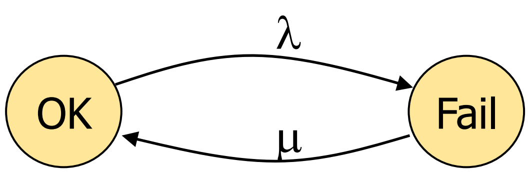
- Definition: CTMC = $(S, \underline{\underline{\mathbf{R}}})$
  - S set of discrete states:
    $s_0, s_1, \ldots, s_n$
  - $\underline{\underline{\mathbf{R}}}: S \times S \rightarrow R_{\geq 0}$ state transition rates




- Notation:
  - $\underline{\underline{\mathbf{Q}}} = \underline{\underline{\mathbf{R}}} - \mathrm{diag}(\underline{E})$ infinitesimal generator matrix
  - $\sigma = s_0, t_0, s_1, t_1, \ldots$ path ($s_i$ is left at $t_i$)
  - $\sigma@t$ the state at time $t$
  - Path(s) set of paths from s
  - $P(s, \sigma)$ the probability of traversing a path $\sigma$ from s

# Application of CTMC: Dependability model

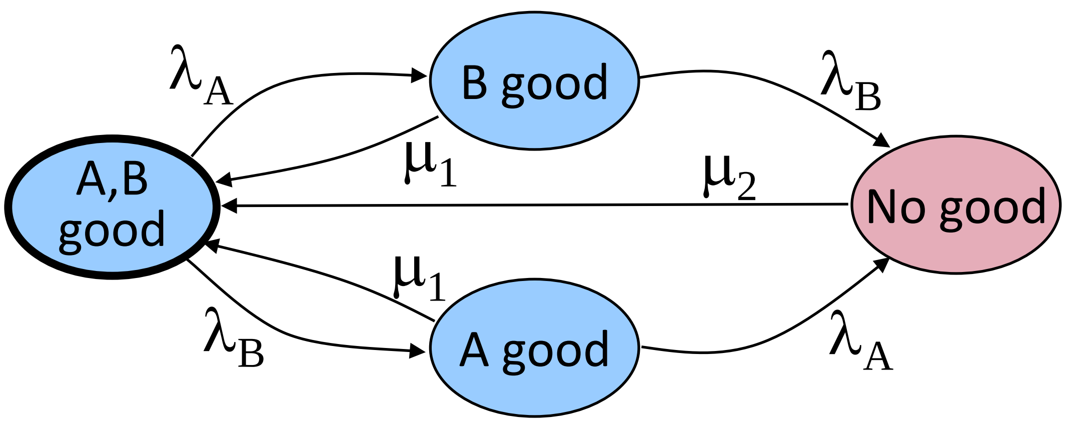
- CTMC states
  - System level states: Combination of component states (fault-free, or failed)
- CTMC transitions
  - Component level fault occurrence:
    Rate of the transition is the component failure rate ($\lambda$)
  - Component level repair:
    Rate of the transition is the component repair rate ($\mu$), which is the reciprocal of the repair time




  - System level repair:
    Rate of the transition is the system repair rate (which is the reciprocal of the system repair time)

# Example: CTMC dependability model

- System consisting of two servers, A and B:
  - The servers may independently fail
  - The servers can be repaired independently of together
- System states: Combination of the server states (good/faulty)
- Transition rates:
  - Failure of server A: $\lambda_A$ failure rate
  - Failure of server B: $\lambda_B$ failure rate
  - Repair of a server: $\mu_1$ repair rate
  - Repair of both servers: $\mu_2$ repair rate



# Solution of a CTMC

- Transient state probabilities:
  - $\pi(s_0, s, t) = P\{\sigma \in Path(s_0) \mid \sigma@t=s\}$ probability that starting from $s_0$ the system is in state $s$ at time $t$
  - $\underline{\pi}(s_0, t)$ starting from $s_0$, the probabilities of the states at $t$
  - Transient state probabilities obtained by solving:

$$\frac{d\,\underline{\pi}(s_0,t)}{dt} = \underline{\pi}(s_0,t)\underline{\underline{Q}}$$

- Steady state probabilities (if exist):
  - $\pi(s_0, s) = \lim_{t\to\infty} \pi(s_0,s,t)$ state probabilities (starting from $s_0$)
  - $\underline{\pi}(s_0)$ steady state probabilities (vector)
  - Steady state probabilities obtained by solving:

$$\underline{\pi}(s_0)\underline{\underline{Q}} = 0 \quad \text{where} \quad \sum_{s} \pi(s_0,s) = 1$$

# Elements of the solution of a Markov chain

- Probability of the holding time of a state:

$$P\{\text{holding s for t}\} = e^{-E(s)t}$$

- Probability of leaving a state:

$$P\{\text{leaving s in t}\} = 1 - e^{-E(s)t}$$

- Probability of a state transition:

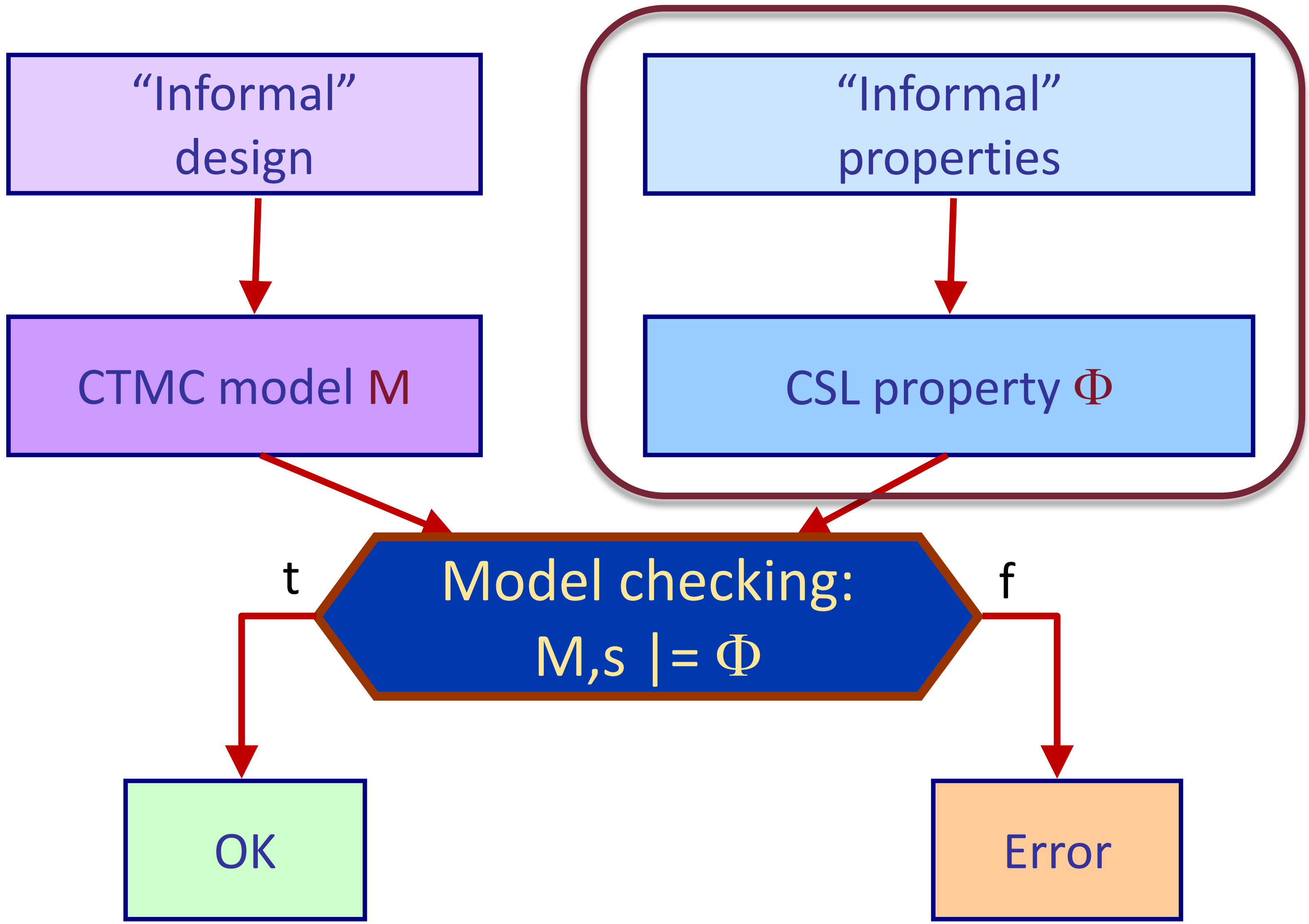
$$P\{\text{transition from s to s' in t}\} = 1 - e^{-R(s,s')t}$$

- Expected value of the time spent in a state:

$$E\{\text{time spent in } s\} = \frac{1}{E(s)}$$

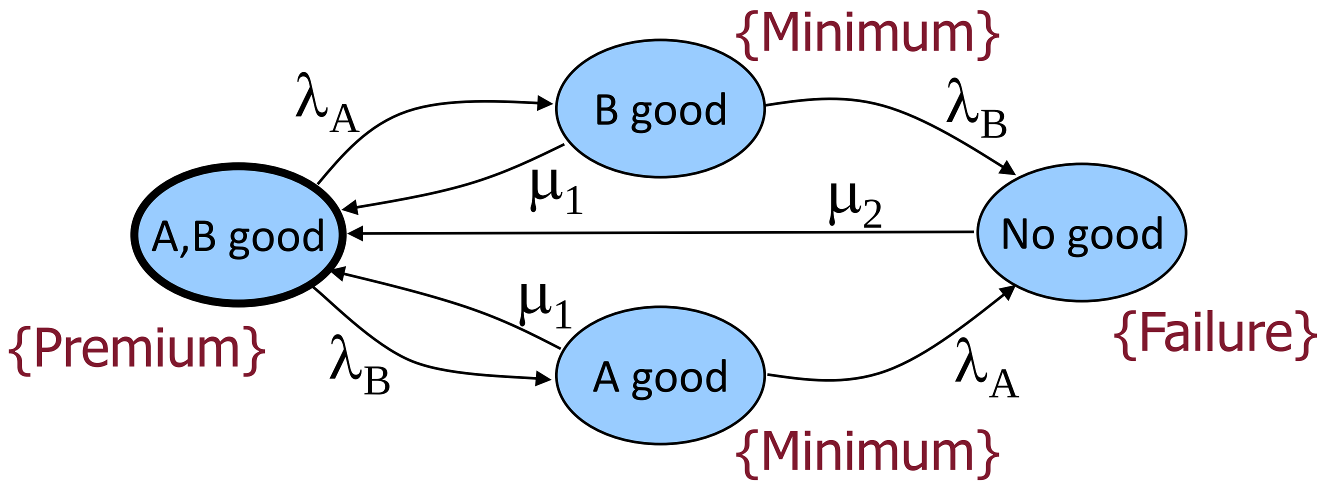
# Formalizing properties

# Formal verification of stochastic properties



# How to formalize QoS properties?

- Modeling: CTMC, simple state-based formalism
  - Extension: Labeling states with atomic propositions




  - For states: Computing steady state or transient probabilities
  - For paths: Computing path traversing probabilities
- Properties: Formalized on the analogy of CTL
  - Specifying probabilities and time intervals for states or paths
  - Result: Continuous Stochastic Logic (CSL)

# Continuous Stochastic Logic

- Extensions with regard to CTL
  - Probability related operators:
    - For steady state: Probability of being in a state partition (set of states) characterized by a state formula
    - For (transient) paths: Probability of executing paths characterized by a path formula
  - Time interval related operators:
    - Extending the operators X and U with time intervals: Occurrence of state(s) characterized by state formula in the given time interval
- Notation:
  - I time interval, e.g., [0, 12), [15,$\infty$)
  - p probability
  - ~ operator for comparison, e.g., $\geq$, $\leq$, <, >
  - $\Phi$ state formula (to be evaluated in state of the CTMC)
  - $\varphi$ path formula (to be evaluated on paths of the CTMC)

# CSL state formula

- The well-formed CSL expressions are the state formula
- Syntax: $\Phi ::= P \mid \neg\Phi \mid \Phi \vee \Phi \mid S_{\sim p}(\Phi) \mid P_{\sim p}(\varphi)$
- Informal semantics of the new operators
  - $S_{\sim p}(\Phi)$ specifies that the steady-state probability of being in state partition characterized by $\Phi$ is $\sim p$

    P{being in state where $\Phi$ holds} ~ p

    - Example: $S_{>0.8}(\mathrm{Minimum} \vee \mathrm{Premium})$
  - $P_{\sim p}(\varphi)$ specifies that the probability of executing a path characterized by $\varphi$ is $\sim p$

    P{executing a path on which $\varphi$ holds} ~p

    - Example: $P_{>0.7}(\mathrm{true}\ \mathrm{U}\ \mathrm{Premium})$

# CSL path formula

- Syntax: $\varphi ::= X^I \Phi \mid \Phi\ U^I\ \Phi$
- Informal semantics of operators
  - $X^I \Phi$ specifies that in the next state reached at time $t \in I$ the state formula $\Phi$ holds
    - Example: $X^{[0,10]}$Premium
  - $\Phi_1\ U^I\ \Phi_2$ specifies that in $t \in I$ a state is reached in which $\Phi_2$ holds and until that state in each preceding state $\Phi_1$ holds
    - Example: Minimum $U^{[5,10]}$ Premium
- Operators introduced as abbreviations:
  - $E\ \varphi = P_{>0}(\varphi)$
  - $A\ \varphi = P_{\geq 1}(\varphi)$
  - $F^I\ \Phi = \text{true}\ U^I\ \Phi$
  - $X\ \Phi = X^I\ \Phi$, $\Phi_1\ U\ \Phi_2 = \Phi_1\ U^I\ \Phi_2$ where $I=[0,\infty)$

# CSL semantics (1)

- $M=(S,\underline{\boldsymbol{R}},L)$ is a CTMC with state labeling
  - $L: S \rightarrow 2^{AP}$ labeling function
- Basic operators:
  - $M,s \models P$ iff $P \in L(s)$
  - $M,s \models \neg\Phi$ iff $M,s \models \Phi$ does not hold
  - $M,s \models \Phi_1 \vee \Phi_2$ iff $M,s \models \Phi_1$ or $M,s \models \Phi_2$
- Probability related operators:
  - $M,s \models S_{\sim p}(\Phi)$ iff $\pi(s, Sat(\Phi)) \sim p$,

    i.e., $M,s \models S_{\sim p}(\Phi)$ iff $\sum_{s' \in Sat(\Phi)} \pi(s,s') \sim p$
  - $M,s \models P_{\sim p}(\varphi)$ iff $P(s, \sigma \mid \sigma \models \varphi) \sim p$,

    i.e., $M,s \models P_{\sim p}(\varphi)$ iff $\sum_{\substack{\sigma \in Path(s) \\ \sigma \models \varphi}} P(s,\sigma) \sim p$

Starting from s, steady state probability of states in which $\Phi$ holds is ~p

Starting from s, probability of paths on which $\varphi$ holds is ~p



MŰEGYETEM 1782

# CSL semantics (2)

- Operators for time intervals:
  - $M,\sigma \models X^I\, \Phi$ iff

    $\exists s_1$: $M,s_1 \models \Phi$ and $t_0 \in I$

  - $M,\sigma \models \Phi_1\, U^I\, \Phi_2$ iff

    $\exists t \in I$: $(\sigma@t \models \Phi_2$ és $\forall u \in [0,t)$: $\sigma@u \models \Phi_1)$

# Outlook: CSL model checking (overview)

- $S_{\sim p}(\Phi)$ formula:
  - Utilizing the steady state solution of the CTMC
- $X^I \Phi$ formula:
  - Utilizing the transient solution of the CTMC (to next state)
- $P_{\sim p}(\varphi)$ or $\Phi_1 \, U^I \, \Phi_2$ formula:
  - Transient solution is needed + time intervals
  - General: Solution of a Volterra integral equation

$$\int_0^t \sum_{s' \in S} \mathbf{R}(s, s') \cdot e^{-\mathbf{E}(s) \cdot x} \cdot Prob(s', \Phi \, \mathcal{U}^{[0, t-x]} \, \Psi) \, dx$$

  - Simplification: Transforming the CTMC and the property to be checked in order to have a problem for which the transient solution of the transformed CTMC is sufficient
    - Transformation: $M \rightarrow M'$, $\Phi \rightarrow \Phi'$
    - To be proved: $M, s \models \Phi$ iff $M', s \models \Phi'$

# Example: Simplification in case of $\Phi_1\ U^{[0,t)}\ \Phi_2$

- Goal: Checking $\Phi_1\ U^{[0,t)}\ \Phi_2$ on model $M$
- Transforming the model from $M$ to $M'$:
  - After reaching states in which $\Phi_2$ holds (before $t$ and through states in which $\Phi_1$ holds), the future behavior is irrelevant for the property; thus all such states in which $\Phi_2$ holds become sink state in $M'$
  - In states for which $\neg(\Phi_1 \vee \Phi_2)$ holds, i.e., counter-example is found, the future behavior is irrelevant for the property; thus all such states become sink state in $M'$
- Transforming the property for M':
  - The following theorem can be proven:

    $M,s \models \Phi_1\ U^{[0,t)}\ \Phi_2$ holds if

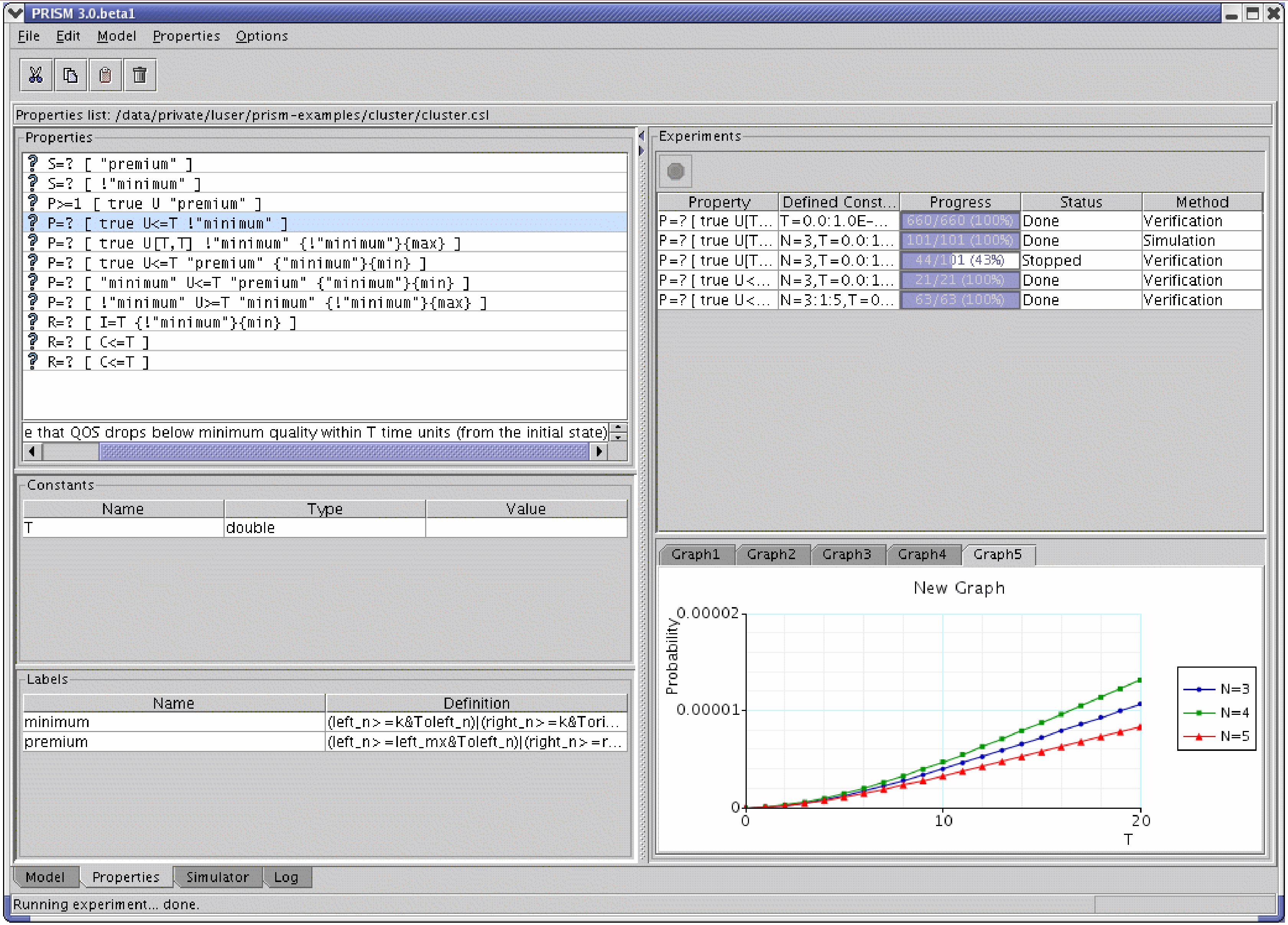
    $M',s \models \text{true}\ U^{[t,t]}\ \Phi_2$ holds (in the transformed model)

    i.e., the transient solution of the transformed model is sufficient

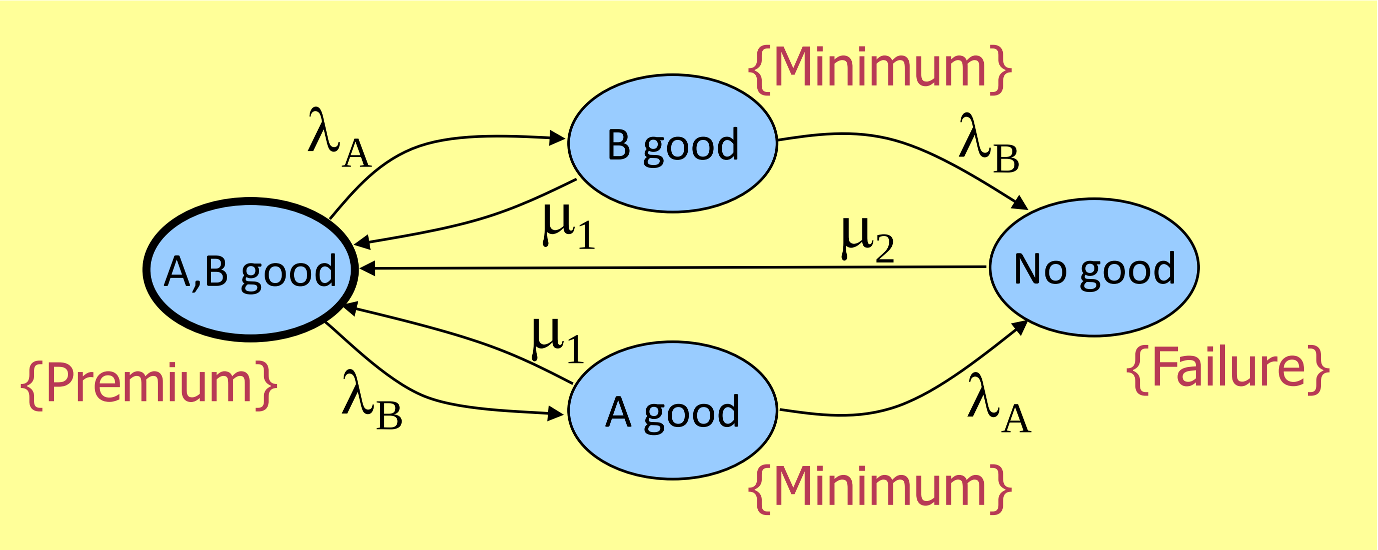
# CSL model checkers

- First implementation:
  ETMCC: Erlangen-Twente Markov Chain Checker (E|-MC²)
  - Supported models: CTMC, Stochastic process algebra
- PRISM: Probabilistic Symbolic Model Checker
  - Supported models: Stochastic Petri nets (GreatSPN extension)
  - Symbolic handling of the state space
- MRMC: Markov Reward Model Checker
  - Discrete time Markov chains are also supported
  - CSRL: CSL extended with reward function
  - Reward: Cost/profit assignment
    - To states: Rate reward (can be integrated for time intervals)
    - To transitions: Impulse reward (can be summed for fired transitions)

# PRISM


PRISM 3.0.beta1
File Edit Model Properties Options
Properties list: /data/private/luser/prism-examples/cluster/cluster.csl
Properties
S=? [ "premium" ]
S=? [ !"minimum" ]
P>=1 [ true U "premium" ]
P=? [ true U<=T !"minimum" ]
P=? [ true U[T,T] !"minimum" {!"minimum"}{max} ]
P=? [ true U<=T "premium" {"minimum"}{min} ]
P=? [ "minimum" U<=T "premium" {"minimum"}{min} ]
P=? [ !"minimum" U>=T "minimum" {!"minimum"}{max} ]
R=? [ I=T {!"minimum"}{min} ]
R=? [ C<=T ]
R=? [ C<=T ]
e that QOS drops below minimum quality within T time units (from the initial state)
Constants
Name Type Value
T double
Labels
Name Definition
minimum (left_n>=k&Toleft_n)|(right_n>=k&Tori...
premium (left_n>=left_mx&Toleft_n)|(right_n>=r...
Experiments
Property Defined Const... Progress Status Method
P=? [ true U[T... T=0.0:1.0E-... 660/660 (100%) Done Verification
P=? [ true U[T... N=3,T=0.0:1... 101/101 (100%) Done Simulation
P=? [ true U[T... N=3,T=0.0:1... 44/101 (43%) Stopped Verification
P=? [ true U<... N=3,T=0.0:1... 21/21 (100%) Done Verification
P=? [ true U<... N=3:1:5,T=0... 63/63 (100%) Done Verification
Graph1 Graph2 Graph3 Graph4 Graph5
New Graph
Probability
0.00002
0.00001
0
0
10
20
T
N=3
N=4
N=5
Model Properties Simulator Log
Running experiment... done.

# Using CSL to formalize QoS properties (1)




Labels to be used: Premium, Minimum, Failure

- Availability of service is greater than 0.99:

$$S_{\geq 0.99}(\text{Premium} \vee \text{Minimum})$$

- In the long run, the probability that the service level is Premium is at least 0.9:

$$S_{\geq 0.9}(\text{Premium})$$

# Using CSL to formalize QoS properties (2)

- It happens with probability less than 0.1, that in 85 time units the service level falls below Minimum:

$$P_{<0.1}(\mathrm{F}^{[0,85]}\ \mathrm{Failure}) = P_{<0.1}(\mathrm{true}\ \mathrm{U}^{[0,85]}\ \mathrm{Failure})$$

- It is possible to reach Premium service level:

$$P_{>0}(\mathrm{F}\ \mathrm{Premium}) = P_{>0}(\mathrm{true}\ \mathrm{U}^{(0,\infty)}\ \mathrm{Premium})$$

- If there is Failure at start, then it happens with probability less than 0.3 that the failure will present after 2 time units:

$$\mathrm{Failure} \Rightarrow P_{<0.3}(\mathrm{F}^{[2,2]}\ \mathrm{Failure})$$

- The probability that the recovery after an initial failure needs more than 15 time units is at most 0.2:

$$\mathrm{Failure} \Rightarrow P_{\leq 0.2}(\mathrm{Failure}\ \mathrm{U}^{[15,\infty)}\ (\mathrm{Minimum} \vee \mathrm{Premium}))$$

# Using CSL to formalize QoS properties (3)

- It happens with probability less than 0.01 that after 9 time units of fault-free operation the system will fail in 1 time unit:

$$P_{<0.01}((\text{Premium} \vee \text{Minimum})\ \mathrm{U}^{[9,10]}\ \text{Failure})$$

- Starting with Minimum service level, it happens with probability more than 0.7 that in 5 time units (keeping at least the Minimum service level) the Premium service level will be provided:

$$\text{Minimum} \Rightarrow \mathrm{P}_{>0.7}(\text{Minimum}\ \mathrm{U}^{[0,5)}\ \text{Premium})$$

# Summary

- Motivation: Checking service quality and timeliness
  - Typical in QoS, SLA
- Basic mathematical model: CTMC, with state labeling
  - It can be mapped from higher-level models
  - Solution: Computing steady state or transient state probabilities
- Formalizing properties: CSL
  - Probability operators for states (in steady state) and executed paths
  - Time intervals for standard path operators
- Model checking
  - Simplification by transforming both the model and the property
- Properties (examples)