

Sztochasztikus temporális logikák

Teljesítmény és szolgáltatásbiztonság jellemzők formalizálása és ellenőrzése

Majzik István

Budapesti Műszaki és Gazdaságtudományi Egyetem

Méréstechnika és Információs Rendszerek Tanszék

<http://www.mit.bme.hu/~majzik/>

Motiváció: Szolgáltatásminőségi követelmények

- Nem tisztán elérhetőségi jellegű követelmények
 - QoS: Quality of Service
 - SLA: Service Level Agreement
- Jellemzők a követelményekre:
 - Adott szolgáltatási szintek **valószínűségei**
 - Példa: Rendelkezésre állás, mint valószínűség (állandósult állapotban)
 - Szolgáltatási szintek fenntartásának (kihagyásának) **időtartama**
 - Példa: Javítási idő maximuma
- Példák összetett QoS követelményekre:
 - Annak a valószínűsége legfeljebb 20%, hogy a hiba utáni helyreállítás több mint 15 időegységet vegyen igénybe.
 - Annak a valószínűsége kisebb 10%-nál, hogy 85 időegység alatt a szolgáltatási szint **Minimum** alá csökken.
 - Több mint 70% annak a valószínűsége, hogy **Minimum** szolgáltatási szint elérése esetén 5 időegységen belül **Premium** szint nyújtható.

Milyen modellek használhatók?

- Teljesítmény- és megbízhatóság modellezés:
 - Sztochasztikus Petri-hálók
 - Sztochasztikus processz algebrák
 - Sztochasztikus aktivitás hálók
- Ezekből **folytonos idejű Markov lánc** képzése és megoldása (mint alacsony szintű formalizmus)
 - Állandósult állapotbeli analízis
 - Tranziens analízis
- Megoldási módok:
 - Analitikus („képlettel”)
 - Numerikus („iterálva”)
 - Szimulációval („kimérve”)

Tevékenységekhez exp. eloszlású időzítés rendelése a kezelhetőség érdekében

Folytonos idő
Diszkrét állapotok
Állapotátmeneti gyakoriság

Markov folyamatok

- Sztochasztikus folyamat:

- Valószínűségekkel jellemezhetően bekövetkező jelenségek modellezése, az idő paraméter függvényében

- Markov folyamat:

$$P\{X(t)=x \mid X(t_n)=x_n, X(t_{n-1})=x_{n-1}, \dots, X(t_0)=x_0\} = P\{X(t)=x \mid X(t_n)=x_n\}$$

minden $t > t_n > t_{n-1} > \dots > t_0$ esetén

- Informálisan:

- A jövőbeli viselkedés (t -ben) csak az aktuális állapottól (t_n -ben) függ, és nem függ a korábbi állapotoktól

- Diszkrét állapotterű Markov folyamatok: **Markov láncok**

- Diszkrét állapotokban való tartózkodás idejével (tartási idő) jellemezhetők a trajektóriák
- Tartási idő **negatív exponenciális eloszlású**
 - Az egyetlen eloszlásfüggvény, ami a Markov tulajdonságot teljesíti
 - Bármely időpillanatban a **maradék tartási idő** statisztikailag független attól, hogy eddig **mennyi időt** töltött a folyamat az adott állapotban

Folytonos idejű Markov láncok (CTMC)

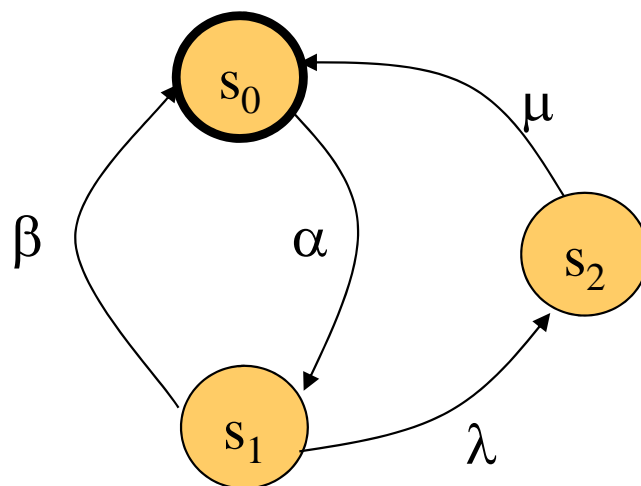
- CTMC: Continuous Time Markov Chain
 - Folytonos idő paraméter, diszkrét állapot tér
- Jelölések, tulajdonságok:
 - Diszkrét állapotok: s_0, s_1, \dots, s_n , a CTMC állapota $S(t)$
 - Állapotátmeneti valószínűség: $Q_{ij}(t_{n-1}, t_n) = P\{S(t_n) = s_j \mid S(t_{n-1}) = s_i\}$
 - Homogén Markov-folyamat: $Q_{ij}(t, t + \Delta t) = Q_{ij}(\Delta t)$
 - Állapotátmeneti valószínűség nem változik az idő függvényében
 - Állapotátmeneti intenzitás (ráta):

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

- Állapot elhagyás összesített rátája: $E(s) = \sum_{s' \in S} R_{s, s'}$
- Állapot tartási ideje: $P\{s\text{-ben marad } t \text{ ideig}\} = e^{-E(s)t}$

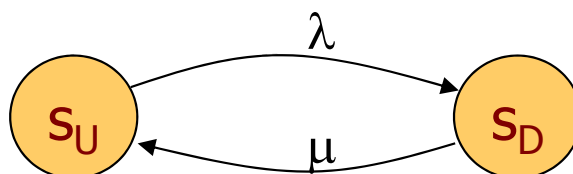
Egy egyszerű CTMC

- CTMC szokásos megjelenítése:
 - Állapotok halmaza (kezdő valószínűségekkel)
 - Minden állapotpárra az **állapotátmeneti intenzitás** (ahol nem nulla, csak ott van feltüntetve)



CTMC alkalmazások

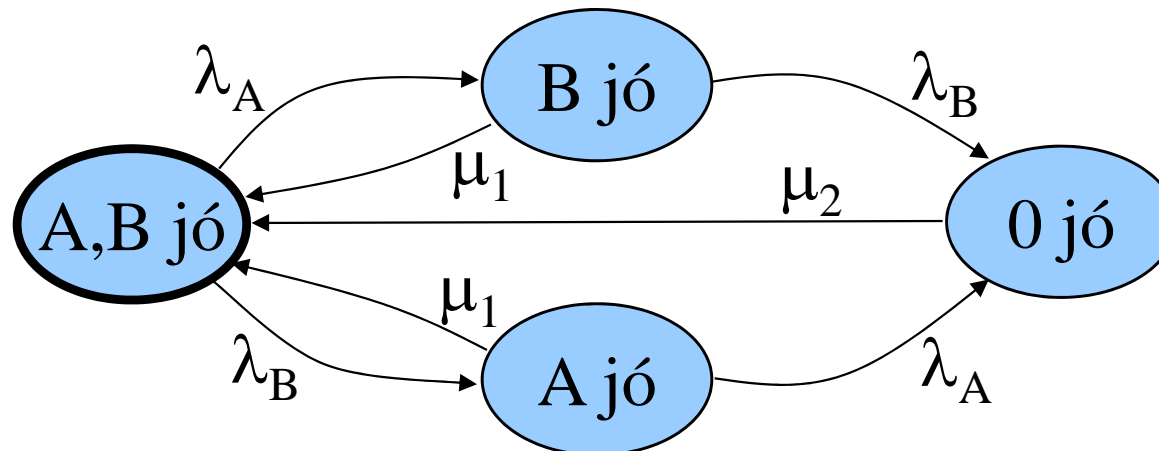
- Megbízhatósági modellezés:
 - Komponens állapottere: Hibamentes s_U vagy hibás s_D állapot
 - Gyakorlati tapasztalat elektronikai komponensekre:
 - A hibamentes állapot tartási ideje exponenciális eloszlással jellemezhető a tipikus használati tartományban
 - Az exp. eloszlásfüggvény paramétere: Meghibásodási tényező, λ
 - A javítási időt is exp. eloszlással számítják (egyszerűsítés), μ



- Teljesítmény modellezés
 - Sorbanállás - kiszolgálás
 - M/M/1 sor: „Markovi” beérkezési és kiszolgálási idők
 - Állapottér mint CTMC vehető fel
 - Sorbanállási hálózatok

Példa: Megbízhatósági modellezés

- Két szerverből (A, B) álló rendszer:
 - Bármelyik szerver meghibásodhat
 - A szerverek külön-külön vagy együtt is javíthatók
 - Rendszerszintű állapotokat modellezünk
- Állapotátmenetek (exponenciális eloszlású időzítés):
 - Az A szerver meghibásodása: λ_A meghibásodási tényező
 - A B szerver meghibásodása: λ_B meghibásodási tényező
 - Egy szerver javítása: μ_1 javítási tényező
 - Teljes rendszer javítása: μ_2 javítási tényező



Folytonos idejű Markov-láncok (jelölések)

- CTMC=(S, \underline{R})

S állapotok halmaza

$\underline{R}: S \times S \rightarrow R_{\geq 0}$ állapotátmeneti intenzitás (ráta) mátrix

- $P\{s\text{-ből } s'\text{-be megy át } t \text{ időn belül}\} = 1 - e^{-R(s,s')t}$
- $\underline{E}(s) = \sum_{s' \in S} R(s,s')$ állapot elhagyás összesített gyakorisága
- $P\{s \text{ állapot elhagyása } t \text{ időn belül}\} = 1 - e^{-E(s)t}$
- $\underline{Q} = \underline{R} - \text{diag}(\underline{E})$ „infinitezimális generátormátrix”

- Útvonalak:

$\sigma = s_0, t_0, s_1, t_1, \dots$ útvonal (t_i időpontban lép ki s_i -ből)

$\sigma@t$ az állapot a t időpillanatban

$\text{Path}(s)$ az s -ből induló útvonalak halmaza

$P(s, \sigma)$ egy útvonal bejárásának valószínűsége

Markov-láncok megoldása

- **Tranziens valószínűségek:**

- $\pi(s,s',t) = P\{\sigma \in \text{Path}(s) \mid \sigma @ t = s'\}$ annak valószínűsége, hogy s -ből indulva a t időpillanatban s' -ben tartózkodik
- $\underline{\pi}(s,t)$: s -ből indulva az állapotok valószínűsége t időpillanatban
- CTMC tranziens megoldása:

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

Kb.: Nincs nyelő állapot

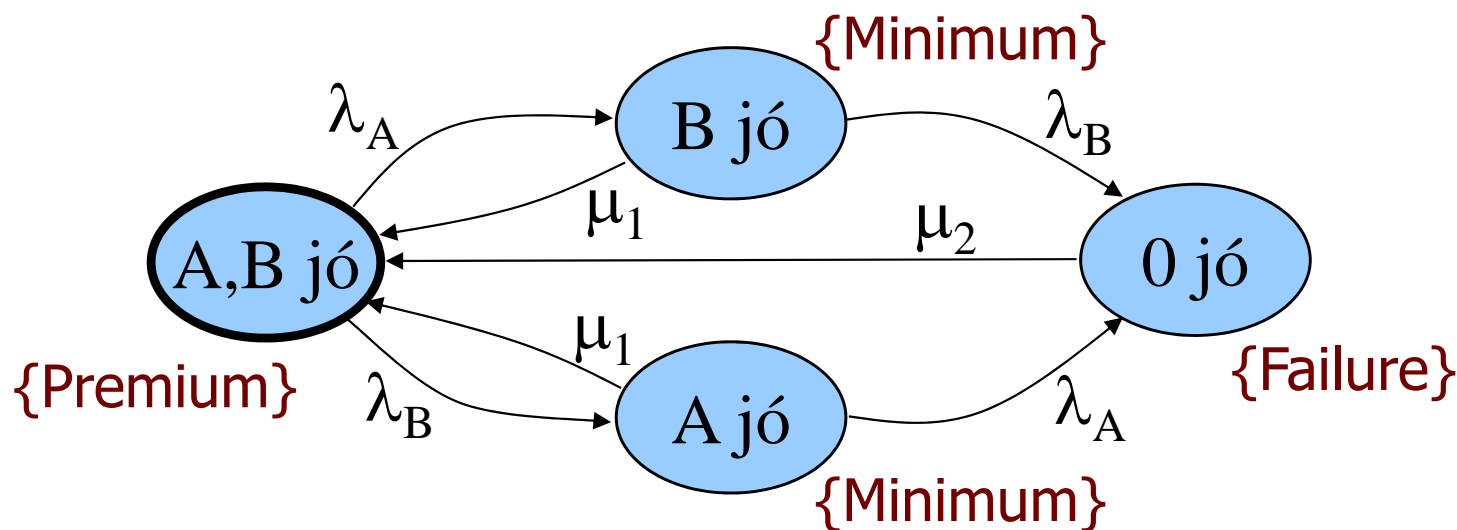
- **Állandósult állapot (véges állapotú és irreducibilis CTMC):**

- $\pi(s,s') = \lim_{t \rightarrow \infty} \pi(s,s',t)$ - s -ből indulva az állapotok valószínűsége
- $\underline{\pi}(s)$ az állapotok valószínűsége (sorvektor)
- $\pi(s,S') = \sum_{s' \in S'} \pi(s,s')$ egy állapothalmaz valószínűsége
- CTMC állandósult állapotbeli megoldása:

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

Hogyan formalizálhatók a követelmények?

- Modell: CTMC, egyszerű állapot-alapú formalizmus
 - Kiterjesztés: Állapotok címkézése atomi kijelentésekkel
 - Ld. Premium, Minimum, Failure a lenti modellen
 - Állapotok: Számítható állandósult vagy tranziens valószínűségek
 - Útvonalak: Számítható bejárési valószínűségek
- Követelmények formalizálása: CTL analógia alapján
 - Állapot és útvonal kifejezések
- CSL: Continuous Stochastic Logic
 - Állapotokra és útvonalakra vonatkozó valószínűségi kifejezések és időtartamok megadása

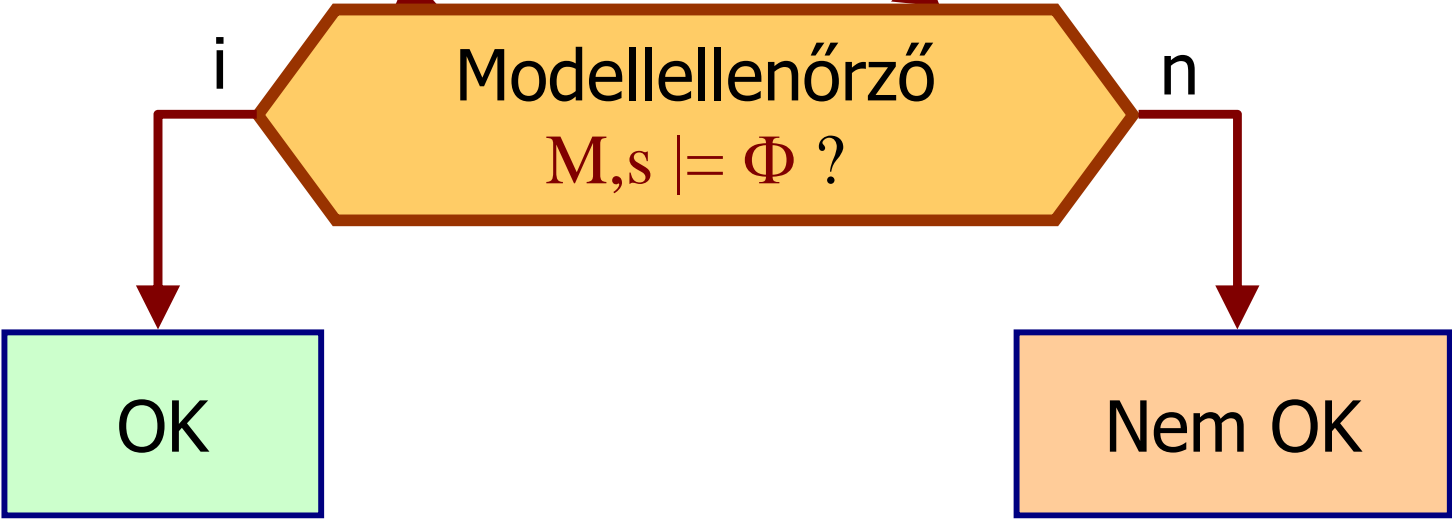


CSL modellellenőrzés

Származtatható sztochasztikus modellekből
(pl. SPN, GSPN, SPA, SAN)

CTMC M

CSL kifejezés Φ



Continuous Stochastic Logic: Szintaxis

- Kiterjesztések a CTL-hez képest:
 - Valószínűségi operátorok:
 - Állandósult állapotban: állapot-kifejezések által megadott állapot-halmazokban való tartózkodás valószínűsége
 - Útvonal-kifejezések által megadott útvonal bejárásának valószínűsége (tranzienst analízis)
 - Időtartományok megadása:
 - Temporális operátorokhoz (X , U) időintervallum megadása: az adott időintervallumon belüli bekövetkezés
- Jelölések:
 - I intervallum, pl. $[0, 12)$, $[15, \infty)$
 - p valószínűség
 - \sim az összehasonlítás operátora, pl. \geq , \leq , $<$, $>$

CSL állapot-kifejezések

- Jelölések:
 - Φ állapot-kifejezések (ezek alkotják a CSL kifejezéseket)
 - φ útvonal-kifejezések
- Szintaxis: $\Phi ::= P \mid \neg\Phi \mid \Phi \vee \Phi \mid S_{\sim p}(\Phi) \mid P_{\sim p}(\varphi)$
 - $S_{\sim p}(\Phi)$ - állandósult állapotban az olyan állapotokban való tartózkodás valószínűsége $\sim p$, ahol Φ igaz
 $P\{\text{olyan állapotban tartózkodik, ahol } \Phi \text{ igaz}\} \sim p$
 - Példa: $S_{>0,8}(\text{Minimum} \vee \text{Premium})$
 - $P_{\sim p}(\varphi)$ - olyan utak bejárásának valószínűsége $\sim p$, amelyeken φ igaz
 $P\{\text{olyan utat jár be, ahol } \varphi \text{ igaz}\} \sim p$
 - Példa: $P_{>0,7}(\text{true} \cup \text{Premium})$

CSL útvonal-kifejezések

- Szintaxis: $\varphi ::= X^I \Phi \mid \Phi U^I \Phi$
 - $X^I \Phi$ - a következő állapotot a $t \in I$ időpillanatban érjük el, és ebben a következő állapotban igaz Φ
 - Példa: $X^{[0,10]} \text{Premium}$
 - $\Phi_1 U^I \Phi_2$ – az útvonal mentén igaz Φ_1 amíg Φ_2 igaz nem lesz a $t \in I$ időpillanatban
 - Példa: $\text{Minimum } U^{[5,10]} \text{Premium}$
- Rövidítések:
 - $E \varphi = P_{>0}(\varphi)$
 - $A \varphi = P_{\geq 1}(\varphi)$
 - $F^I \Phi = \text{true } U^I \Phi$
 - $X \Phi = X^I \Phi, \quad \Phi_1 U \Phi_2 = \Phi_1 U^I \Phi_2 \quad \text{ahol } I = [0, \infty)$

CSL szemantika

- $M=(S, \underline{R}, L)$ egy CTMC az állapotok címkézésével
 - $L: S \rightarrow 2^{AP}$ állapot címkézés

- Alap operátorok:

- $M, s \models P$ a.cs.a. $P \in L(s)$
- $M, s \models \neg \Phi$ a.cs.a. nem igaz $M, s \models \Phi$
- $M, s \models \Phi_1 \vee \Phi_2$ a.cs.a. $M, s \models \Phi_1$ vagy $M, s \models \Phi_2$

- Állapot kvantorok:

- $M, s \models S_{\sim p}(\Phi)$ a.cs.a. $\pi(s, \text{Sat}(\Phi)) \sim p$,

s-ből indulva $\text{Sat}(\Phi)$ áll. állapotban való tartózkodás vsz. $\sim p$

azaz $s \in \text{Sat}(S_{\sim p}(\Phi))$ a.cs.a. $\sum_{s' \in \text{Sat}(\Phi)} \pi(s, s') \sim p$

- $M, s \models P_{\sim p}(\varphi)$ a.cs.a. $P(s, \sigma \mid \sigma \models \varphi) \sim p$,

$\sigma \models \varphi$ útvonal bejárás vsz. $\sim p$

azaz $s \in \text{Sat}(P_{\sim p}(\varphi))$ a.cs.a. $\sum_{\substack{\sigma \in \text{Path}(s) \\ \sigma \models \varphi}} P(s, \sigma) \sim p$

CSL szemantika (folytatás)

- Útvonal kvantorok:

- $M, \sigma \models X^l \Phi$ a.cs.a.

$$\exists s_1: M, s_1 \models \Phi \text{ és } t_0 \in I$$

- $M, \sigma \models \Phi_1 U^l \Phi_2$ a.cs.a.

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

CSL modellellenőrzés

- $S_{\sim p}(\Phi)$ esetén:
 - Állandósult állapotbeli CTMC megoldásból származik
- $X^1 \Phi$ esetén:
 - CTMC tranziens megoldás (következő állapotba lépés)
- $P_{\sim p}(\varphi)$ illetve $\Phi_1 \cup^1 \Phi_2$ esetén:
 - Tranziens megoldás kell, de időintervallumokra
 - Általános: Volterra integrál-egyenlet megoldása

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

- Egyszerűsítés: CTMC és követelmény átalakítás úgy, hogy elég legyen t -re egy tranziens analízis
 - Átalakítás: $M \rightarrow M', \Phi \rightarrow \Phi'$
 - Bizonyítandó: $M, s \models \Phi$ ekvivalens $M', s \models \Phi'$

Az egyszerűsítés illusztrálása $\Phi_1 \cup^{[0,t)} \Phi_2$ esetén

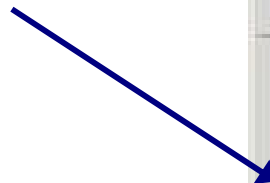
- Célkitűzés: $\Phi_1 \cup^{[0,t)} \Phi_2$ ellenőrzése M modellen
- A modell átalakítása M -ről M' -re:
 - Φ_2 -t teljesítő állapotok (Φ_1 teljesítése mentén, t előtt) elérése után a viselkedés nem érdekes, így minden Φ_2 tulajdonságú állapot **nyelő** lesz M' -ben
 - $\neg (\Phi_1 \vee \Phi_2)$ esetén, tehát ha egyiket sem teljesíti, akkor a további viselkedés nem érdekes, így ezek is **nyelők** lesznek M' -ben
- A követelmény átalakítása M' esetén:
 - Bizonyítható tétel:
 $M, s \models \Phi_1 \cup^{[0,t)} \Phi_2$ ellenőrzése ekvivalens
 $M', s \models \text{true} \cup^{[t,t]} \Phi_2$ ellenőrzésével;
itt t -re tranziens analízis elég az ellenőrzéshez!

CSL modellellenőrzők

- Az első megvalósítás:
ETMCC: Erlangen-Twente Markov Chain Checker (E|-MC²)
 - Markov-láncok
 - Sztochasztikus processz algebrák
- PRISM: Probabilistic Symbolic Model Checker
 - GreatSPN kiterjesztés
 - BDD alapú reprezentációval kombinálva
- MRMC Markov Reward Model Checker
 - Diszkrét idejű Markov-lánc is használható
 - CSRL: CSL kiterjesztése reward hozzárendeléssel
 - Reward: Költség/haszon megadása
 - Állapotokhoz: Rate reward (integrálható időtartamra)
 - Átmenetekhez: Impulse reward (összegezhető tüzelésekre)

ETMCC

CSL kifejezések



The screenshot shows the ETMCC v1.3 application window. The title bar reads 'ETMCC v1.3'. The menu bar includes 'File', 'Run', 'Options', and 'About'. The 'Current Properties' section contains a dropdown menu with the CSL expression $P(>0.4) [a U \leq 3 b]$ and a checked 'Verify All' checkbox. To the right of the dropdown are a play button and a square button. The main text area displays the following output:

```
Verifier: Time consumption: 0.0 seconds.  
Verifier: CheckingProbTimedUntil P(>0.4)[a U<=3.0 b]  
GraphAnalysis: Computing Exist Until.  
GraphAnalysis: Computing Always Until.  
ProbPathGS: Running with Accuracy = 1.0E-4, MaxLoopCount= 1000000  
ProbPathGS: Loops: 10  
VerifyProbTimedUntil: Running transient analysis with Accuracy = 1.0E-8  
Verifier: Time consumption: 0.11 seconds.  
RuntimeTask: Time consumption for formula P(>0.4) [a U<=3 b] : 0.11 seconds.  
RuntimeTask: Verification terminated.  
Output written to standard.log
```

The status bar at the bottom shows: 'Status: IDLE', '#States 11', '#Transitions 19', and 'Memory Usage: 392 Bytes'.

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&Tolleft_n)(right_n>=k&Tori...
premium	(left_n>=left_mx&Tolleft_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

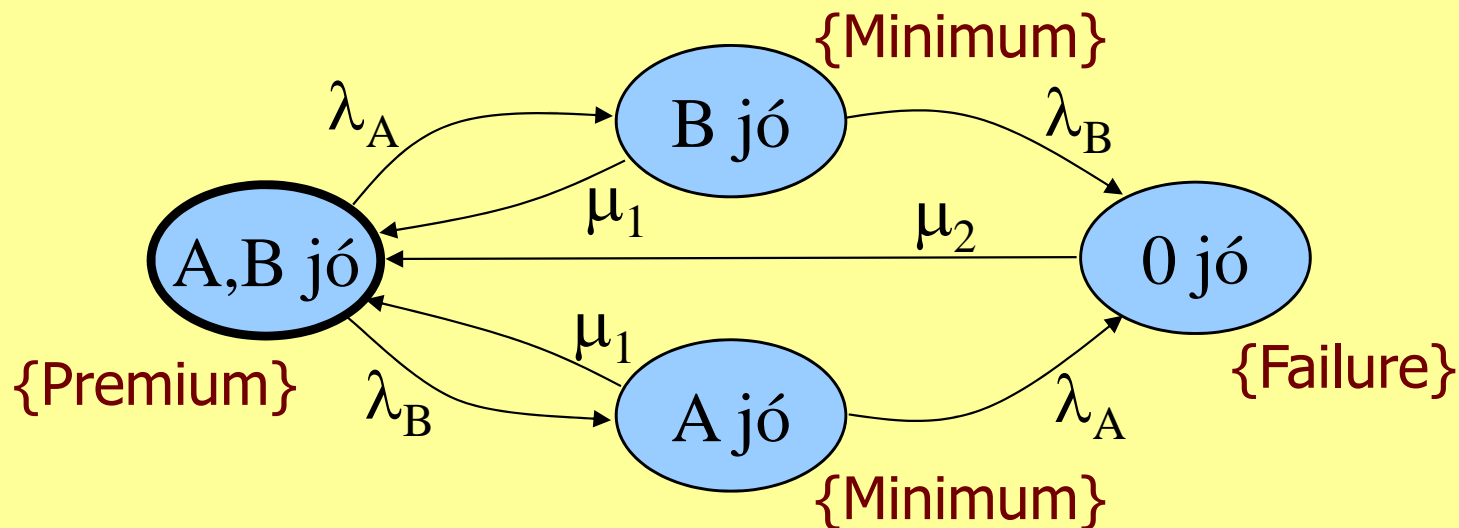
T

Legend: N=3 (blue circles), N=4 (green squares), N=5 (red triangles)

Model Properties Simulator Log

Running experiment... done.

CSL használata QoS formalizálására I.



- **Követelmények:**

- Rendelkezésre állás nagyobb 99%-nál:

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

- Hosszú távon legalább 90% valószínűséggel Premium szolgáltatás:

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

CSL használata QoS formalizálására II.

- Követelmények (folytatás):

- Annak a valószínűsége kisebb 10%-nál, hogy 85 időegység alatt a szolgáltatási szint Minimum alatti lesz:

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

- Lehetőség van a Premium szolgáltatás szint elérésére:

$$P_{\geq 1}(F \text{ Premium}) = P_{\geq 1}(\text{true } U^{(0,\infty)} \text{ Premium})$$

- Ha kezdetben hibás, akkor a hiba kisebb mint 30% valószínűséggel áll fenn 2 időegység múlva:

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

- Annak a valószínűsége legfeljebb 20%, hogy hiba esetén a helyreállítás több mint 15 időegységet vegyen igénybe:

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

CSL használata QoS formalizálására III.

- Követelmények (folytatás):

- 1%-nál kisebb a valószínűsége, hogy 9 időegység alatti folyamatos működés után egy időegységben belül hibásodik meg:

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

- Annak a valószínűsége, hogy Minimum szolgáltatási szint esetén 5 időegységben belül (ezalatt legalább a Minimum szintet megtartva) Premium szint nyújtható, több mint 70%:

$$\text{Minimum} \Rightarrow P_{>0.7}(\text{Minimum} U^{[0,5]} \text{Premium})$$

Összefoglalás

- Motiváció: Szolgáltatásminőségi követelmények verifikációja
 - QoS, SLA
- Alapszintű modell: CTMC, állapotcímkezéssel
 - Magasabb szintű modellekből leképezhető
 - Megoldás: Állandósult állapotbeli és tranziens analízis
- Követelmények formalizálása: CSL
 - Szintaxis: Állapot- és útvonalkifejezések
 - Szemantika: CTMC fogalmakkal
- Modellellenőrzés
 - Modell és követelmény együttes átalakítása
- Eszközök
- Követelmények (példák)