

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 állapot-elérhetőségi jellegű követelmények
 - QoS: Quality of Service
 - SLA: Service Level Agreement
- 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 indítás után 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.
- Jellemzők a követelményekre:
 - Adott állapotok (pl. szolgáltatási szintek) **valószínűségei**
 - Példa: Rendelkezésre állás, mint valószínűség (állandósult állapotban)
 - Állapotok (szolgáltatási szintek) fenntartásának **időtartama**
 - Példa: Javítási idő maximuma

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 $S(t)$ állapottal:
$$P\{S(t)=s \mid S(t_n)=s_n, S(t_{n-1})=s_{n-1}, \dots, S(t_0)=s_0\} = P\{S(t)=s \mid S(t_n)=s_n\}$$

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
 - Állapotok tartási ideje **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** már a folyamat az adott állapotban

Folytonos idejű Markov láncok (CTMC)

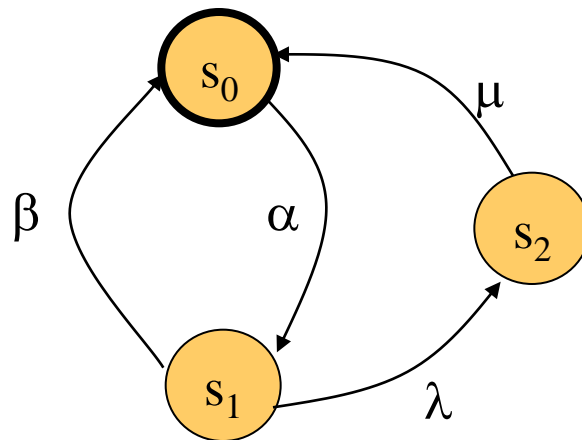
- CTMC: Continuous Time Markov Chain
 - Folytonos idő paraméter, diszkrét állapottér
- Jelölések, tulajdonságok:
 - Diszkrét állapotok: s_0, s_1, \dots, s_n , a CTMC állapota $S(t)$
 - Állapotátmeneti valószínűség: $Q_{ij}(t_{n-1}, t_n) = P\{S(t_n)=s_j \mid S(t_{n-1})=s_i\}$
 - Homogén Markov-folyamat: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

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

- Állapot elhagyás összesített rátája: $E(s) = \sum_{s' \in S} R_{s, s'}$

Egy egyszerű CTMC

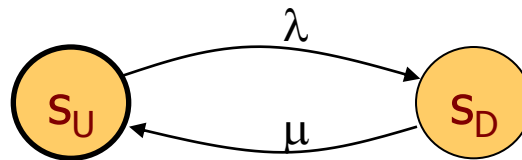
- CTMC szokásos megjelenítése:
 - Állapotok halmaza (kezdő valószínűségekkel)
 - Minden állapotpárra az **állapotátmeneti intenzitás** (ahol nem nulla, csak ott van feltüntetve állapotátmenet)



CTMC alkalmazások

- Megbízhatósági modellezés:

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

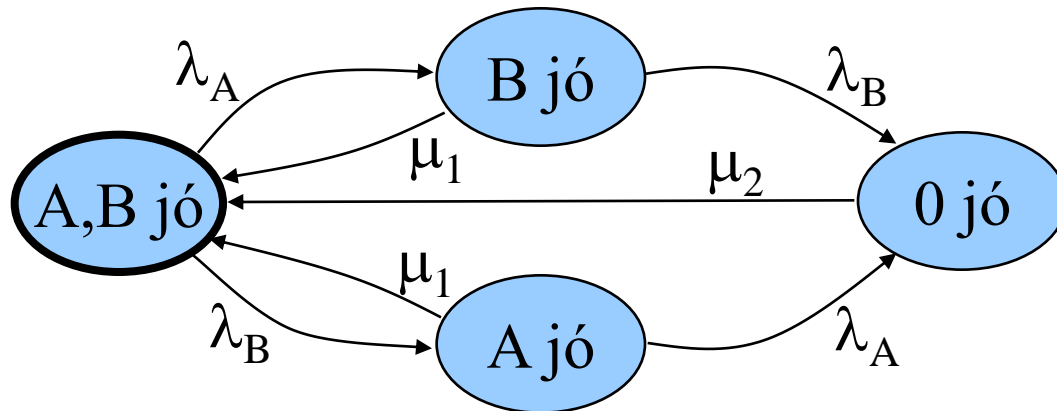


- Teljesítmény modellezés

- 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 hibás 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

- $\underline{E}(s) = \sum_{s' \in S} R(s, s')$ állapot elhagyás összesített intenzitása
- $\underline{Q} = \underline{R} - \text{diag}(\underline{E})$ „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:

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

- Állandósult állapot: ha 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:

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

A megoldás elemei

- Egy állapot tartási ideje:

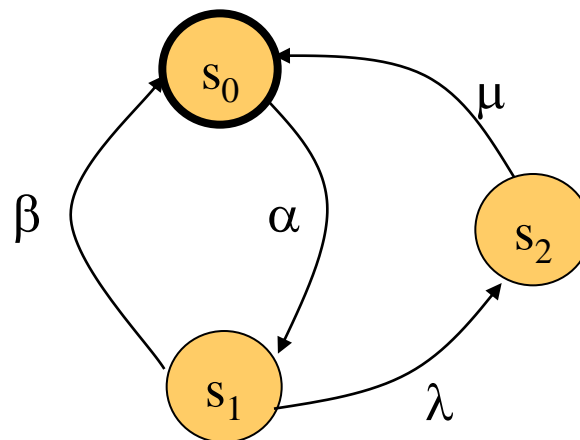
$$P\{\text{s-ben marad } t \text{ ideig}\} = e^{-E(s)t}$$

- Másképp: Egy állapot elhagyásának valószínűségére:

$$P\{\text{s elhagyása } t \text{ időn belül}\} = 1 - e^{-E(s)t}$$

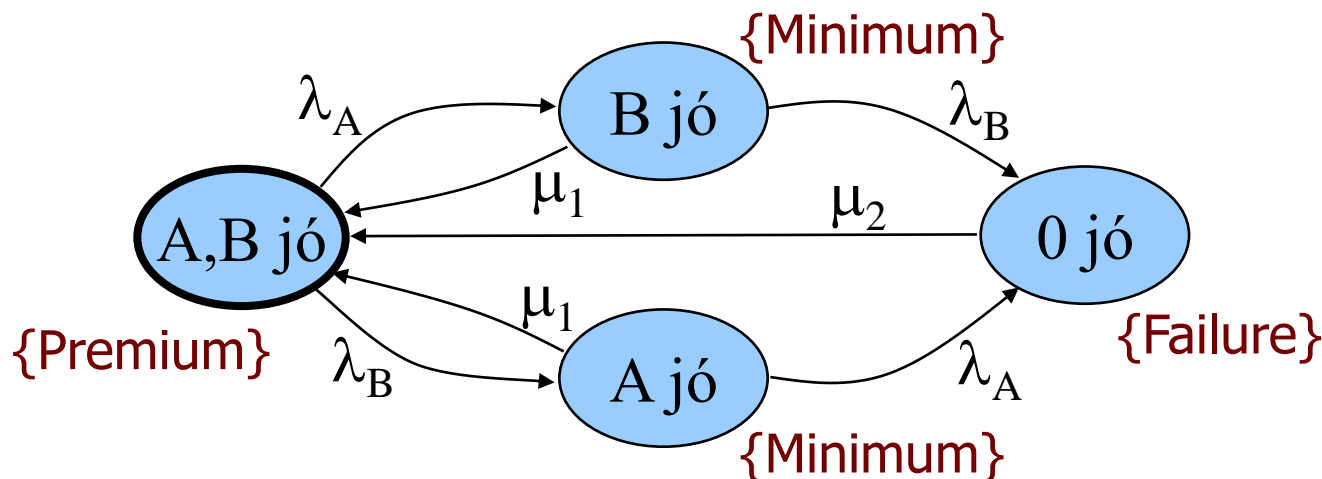
- Egy konkrét állapotátmenet valószínűsége:

$$P\{\text{s-ből s'-be megy át } t \text{ időn belül}\} = 1 - e^{-R(s,s')t}$$



Hogyan formalizálhatók a követelmények?

- Modell: CTMC, egyszerű állapot-alapú formalizmus
 - Kiterjesztés: Állapotok címkézése atomi kijelentésekkel



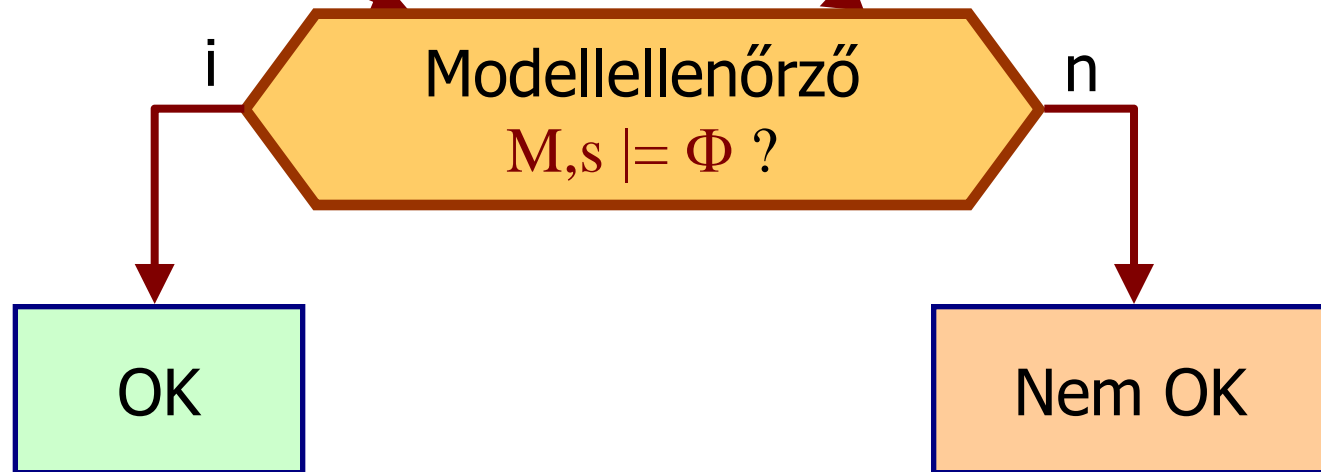
- Állapotokra: Számítható állandósult vagy tranziens valószínűségek
- Útvonalakra: Számítható bejárás valószínűségek
- Követelmények formalizálása: CTL analógia alapján
 - **CSL: Continuous Stochastic Logic**
 - Állapotokra és útvonalakra vonatkozó valószínűségi kifejezések és időtartamok megadása

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 állapotra: Állapot-kifejezések által megadott állapot-halmazokban való tartózkodás valószínűsége
 - (Tranziens) útvonalakra: Útvonal-kifejezések által megadott útvonal bejárásának valószínűsége
 - Időtartományok megadása:
 - X és U temporális operátorokhoz 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 a Φ tulajdonságú állapotokban való tartózkodás valószínűsége $\sim p$
 $P\{\text{olyan állapotban tartózkodik, ahol } \Phi \text{ igaz}\} \sim p$
 - Példa: $S_{>0,8}(\text{Minimum} \vee \text{Premium})$
 - $P_{\sim p}(\varphi)$ – a φ tulajdonságú utak bejárásának valószínűsége $\sim p$
 $P\{\text{olyan utat jár be, ahol } \varphi \text{ igaz}\} \sim p$
 - 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: $X^{[0,10]} \text{Premium}$
 - $\Phi_1 U^I \Phi_2$ – a $t \in I$ időpillanatban elérünk egy olyan állapotba, ahol Φ_2 igaz, és az odavezető úton Φ_1 igaz
 - 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. $\pi(s, \text{Sat}(\Phi)) \sim p,$

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

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

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

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

φ tulajdonságú útvonal bejárás vsz. $\sim p$

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

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

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

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

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 (ez már ellenpélda), így ezek is **nyelők** lesznek M' -ben
- A követelmény átalakítása M' esetén:
 - Bizonyítható tétel:
 $M, s \models \Phi_1 \cup^{[0,t)} \Phi_2$ ellenőrzésének eredménye ekvivalens
 $M', s \models \text{true} \cup^{[t,t]} \Phi_2$ ellenőrzésével (a módosított modellen!);
azaz a módosított modellen t -re tranziens analízis elég

CSL modellellenőrzők

- Az első megvalósítás:
ETMCC: Erlangen-Twente Markov Chain Checker (E|-MC²)
 - Markov-láncok
 - Sztochasztikus processz algebrák
- PRISM: Probabilistic Symbolic Model Checker
 - GreatSPN kiterjesztése
 - Szimbolikus állapottér kezelés
- 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 a tüzelő átmenetekre)

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

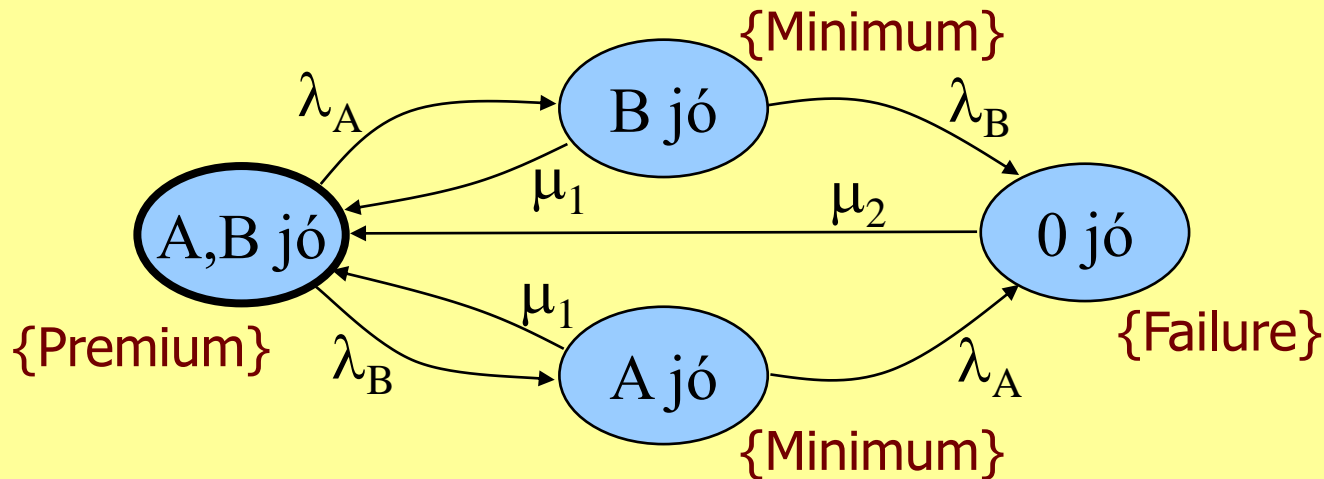
New Graph

T	N=3	N=4	N=5
0	0.000000	0.000000	0.000000
5	0.000001	0.000002	0.000001
10	0.000002	0.000004	0.000002
15	0.000003	0.000006	0.000003
20	0.000004	0.000008	0.000004

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:

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

- 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 kezdeti hiba esetén a helyreállítás több mint 15 időegységet igényel:

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

CSL használata QoS formalizálására III.

- Követelmények (folytatás):

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

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

- Minimum szolgáltatási szinttel indulva annak a valószínűsége több mint 70%, hogy 5 időegységen belül (ezalatt legalább a Minimum szintet megtartva) Premium szint nyújtható

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

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 útvonal kifejezések
 - Szemantika: CTMC fogalmakkal
- Modellellenőrzés
 - Modell és követelmény együttes átalakításával egyszerűsíthető
- Eszközök
- Követelmények (példák)