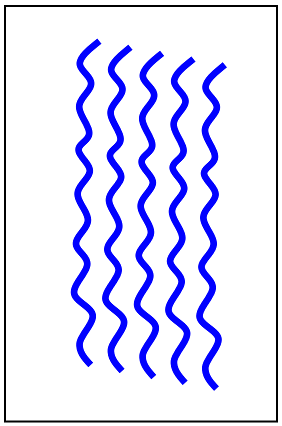
Learning Continuous Time Markov Chains from Sample Executions

Koushik Sen
Mahesh Viswanathan
Gul Agha
University of Illinois at Urbana Champaign

Motivation

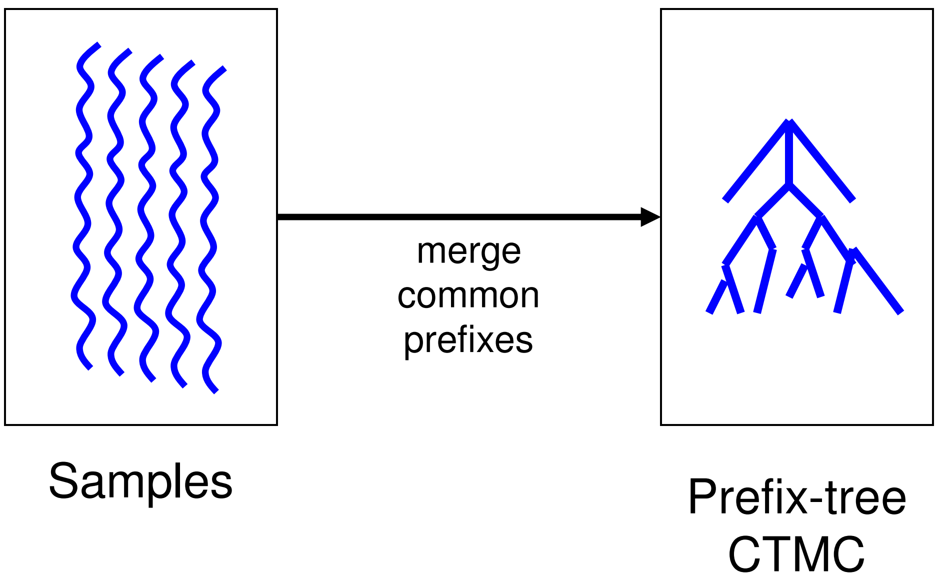
- Continuous-time Markov Chains (CTMC) are widely used
  - to model stochastic systems
  - to analyze performance and reliability
- Model $\rightarrow$ Analyze $\rightarrow$ Implement
- Implementation may not match the model
  - bugs introduced during coding
  - estimated values of parameters may differ from actual values
- Learn model (CTMC) from the sample execution of the implementation
  - learned model can be used for
    - performance evaluation
    - model-checking
    - generate smaller abstract model of the system

Overview of Our Approach

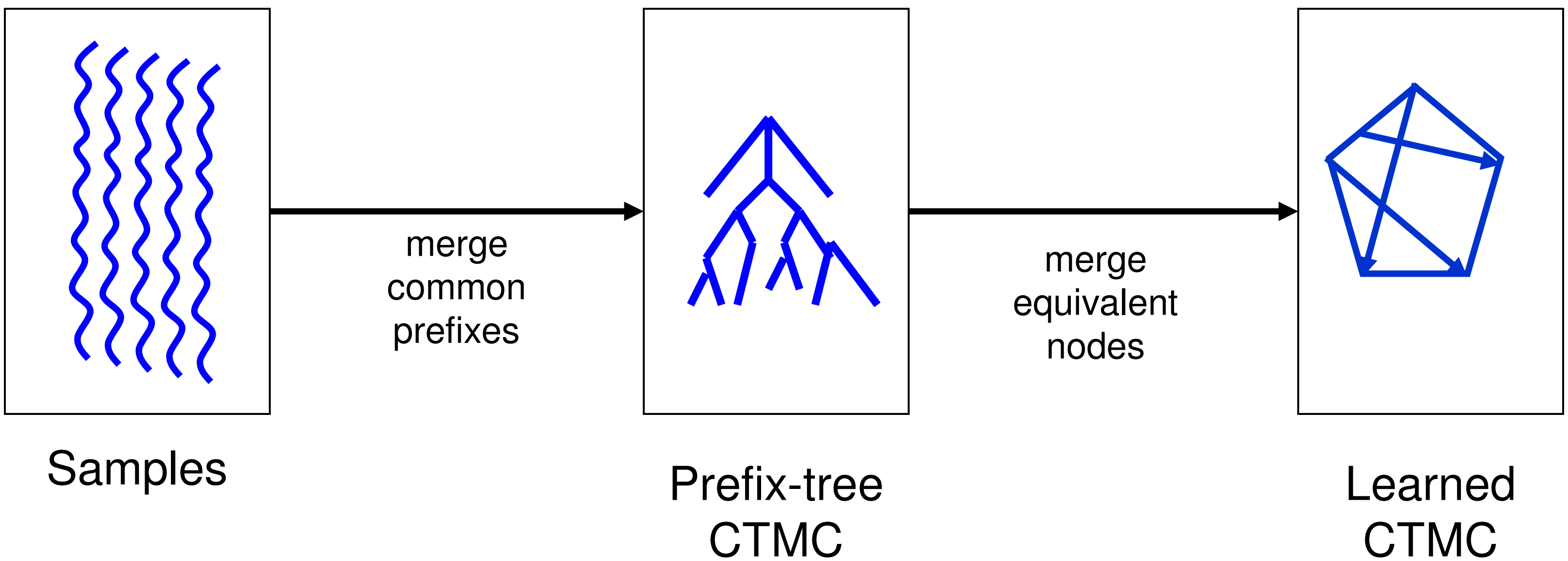


Samples

Overview of Our Approach



Overview of Our Approach



Edge Labeled Continuous-time Markov Chains ($CTML_L$)

- $M = (S,\Sigma,s_0,\delta,\rho,L)$
  - S : finite set of States
  - Σ : finite set of edge labels
  - $s_0 \in S$: initial state
  - $\delta : S \times \Sigma \to S$: state and edge label to next state
    - deterministic : $\delta(s,a)$ is unique
    - partial function
  - $\rho : S \times \Sigma \to R_{\geq 0}$: rates associated with transitions
  - $L : S \to 2^{AP}$: state to a set of atomic propositions

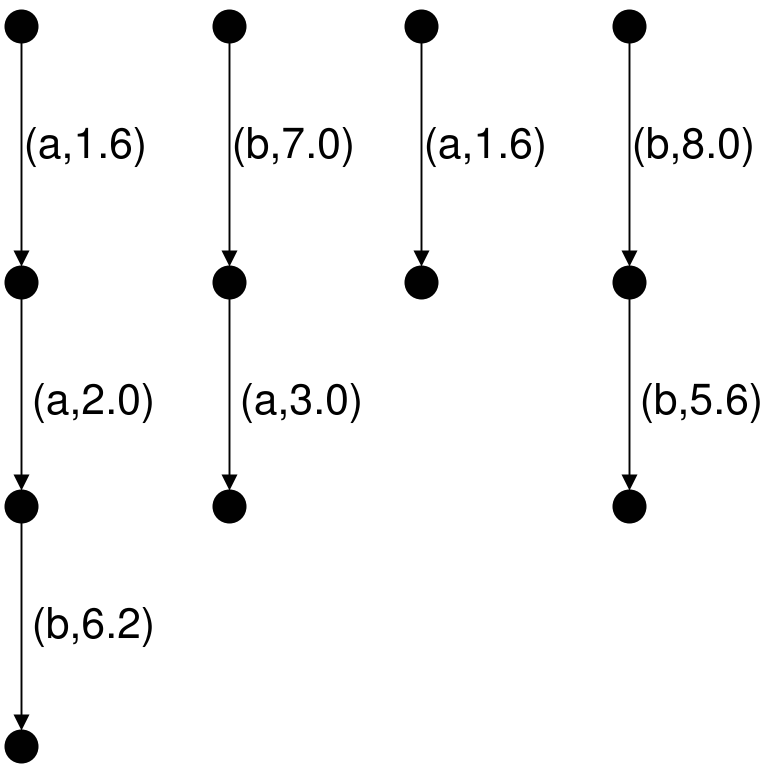
Semantics of $CTMC_L$

- $E(s) = \sum_{a \in \Sigma} \rho(s,a)$
- Probability of taking edge a from state s
  - $P(s,a) = \rho(s,a)/E(s)$
- Probability of leaving the state s within t units of time
  - $(1 - e^{-E(s)t})$
- Probability to move from s along edge a within t time units
  - $P(s,a)(1 - e^{-E(s)t})$
- Path is an infinite sequence
  - $l_0 \rightarrow_{(a_1,t1)} l_1 \rightarrow_{(a2,t2)} l_2 \rightarrow_{(a3,t3)} \cdots$

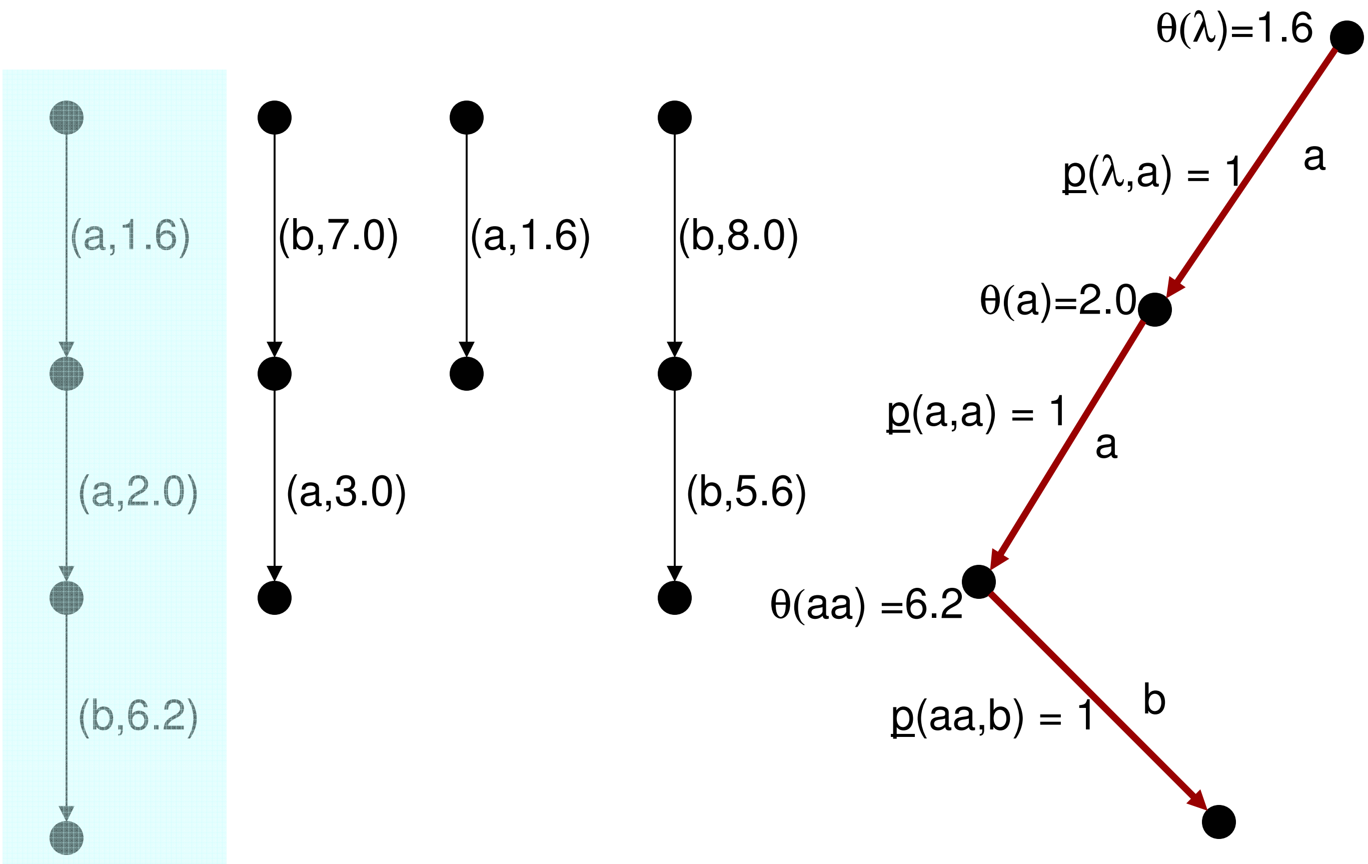
Generating Samples

- Executions of $CTMC_L$ are typically infinite
  - We need executions of finite length
- Generate samples from Finitary Edge Labeled Continuous-time Markov Chains ($CTMC^f_L$)
  - $F = (M,p)$
    - M : $CTMC_L$
    - p : stopping probability at any state s (say $p = 0.1$)
  - Technical tool
  - Paths are of finite length
  - Learn $CTMC^f_L$ instead of $CTMC_L$

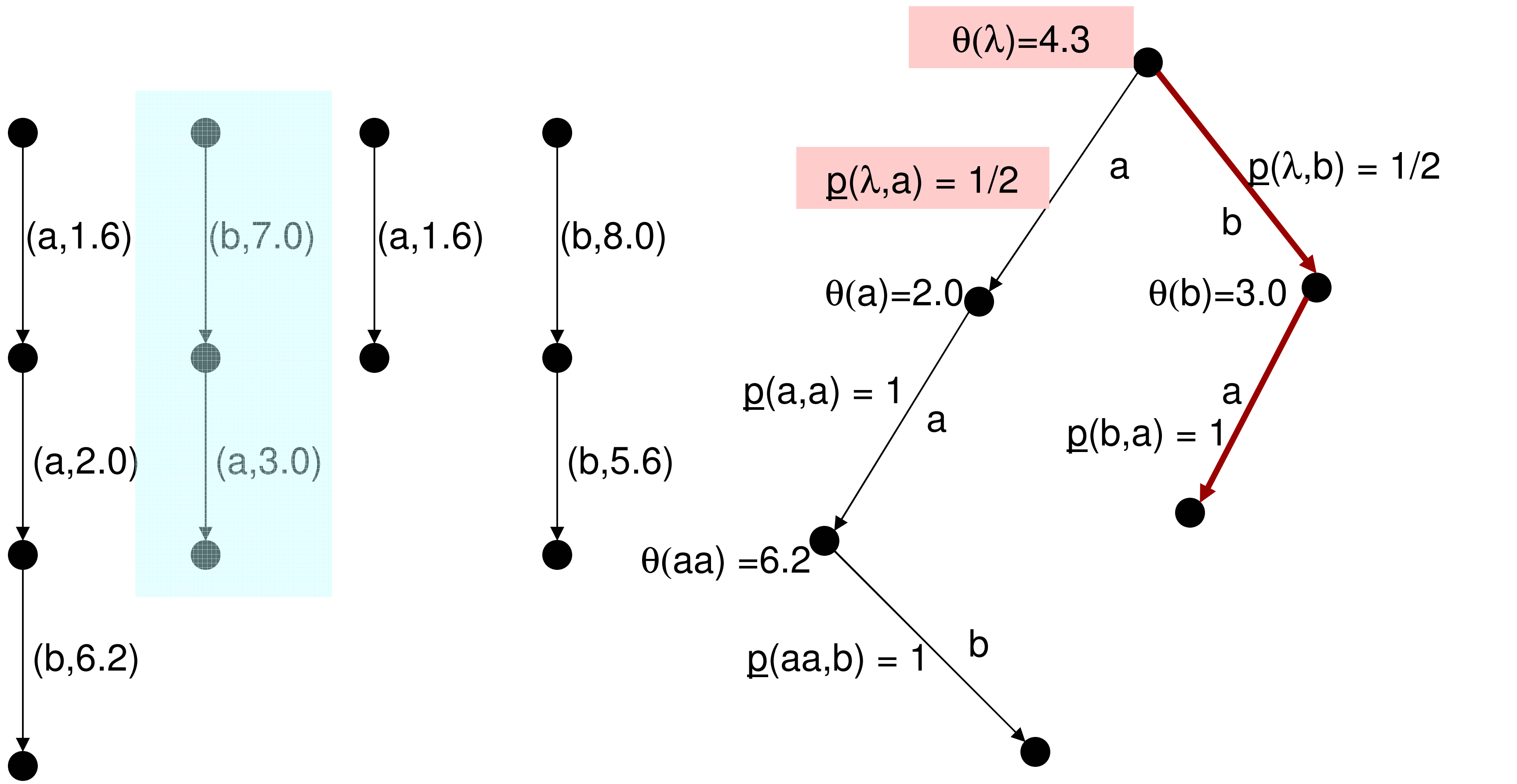
From Samples to Prefix-tree $CTMC^{f}_{L}$



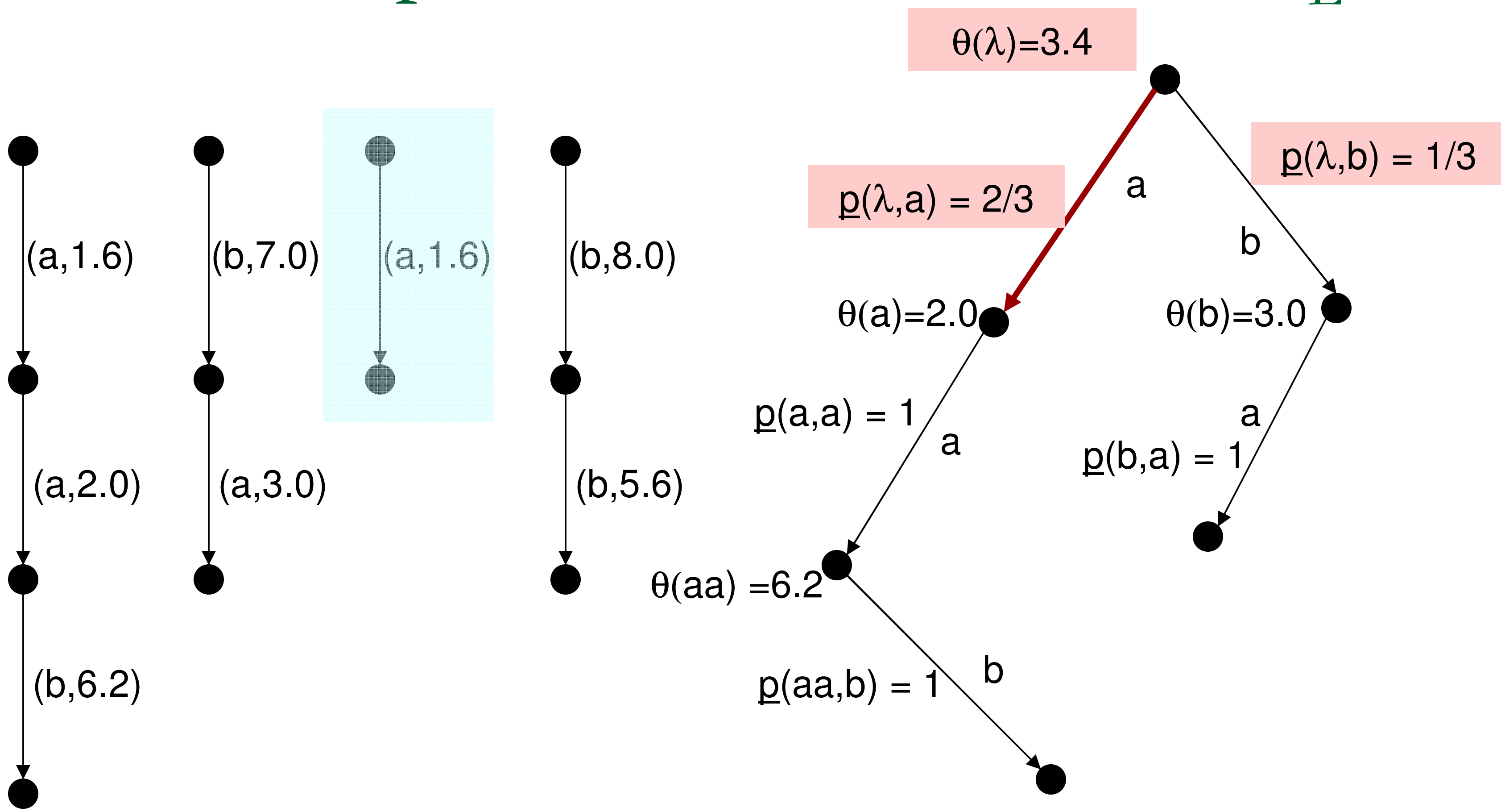
From Samples to Prefix-tree $CTMC^f_L$



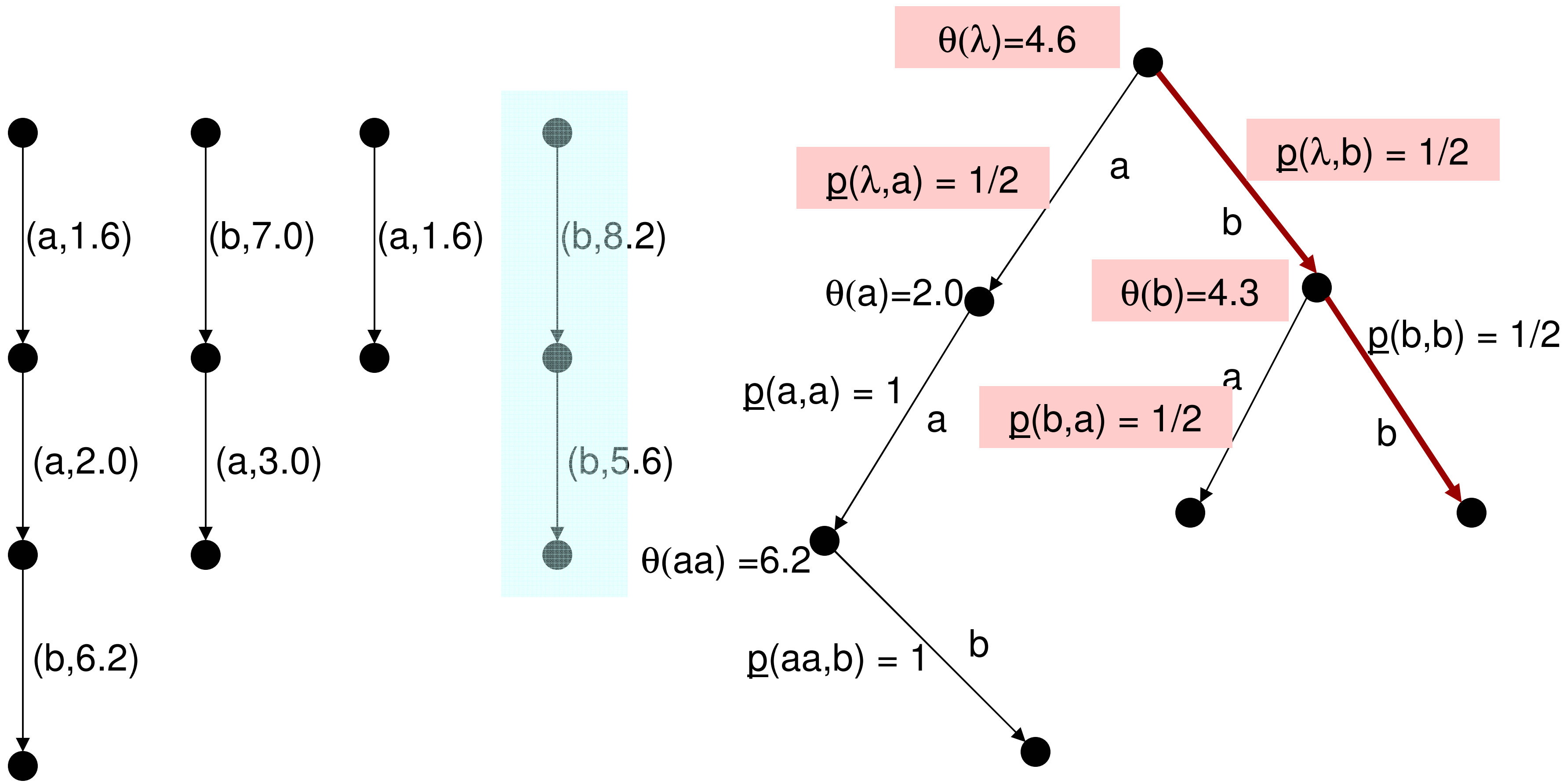
From Samples to Prefix-tree $CTMC^f_L$



From Samples to Prefix-tree $CTMC^f_L$



From Samples to Prefix-tree $CTMC^f{}_L$



Equivalent States

- Given $M = (S,\Sigma,s_0,\delta,\rho,L)$
- $R \subseteq S \times S$ is a stable relation
- $(s,s') \in R$ if and only if
  - $L(s) = L(s')$
  - $E(s) = E(s')$
  - $\forall a \in \Sigma$, if $\exists t \in S$ s.t. $\delta(s,a)=t$ then $\exists t' \in S$ s.t. $\delta(s',a)=t'$, $P(s,a)=P(s',a)$, and $(t,t') \in R$
  - vice-versa
- $s \equiv s'$ if and only if $\exists R$ s.t. $(s,s') \in R$

Equivalent $CTMC^f_L$

- Given a $CTMC^f_L$ F = (M,p), let F' = (M',p) be the $CTMC^f_L$ obtained by merging equivalent states.
  - The probability space defined by M and M' are the same

Learning Algorithm

- Given I^+ : a multi-set of finite sample executions from a $CTMC^f_L$
- A = Prefix tree $CTMC^f_L$ of I^+
- Merge states in A according to lexicographic order
  - merge if two states are equivalent (how?)
  - determinize after each merge
  - continue till no more merge is possible

Statistical Equivalence of States

- A = PCTMC(I^+) is built from experimental data
  - cannot test exact equivalence of two states in A
  - $s \equiv s'$ is replaced by $s \approx s'$ (compatible)
    - $L(s) = L(s')$
    - $E(s) \sim E(s')$ [statistical test]
    - $\forall a \in \Sigma$,
      - $P(s,a) \sim P(s',a)$ [statistical test]
      - $\delta(s,a) \approx \delta(s',a)$

$E(s) \sim E(s')$ with error α

- Check if means 1/E(s) and 1/E(s') of two exponential distributions are same
- Two exponential distributions with means θ_1 and θ_2
  - $H_0 : \theta_1 = \theta_2$ (or $\theta_1/\theta_2 = 1$)
  - $H_a : \theta_1 \neq \theta_2$ (or $\theta_1/\theta_2 \neq 1$)
- Let $x_1, x_2, \ldots, x_n$ are n samples from exponential distribution with mean θ_1
  - $\underline{\theta_1} = \sum_{i=1 \text{ to } n} x_i/n$
- Let $y_1, y_2, \ldots, y_m$ are m samples from exponential distribution with mean θ_2
  - $\underline{\theta_2} = \sum_{i=1 \text{ to } m} y_i/m$
- Accept H_a against H_0 if $\underline{\theta_1}/\underline{\theta_2}$ is not in $[r_{min}, r_{max}]$
  - $r_{min} = \mu - \sigma/\alpha$ and $r_{max} = \mu + \sigma/\alpha$
    - μ and σ are mean and standard deviation of $Z \sim F(2n,2m)$
  - $Prob[r_{min} \leq Z \leq r_{max}] < 1 - \alpha$

$P(s,a) \sim P(s',a)$ within error α

- f_1 tries are 1 out of n_1 tries from a Bernoulli distribution with mean p_1
- f_2 tries are 1 out of n_2 tries from a Bernoulli distribution with mean p_2
- p_1 and p_2 are statistically same if

$$\left|\frac{f_1}{n_1} - \frac{f_2}{n_2}\right| < \sqrt{\frac{1}{2}\log\frac{2}{\alpha}}\left(\frac{1}{\sqrt{n_1}} + \frac{1}{\sqrt{n_2}}\right)$$

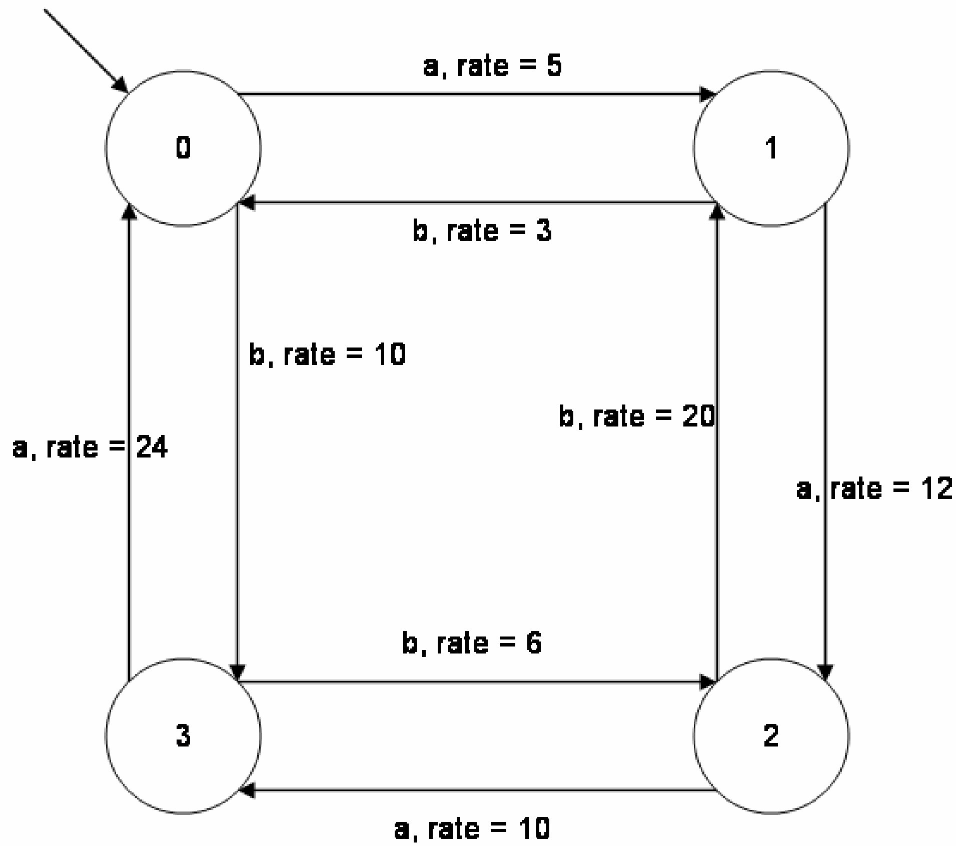
(Hoeffding bound)

Learning in the Limit

- Non-zero probability of getting structurally complete sample I^+
  - for every s there is a $\tau \in I^+$ such that τ contains s
  - for every transition (s,a) there is a $\tau \in I^+$ such that τ traverses (s,a)
- Error goes to 0 as $|I^+|$ goes to ∞
  - Type I error : compatible returns false when two states are equivalent
  - Type II error : compatibility returns true when two states are not equivalent

Tool and Experiments

- Implemented as a sub-component of the tool VESTA (Verification based on Statistical Analysis)
  - http://osl.cs.uiuc.edu/~ksen/vesta/
- Symmetric CTMC
  - 600 samples and α=0.0001



Conclusion and Related Work

- Similar learning algorithms
  - Regular languages– RPNI (Oncina and Garcia'92)
  - Stochastic regular grammar- ALERGIA (Carrasco and Oncina'94)
  - Continuous-time Hidden Markov Model- Wei, Wang, Towsley'02
    - Fix the size of HMM that they want to learn
- Our approach may not scale for large systems
  - requires lot of samples
  - Can we combine this approach with statistical model-checking?

Prefix Tree $CTMC^f_L$ for I^+ ($PCTMC(I^+)$)

- I^+ = multi-set of finite traces $\tau = l_0 \rightarrow_{(a_1,t_1)} l_1 \rightarrow_{(a_2, t_2)} l_2 \ldots \rightarrow_{(a_n,t_n)} l_n$
- S = all prefixes of I^+
  - $Pr(\tau) = \{\lambda, a_1, a_1 a_2, \ldots, a_1 a_2 \ldots a_n\}$ (all prefixes)
  - I^+ : multi-set of samples

Prefix Tree $CTMC^f_L$ for I^+ (PCTMC(I^+))

- I^+ = multi-set of finite traces $\tau = l_0 \to_{(a_1,t_1)} l_1 \to_{(a_2,t_2)} l_2 \ldots \to_{(a_n,t_n)} l_n$
- S = all prefixes of I^+
- $s_0 = \lambda$
- $\delta(x,a) = xa$ if $xa \in S$
  $= \bot$ otherwise
- $P(x,a) = \underline{p}(x,a,I^+)$
  - $n(x,I^+)$ = number of τ in I^+ such that x is prefix of τ
  - $n'(x,I^+) = n(x,I^+)$ - number x in I^+
  - $\underline{p}(x,a,I^+) = n(xa,I^+)/n'(x,I^+)$ if $n'(x,I^+) > 0$
    $= 0$ otherwise

Prefix Tree $CTMC^{f}_{L}$ for I^{+} (PCTMC(I^{+}))

- I^{+} = multi-set of finite traces $\tau = I_0 \rightarrow_{(a_1,t_1)} I_1 \rightarrow_{(a_2,t_2)} I_2 \ldots \rightarrow_{(a_n,t_n)} I_n$
- S = all prefixes of I^{+}
- $s_0 = \lambda$
- $\delta(x,a) = xa$ if $xa \in S$

  $= \perp$ otherwise
- $P(x,a) = \underline{p}(x,a,I^{+})$
- $E(x) = 1/\underline{\theta}(x,I^{+})$
  - $\theta(x,a,\tau) = t_i$ where $x = a_1 a_2 \ldots a_{i-1}$ and $a = a_i$
  - $\underline{\theta}(x,I^{+}) = \sum_{a\in\Sigma}\sum_{\tau\in I^{+}} \theta(x,a,\tau)/n'(x,I^{+})$ if $n'(x,I^{+}) > 0$

    $= 0$ otherwise

Prefix Tree $CTMC^f_L$ for I^+ ($PCTMC(I^+)$)

- I^+ = multi-set of finite traces $\tau = l_0 \rightarrow_{(a_1,t_1)} l_1 \rightarrow_{(a_2,t_2)} l_2 \dots \rightarrow_{(a_n,t_n)} l_n$
- S = all prefixes of I^+
- $s_0 = \lambda$
- $\delta(x,a) = xa$ if $xa \in S$

  $= \perp$ otherwise
- $P(x,a) = \underline{p}(x,a,I^+)$
- $E(x) = 1/\underline{\theta}(x,I^+)$
- $\rho(x,a) = P(x,a)E(x)$
- $L(x) = L(x,I^+)$
- p is stopping probability with which I^+ are generated