

# Expanding the Boundaries of Performance Requirement Representation with Performance Trees

Tamas Suto

Department of Computing  
Imperial College London

PASTA'06  
30 June 2006

# Current Boundaries of Query Representation

- Existing techniques for specifying performance queries are based on stochastic logics
- Steep learning curve – not straightforward to use
- Designed for model checking – constrained expressiveness
- Opportunities for improvement:
  - Need to ease usability
  - Need to expand scope of expressiveness

# A Way to Go

- Maintain expressiveness of stochastic logics
- Do not restrict to model checking-specific queries – incorporate quantitative analysis capabilities
- Use a graphical specification mechanism instead of logic
- Design a new formalism, incorporating all of the above, independent of stochastic logics

# Query Specification with CSL

- Traditional performance query specification is based on CSL
- Operates on the state level (on CTMCs)

## Definition

$$\begin{aligned} \sigma &\stackrel{\text{def}}{=} tt \mid a \mid \neg\sigma \mid \sigma \wedge \sigma \mid \mathcal{S}_{\bowtie p}(\sigma) \mid \mathcal{P}_{\bowtie p}(\varphi) \\ \varphi &\stackrel{\text{def}}{=} \mathcal{X}^l\sigma \mid \sigma \mathcal{U}^l\sigma \end{aligned}$$

- Can express steady-state, path-based and nested constraints

# Query Specification with aCSL

- An augmented version of CSL that allows the incorporation of actions into performance queries
- Operates on the state level (on CTMCs)

## Definition

$$\begin{aligned} \sigma &\stackrel{\text{def}}{=} tt \mid a \mid \neg\sigma \mid \sigma \wedge \sigma \mid S_{\text{Exp}}(\sigma) \mid P_{\text{Exp}}(\varphi) \\ \varphi &\stackrel{\text{def}}{=} \sigma AU^{<t}\sigma \mid \sigma AU_B^{<t}\sigma \end{aligned}$$

- Can express steady-state, path-based and nested constraints with action restrictions

# Query Specification with eCSL

- A significantly enhanced version of CSL that distances itself from path-based constraints
- Operates on the model level (on SM-SPNs)

## Definition

$$\sigma \stackrel{\text{def}}{=} tt \mid \neg\sigma \mid \sigma \wedge \sigma \mid p[N]$$

$$\varphi \stackrel{\text{def}}{=} tt \mid \neg\varphi \mid \varphi \wedge \varphi \mid \mathcal{S}_\rho(\sigma) \mid \mathcal{I}_\rho^T(\sigma, \sigma) \mid \mathcal{P}_\rho^T(\sigma, \sigma)$$

- Can express steady-state, transient and passage time requirements

# Query Specification with Performance Trees

- Graphical query representation formalism
- Offers full expressiveness of CSL, aCSL and eCSL
- Incorporates measure extraction capabilities
- Can deal with high-level concepts that were not expressible previously

# Query Specification with Performance Trees

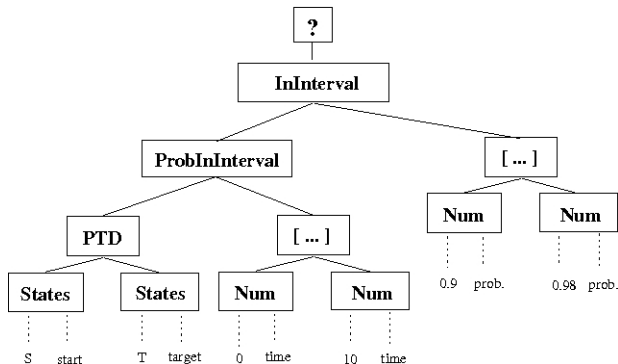
## Example

*“Does a passage between the set of states  $S$  and the set of states  $T$  occur within the time interval  $[0, 10]$  with probability lying in the range  $[0.9, 0.98]$ ?”*

# Query Specification with Performance Trees

## Example

*“Does a passage between the set of states  $S$  and the set of states  $T$  occur within the time interval  $[0, 10]$  with probability lying in the range  $[0.9, 0.98]$ ?”*



# Query Specification with Performance Trees

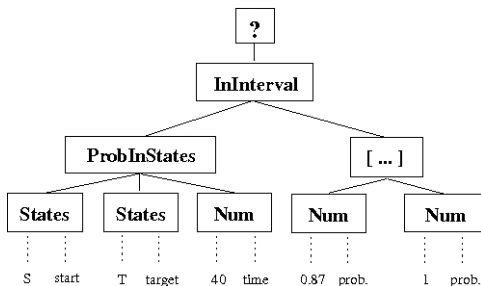
## Example

*“Is the probability that the system will be in the set of states  $T$  at time instant 40, given that the system has originally started from the set of states  $S$ , greater than 0.87?”*

# Query Specification with Performance Trees

## Example

*“Is the probability that the system will be in the set of states  $T$  at time instant 40, given that the system has originally started from the set of states  $S$ , greater than 0.87?”*



# Query Specification with Performance Trees

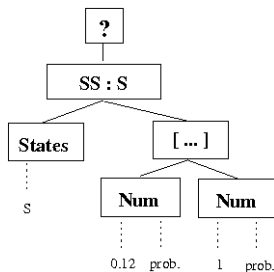
## Example

*“Out of the set of states  $S$ , which states have a steady-state probability greater than 0.12?”*

# Query Specification with Performance Trees

## Example

*“Out of the set of states  $S$ , which states have a steady-state probability greater than 0.12?”*



# Query Specification with Performance Trees

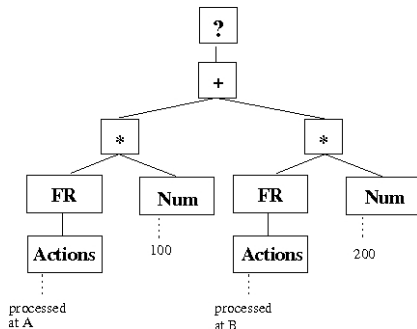
## Example

*“What is the productivity of the system, defined as the sum of the mean firing rate of action ‘processed at A’ multiplied by 100, and the mean firing rate of action ‘processed at B’ multiplied by 200?”*

# Query Specification with Performance Trees

## Example

*“What is the productivity of the system, defined as the sum of the mean firing rate of action ‘processed at A’ multiplied by 100, and the mean firing rate of action ‘processed at B’ multiplied by 200?”*



# Query Specification with Performance Trees

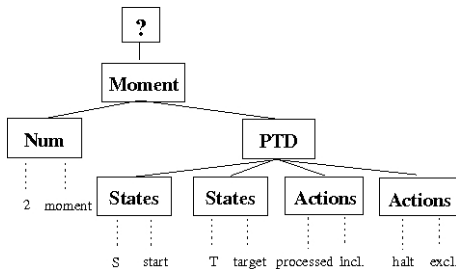
## Example

*“What is the variance of the passage time defined over the set of start states  $S$  and the set of target states  $T$ , with the constraint that action ‘processed’ takes place at least once and that the action ‘halt’ does not occur during the passage?”*

# Query Specification with Performance Trees

## Example

*“What is the variance of the passage time defined over the set of start states  $S$  and the set of target states  $T$ , with the constraint that action ‘processed’ takes place at least once and that the action ‘halt’ does not occur during the passage?”*



# Query Specification with Performance Trees

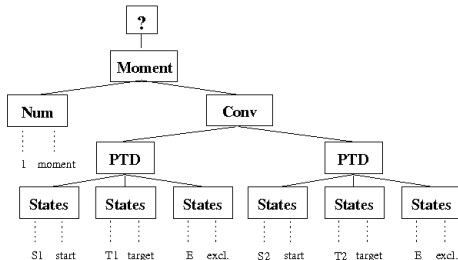
## Example

*“What is the average time required to complete the passage defined by the convolution of the passage from the set of start states  $S1$  to the set of target states  $T1$  with the passage from the set of start states  $S2$  to the set of target states  $T2$ , having the additional constraint that the set of states  $E$  is excluded from both passages?”*

# Query Specification with Performance Trees

## Example

*“What is the average time required to complete the passage defined by the convolution of the passage from the set of start states  $S1$  to the set of target states  $T1$  with the passage from the set of start states  $S2$  to the set of target states  $T2$ , having the additional constraint that the set of states  $E$  is excluded from both passages?”*



# Query Specification with Performance Trees

## 1 Operation Nodes

<b>?</b>	: result of query
<b>;</b>	: sequential evaluation
$\vee$	: conjunction or disjunction
$\neg$	: negation
$\bowtie$	: binary comparison ( $<$ , $\leq$ , $=$ , $\geq$ , $>$ )
$\oplus$	: arithmetic operation ( $+$ , $-$ , $*$ , $\div$ )
<b>PTD</b>	: passage time density
<b>Dist</b>	: passage time distribution
<b>Conv</b>	: convolution
<b>Moment</b>	: higher moment
<b>SS:P</b>	: steady-state probability of a set of states
<b>SS:S</b>	: set of states having a certain steady-state prob.
<b>FR</b>	: firing rate of a transition / action

# Query Specification with Performance Trees

## 1 Operation Nodes (ctd.)

- InInterval** : is numerical value in a given range?
- ProbInInterval** : prob. with which passage occurs in a certain amount of time
- ProbInStates** : transient probability measure
- StatesAtTime** : set of states that the system is occupying at a given time with a certain probability

## 2 Value Nodes

- [...]** : numerical interval
- States** : set of states
- Actions** : set of actions
- Num** : numerical value
- Bool** : boolean value

# Stochastic Logics vs. Performance Trees

- One-inch logics – very concise, yet very restricted
- Hard to work with them – require expert understanding
- Lack a desired level of expressiveness
- Are not equipped for measure extraction

# Stochastic Logics vs. Performance Trees

- More natural to specify queries as PTs than as SLs
- Can represent a large variety of performance queries
- Subsume CSL, aCSL and eCSL
- Can be used for model checking and measure extraction
- Defined on an abstract level – can be applied to general stochastic systems
- Have an extensive syntax and type system

# Work in Progress

- Semantics
- Application to different underlying modelling formalisms
- Tool implementation

Thank you for your attention.

Any questions?