A Logic for Graphs with QoS

VODCA 2004
11-12 September 2004

Alberto Lluch Lafuente, Gianlugi Ferrari
Dipartimento di Informatica, Università di Pisa
Introduction

- We are interested in Specification and Verification of WAN applications.
- Systems involving concurrency, distribution, mobility and ... QoS.
- QoS: non-functional aspects, like network bandwidth, prize, etc.
- Service Overlay Network Applications are an example.
- Graph Transformation Systems are suitable modeling formalisms.
- Transition system where each state is (or has a associated) a graph representing, for instance, the interconnection between components.
Graph Logics

- Graph Logics allow to express properties of graphs.
- Examples: MSO (and its fragments), GL.
Temporal Logics

- Temporal logics allow to reason about the ordering of events in time.
- Examples: $\mu$-calculus, CTL*, CTL, LTL.
Logics for GTSs

- Allow to reason about the evolution of graphs in time.
- Examples: Courcelle, Baldan et. al, A. Rensink, etc.
- Graphs enriched with QoS attributes.
- We need logics to reason about QoS.
Our formal model for QoS: c-semirings

- Each attribute represented by a (partially) ordered set of values.
- An operation to select among values (generalized addition, or min).
- An operation to combine values (generalized multiplication).
- Support of composition of different attributes (multicriteria).
C-semirings: Definition

A c-semiring is a tuple $\langle A, +, \times, 0, 1 \rangle$ such that:

- $A$ is a set;
- $0$ and $1$ are elements of $A$;
- $+: 2^A \to A$ is defined over (possibly infinite) sets of elements of $A$ as follows:\footnote{When $+$ is applied to a set with two elements we use $+$ as binary operator in infix notation, while in all other cases we use symbol $\sum$ in prefix notation.}: $\sum\{a\} = a$, $\sum\emptyset = 0$, $\sum A = 1$ and $\sum(\bigcup A_i) = \sum\{\sum A_i\}$, for $A_i \subseteq A$, $i \geq 0$;
- $\times : A \times A \to A$ is a binary associative, commutative operation that distributes over $+$, has $1$ as unit element and $0$ as absorbing element.
## C-semirings: Examples

C-semirings are the formal structure of many QoS attributes.

<table>
<thead>
<tr>
<th>$A$</th>
<th>$+$</th>
<th>$\times$</th>
<th>0</th>
<th>1</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>${ T, F }$</td>
<td>$\lor$</td>
<td>$\land$</td>
<td>$F$</td>
<td>$T$</td>
<td>Network and service availability</td>
</tr>
<tr>
<td>$\mathbb{R}^+$</td>
<td>$\min$</td>
<td>$+$</td>
<td>$+\infty$</td>
<td>0</td>
<td>Price, propagation delay</td>
</tr>
<tr>
<td>$\mathbb{N}^+$</td>
<td>$\min$</td>
<td>$+$</td>
<td>$+\infty$</td>
<td>0</td>
<td>Price, propagation delay</td>
</tr>
<tr>
<td>$\mathbb{R}^+$</td>
<td>$\max$</td>
<td>$\min$</td>
<td>0</td>
<td>$+\infty$</td>
<td>Bandwidth</td>
</tr>
<tr>
<td>$[0, 1]$</td>
<td>$\max$</td>
<td>$\cdot$</td>
<td>0</td>
<td>1</td>
<td>Performance</td>
</tr>
<tr>
<td>$[0, 1]$</td>
<td>$\max$</td>
<td>$\min$</td>
<td>0</td>
<td>1</td>
<td>Performance, security degree</td>
</tr>
<tr>
<td>$2^N$</td>
<td>$\lor$</td>
<td>$\land$</td>
<td>$\emptyset$</td>
<td>$N$</td>
<td>Capabilities, access rights</td>
</tr>
</tbody>
</table>
• A formula is evaluated as a c-semiring value.
• Basically, \( \vee \) and \( \wedge \) are substituted by \( + \) and \( \times \).
• Eventually \( x \) and \( y \) connected
  \( \Rightarrow \) Optimal QoS of connection between \( x \) and \( y \) ever achieved.
Graph Logic for QoS

Again...

- A formula is evaluated as a c-semiring value.
- Basically, ∨ and ∧ are substituted by + and ×.
- $x$ are $y$ connected
  $\Rightarrow$ Best QoS of connection between $x$ and $y$. 
Graph Model: Definition

A graph is a tuple \( \langle X \cup E, \text{edge}, C, K, I \rangle \), where:

- \( X \subseteq \mathcal{X} \) is a (finite) set of nodes,
- \( E \subseteq \mathcal{E} \) is a (finite) set of edges,
- \( \text{edge} : E \to X \times X \) associates edges with source and target,
- \( C \) is a c-semiring,
- \( K \subseteq \mathcal{K} \) is a set of cost function symbols,
- \( I \) is an interpretation of symbols.
**Graph Model: Decomposition**

A decomposition of $G$ is a pair of graphs $G_1, G_2$ such that $E_1 \cup E_2 = E$, $\text{edge}_1 \cup \text{edge}_2 = \text{edge}$.

The set of all decompositions of a graph $G$ will be denoted by $\Theta(G)$. The size of $\Theta(G)$ is $2^{|E|}$.
**Graph Logic: Syntax**

Let $V_X$ be a set of node variables, and $V_R$ be a set of recursion variables.

$$
\phi ::= \text{nil} \mid k(\xi) \mid k(\xi, \xi) \mid \phi \mid \phi \mid \phi \parallel \phi \quad \text{spatial operators}
$$

$$
\phi + \phi \mid \phi \times \phi \mid f(\phi, \ldots, \phi) \quad \text{c-semiring operators}
$$

$$
\sum x.\phi \mid \prod x.\phi \quad \text{quantification}
$$

$$
\text{r}(\bar{\xi}) \mid (\mu r(\overline{x}).\phi)\bar{\xi} \mid (\nu r(\overline{x}).\phi)\bar{\xi} \quad \text{fixpoints}
$$

where $x \in \mathcal{X}$, $x \in V_X$, $\xi \in \mathcal{X} \cup V_X$, $k \in \mathcal{K}$, $f \in \mathcal{F}$, $r \in V_R$. 

15
Graph Logic: Semantics (void and costs)

\[ [\text{nil}]_{\sigma;\rho}(G) = (E = \emptyset) \]

1 if the graph has no edges, 0 otherwise.

\[ [k(\xi)]_{\sigma;\rho}(G) = I(k)(x\sigma) \]

\( k \)-cost of node

\[ [k(\xi_1, \xi_2)]_{\sigma;\rho}(G) = (E = \{e\}) \times (\text{edge}(e) = (\xi_1\sigma, \xi_2\sigma)) \times I(k)(e) \]

\( k \)-cost of edge (if it is the only one in the graph, else 0).
Graph Logic: Semantics (decomposition)

\[
\llbracket \phi_1 \mid \phi_2 \rrbracket_{\sigma; \rho}(G) = \sum_{(G_1, G_2) \in \Theta(G)} \{\llbracket \phi_1 \rrbracket_{\sigma; \rho}(G_1) \times \llbracket \phi_2 \rrbracket_{\sigma; \rho}(G_2)\}
\]

\[
\llbracket \phi_1 \parallel \phi_2 \rrbracket_{\sigma; \rho}(G) = \prod_{(G_1, G_2) \in \Theta(G)} \{\llbracket \phi_1 \rrbracket_{\sigma; \rho}(G_1) + \llbracket \phi_2 \rrbracket_{\sigma; \rho}(G_2)\}
\]
Graph Logic: Semantics (c-semiring operations)

\[
\begin{align*}
[\phi_1 + \phi_2]_{\sigma;\rho}(G) &= [\phi_1]_{\sigma;\rho}(G) + [\phi_2]_{\sigma;\rho}(G) \\
[\phi_1 \times \phi_2]_{\sigma;\rho}(G) &= [\phi_1]_{\sigma;\rho}(G) \times [\phi_2]_{\sigma;\rho}(G) \\
[f(\phi_1, \ldots, \phi_n)]_{\sigma;\rho}(G) &= I(f)([\phi_1]_{\sigma;\rho}(G), \ldots, [\phi_n]_{\sigma;\rho}(G))
\end{align*}
\]
Graph Logic: Semantics (Quantification)

\[ \sum_{x} \phi \]_{\sigma; \rho}(G) = \sum_{x \in X} \mathcal{L}_{\sigma; \rho}(G)

\[ \prod_{x} \phi \]_{\sigma; \rho}(G) = \prod_{x \in X} \mathcal{L}_{\sigma; \rho}(G)
Graph Logic: Semantics (fixpoints)

\[
[r(\overline{\xi})]_{\sigma; \rho}(G) = r\rho(\overline{\xi}\sigma)
\]

\[
[(\mu r(\overline{x}).\phi)\overline{\xi}]_{\sigma; \rho}(G) = \text{lfp}(\lambda s. \lambda \overline{y}. [[\phi]_{\sigma[\overline{y}/\overline{x}], \rho[s/r]}](\overline{\xi}\sigma))(G)
\]

\[
[(\nu r(\overline{x}).\phi)\overline{\xi}]_{\sigma; \rho}(G) = \text{gfp}(\lambda s. \lambda \overline{y}. [[\phi]_{\sigma[\overline{y}/\overline{x}], \rho[s/r]}](\overline{\xi}\sigma))(G)
\]
There is a path from $x$ to $y$

$$\text{path}(x, y) \equiv \mu r(x, y). (x = y) + \sum z.e(x, z)|r(z, y).$$

QoS of optimal path from $x$ to $y$

$$\text{costp}(x, y) \equiv \mu r(x, y). (x = y) + \sum z.cost(x, z)|r(z, y)$$
Graph Logic: Verification

- GL is PSPACE-complete (with and without recursion).

- Without recursion (abstracting from c-semiring operations):
  - \texttt{nil}, \( k(\xi) \) and \( k(\xi, \xi) \) can be done in \( O(1) \).
  - \( \sum x.\phi \) and \( \prod x.\phi \) require \( O(|X|) \).
  - \( \phi|\phi \) and \( \phi||\phi \) require to consider all the \( 2^{|E|} \) decompositions.
    \( \Rightarrow \) this leads to \( O(2^{|E|}|\phi|) \).

- With recursion:
  - Iteration might not terminate.
  - Combine recursion with composition.
    \( \Rightarrow \) Conjecture: Guarantees termination.
    \( \Rightarrow \) Open question: Limits expressivity of practical problems?
Conclusion

- We have extended a graph logic to include QoS attributes.
- We have temporal logics to reason about QoS in time.
- We can define a logic to reason about spatial, temporal and QoS properties of systems.