Decomposition of Distributed Nonmonotonic Multi-Context Systems

Seif El-Din Bairakdar  Minh Dao-Tran  Thomas Eiter
Michael Fink  Thomas Krennwallner

KBS Group, Institute of Information Systems, Vienna University of Technology

JELIA 2010
Sep 15, 2010
Overview

Heterogeneous & Nonmonotonic Multi-Context Systems

Decomposition of Nonmonotonic Multi-Context Systems

Experiments

Conclusions
Multi-Context Systems (MCS)

- MCSen introduced by [Giunchiglia and Serafini, 1994]:
  - represent inter-contextual information flow
  - express reasoning w.r.t. contextual information
  - allow decentralized, pointwise information exchange
  - monotonic, homogeneous logic

- Framework extended for integrating heterogeneous and nonmonotonic logics [Brewka and Eiter, 2007]
Syntax of Multi-Context Systems

- multi-context system
  - a collection $M = (C_1, \ldots, C_n)$ of contexts
- context $C_i = (L_i, kb_i, br_i)$
  - $L_i$: a logic
  - $kb_i$: a knowledge base of logic $L_i$
  - $br_i$: a set of bridge rules
Syntax of Multi-Context Systems

▶ multi-context system
  ▶ a collection $M = (C_1, \ldots, C_n)$ of contexts

▶ context $C_i = (L_i, kb_i, br_i)$
  ▶ $L_i$: a logic
  ▶ $kb_i$: a knowledge base of logic $L_i$
  ▶ $br_i$: a set of bridge rules

▶ logic $L = (\text{KB}_L, \text{BS}_L, \text{ACC}_L)$
  ▶ $\text{KB}_L$: set of well-formed knowledge bases
  ▶ $\text{BS}_L$: is the set of possible belief sets
  ▶ $\text{ACC}_L$: acceptability function $\text{KB}_L \mapsto 2^{\text{BS}_L}$
    Which belief sets are accepted by a knowledge base?
Syntax of Multi-Context Systems (bridge rules)

- **multi-context system**
  
  \[ M = (C_1, \ldots, C_n) \]

- **context**

  \[ C_i = (L_i, kb_i, br_i) \]

- **logic**

  \[ L_i = (KB_i, BS_i, ACC_i) \]

- **Bridge rule** \( r \in br_i \) of a context \( C_i \)

  \[ s \leftarrow (c_1 : p_1), \ldots, (c_j : p_j), \not (c_{j+1} : p_{j+1}), \ldots, \not (c_m : p_m) \]

  \( (c_k : p_k) \) looks at belief \( p_k \) in context \( C_{c_k} \)

  \( r \) is applicable \( \iff \) positive/negative beliefs are present/absent

  we add the head \( s \) to \( kb_i \) if \( r \) is applicable
Semantics of Multi-Context Systems

- multi-context system
  \[ M = (C_1, \ldots, C_n) \]

- context
  \[ C_i = (L_i, kb_i, br_i) \]

- logic
  \[ L_i = (KB_i, BS_i, ACC_i) \]

- knowledge base of a context \( C_i \)
  \[ kb_i \in KB_i \]

- set of bridge rules \( br_i \) of a context \( C_i \) of form
  \[ s \leftarrow (c_1 : p_1), \ldots, (c_j : p_j), \text{not} (c_{j+1} : p_{j+1}), \ldots, \text{not} (c_m : p_m) \]

- Contexts \( C_1, \ldots, C_n \) are knowledge bases with semantics in terms of accepted belief sets

- \( S = (S_1, \ldots, S_n) \) is a belief state of \( M \) with each \( S_i \in BS_i \)
Semantics of Multi-Context Systems

- **multi-context system**
  \[ M = (C_1, \ldots, C_n) \]

- **context**
  \[ C_i = (L_i, kb_i, br_i) \]

- **logic**
  \[ L_i = (KB_i, BS_i, ACC_i) \]

- **Equilibrium semantics**
  - A belief state \( S = (S_1, \ldots, S_n) \) with \( S_i \in BS_i \)
    makes certain bridge rules applicable,
    add applicable bridge heads to \( kb_i \)
  \[ S \text{ is an equilibrium} \iff \]
  each \( kb_i \) plus acceptable bridge heads from \( br_i \) accepts \( S_i \)
  \[ S_i \in ACC_i(kb_i \cup \{ \text{head}(r) \mid r \in app(br_i, S) \}) \]
The Diamond Example

\[ M = (C_1, C_2, C_3, C_4), \text{ were each } L_i \text{ of } C_i \text{ is an ASP logic} \]

\[
\begin{align*}
C_1 & : kb_1 = \emptyset \\
& \quad \text{br}_1 = \{ a \leftarrow (2 : b), (3 : c) \}
\end{align*}
\]

\[
\begin{align*}
C_2 & : kb_2 = \emptyset \\
& \quad \text{br}_2 = \{ b \leftarrow (4 : g) \}
\end{align*}
\]

\[
\begin{align*}
C_3 & : kb_3 = \{ c \leftarrow d \} \\
& \text{br}_3 = \{ c \lor e \leftarrow \text{not} (4 : f) \}
\end{align*}
\]

\[
\begin{align*}
C_4 & : kb_4 = \{ f \lor g \} \\
& \quad \text{br}_4 = \emptyset
\end{align*}
\]

Equilibria:
- \( (\emptyset, \emptyset, \emptyset, \{f\}) \)
- \( (\emptyset, \{b\}, \{e\}, \{g\}) \)
- \( (\{a\}, \{b\}, \{c, d\}, \{g\}) \)
Towards Distributed Equilibria Building for MCS

Obstacles:

- abstraction of contexts
- information hiding and security aspects
- lack of system topology
- cycles between contexts

We need to capture:

- dependencies between contexts
- representation of partial knowledge
- combination/join of local results
Import Neighborhood & Closure

Import neighborhood of $C_k$

$In(k) = \{c_i \mid (c_i : p_i) \in B(r), r \in br_k\}$
Import Neighborhood & Closure

Import neighborhood of $C_k$

$$In(k) = \{c_i \mid (c_i : p_i) \in B(r), r \in br_k\}$$

Import closure $IC(k)$ of $C_k$ is the smallest set $S$ such that
(i) $k \in S$ and
(ii) for all $i \in S$, $In(i) \subseteq S$. 
Partial Belief States and Equilibria

Let $M = (C_1, \ldots, C_n)$ be an MCS, and let $\epsilon \notin \bigcup_{i=1}^{n} BS_i$.
Partial Belief States and Equilibria

Let $M = (C_1, \ldots, C_n)$ be an MCS, and let $\epsilon \notin \bigcup_{i=1}^{n} BS_i$

A partial belief state of $M$ is a sequence $S = (S_1, \ldots, S_n)$, where $S_i \in BS_i \cup \{\epsilon\}$, for $1 \leq i \leq n$
Partial Belief States and Equilibria

Let $M = (C_1, \ldots, C_n)$ be an MCS, and let $\epsilon \notin \bigcup_{i=1}^{n} \text{BS}_i$

A partial belief state of $M$ is a sequence $S = (S_1, \ldots, S_n)$, where $S_i \in \text{BS}_i \cup \{\epsilon\}$, for $1 \leq i \leq n$

$S = (S_1, \ldots, S_n)$ is a partial equilibrium of $M$ w.r.t. a context $C_k$ iff for $1 \leq i \leq n$,

- if $i \in \text{IC}(k)$ then $S_i \in \text{ACC}_i(kb_i \cup \{\text{head}(r) \mid r \in \text{app}(br_i, S)\})$
- otherwise, $S_i = \epsilon$

Intuitively, partial equilibria w.r.t. a context $C_k$ cover the reachable contexts of $C_k$
Example

Evaluation of an Multi-Context System with the DMCS algorithm
Input: interface variables $V = \{a, b, c, f, g\}$.

\[ \begin{align*}
kb_1 &= \emptyset \\
br_1 &= \{a \leftarrow (2 : b), (3 : c)\} \\
kb_2 &= \emptyset \\
br_2 &= \{b \leftarrow (4 : g)\} \\
kb_3 &= \{c \leftarrow d\} \\
\quad &\quad \{d \leftarrow c\} \\
br_3 &= \{c \lor e \leftarrow not (4 : f)\} \\
kb_4 &= \{f \lor g \leftarrow\} \\
br_4 &= \emptyset
\end{align*} \]
Example

Evaluation of an Multi-Context System with the DMCS algorithm
Input: interface variables $V = \{a, b, c, f, g\}$.
Example

Evaluation of an Multi-Context System with the DMCS algorithm
Input: interface variables \( V = \{a, b, c, f, g\} \).
Example

Evaluation of an Multi-Context System with the DMCS algorithm
Input: interface variables $V = \{a, b, c, f, g\}$.

\[
\begin{align*}
v & \leftarrow & C_1 \\
k_b_1 & = & \emptyset \\
br_1 & = & \{a \leftarrow (2 : b), (3 : c)\} \\
C_2 & \leftarrow & C_1 \\
k_b_2 & = & \emptyset \\
br_2 & = & \{b \leftarrow (4 : g)\} \\
C_3 & \leftarrow & C_2 \\
k_b_3 & = & \{c \leftarrow d\} \\
d & \leftarrow & c \\
br_3 & = & \{c \lor e \leftarrow \text{not}(4 : f)\} \\
C_4 & \leftarrow & C_3 \\
k_b_4 & = & \{f \lor g \leftarrow\} \\
br_4 & = & \emptyset
\end{align*}
\]
Example

Evaluation of an Multi-Context System with the DMCS algorithm
Input: interface variables $V = \{a, b, c, f, g\}$. 

$kb_1 = \emptyset$
$br_1 = \{a \leftarrow (2 : b), (3 : c)\}$

$kb_2 = \emptyset$
$br_2 = \{b \leftarrow (4 : g)\}$

$kb_3 = \{c \leftarrow d\}$
$br_3 = \{c \lor e \leftarrow \text{not}(4 : f)\}$

$kb_4 = \{f \lor g \leftarrow\}$
$br_4 = \emptyset$

$kb_1 = \emptyset$
$br_1 = \{a \leftarrow (2 : b), (3 : c)\}$

$kb_2 = \emptyset$
$br_2 = \{b \leftarrow (4 : g)\}$

$kb_3 = \{c \leftarrow d\}$
$br_3 = \{c \lor e \leftarrow \text{not}(4 : f)\}$

$kb_4 = \{f \lor g \leftarrow\}$
$br_4 = \emptyset$

\begin{align*}
(\epsilon, \{b\}, \{c\}, \{\neg f, g\}) \\
(\epsilon, \{b\}, \{\neg c\}, \{\neg f, g\}) \\
(\epsilon, \{\neg b\}, \{\neg c\}, \{f, \neg g\}) \\
(\epsilon, \{b\}, \epsilon, \{\neg f, g\}) \\
(\epsilon, \{\neg b\}, \epsilon, \{f, \neg g\}) \\
(\epsilon, \epsilon, \{\neg c, \neg d, e\}, \{\neg f, g\}) \\
(\epsilon, \epsilon, \{c, d, \neg e\}, \{\neg f, g\}) \\
(\epsilon, \epsilon, \{\neg c, \neg d, \neg e\}, \{f, \neg g\}) \\
(\epsilon, \epsilon, \{\neg f, g\}) \\
(\epsilon, \epsilon, \epsilon, \{f, \neg g\})
\end{align*}
Example

Evaluation of an Multi-Context System with the DMCS algorithm
Input: interface variables \( V = \{a, b, c, f, g\} \).
Motivation for MCS Decomposition

Scalability issues with the basic evaluation algorithm DMCS
- unaware of global context dependencies, only know (local) import neighborhood
- a context $C_i$ returns a possibly huge set of partial belief states, which are the join of neighbor belief states of $C_i$ plus local belief sets

We address these issues by
- capturing inter-context dependencies (topology)
- providing a decomposition based on biconnected components
- characterizing minimal interface variables in each component
- develop the DMCSOPT algorithm which operates on query plans
A group of 4 scientists.

Problem: How to go home?

Possible solutions:
- Car: slower than train
- Train: should bring some food

Mr. 3 and Ms. 4 have additional information from Mr. 5 and Ms. 6.
A group of 4 scientists.

Problem: How to go home?

Possible solutions:
  - Car
  - Train
A group of 4 scientists.

Problem: How to go home?

Possible solutions:
- Car: slower than train
- Train: should bring some food

Mr. 3 and Ms. 4 have additional information from Mr. 5 and Ms. 6
Ms. 4 just got married to Mr. 5.

Mr. 5 wants his wife to come back as soon as possible.

\[ kb_4 = \{ car_4 \lor train_4 \leftarrow \} \]
\[ br_4 = \{ train_4 \leftarrow (5 : want \_sooner_5) \} \]

\[ kb_5 = \{ want \_sooner_5 \leftarrow soon_5 \} \]
\[ br_5 = \{ soon_5 \leftarrow (4 : train_4) \} \]
Scientist Group Example (ctd.)

- Mr. 3 has a daughter, Ms. 6.
- Mr. 3 is responsible for buying provisions, if they go by train.
- If Ms. 6 is sick, then Mr. 3 must attend to her as fast as possible.

\[
kb_3 = \begin{cases} 
  \text{car}_3 \lor \text{train}_3 \leftarrow \\
  \text{train}_3 \leftarrow \text{urgent}_3 \\
  \text{sandwiches}_3 \lor \text{chocolate}_3 \lor \text{peanuts}_3 \leftarrow \text{train}_3 \\
  \text{coke}_3 \lor \text{juice}_3 \leftarrow \text{train}_3 
\end{cases}
\]

\[
br_3 = \begin{cases} 
  \text{urgent}_3 \leftarrow (6 : \text{sick}_6) \\
  \text{train}_3 \leftarrow (4 : \text{train}_4)
\end{cases};
\]

\[
kb_6 = \{ \text{sick}_6 \lor \text{fit}_6 \leftarrow \}
\]

\[
br_6 = \emptyset.
\]
Scientist Group Example (ctd.)

- Ms. 1 is leader of group.
- Ms. 1 is allergic to peanuts.
- Mr. 2 wants to get home somehow and doesn’t want coke.

\[ kb_1 = \{ \text{car}_1 \leftarrow \text{not train}_1 \} \]
\[ br_1 = \{ \text{train}_1 \leftarrow (2 : \text{train}_2), (3 : \text{train}_3) \}
\{ \text{peanuts}_1 \leftarrow (3 : \text{chocolate-peanuts}_3) \} \]

\[ kb_2 = \{ \text{not car}_2, \text{not train}_2 \} \text{ and } \]
\[ br_2 = \{ \text{car}_2 \leftarrow (3 : \text{car}_3), (4 : \text{car}_4) \}
\{ \text{train}_2 \leftarrow (3 : \text{train}_3), (4 : \text{train}_4), \not(3 : \text{coke}_3) \} \]
Scientist Group Example (ctd.)

- Ms. 1 is leader of group.
- Ms. 1 is allergic to peanuts.
- Mr. 2 wants to get home somehow and doesn’t want coke.

\[
kb_1 = \{ \text{car}_1 \leftarrow \text{not train}_1, \text{peanuts}_1 \newline \downarrow \leftarrow \text{peanuts}_1 \} \\
br_1 = \{ \text{train}_1 \leftarrow (2 : \text{train}_2), (3 : \text{train}_3) \newline \text{peanuts}_1 \leftarrow (3 : \text{chocolate.peanuts}_3) \}\]

\[
kb_2 = \{ \downarrow \leftarrow \text{not car}_2, \text{not train}_2 \} \text{ and } \\
br_2 = \{ \text{car}_2 \leftarrow (3 : \text{car}_3), (4 : \text{car}_4) \newline \text{train}_2 \leftarrow (3 : \text{train}_3), (4 : \text{train}_4), \newline \text{not} (3 : \text{coke}_3) \} \\
\]

One equilibrium is \( S = (\{ \text{train}_1 \}, \{ \text{train}_2 \}, \{ \text{train}_3, \text{urgent}_3, \text{juice}_3, \text{sandwiches}_3 \}, \{ \text{train}_4 \}, \{ \text{soon}_5, \text{want}_soon\text{er}_5 \}, \{ \text{sick}_6 \}) \)
Ms. 1 decides after gathering information.
Ms. 1 decides after gathering information.
Mr. 3 and Ms. 4 do not want to bother the others.
A graph is weakly connected if replacing every directed edge by an undirected edge yields a connected graph.

A vertex $c$ of a weakly connected graph $G$ is a cut vertex, if $G \setminus c$ is disconnected (3 and 4 are cut vertices)
Based on cut vertices, we can decompose the MCS into a block tree: provides a “high-level” view of the dependencies (edge partitioning)
Scientist Group Example (ctd.)

- Based on cut vertices, we can decompose the MCS into a **block tree**: provides a “high-level” view of the dependencies (edge partitioning)
- The block tree of our example is:

- $B_1$ induced by $\{1, 2, 3, 4\}$
- $B_2$ induced by $\{4, 5\}$
- $B_3$ induced by $\{3, 6\}$
Optimization: Creating Acyclic Topologies

cycle breaking by creating a spanning tree of a cyclic MCS
Optimization: Creating Acyclic Topologies

cycle breaking by creating a spanning tree of a cyclic MCS

![Graph example]
Optimization: Creating Acyclic Topologies

cycle breaking by creating a spanning tree of a cyclic MCS

ear decomposition $P = \langle P_0, \rangle$
Optimization: Creating Acyclic Topologies

cycle breaking by creating a spanning tree of a cyclic MCS

ear decomposition \( P = \langle P_0, P_1, \rangle \)
Optimization: Creating Acyclic Topologies

cycle breaking by creating a spanning tree of a cyclic MCS

\[ P = \langle P_0, P_1, P_2, \ldots \rangle \]

ear decomposition
Optimization: Creating Acyclic Topologies

cycle breaking by creating a spanning tree of a cyclic MCS

ear decomposition $P = \langle P_0, P_1, P_2, P_3, \rangle$
Optimization: Creating Acyclic Topologies

cycle breaking by creating a spanning tree of a cyclic MCS

\[ P = \langle P_0, P_1, P_2, P_3, P_4 \rangle \]

*ear decomposition*
Optimization: Creating Acyclic Topologies

cycle breaking by creating a spanning tree of a cyclic MCS

\[ P = \langle P_0, P_1, P_2, P_3, P_4 \rangle \]

cycle breaker edges \( cb(G, P) \): remove last edge from each path \( P_i \) in \( G \)

Intuition: for a removed edge \((\ell, t)\), guess at leaf \( C_\ell \) the variables at \( C_t \)
**Optimization: Avoiding Unnecessary Calls**

*transitive reduction* of a digraph $G$ is the graph $G^-$ with the smallest set of edges whose transitive closure $G^+$ equals the one of $G$.
transitive reduction of a digraph $G$ is the graph $G^{-}$ with the smallest set of edges whose transitive closure $G^{+}$ equals the one of $G$.

Intuitively, the transitive reduction of an acyclic graph is unique, and one can evaluate the contexts using a topological sort of the contexts.
Scientist Group Example (ctd.)

- $B_1$: acyclic $\rightarrow$ apply transitive reduction
- $B_2$: cyclic $\rightarrow$ apply ear decomposition, remove last edge from each ear, then apply transitive reduction (already reduced)
- $B_3$: acyclic and already reduced
In a pruned block $B'$, take all variables from

- the minimal interface in $B'$
- child cut vertices $c$
- removed edges $E$

Outcome: query plan for the MCS to restrict calls and partial belief states
$S = ( \{ \text{train}_1 \}, \{ \text{train}_2 \}, \{ \text{train}_3, \text{urgent}_3, \text{juice}_3, \text{sandwiches}_3 \}, \{ \text{train}_4 \}, \{ \text{soon}_5, \text{want \_ sooner}_5 \}, \{ \text{sick}_6 \} )$

$T = ( \{ \text{train}_1 \}, \{ \text{train}_2 \}, \{ \text{train}_3, \text{juice}_3, \text{sandwiches}_3 \}, \{ \text{train}_4 \}, \{ \text{soon}_5, \text{want \_ sooner}_5 \}, \{ \text{fit}_6 \} )$

$U = ( \{ \text{car}_1 \}, \{ \text{car}_2 \}, \{ \text{car}_3 \}, \{ \text{car}_4 \}, \emptyset, \{ \text{fit}_6 \} )$
Scientist Group Example (ctd.)

\[
S_{B_1} = (\{\text{train}_1\}, \{\text{train}_2\}, \{\text{train}_3, \text{juice}_3, \text{sandwiches}_3\}, \{\text{train}_4\}, \emptyset, \emptyset) \\
U_{B_1} = (\{\text{car}_1\}, \{\text{car}_2\}, \{\text{car}_3\}, \{\text{car}_4\}, \emptyset, \emptyset)
\]
### Experiments

<table>
<thead>
<tr>
<th></th>
<th>$A_\phi$</th>
<th>$A_\bowtie$</th>
<th>$A_{\leftrightarrow}$</th>
<th>$A_\Sigma (\sigma)$</th>
<th>$# (\sigma)$</th>
<th>$B_\phi$</th>
<th>$B_\bowtie$</th>
<th>$B_{\leftrightarrow}$</th>
<th>$B_\Sigma (\sigma)$</th>
<th>$# (\sigma)$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>D</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>0.9</td>
<td>0.0</td>
<td>0.0</td>
<td>1.0 (0.2)</td>
<td>28 (17.6)</td>
<td>0.8</td>
<td>8.4</td>
<td>0.0</td>
<td>9.4 (5.5)</td>
<td>3136 (3155.8)</td>
</tr>
<tr>
<td>25</td>
<td>11.2</td>
<td>0.5</td>
<td>0.0</td>
<td>12.8 (1.3)</td>
<td>17 (18.9)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>51.1</td>
<td>3.7</td>
<td>0.0</td>
<td>59.5 (8.9)</td>
<td>58 (49.7)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>R</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>0.1 (0.0)</td>
<td>3.5 (3.4)</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>0.2 (0.1)</td>
<td>300 (694.5)</td>
</tr>
<tr>
<td>13</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>0.2 (0.1)</td>
<td>6 (1.2)</td>
<td>0.1</td>
<td>1.5</td>
<td>1.9</td>
<td>3.9 (5.3)</td>
<td>5064 (21523.8)</td>
</tr>
<tr>
<td>301</td>
<td>4.1</td>
<td>0.1</td>
<td>2.1</td>
<td>10.2 (2.2)</td>
<td>8 (4.9)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Z</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>0.6</td>
<td>0.1</td>
<td>0.0</td>
<td>0.7 (0.2)</td>
<td>34 (41.8)</td>
<td>5.5</td>
<td>4.2</td>
<td>0.0</td>
<td>11.5 (4.0)</td>
<td>3024 (1286.8)</td>
</tr>
<tr>
<td>151</td>
<td>8.9</td>
<td>22.3</td>
<td>0.4</td>
<td>32.2 (7.3)</td>
<td>33 (28.5)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>301</td>
<td>21.6</td>
<td>99.5</td>
<td>1.7</td>
<td>124.3 (20.6)</td>
<td>22 (41.4)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>H</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>0.2</td>
<td>0.0</td>
<td>0.0</td>
<td>0.2 (0.0)</td>
<td>28 (44.4)</td>
<td>1.1</td>
<td>0.9</td>
<td>0.0</td>
<td>2.0 (1.3)</td>
<td>684 (1308.0)</td>
</tr>
<tr>
<td>101</td>
<td>1.8</td>
<td>0.3</td>
<td>0.3</td>
<td>3.8 (1.0)</td>
<td>48 (76.6)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>301</td>
<td>7.8</td>
<td>2.0</td>
<td>2.4</td>
<td>25.1 (8.7)</td>
<td>38 (34.2)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table:** Runtime for DMCSOPT ($A_x$) and DMCS ($B_x$), timeout 180 secs (—)

Random instances with $n$ contexts and topologies:

- **Diamond**
- **Ring**
- **Zig-zag**
- **House**

**Timings:**

- clasp ($\phi$), Belief state combination ($\bowtie$) and transfer ($\leftrightarrow$); No. of partial equilibria: $\#$
Experiments

Parameter $Pi=(n,s,b,r)$  # equilibria

Diamond

DMCSOPT

DMCS

#28

#3136

#17

#58
Experiments

Parameter $\Pi = (n, s, b, r)$  # equilibria

Ring

<table>
<thead>
<tr>
<th>Parameter</th>
<th>$n$</th>
<th>$s$</th>
<th>$b$</th>
<th>$r$</th>
<th># equilibria</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10</td>
<td>10</td>
<td>5</td>
<td>5</td>
<td>#300</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>10</td>
<td>5</td>
<td>5</td>
<td>#6</td>
</tr>
<tr>
<td></td>
<td>301</td>
<td>10</td>
<td>5</td>
<td>5</td>
<td>#8</td>
</tr>
</tbody>
</table>

- DMCSOPT
- DMCS

Evaluation time / secs (logscale)
Experiments

Parameter $Pi=(n,s,b,r)$    # equilibria

Zigzag

DMCSOPT
DMCS

Evaluation time / secs (logscale)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>13-10-5-5</th>
<th>151-10-5-5</th>
<th>301-10-5-5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>1</td>
<td>10</td>
<td>100</td>
</tr>
<tr>
<td>1</td>
<td>13-10-5-5</td>
<td>151-10-5-5</td>
<td>301-10-5-5</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1000</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Bar chart showing evaluation time for different parameters.
Experiments

Parameter $Pi=(n,s,b,r)$    # equilibria

House

<table>
<thead>
<tr>
<th>Parameter</th>
<th># equilibria</th>
</tr>
</thead>
<tbody>
<tr>
<td>9-10-5-5</td>
<td>#28</td>
</tr>
<tr>
<td>13-10-5-5</td>
<td>#77</td>
</tr>
<tr>
<td>41-10-5-5</td>
<td>#26</td>
</tr>
<tr>
<td>101-10-5-5</td>
<td>#48</td>
</tr>
<tr>
<td>301-10-5-5</td>
<td>#38</td>
</tr>
</tbody>
</table>

DMCSOPT (green) and DMCS (red) evaluation times (log scale).
Experiments

Parameter $\Pi=(n,s,b,r)$ # equilibria

Binary Tree

Evaluation time / secs (logscale)

DMCSOPT

DMCS

#8 #9.5 #7 #4
Conclusions

- MCS is a general framework for integrating diverse formalisms
- First attempt for distributed MCS evaluation
- Initial experiments with a prototype implementation
- Decomposition technique is encouraging: binary tree with $n = 600$ evaluated in 176secs ($\# = 4$)
Conclusions

- MCS is a general framework for integrating diverse formalisms
- First attempt for distributed MCS evaluation
- Initial experiments with a prototype implementation
- Decomposition technique is encouraging: binary tree with $n = 600$ evaluated in 176secs ($\# = 4$)

Future work:

- improve scalability
  - approximation semantics
  - syntactic restrictions
  - specialized algorithms for some types of topologies
- dynamic multi-context systems
Gerhard Brewka and Thomas Eiter.  
Equilibria in heterogeneous nonmonotonic multi-context systems.  

Fausto Giunchiglia and Luciano Serafini.  
Multilanguage hierarchical logics or: How we can do without modal logics.  