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

Knowledge-Based Systems Group

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 $L = (\mathbf{KB}_L, \mathbf{BS}_L, \mathbf{ACC}_L)$
  - $\mathbf{KB}_L$: set of well-formed knowledge bases
  - $\mathbf{BS}_L$: is the set of possible belief sets
  - $\mathbf{ACC}_L$: acceptability function $\mathbf{KB}_L \mapsto 2^{\mathbf{BS}_L}$
    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*

$$L_i = (\mathbf{KB}_i, \mathbf{BS}_i, \mathbf{ACC}_i)$$

- Bridge rule $r \in br_i$ of a context C_i

$$\begin{aligned} s \leftarrow {} & (c_1 : p_1), \ldots, (c_j : p_j), \\ & not\,(c_{j+1} : p_{j+1}), \ldots, not\,(c_m : p_m) \end{aligned}$$

  - $(c_k : p_k)$ looks at belief p_k in context C_{c_k}
  - r is applicable :⇔ 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*
$$L_i = (\mathbf{KB}_i, \mathbf{BS}_i, \mathbf{ACC}_i)$$
- *knowledge base* of a context C_i
$$kb_i \in \mathbf{KB}_i$$
- *set of bridge rules* br_i of a context C_i of form
$$s \leftarrow (c_1 : p_1), \ldots, (c_j : p_j), not\,(c_{j+1} : p_{j+1}), \ldots, not\,(c_m : p_m)$$
- 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 $S_i \in \mathbf{BS}_i$

Semantics of Multi-Context Systems

- *multi-context system*

$$M = (C_1, \ldots, C_n)$$

- *context*

$$C_i = (L_i, kb_i, br_i)$$

- *logic*

$$L_i = (\mathbf{KB}_i, \mathbf{BS}_i, \mathbf{ACC}_i)$$

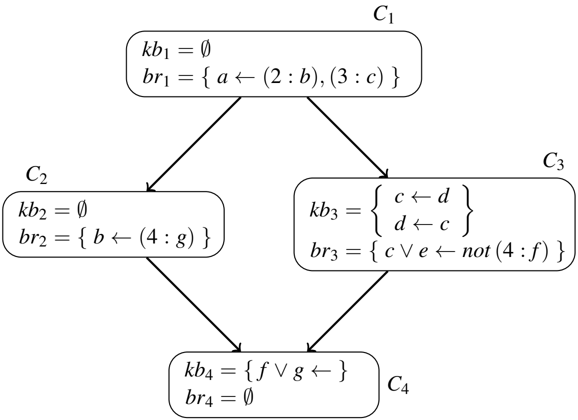
- Equilibrium semantics
  - A belief state $S = (S_1, \ldots, S_n)$ with $S_i \in \mathbf{BS}_i$
    ... makes certain bridge rules applicable,
    ... add applicable bridge heads to kb_i

$\Rightarrow$ S is an equilibrium :$\Leftrightarrow$
each kb_i plus acceptable bridge heads from br_i accepts S_i

$$S_i \in \mathbf{ACC}_i(kb_i \cup \{head(r) \mid r \in app(br_i, S)\})$$

The Diamond Example

$M = (C_1, C_2, C_3, C_4)$, were each L_i of C_i is an ASP logic



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

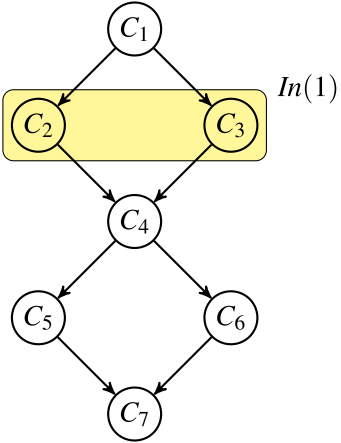
We need to capture:

- dependencies between contexts
- representation of partial knowledge
- combination/join of local results

Import Neighborhood & Closure

Import neighborhood of C_k

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

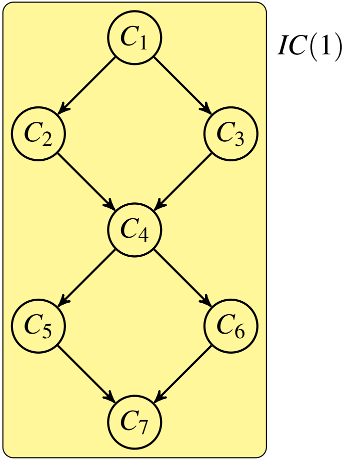


Import Neighborhood & Closure

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 $\epsilon \notin \bigcup_{i=1}^{n} \mathbf{BS}_i$

Partial Belief States and Equilibria

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

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

Partial Belief States and Equilibria

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

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

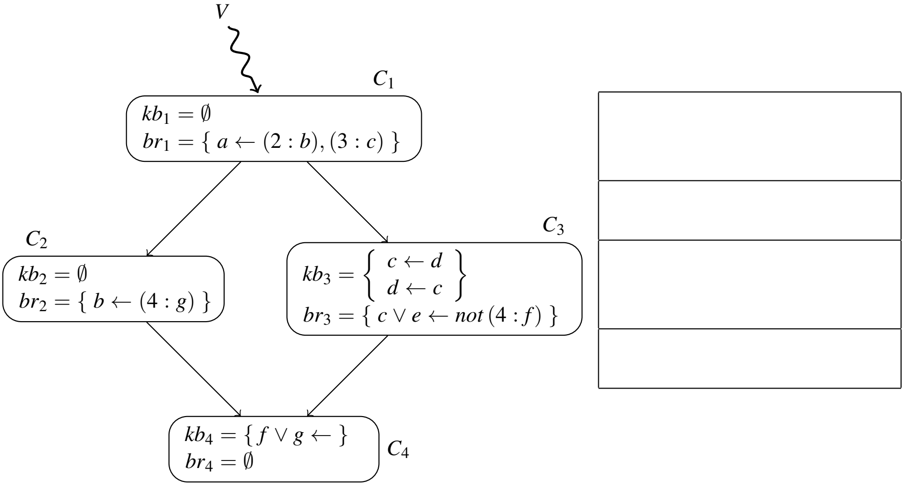
$S = (S_1, \ldots, S_n)$ is a partial equilibrium of M w.r.t. a context C_k iff for $1 \leq i \leq n$,

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

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

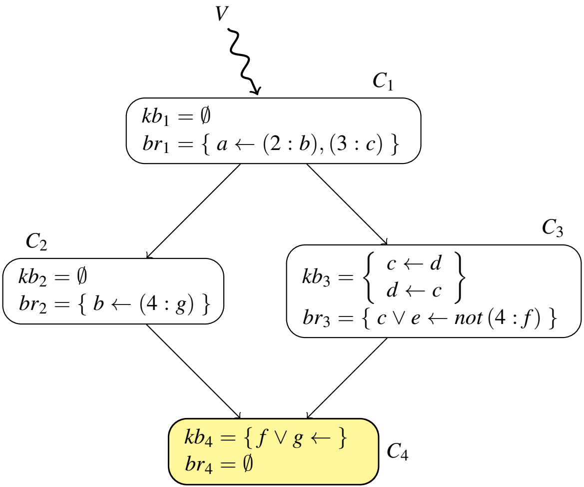
Example

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



Example

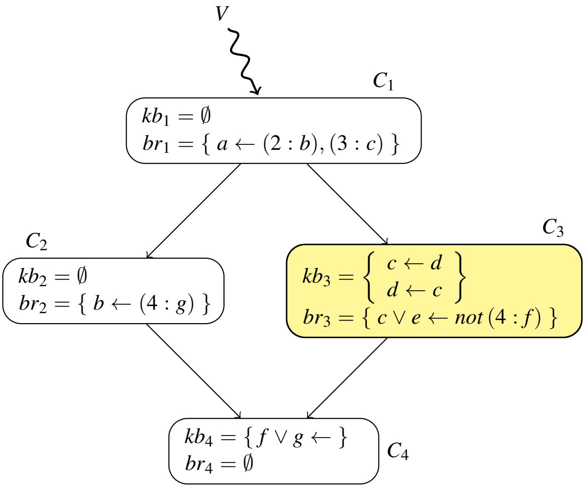
Evaluation of an Multi-Context System with the DMCS algorithm
Input: interface variables $V = \{a, b, c, f, g\}$.



$(\epsilon, \epsilon, \epsilon, \{\neg f, g\})$ $(\epsilon, \epsilon, \epsilon, \{f, \neg g\})$

Example

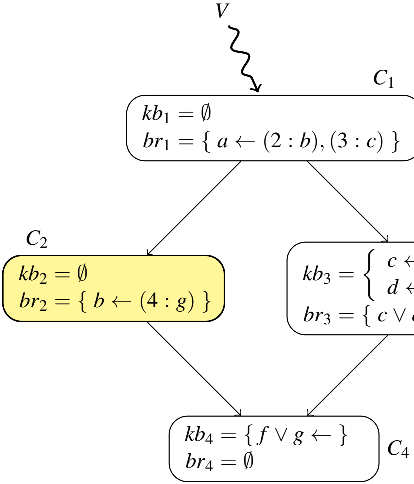
Evaluation of an Multi-Context System with the DMCS algorithm
Input: interface variables $V = \{a, b, c, f, g\}$.



$(\epsilon, \epsilon, \{\neg c, \neg d, e\}, \{\neg f, g\})$ $(\epsilon, \epsilon, \{c, d, \neg e\}, \{\neg f, g\})$ $(\epsilon, \epsilon, \{\neg c, \neg d, \neg e\}, \{f, \neg g\})$
$(\epsilon, \epsilon, \epsilon, \{\neg f, g\})$ $(\epsilon, \epsilon, \epsilon, \{f, \neg g\})$

Example

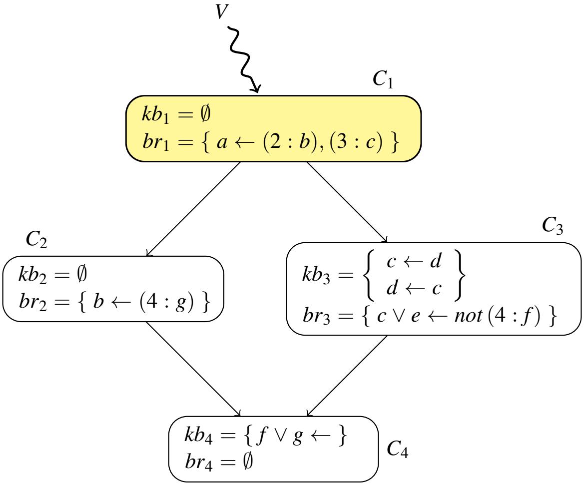
Evaluation of an Multi-Context System with the DMCS algorithm
Input: interface variables $V = \{a, b, c, f, g\}$.



$(\epsilon, \{b\}, \epsilon, \{\neg f, g\})$ $(\epsilon, \{\neg b\}, \epsilon, \{f, \neg g\})$
$(\epsilon, \epsilon, \{\neg c, \neg d, e\}, \{\neg f, g\})$ $(\epsilon, \epsilon, \{c, d, \neg e\}, \{\neg f, g\})$ $(\epsilon, \epsilon, \{\neg c, \neg d, \neg e\}, \{f, \neg g\})$
$(\epsilon, \epsilon, \epsilon, \{\neg f, g\})$ $(\epsilon, \epsilon, \epsilon, \{f, \neg g\})$

Example

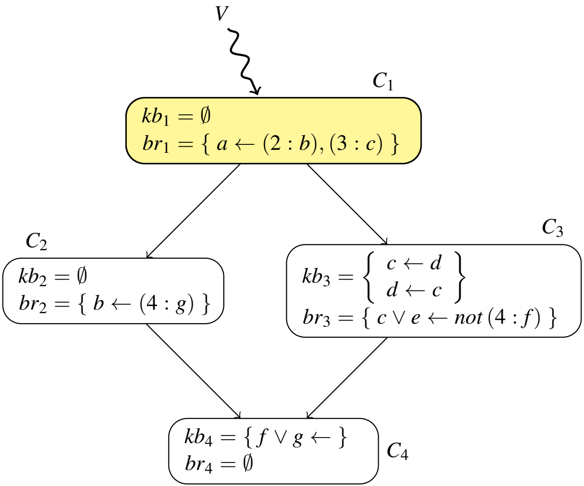
Evaluation of an Multi-Context System with the DMCS algorithm
Input: interface variables $V = \{a, b, c, f, g\}$.



$(\epsilon, \{b\}, \{c\}, \{\neg f, g\})$ $(\epsilon, \{b\}, \{\neg c\}, \{\neg f, g\})$ $(\epsilon, \{\neg b\}, \{\neg c\}, \{f, \neg g\})$
$(\epsilon, \{b\}, \epsilon, \{\neg f, g\})$ $(\epsilon, \{\neg b\}, \epsilon, \{f, \neg g\})$
$(\epsilon, \epsilon, \{\neg c, \neg d, e\}, \{\neg f, g\})$ $(\epsilon, \epsilon, \{c, d, \neg e\}, \{\neg f, g\})$ $(\epsilon, \epsilon, \{\neg c, \neg d, \neg e\}, \{f, \neg g\})$
$(\epsilon, \epsilon, \epsilon, \{\neg f, g\})$ $(\epsilon, \epsilon, \epsilon, \{f, \neg g\})$

Example

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



$(\{a\}, \{b\}, \{c\}, \{\neg f, g\})$ $(\{\neg a\}, \{b\}, \{\neg c\}, \{\neg f, g\})$ $(\{\neg a\}, \{\neg b\}, \{\neg c\}, \{f, \neg g\})$
$(\epsilon, \{b\}, \epsilon, \{\neg f, g\})$ $(\epsilon, \{\neg b\}, \epsilon, \{f, \neg g\})$
$(\epsilon, \epsilon, \{\neg c, \neg d, e\}, \{\neg f, g\})$ $(\epsilon, \epsilon, \{c, d, \neg e\}, \{\neg f, g\})$ $(\epsilon, \epsilon, \{\neg c, \neg d, \neg e\}, \{f, \neg g\})$
$(\epsilon, \epsilon, \epsilon, \{\neg f, g\})$ $(\epsilon, \epsilon, \epsilon, \{f, \neg g\})$

Motivation for MCS Decomposition

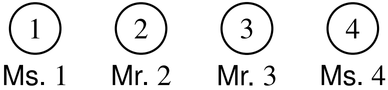
Scalability issues with the basic evaluation algorithm DMCS

- 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

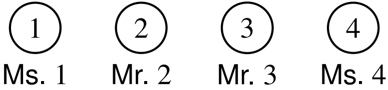
- capturing inter-context dependencies (topology)
- providing a decomposition based on biconnected components
- characterizing minimal interface variables in each component
- develop the DMCSOPT algorithm which operates on query plans

Scientist Group Example



- A group of 4 scientists.
- Problem: How to go home?

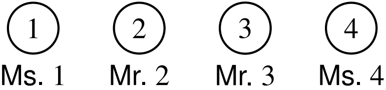
Scientist Group Example



- A group of 4 scientists.
- Problem: How to go home?

- Possible solutions:
  - Car
  - Train

Scientist Group Example



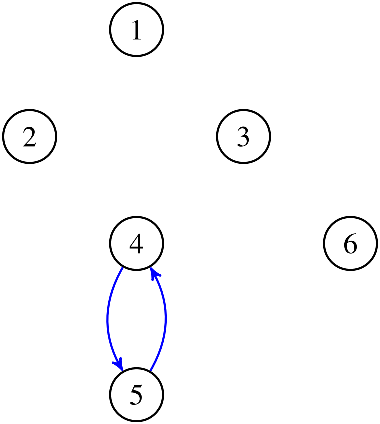
- A group of 4 scientists.
- Problem: How to go home?
- Possible solutions:
  - Car: slower than train
  - Train: should bring some food
- Mr. 3 and Ms. 4 have additional information from Mr. 5 and Ms. 6

Scientist Group Example (ctd.)

- Ms. 4 just got married to Mr. 5.
- Mr. 5 wants his wife to come back as soon as possible.

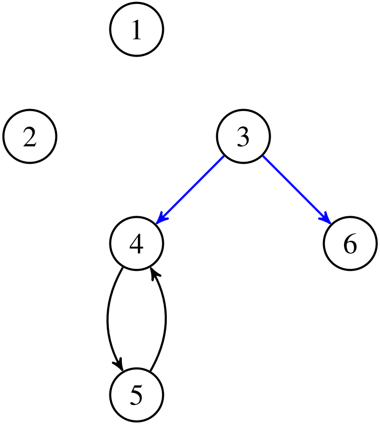
$$kb_4 = \{car_4 \vee train_4 \leftarrow \}$$
$$br_4 = \{train_4 \leftarrow (5 : want\_sooner_5)\}$$

$$kb_5 = \{want\_sooner_5 \leftarrow soon_5\}$$
$$br_5 = \{soon_5 \leftarrow (4 : train_4)\}$$



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.



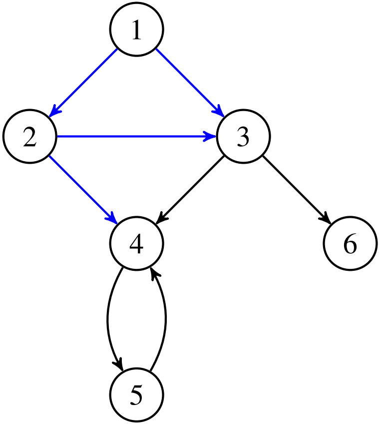
$$kb_3 = \left\{ \begin{array}{r} car_3 \vee train_3 \leftarrow \\ train_3 \leftarrow urgent_3 \\ sandwiches_3 \vee chocolate\_peanuts_3 \leftarrow train_3 \\ coke_3 \vee juice_3 \leftarrow train_3 \end{array} \right\}$$

$$br_3 = \left\{ \begin{array}{r} urgent_3 \leftarrow (6 : sick_6) \\ train_3 \leftarrow (4 : train_4) \end{array} \right\};$$

$$kb_6 = \{ sick_6 \vee fit_6 \leftarrow \}$$
$$br_6 = \emptyset.$$

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.



$$kb_1 = \left\{ \begin{array}{c} car_1 \leftarrow not\ train_1 \\ \bot \leftarrow peanuts_1 \end{array} \right\}$$

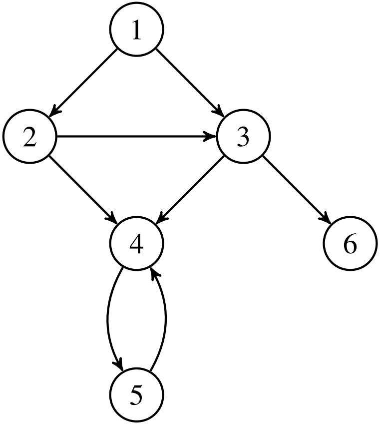
$$br_1 = \left\{ \begin{array}{c} train_1 \leftarrow (2 : train_2), (3 : train_3) \\ peanuts_1 \leftarrow (3 : chocolate\_peanuts_3) \end{array} \right\}$$

$kb_2 = \{\bot \leftarrow not\, car_2, not\, train_2\}$ and

$$br_2 = \left\{ \begin{array}{c} car_2 \leftarrow (3 : car_3), (4 : car_4) \\ train_2 \leftarrow (3 : train_3), (4 : train_4), \\ not\ (3 : coke_3) \end{array} \right\}$$

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.



$$kb_1 = \left\{ \begin{array}{c} car_1 \leftarrow not\ train_1 \\ \bot \leftarrow peanuts_1 \end{array} \right\}$$

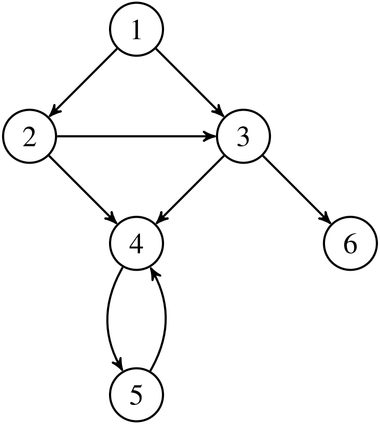
$$br_1 = \left\{ \begin{array}{c} train_1 \leftarrow (2 : train_2), (3 : train_3) \\ peanuts_1 \leftarrow (3 : chocolate\_peanuts_3) \end{array} \right\}$$

$kb_2 = \{\bot \leftarrow not\ car_2, not\ train_2\}$ and

$$br_2 = \left\{ \begin{array}{c} car_2 \leftarrow (3 : car_3), (4 : car_4) \\ train_2 \leftarrow (3 : train_3), (4 : train_4), \\ not\ (3 : coke_3) \end{array} \right\}$$

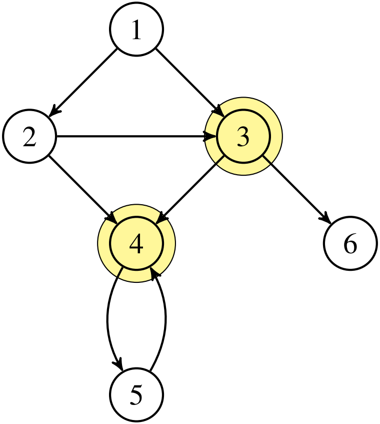
One equilibrium is $S = (\{train_1\}, \{train_2\}, \{train_3, urgent_3, juice_3, sandwiches_3\}, \{train_4\}, \{soon_5, want\_sooner_5\}, \{sick_6\})$

Scientist Group Example (ctd.)



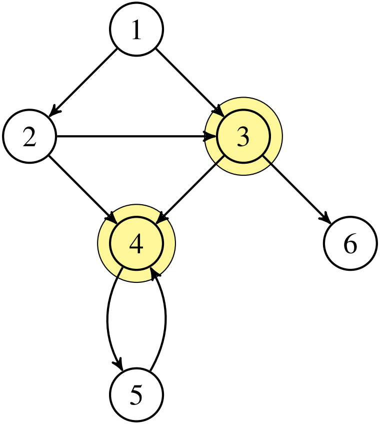
- Ms. 1 decides after gathering information.

Scientist Group Example (ctd.)



- Ms. 1 decides after gathering information.
- Mr. 3 and Ms. 4 do not want to bother the others.

Scientist Group Example (ctd.)



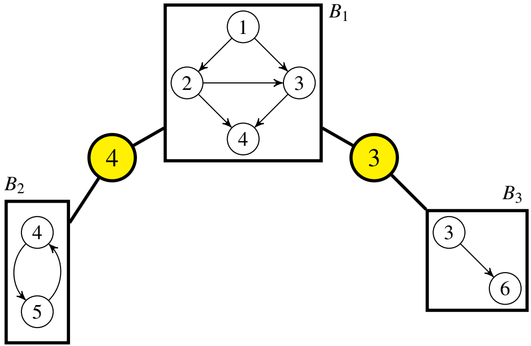
- A graph is weakly connected if replacing every directed edge by an undirected edge yields a connected graph.
- A vertex c of a weakly connected graph G is a *cut vertex*, if $G \backslash c$ is disconnected (3 and 4 are cut vertices)

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)

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:



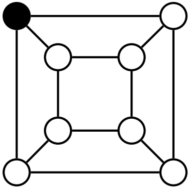
- B_1 induced by $\{1, 2, 3, 4\}$
- B_2 induced by $\{4, 5\}$
- B_3 induced by $\{3, 6\}$

Optimization: Creating Acyclic Topologies

cycle breaking by creating a spanning tree of a cyclic MCS

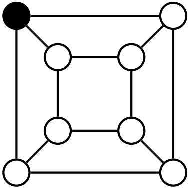
Optimization: Creating Acyclic Topologies

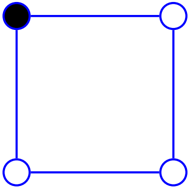
cycle breaking by creating a spanning tree of a cyclic MCS



Optimization: Creating Acyclic Topologies

cycle breaking by creating a spanning tree of a cyclic MCS

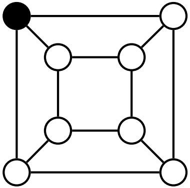


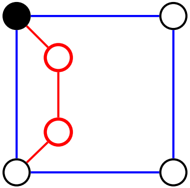


ear decomposition $P = \langle\ P_0,\qquad\qquad \rangle$

Optimization: Creating Acyclic Topologies

cycle breaking by creating a spanning tree of a cyclic MCS

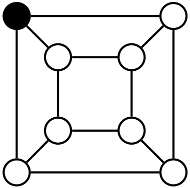


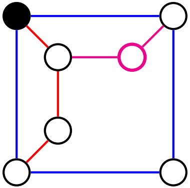


ear decomposition $P = \langle\ P_0, P_1, \qquad \rangle$

Optimization: Creating Acyclic Topologies

cycle breaking by creating a spanning tree of a cyclic MCS

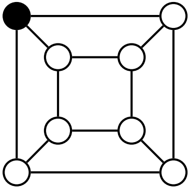


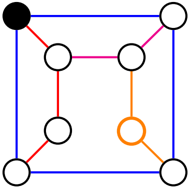


ear decomposition $P = \langle\, P_0, P_1, P_2, \qquad \rangle$

Optimization: Creating Acyclic Topologies

cycle breaking by creating a spanning tree of a cyclic MCS

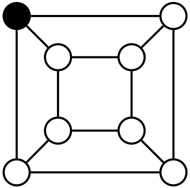


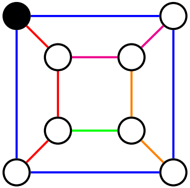


ear decomposition $P = \langle\, P_0, P_1, P_2, P_3, \qquad \rangle$

Optimization: Creating Acyclic Topologies

cycle breaking by creating a spanning tree of a cyclic MCS

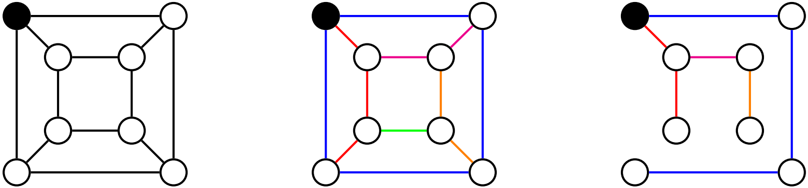




ear decomposition $P = \langle\, P_0, P_1, P_2, P_3, P_4 \,\rangle$

Optimization: Creating Acyclic Topologies

cycle breaking by creating a spanning tree of a cyclic MCS



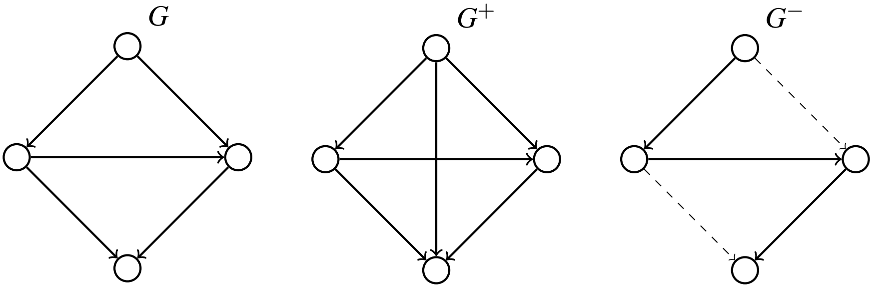
ear decomposition $P = \langle P_0, P_1, P_2, P_3, P_4 \rangle$

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

Intuition: for a removed edge (ℓ, t), guess at leaf C_ℓ the variables at C_t

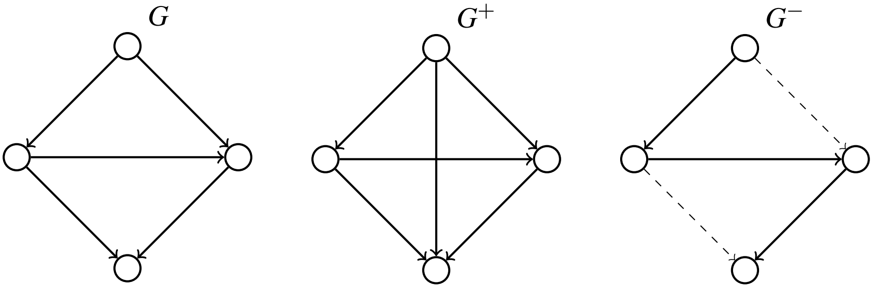
Optimization: Avoiding Unnecessary Calls

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



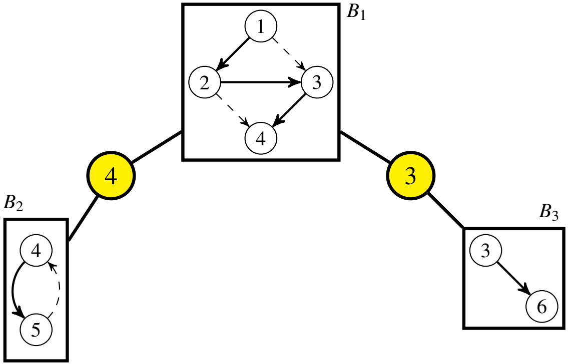
Optimization: Avoiding Unnecessary Calls

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



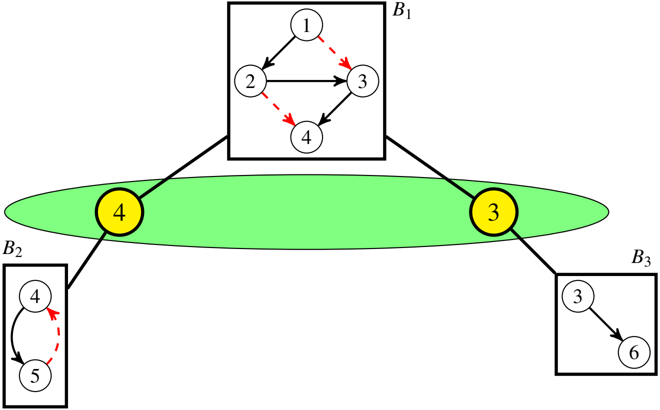
Intuitively, the transitive reduction of an acyclic graph is unique, and one can evaluate the contexts using a topological sort of the contexts.

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

Optimization: Minimal Interface

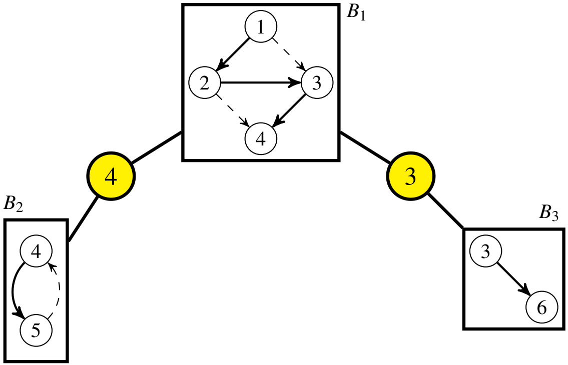


In a pruned block B', take all variables from

- 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

Scientist Group Example (ctd.)

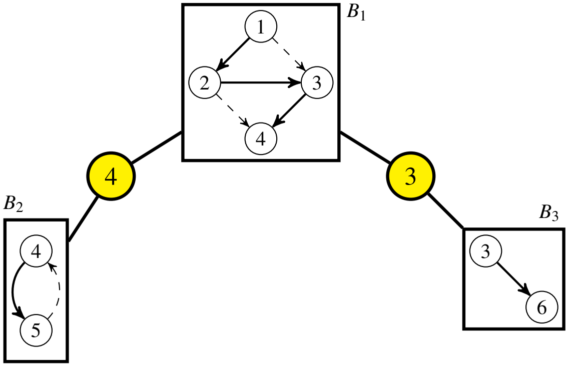


$$S = (\ \{train_1\}, \{train_2\}, \{train_3, urgent_3, juice_3, sandwiches_3\}, \{train_4\}, \{soon_5, want\_sooner_5\}, \{sick_6\})$$

$$T = (\ \{train_1\}, \{train_2\}, \{train_3, juice_3, sandwiches_3\}, \{train_4\}, \{soon_5, want\_sooner_5\}, \{fit_6\})$$

$$U = (\ \{car_1\}, \{car_2\}, \{car_3\}, \{car_4\}, \emptyset, \{fit_6\})$$

Scientist Group Example (ctd.)



$$\text{“}S|_{B_1}\text{”} = (\{train_1\}, \{train_2\}, \{train_3, juice_3, sandwiches_3\}, \{train_4\}, \emptyset, \emptyset)$$

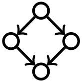
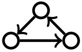
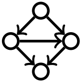
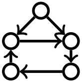
$$\text{“}U|_{B_1}\text{”} = (\{car_1\}, \{car_2\}, \{car_3\}, \{car_4\}, \emptyset, \emptyset)$$

Experiments

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

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

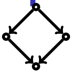
Random instances with n contexts and topologies:

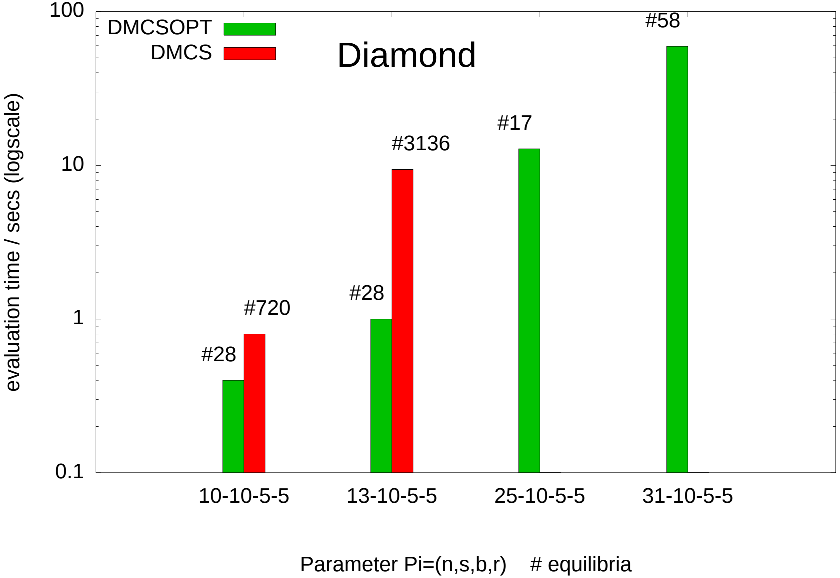
*D*iamond , *R*ing , *Z*ig-zag , *H*ouse 

Timings:

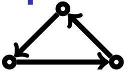
clasp (ϕ), Belief state combination ($\bowtie$) and transfer ($\leftrightarrow$); No. of partial equilibria: #

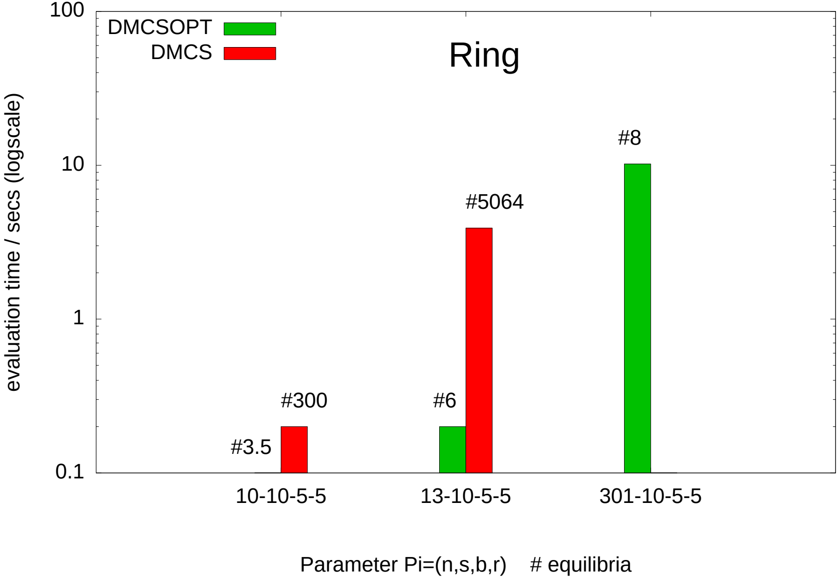
Experiments



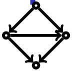


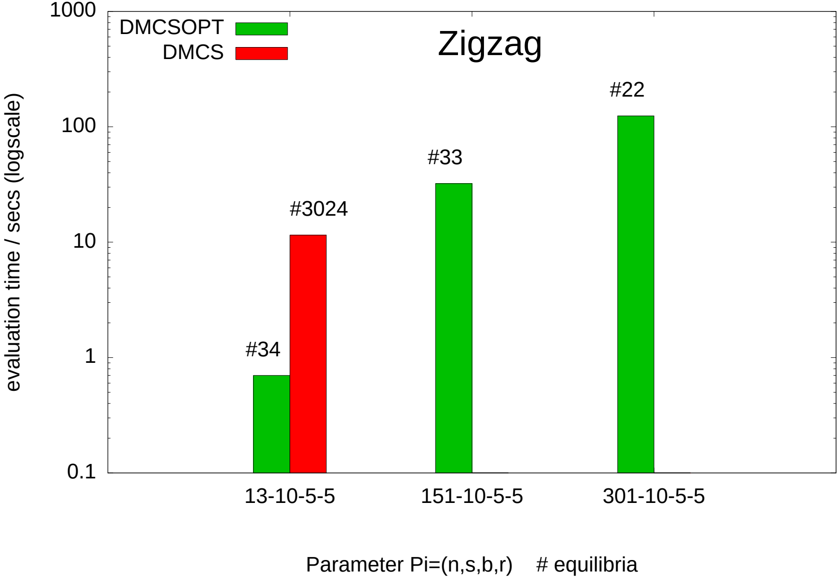
Experiments



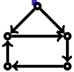


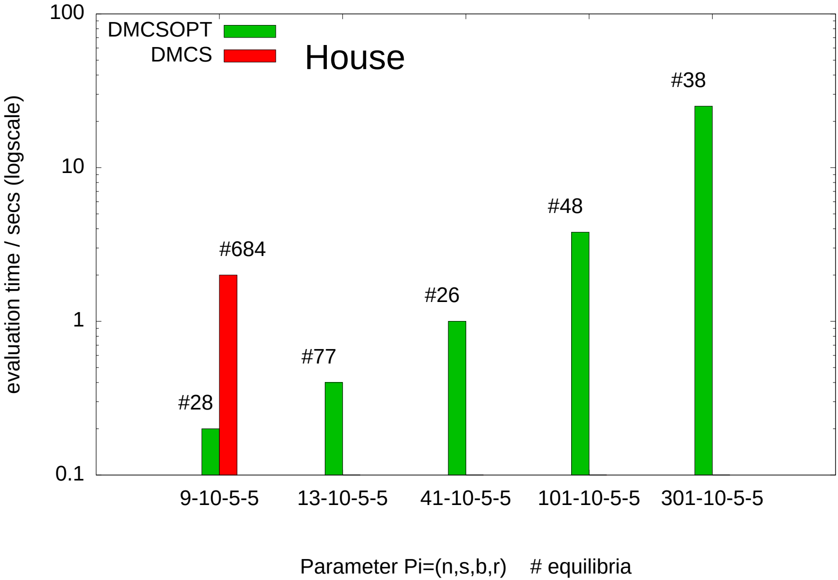
Experiments



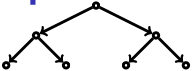


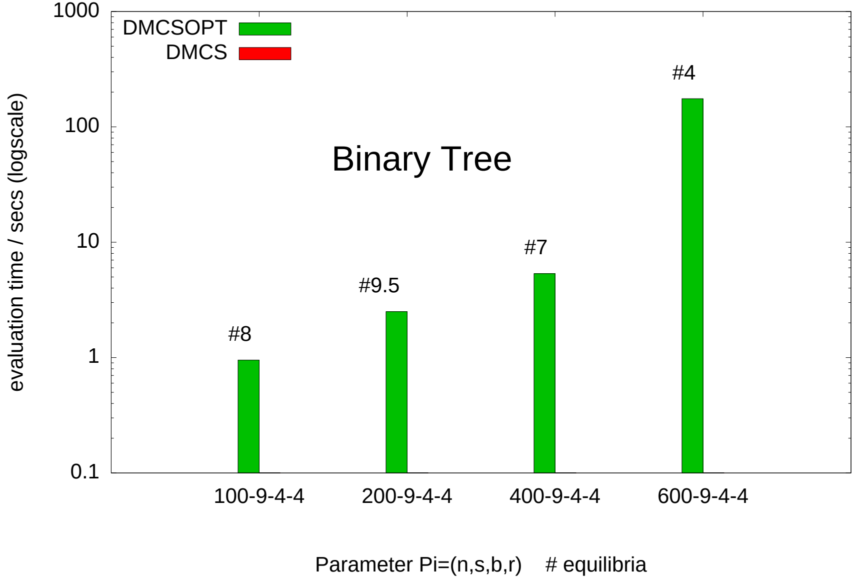
Experiments





Experiments





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

References I

- Gerhard Brewka and Thomas Eiter.
  Equilibria in heterogeneous nonmonotonic multi-context systems.
  In *AAAI'07*, pages 385–390. AAAI Press, 2007.
- Fausto Giunchiglia and Luciano Serafini.
  Multilanguage hierarchical logics or: How we can do without modal logics.
  Artificial Intelligence, 65(1):29–70, 1994.