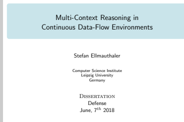
Multi-Context Reasoning in Continuous Data-Flow Environments

Stefan Ellmauthaler

Computer Science Institute
Leipzig University
Germany

DISSERTATION
Defense
June, 7^{th} 2018



Sehr geehrte Mitglieder der Verteidigungskommission, sehr geehrte Anwesende, auch ich möchte Sie zu diesem Vortrag begrüßen und mich allseits für das Interesse daran bedanken.

Da einige Zuhörer im Publikum nur deutsch, respektive englisch, sprechen, habe ich mich dazu entschieden die Folien in der Sprache der Dissertation, also englisch zu formulieren, während ich als Vortragssprache deutsch gewählt habe.

Der Titel meiner Dissertation ist **Multi-Context Reasoning in Continuous Data-Flow Environments** und mein Ziel ist es gewesen, **Grundlagen** für intelligentes Schließen in Fortwährend mit Informationen gefütterten Systemen zu ergründen.

Introduction

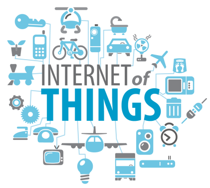
- The world gets more and more connected
  - Mobile devices (phones, notebooks, ...)
  - Electronic devices (fridges, stoves, TVs, doors, ...)
  - Tools (CCTVs, warehouse parts, ...)
  - "Things" (wares, items, bits, ...)
- The World Wide Web Consortium (W3C) works on a standardisation

  Web of Things (WoT)
- Industry 4.0 & WoT depends heavily on the idea of Internet of Things (IoT)

IoT Consequences

- many connected devices
- huge amount of collected data
- continuous exchange of information between devices and systems

⇒ An environment with Continuous Data-Flow



Assisted Living

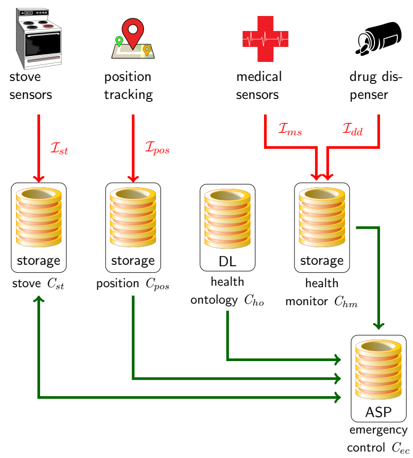
Real World Application

Basic Idea

- Healthcare for patients needs high amount of personal monitoring
- Enhance an apartment with an AI which monitors daily living activities of the inhabitants
- Coordinate services by outside healthcare providers
- Provide supervision and assistance to ensure the inhabitants
  - health,
  - safety, and
  - well-being

Assisted Living

Our Example



Goals

and their Realisation

Find a way to handle the

- integration of knowledge and beliefs from KR-formalisms
- dynamics which occur over time

Realised by

- reactive Multi-Context Systems, an extension of managed Multi-Context Systems
- asynchronous Multi-Context Systems, which can be seen as a language to model concurrent reasoning tasks

Outline

Outline

Logic — an Abstract Representation

- An abstract way to define a Logic
- Capable of realising monotone and non-monotone logics
- Representing different number of values (e.g. binary, many valued, fuzzy values, ...)

Definition (Logic [Brewka and Eiter, 2007])

A logic is a triple $L = \langle KB, BS, \mathbf{acc} \rangle$, where

- KB is a set of knowledge bases,
- BS is a set of belief sets, and
- $\mathbf{acc} : KB \mapsto 2^{BS}$, the *acceptance funtion* is a function which assigns to each knowledge base a set of belief sets.

Logic — to Represent KRR-Formalisms

Description Logic

- Decidable FO logic fragment
  - Concepts & Roles
  - $\mathcal{T}$Box & $\mathcal{A}$Box
- Monotone
- Many different versions ($\mathcal{AL}$, $\mathcal{ALC}$, $\mathcal{SHIF}$, $\mathcal{SROIC}$, ...)

Answer Set Programming

- Rule-Based KR formalism
  - Predicates, Default negation
  - Set of Rules
- Non-monotone
- Normal, disjunctive, negated ASP (w/ optimisation)

Represent KRR Formalisms

Description Logic $\mathcal{AL}$

$L_d = \langle KB_d, BS_d, \mathbf{acc}_d \rangle$

- KB_d are all ontologies
- BS_d is the set of deductively closed subsets in $\mathcal{AL}$
- $\mathbf{acc}_d$ is a mapping of $kb \in KB_d$ to $M \subseteq 2^{BS_d}$, s.t. $\forall_{m \in M} kb \models m$ holds.

Answer Set Programming

$L_{asp} = \langle KB_{asp}, BS_{asp}, \mathbf{acc}_{asp} \rangle$

- Let A be the set of all possible ground atoms
- KB_{asp} is the set of all answer programs over A.
- $BS_{asp} = 2^A$
- $\mathbf{acc}_{asp}$ maps each ASP program to its answer sets

2018-06-06

DISSERTATION
└Preliminaries
└Represent KRR Formalisms

All ontologies is the set of all well-formed description logic knowledge bases over $\mathcal{AL}$

Outline

Origins — from mMCS via [r/e]MCS to rMCS

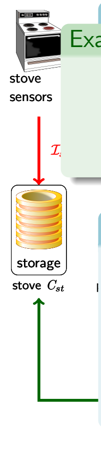
reactive Multi-Context Systems

- based on managed Multi-Context Systems [Brewka et al., 2011]
- old version got presented at ECAI 2014 [Brewka et al., 2014]
- evolving Multi-Context Systems at ECAI 2014 [Gonçalves et al., 2014]
- ⇒ complete redefinition of rMCS

Current reactive Multi-Context Systems

- less complicated, cycle-free definitions
- a generalisation of managed Multi-Context Systems
- declarative and operative bridge rules
- results on inconsistency management
- results on complexity
- results on simulating other approaches

Syntax — Building Blocks



Definition (Context)

Example

$$\text{setTemp}(hot) \leftarrow st{::}\text{tmp}(T), 42 < T$$
$$\mathbf{next}(\text{setPower}(off)) \leftarrow ec{:}\text{turnOff}(stove)$$
$$\mathbf{next}(\text{setPower}(off)) \leftarrow st{::}\text{switch}, st{:}\text{pw}$$

Definition (Bridge Rule)

Let $C = \langle C_1, \ldots, C_n \rangle$ be a tuple of contexts and IL $= \langle IL_1, \ldots, IL_k \rangle$ a tuple of input languages. A bridge rule for C_i over C and IL, $i \in \{1, \ldots, n\}$, is of the form

$$op \leftarrow a_1, \ldots, a_j, \mathbf{not}\ a_{j+1}, \ldots, \mathbf{not}\ a_m \text{ or}$$
$$\mathbf{next}(op) \leftarrow a_1, \ldots, a_j, \mathbf{not}\ a_{j+1}, \ldots, \mathbf{not}\ a_m$$

Syntax

Definition (Reactive Multi-Context System)

A *reactive Multi-Context System* is a tuple $M = \langle \mathsf{C}, \mathsf{IL}, \mathsf{BR} \rangle$, where

- $\mathsf{C} = \langle C_1, \ldots, C_n \rangle$ is a tuple of contexts;
- $\mathsf{IL} = \langle IL_1, \ldots, IL_k \rangle$ is a tuple of input languages;
- $\mathsf{BR} = \langle BR_1, \ldots, BR_n \rangle$ is a tuple such that each BR_i, $i \in \{1, \ldots, n\}$, is a set of bridge rules for C_i over C and IL.

Semantics

Current Snapshot

Definition (Configuration of Knowledge Bases)

Let $M = \langle \mathsf{C}, \mathsf{IL}, \mathsf{BR} \rangle$ be an rMCS, such that $\mathsf{C} = \langle C_1, \ldots, C_n \rangle$. A configuration of knowledge bases for M is a tuple $\mathsf{KB} = \langle kb_1, \ldots, kb_n \rangle$, such that $kb_i \in KB_i$, for each $i \in \{1, \ldots, n\}$. We use Con_M to denote the set of all configurations of knowledge bases for M.

Definition (Belief State)

Let $M = \langle \langle C_1, \ldots, C_n \rangle, \mathsf{IL}, \mathsf{BR} \rangle$ be an rMCS. Then, a belief state for M is a tuple $\mathsf{B} = \langle B_1, \ldots, B_n \rangle$ such that $B_i \in BS_i$, for each $i \in \{1, \ldots, n\}$. We use Bel_M to denote the set of all belief states for M.

Definition (Input)

Let $M = \langle \mathsf{C}, \langle IL_1, \ldots, IL_k \rangle, \mathsf{BR} \rangle$ be an rMCS. Then an input for M is a tuple $\mathsf{I} = \langle I_1, \ldots, I_k \rangle$ such that $I_i \subseteq IL_i$, $i \in \{1, \ldots, k\}$. The *set of all inputs for* M is denoted by Inp_M.

Semantics

One-Shot Reasoning

- Only utilise Declarative Bridge Rules
- A belief state is an Equilibrium if
  - the updated knowledge base
    (i.e. the management function result on the belief state, the input, and the current configuration)
  - has the belief state as one of the accepted belief states
    (i.e. it is part of the deductive closure of the semantics)

Definition (Equilibrium)

Let $M = \langle \langle C_1, \ldots, C_n \rangle, \mathsf{IL}, \mathsf{BR} \rangle$ be an rMCS, $\mathsf{KB} = \langle kb_1, \ldots, kb_n \rangle$ a configuration of knowledge bases for M, and I an input for M. Then, a belief state $\mathsf{B} = \langle B_1, \ldots, B_n \rangle$ for M is an equilibrium of M given KB and I if, for each $i \in \{1, \ldots, n\}$, we have that

$$B_i \in \mathbf{acc}_i(kb'), \text{ where } kb' = \mathbf{mng}_i(\mathbf{app}_i^{now}(\mathsf{I}, \mathsf{B}), kb_i).$$

Semantics

Reactive Reasoning

- Extend the concept of the Input, to be an Input Stream
- Operative Bridge Rules allow configuration changes
- Updates are based on the previously computed Equilibrium
- Results represented as Equilibria Stream and its dual Configuration Stream

Semantics

Reactive Reasoning

Definition (Update Function)

Let $M = \langle \mathsf{C}, \mathsf{IL}, \mathsf{BR} \rangle$ be an rMCS such that $\mathsf{C} = \langle C_1, \ldots, C_n \rangle$, $\mathsf{KB} = \langle kb_1, \ldots, kb_n \rangle$ a configuration of knowledge bases for M, I an input for M, and B a belief state for M.
Then, $\mathbf{upd}_M(\mathsf{KB}, \mathsf{I}, \mathsf{B}) = \langle kb'_1, \ldots, kb'_n \rangle$ is the update function for M, such that for each $i \in \{1 \ldots, n\}$, $kb'_i = \mathbf{mng}_i(\mathbf{app}_i^{next}(\mathsf{I}, \mathsf{B}), kb_i)$ holds.

Definition (Input Stream)

Let $M = \langle \mathsf{C}, \mathsf{IL}, \mathsf{BR} \rangle$ be an rMCS such that $\mathsf{IL} = \langle IL_1, \ldots, IL_k \rangle$. An input stream for M (until τ) is a function $\mathcal{I} : [1..\tau] \to \mathsf{Inp}_M$ where $\tau \in \mathbb{N} \cup \{\infty\}$.

Semantics

Equilibria Stream

Definition (Equilibria Stream)

Let $M = \langle \mathsf{C}, \mathsf{IL}, \mathsf{BR} \rangle$ be an rMCS, KB a configuration of knowledge bases for M, and $\mathcal{I}$ an input stream for M until τ where $\tau \in \mathbb{N} \cup \{\infty\}$. Then, an equilibria stream of M given KB and $\mathcal{I}$ is a function $\mathcal{B} : [1..\tau] \to \mathsf{Bel}_M$ such that

- $\mathcal{B}^t$ is an equilibrium of M given $\mathcal{KB}^t$ and $\mathcal{I}^t$, where $\mathcal{KB}^t$ is inductively defined as
  - $\mathcal{KB}^1 = \mathsf{KB}$
  - $\mathcal{KB}^{t+1} = \mathbf{upd}_M(\mathcal{KB}^t, \mathcal{I}^t, \mathcal{B}^t)$.

In a dual manner, we will refer to the function $\mathcal{KB} : [1..\tau] \to \mathsf{Con}_M$ as the *configurations stream of* M *given* KB, $\mathcal{I}$, *and* $\mathcal{B}$.

Computational Complexity and Expressiveness

Theorem

The table below summarizes the complexities of membership of problems $Q^\exists$ and $Q^\forall$ for finite input streams (until some $\tau \in \mathbb{N}$) depending on the context complexity. Hardness also holds if it holds for the context complexity.

$\mathcal{CC}(M, k{:}b)$	$Q^\exists$	$Q^\forall$
$\mathbf{P}$	$\mathbf{NP}$	$\mathbf{coNP}$
$\mathbf{\Delta}_i^{\mathrm{P}} (i \geq 2)$	$\mathbf{\Sigma}_i^{\mathrm{P}}$	$\mathbf{\Pi}_i^{\mathrm{P}}$
$\mathbf{\Sigma}_i^{\mathrm{P}} (i \geq 1)$	$\mathbf{\Sigma}_i^{\mathrm{P}}$	$\mathbf{\Pi}_i^{\mathrm{P}}$
PSPACE	**PSPACE**	**PSPACE**
EXPTime	**EXPTime**	**EXPTime**

Outline

Inconsistencies

Due to the combination of

- different Knowledge Representation formalisms,
- changing Knowledge Bases,
- incontestable Facts (input Stream), and
- no major restrictions to Bridge Rules

no belief state is an Equilibrium

Types of Inconsistencies

Local Inconsistency

- one Context formalism produces an inconsistency
- may be solved via inconsistency management techniques for the given formalism

Global Inconsistency

- one or more Concepts of the rMCS is introducing the inconsistency
- different reasons, like
  - Bridge Rules whose operations lead to inconsistencies in the context,
  - Bridge Rules which make themself unapplicable,
  - the absence of a belief state which can be accepted by every context

Deal with Global Inconsistencies

- Ensure consistency
- Repair inconsistent cases
- Work with inconsistent cases

Definition (Consistency)

Let M be an rMCS, KB a configuration of knowledge bases for M, and $\mathcal{I}$ an input stream for M. Then,

- M is consistent with respect to KB and $\mathcal{I}$ if there exists an equilibria stream of M given KB and $\mathcal{I}$.
- M is strongly consistent with respect to KB if, for every input stream $\mathcal{I}$ for M, M is consistent with respect to KB and $\mathcal{I}$.

Coherence and Consistency

Definition (Total Coherence)

A context C_i is totally coherent if $\mathbf{acc}_i(kb) \neq \emptyset$, for every $kb \in KB_i$.

Proposition

Let $M = \langle\langle C_1, \ldots, C_n\rangle, \mathsf{IL}, \mathsf{BR}\rangle$ be an acyclic rMCS such that every C_i, $1 \leq i \leq n$, is totally coherent, and KB a configuration of knowledge bases for M. Then, M is strongly consistent with respect to KB.

Repair

Definition (Repair)

Let $M = \langle \mathsf{C}, \mathsf{IL}, \langle BR_1, \ldots, BR_n\rangle\rangle$ be an rMCS, KB a configuration of knowledge bases for M, $\mathcal{I}$ an input stream for M until τ where $\tau \in \mathbb{N} \cup \{\infty\}$, and $ABR_M = \bigcup_{1 \leq i \leq n} BR_i$ the set of all bridge rules of M. Then, a repair for M given KB and $\mathcal{I}$ is a function $\mathcal{R} : [1..\tau] \to 2^{ABR_M}$ such that there exists a function $\mathcal{B} : [1..\tau] \to \mathsf{Bel}_M$ such that

- $\mathcal{B}^t$ is an equilibrium of $M[\mathcal{R}^t]$ given $\mathcal{KB}^t$ and $\mathcal{I}^t$, where $\mathcal{KB}^t$ is inductively defined as
  - $\mathcal{KB}^1 = \mathsf{KB}$
  - $\mathcal{KB}^{t+1} = \mathbf{upd}_{M[\mathcal{R}^t]}(\mathcal{KB}^t, \mathcal{I}^t, \mathcal{B}^t)$.

We refer to $\mathcal{B}$ as a repaired equilibria stream of M given KB, $\mathcal{I}$ and $\mathcal{R}$.

Repair

Proposition

Let $M = \langle\langle C_1, \ldots, C_n\rangle, \mathsf{IL}, \mathsf{BR}\rangle$ be an rMCS such that each C_i, $i \in \{1, \ldots, n\}$, is totally coherent, KB a configuration of knowledge bases for M, and $\mathcal{I}$ an input stream for M until τ. Then, there exists $\mathcal{R} : [1..\tau] \to 2^{br_M}$ and $\mathcal{B} : [1..\tau] \to \mathsf{Bel}_M$ such that $\mathcal{B}$ is a repaired equilibria stream given KB, $\mathcal{I}$ and $\mathcal{R}$.

Types of Repairs

Let $\mathcal{R}$ be a repair for some rMCS M given KB and $\mathcal{I}$. We say that $\mathcal{R}$ is a:

Minimal Repair if there is no repair $\mathcal{R}_a$ for M given KB and $\mathcal{I}$ such that $\mathcal{R}_a \subset \mathcal{R}$.

Global Repair if $\mathcal{R}^i = \mathcal{R}^j$ for every $i, j \leq \tau$.

Minimal Global Repair if $\mathcal{R}$ is global and there is no global repair $\mathcal{R}_a$ for M given KB and $\mathcal{I}$ such that $\mathcal{R}_a \subset \mathcal{R}$.

Incremental Repair if $\mathcal{R}^i \subseteq \mathcal{R}^j$ for every $i \leq j \leq \tau$.

Minimally Incremental Repair if $\mathcal{R}$ is incremental and there is no incremental repair $\mathcal{R}_a$ and $j \leq \tau$ such that $\mathcal{R}_a^i \subset \mathcal{R}^i$ for every $i \leq j$.

Partial Equilibria Stream

Definition (Partial Equilibria Stream)

Let $M = \langle \mathsf{C}, \mathsf{IL}, \mathsf{BR}\rangle$ be an rMCS, $\mathsf{KB} = \langle kb_1, \ldots, kb_n\rangle$ a configuration of knowledge bases for M, and $\mathcal{I}$ an input stream for M until τ where $\tau \in \mathbb{N} \cup \{\infty\}$. Then, a partial equilibria stream of M given KB and $\mathcal{I}$ is a partial function $\mathcal{B} : [1..\tau] \twoheadrightarrow \mathsf{Bel}_M$ such that

- $\mathcal{B}^t$ is an equilibrium of M given $\mathcal{KB}^t$ and $\mathcal{I}^t$,
- or $\mathcal{B}^t$ is undefined.

$\mathcal{KB}^t$ is inductively defined as

- $\mathcal{KB}^1 = \mathsf{KB}$
- $\mathcal{KB}^{t+1} = \begin{cases} \mathbf{upd}_M(\mathcal{KB}^t, \mathcal{I}^t, \mathcal{B}^t), & \text{if } \mathcal{B}^t \text{ is not undefined.} \\ \mathcal{KB}^t, & \text{otherwise.} \end{cases}$

Partial Equilibria Stream

Results

Proposition

Every equilibria stream of M given KB and $\mathcal{I}$ is a partial equilibria stream of M given KB and $\mathcal{I}$.

Proposition

Let M be an rMCS, KB a configuration of knowledge bases for M, and $\mathcal{I}$ an input stream for M until τ. Then, there exists $\mathcal{B} : [1..\tau] \nrightarrow \mathsf{Bel}_M$ such that $\mathcal{B}$ is a partial equilibria stream given KB and $\mathcal{I}$.

Outline

Recap

Contributions in the Talk

Two important topics in AI and KRR

- Knowledge Integration, Belief Exchange
- Handling of dynamics in Data-Flow Environments

reactive Multi-Context Systems

- Introduction of reactive Multi-Context Systems
  - Extension and Redefinition of managed Multi-Context Systems
  - Declarative and Operational Bridge Rules
- Complexity results
- Study on Inconsistency Management
  - Ensure Consistency (total coherence & acyclicity)
  - Repair cyclic rMCS
  - Work with incoherent contexts and partial equilibria streams

Further Contributions

- Introduction of reactive Multi-Context Systems
  - Historical development of reactive Multi-Context Systems
  - Modelling techniques and Considerations
  - Model other approaches from the literature
- Introduction of asynchronous Multi-Context Systems
  - Modelling language for (concurrent) Computation and Reasoning
  - Paradigm shift of Bridge Rules to Output Rules
  - Asynchronous computation mode, without synchronised agreement
  - Approach to pre-filter incoming stream data (Data Packing)
  - Methods to work on partial results and control the flow of computation

Own Publications

Journals

- Brewka, G., Ellmauthaler, S., Strass, H., Wallner, J. P., and Woltran, S. (2017).
  Abstract dialectical frameworks. an overview.
  IfCoLog Journal of Logics and their Applications Volume 4, Number 8. Formal Argumentation, 4(8):2263–2317.
- Brewka, G., Ellmauthaler, S., Gonalves, R., Knorr, M., Leite, J., and Pührer, J. (2018).
  Reactive multi-context systems: Heterogeneous reasoning in dynamic environments.
  Artificial Intelligence, 256:68–104.

Conferences

- 4 on ADFs
- 2 on reactive Multi-Context Systems

Outlook

- Inconsistency measurement [McIlraith and Weinberger, 2018]
- Implementation of rMCS
- Utilisation of rMCS (and aMCS)
  for a formal description language of distributed reasoning systems

Thank you for your interest

References I

... a selection

Description Logic

Syntax

Definition (Concept Descriptions for $\mathcal{ALC}$)

$$
\begin{array}{rrl}
C, D \rightarrow & A \mid & \text{(atomic concept)} \\
& \top \mid & \text{(universal concept)} \\
& \bot \mid & \text{(bottom concept)} \\
& \neg A \mid & \text{(atomic negation)} \\
& C \sqcap D \mid & \text{(intersection)} \\
& \forall R.C \mid & \text{(value restriction)} \\
& \exists R.\top \mid & \text{(limited existential quantification)} \\
& & \\
& \neg C \mid & \text{(negation - } \mathcal{C}\text{)} \\
& C \sqcup D \mid & \text{(union - } \mathcal{U}\text{)} \\
& \exists R.C \mid & \text{(full existential quantifier - } \mathcal{E}\text{)}
\end{array}
$$

Description Logic

Semantics

Definition (Semantics of $\mathcal{ALC}$ Concepts)

The *Interpretation* $\mathcal{I}$ is a tuple $\langle \Delta^{\mathcal{I}}, val \rangle$, where $\Delta^{\mathcal{I}}$ is a nonempty set, the *domain of the interpretation*. val is a valuation function which assigns to every atomic Concept A a set $A^{\mathcal{I}} \in \Delta^{\mathcal{I}}$ and to each atomic Role R a binary relation $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$.

$$
\begin{array}{rcl}
\top^{\mathcal{I}} & = & \Delta^{\mathcal{I}} \\
\bot^{\mathcal{I}} & = & \emptyset \\
(\neg A)^{\mathcal{I}} & = & \Delta^{\mathcal{I}} \setminus A^{\mathcal{I}} \\
(C \sqcap D)^{\mathcal{I}} & = & C^{\mathcal{I}} \cap D^{\mathcal{I}} \\
\forall R.C^{\mathcal{I}} & = & \{a \in \Delta^{\mathcal{I}} \mid \forall (b).(a,b) \in R^{\mathcal{I}} \rightarrow b \in C^{\mathcal{I}}\} \\
\exists R.\top^{\mathcal{I}} & = & \{a \in \Delta^{\mathcal{I}} \mid \exists (b).(a,b) \in R^{\mathcal{I}}\} \\
(\neg C)^{\mathcal{I}} & = & \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}} \\
(C \sqcup D)^{\mathcal{I}} & = & C^{\mathcal{I}} \cup D^{\mathcal{I}} \\
\exists R.C^{\mathcal{I}} & = & \{a \in \Delta^{\mathcal{I}} \mid \exists (b).(a,b) \in R^{\mathcal{I}} \wedge b \in C^{\mathcal{I}}\}
\end{array}
$$

Description Logic

Reasoning tasks

Reasoning with $\mathcal{ALC}$

Let C and D be two Concepts, then

- C subsumes D $\iff$ $C \sqcap \neg D$ is unsatisfiable,
- C is equivalent to D $\iff$ $C \sqcap \neg D$ and $\neg D \sqcap C$ are unsatisfiable, and
- C is disjoint to D $\iff$ $C \sqcap D$ is unsatisfiable.

This also holds with respect to a terminology $\mathcal{T}$.

- Tableaux-Calculus can decide satisfiability for a given $\mathcal{A}$Box
- $\mathcal{T}$Box can be transformed into $\mathcal{A}$Box

Answer Set Programming

Syntax

Definition (Normal Logic Program Rule)

A normal logic program rule r is of the form

$$a \leftarrow b_1, \ldots, b_n, not\ c_1, \ldots, not\ c_m$$

where a, $b_1, \ldots, b_n$ and $c_1, \ldots, c_m$ are ground atoms.

- a is the head of the rule ($hd(R)$).
- $body^+(r)$ is the set of positive atoms $b_1, \ldots, b_m$ and $body^-(r)$ is the set of negated atoms $b_{m+1}, \ldots, b_n$.
- The whole body of a rule R is $body(r) = body^+(r) \cup body^-(r)$.
- If the body is empty, the rule is a *fact*. "$\bot \leftarrow$" can be omitted. An empty head means that the rule implies $\bot$.

Answer Set Programming

Semantics

Definition (Stable Model Semantic for Normal Logic Programs)

Let S be a finite set of atoms and P a normal logic program.

- S is closed under P $\iff$ $a \in S$ whenever there is a rule $r \in P$ such that $head(r) = a$, $body^+(r) \subseteq S$, and $body^-(r) \cap S = \emptyset$.
- Let R be a derivation of rules of P. The set S defeats a rule $r_i \in R$ $\iff$ $S \cap\ body^-(r_i) \neq \emptyset$. A valid derivation in S contains no defeated rules.
- S is grounded in P $\iff$ $a \in S$ implies that $a \in atoms_d(R)$ holds in one derivation of P valid in S.

S is called a *stable model* of P if it is closed and grounded in P. Such a set is also called an *answer set*. We will write $AS(P)$ to denote the set of answer sets of P.

managed Multi-Context System

Definition (Logic Suite)

A *logic suite* $LS = (KB_{LS}, BS_{LS}, ACC_{LS})$ consists of the set BS_{LS} of possible belief sets, the set KB_{LS} of well-formed knowledge-bases, and a nonempty set ACC_{LS} of possible semantics of LS, i.e. $\mathbf{acc}_{LS} \in ACC_{LS}$ implies $\mathbf{acc}_{LS} : KB_{LS} \rightarrow 2^{BS_{LS}}$.

Definition (Management Function)

A management function over a logic suite LS and a management base OP is a function $mng : 2^{F_{LS}^{OP}} \times KB_{LS} \rightarrow 2^{KB_{LS} \times ACC_{LS}} \setminus \{\emptyset\}$.

managed Multi-Context System

Definition (Managed Multi-Context System)

A managed Multi-Context System M is a collection $(C_1, \ldots, C_n)$ of managed contexts where, for $1 \leq i \leq n$, each managed context C_i is a quintuple $C_i = (LS_i, kb_i, br_i, OP_i, mng_i)$ such that

- $LS_i = (BS_{LS_i}, KB_{LS_i}, ACC_{LS_i})$ is a logic suite,
- $kb_i \in KB_{LS_i}$ is a knowledge base,
- OP_i is a management base,
- br_i is a set of bridge rules for C_i, with the form

$$op_i \leftarrow (c_1 : p_1), \ldots, (c_j : p_j), not(c_{j+1} : p_{j+1}), \ldots, not(c_m : p_m).$$

  such that $op_i \in F_{LS_i}^{OP_i}$ and for all $1 \leq k \leq m$ there exists a context $c_k \in (C_1, \ldots, C_n)$ such that $p_k \in\ S \in\ BS_{LS_{c_k}}$, and
- mng_i is a management function over LS_i and OP_i.

managed Multi-Context System

Definition (Equilibria for managed Multi-Context Systems)

Let $M = (C_1, \ldots, C_n)$ be a managed multi-context system. A belief state $\mathrm{B} = (b_1, \ldots, b_n)$ is an equilibrium of M iff for every $1 \leq i \leq n$ there exists some $(kb_i', \mathbf{acc}_{LS_i}) \in mng_i(app_i(S), kb_i)$ such that $S_i \in \mathbf{acc}_{LS_i}(kb_i')$.

reactive Multi-Context System — Syntax

Definition (Context)

A context is a triple $C = \langle L, OP, \mathbf{mng} \rangle$ where

- $L = \langle KB, BS, \mathbf{acc} \rangle$ is a logic,
- OP is a *set of operations*,
- $\mathbf{mng} : 2^{OP} \times KB \to KB$ is a *management function*.

reactive Multi-Context System — Syntax

Definition (Bridge Rule)

Let $\mathsf{C} = \langle C_1, \dots, C_n \rangle$ be a tuple of contexts and $\mathsf{IL} = \langle IL_1, \dots, IL_k \rangle$ a tuple of input languages. A *bridge rule for C_i over* C *and* IL, $i \in \{1, \dots, n\}$, is of the form

$$\mathbf{op} \leftarrow a_1, \dots, a_j, \mathbf{not}\ a_{j+1}, \dots, \mathbf{not}\ a_m \qquad (1)$$

such that $\mathbf{op} = op$ or $\mathbf{op} = \mathbf{next}(op)$ for $op \in OP_i$, $j \in \{0, \dots, m\}$, and every *atom* a_ℓ, $\ell \in \{1, \dots, m\}$, is one of the following:

- a *context atom* $c{:}b$ with $c \in \{1, \dots, n\}$ and $b \in B$ for some $B \in BS_c$
- an *input atom* $s{::}b$ with $s \in \{1, \dots, k\}$ and $b \in IL_s$.

For a bridge rule r of the form (1) $\mathrm{head}(r)$ denotes $\mathbf{op}$, the *head of* r, while $\mathrm{body}(r) = \{a_1, \dots, a_j, \mathbf{not}\ a_{j+1}, \dots, \mathbf{not}\ a_m\}$ is the *body of* r. A *literal* is either an atom or an atom preceded by $\mathbf{not}$.

reactive Multi-Context System — Syntax

Definition (Reactive Multi-Context System)

A *reactive Multi-Context System* is a tuple $M = \langle \mathsf{C}, \mathsf{IL}, \mathsf{BR} \rangle$, where

- $\mathsf{C} = \langle C_1, \dots, C_n \rangle$ is a tuple of contexts;
- $\mathsf{IL} = \langle IL_1, \dots, IL_k \rangle$ is a tuple of input languages;
- $\mathsf{BR} = \langle BR_1, \dots, BR_n \rangle$ is a tuple such that each BR_i, $i \in \{1, \dots, n\}$, is a set of bridge rules for C_i over C and IL.

reactive Multi-Context System — Semantics

Definition (Belief State)

Let $M = \langle \langle C_1, \dots, C_n \rangle, \mathsf{IL}, \mathsf{BR} \rangle$ be an rMCS. Then, a *belief state for M* is a tuple $\mathsf{B} = \langle B_1, \dots, B_n \rangle$ such that $B_i \in BS_i$, for each $i \in \{1, \dots, n\}$. We use Bel_M to denote the set of all belief states for M.

Definition (Input)

Let $M = \langle \mathsf{C}, \langle IL_1, \dots, IL_k \rangle, \mathsf{BR} \rangle$ be an rMCS. Then an *input for M* is a tuple $\mathsf{I} = \langle I_1, \dots, I_k \rangle$ such that $I_i \subseteq IL_i$, $i \in \{1, \dots, k\}$. The *set of all inputs for M* is denoted by Inp_M.

reactive Multi-Context System — Semantics

Definition (Satisfaction of Literals)

Let $M = \langle \mathsf{C}, \mathsf{IL}, \mathsf{BR} \rangle$ be an rMCS, such that $\mathsf{C} = \langle C_1, \dots, C_n \rangle$ and $\mathsf{IL} = \langle IL_1, \dots, IL_k \rangle$. Given an input $\mathsf{I} = \langle I_1, \dots, I_k \rangle$ for M and a belief state $\mathsf{B} = \langle B_1, \dots, B_n \rangle$ for M, we define the satisfaction of literals as:

- $\langle \mathsf{I}, \mathsf{B} \rangle \models a_\ell$ if a_ℓ is of the form $c{:}b$ and $b \in B_c$;
- $\langle \mathsf{I}, \mathsf{B} \rangle \models a_\ell$ if a_ℓ is of the form $s{::}b$ and $b \in I_s$;
- $\langle \mathsf{I}, \mathsf{B} \rangle \models \mathbf{not}\ a_\ell$ if $\langle \mathsf{I}, \mathsf{B} \rangle \not\models a_\ell$.

Let r be a bridge rule for C_i over C and IL. Then

- $\langle \mathsf{I}, \mathsf{B} \rangle \models \mathrm{body}(r)$ if $\langle \mathsf{I}, \mathsf{B} \rangle \models l$ for every $l \in \mathrm{body}(r)$.

reactive Multi-Context System — Semantics

Definition (Applicable Operators)

Let $M = \langle \mathsf{C}, \mathsf{IL}, \mathsf{BR} \rangle$ be an rMCS, such that $\mathsf{C} = \langle C_1, \dots, C_n \rangle$ and $\mathsf{BR} = \langle BR_1, \dots, BR_n \rangle$. Given an input I for M and a belief state B for M, we define, for each $i \in \{1, \dots, n\}$, the sets

- $\mathbf{app}_i^{now}(\mathsf{I}, \mathsf{B}) = \{\mathrm{head}(r) \mid r \in BR_i, \langle \mathsf{I}, \mathsf{B} \rangle \models \mathrm{body}(r), \mathrm{head}(r) \in OP_i\}$;
- $\mathbf{app}_i^{next}(\mathsf{I}, \mathsf{B}) = \{op \mid r \in BR_i, \langle \mathsf{I}, \mathsf{B} \rangle \models \mathrm{body}(r), \mathrm{head}(r) = \mathbf{next}(op)\}$.

reactive Multi-Context System — Semantics

Definition (Configuration of Knowledge Bases)

Let $M = \langle \mathsf{C}, \mathsf{IL}, \mathsf{BR} \rangle$ be an rMCS, such that $\mathsf{C} = \langle C_1, \dots, C_n \rangle$. A *configuration of knowledge bases for M* is a tuple $\mathsf{KB} = \langle kb_1, \dots, kb_n \rangle$, such that $kb_i \in KB_i$, for each $i \in \{1, \dots, n\}$. We use Con_M to denote the set of all configurations of knowledge bases for M.

Definition (Equilibrium)

Let $M = \langle \langle C_1, \dots, C_n \rangle, \mathsf{IL}, \mathsf{BR} \rangle$ be an rMCS, $\mathsf{KB} = \langle kb_1, \dots, kb_n \rangle$ a configuration of knowledge bases for M, and I an input for M. Then, a belief state $\mathsf{B} = \langle B_1, \dots, B_n \rangle$ for M is an *equilibrium* of M given KB and I if, for each $i \in \{1, \dots, n\}$, we have that

$$B_i \in \mathbf{acc}_i(kb'), \text{ where } kb' = \mathbf{mng}_i(\mathbf{app}_i^{now}(\mathsf{I}, \mathsf{B}), kb_i).$$

reactive Multi-Context System — Semantics

Definition (Update Function)

Let $M = \langle \mathsf{C}, \mathsf{IL}, \mathsf{BR} \rangle$ be an rMCS such that $\mathsf{C} = \langle C_1, \dots, C_n \rangle$, $\mathsf{KB} = \langle kb_1, \dots, kb_n \rangle$ a configuration of knowledge bases for M, I an input for M, and B a belief state for M.
Then, $\mathbf{upd}_M(\mathsf{KB}, \mathsf{I}, \mathsf{B}) = \langle kb'_1, \dots, kb'_n \rangle$ is the *update function for M*, such that for each $i \in \{1 \dots, n\}$, $kb'_i = \mathbf{mng}_i(\mathbf{app}_i^{next}(\mathsf{I}, \mathsf{B}), kb_i)$ holds.

Definition (Input Stream)

Let $M = \langle \mathsf{C}, \mathsf{IL}, \mathsf{BR} \rangle$ be an rMCS such that $\mathsf{IL} = \langle IL_1, \dots, IL_k \rangle$. An **input stream for** M (until τ) is a function $\mathcal{I} : [1..\tau] \to \mathsf{Inp}_M$ where $\tau \in \mathbb{N} \cup \{\infty\}$.

reactive Multi-Context System — Semantics

Definition (Equilibria Stream)

Let $M = \langle \mathsf{C}, \mathsf{IL}, \mathsf{BR} \rangle$ be an rMCS, KB a configuration of knowledge bases for M, and $\mathcal{I}$ an input stream for M until τ where $\tau \in \mathbb{N} \cup \{\infty\}$. Then, an *equilibria stream of M given* KB *and $\mathcal{I}$* is a function $\mathcal{B} : [1..\tau] \to \mathsf{Bel}_M$ such that

- $\mathcal{B}^t$ is an equilibrium of M given $\mathcal{KB}^t$ and $\mathcal{I}^t$, where $\mathcal{KB}^t$ is inductively defined as
  - $\mathcal{KB}^1 = \mathsf{KB}$
  - $\mathcal{KB}^{t+1} = \mathbf{upd}_M(\mathcal{KB}^t, \mathcal{I}^t, \mathcal{B}^t)$.

In a dual manner, we will refer to the function $\mathcal{KB} : [1..\tau] \to \mathsf{Con}_M$ as the *configurations stream of M given* KB, *$\mathcal{I}$, and $\mathcal{B}$*.

reactive Multi-Context System — Variables

Definition (Rule Schemata)

Given a tuple of input languages $\mathsf{IL} = \langle IL_1, \ldots, IL_m \rangle$ and the tuple of contexts $\mathsf{C} = \langle C_1, \ldots, C_n \rangle$. Let $\mathcal{A}$ be the alphabet of symbols occuring in all possible bridge rules for all contexts $C_i \in \{C_1, \ldots, C_n\}$ over C and IL, the set of symbols $\mathcal{P}$ be a set of parameters, such that $\mathcal{A} \cap \mathcal{P} = \emptyset$.

- An *instantiation term* is a string built upon the alphabet $\mathcal{A}$ for C and IL, and
- an *instantiation condition* for C and IL is a predicate $ic(T_1, \ldots, T_o)$, where T_1 to T_o are strings constructed over the alphabet $\mathcal{A} \cup \mathcal{P}$.

A *rule schemata* R for C and IL with parameters $\mathcal{P}$ is of the form

$$H \leftarrow A_1, \ldots, A_p, \mathbf{not}\ A_{p+1}, \ldots, \mathbf{not}\ A_q, D_1, \ldots, D_r \qquad (2)$$

such that $H, A_1, \ldots, A_q$ are strings over $\mathcal{A} \cup \mathcal{P}$ and D_j is an instantion condition for each $j \in \{1, \ldots, r\}$.

reactive Multi-Context System — Variables

Definition (Rule Instantiation)

A bridge rule

$$r = op \leftarrow a_1, \ldots, a_j, \mathbf{not}\ a_{j+1}, \ldots, \mathbf{not}\ a_k$$

for C_i over $\mathsf{C} = \langle C_1, \ldots, C_n \rangle$ and $\mathsf{IL} = \langle IL_1, \ldots, IL_m \rangle$ is called an *instance* of a rule schemata R of the form (2) for C and IL with parameters $\mathcal{P}$ if $r = (H \leftarrow A_1, \ldots, A_p, \mathbf{not}\ A_{p+1}, \ldots, \mathbf{not}\ A_q)\sigma$ holds for a uniform substitution σ, such that σ substitutes every parameter with its instantiation terms, and for each $D_j = ic(T_1, \ldots, T_o)$, $j \in \{1, \ldots, r\}$, the predicate $ic(T_1\sigma, \ldots, T_o\sigma)$ holds.

Inconsistency Management — Coherence

Definition (Consistency)

Let M be an rMCS, KB a configuration of knowledge bases for M, and $\mathcal{I}$ an input stream for M. Then, M is *consistent* with respect to KB and $\mathcal{I}$ if there exists an equilibria stream of M given KB and $\mathcal{I}$. M is *strongly consistent* with respect to KB if, for every input stream $\mathcal{I}$ for M, M is consistent with respect to KB and $\mathcal{I}$.

Definition (Total Coherence)

A context C_i is *totally coherent* if $\mathbf{acc}_i(kb) \neq \emptyset$, for every $kb \in KB_i$.

Inconsistency Management — Acyclicity

Definition (Information Dependence)

Given an rMCS $M = \langle\langle C_1, \ldots, C_n \rangle, \mathsf{IL}, \mathsf{BR} \rangle$, $\lhd_M$ is the binary relation over contexts of M such that $(C_i, C_j) \in \lhd_M$ if there is a bridge rule $r \in BR_i$ and $j{:}b \in \mathrm{body}(r)$ for some b. If $(C_i, C_j) \in \lhd_M$, also denoted by $C_i \lhd_M C_j$, we say that C_i depends on C_j in M, dropping the reference to M whenever unambiguous.

Definition (Acyclicity)

An rMCS M is called *acyclic* if the transitive closure of $\lhd_M$ is irreflexive.

Inconsistency Management

Proposition

Let $M = \langle\langle C_1, \ldots, C_n \rangle, \mathsf{IL}, \mathsf{BR} \rangle$ be an acyclic rMCS such that every C_i, $1 \leq i \leq n$, is totally coherent, and KB *a configuration of knowledge bases for M. Then, M is strongly consistent with respect to* KB.

STARQL — Definition

SELECT $selectClause(\vec{x}, \vec{y})$
FROM $listOfWindowedStreamExpressions$
USING $listOfResources$
WHERE $\Psi(\vec{x})$
SEQUENCE BY $seqMethod$
HAVING $\Phi(\vec{x}, \vec{y})$

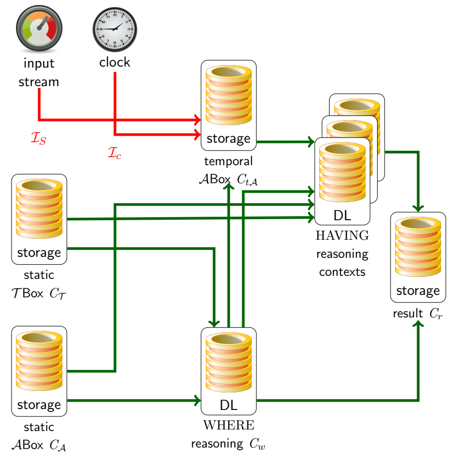
STARQL — Example

Example

```
CREATE STREAM S_out AS

SELECT {?sens rdf:type MonIncTemp}<NOW>
FROM S 0s<-[NOW-2s, NOW]->1s
USING  STATIC ABOX <http://example.org/staticABox>,
       TBOX <http://example.org/TBox>
WHERE { ?sens rdf:type TempSensor }
SEQUENCE BY StdSeq AS SEQ1
HAVING FORALL i<= j in SEQ1 ,x,y:
       IF ( { ?sens rd ?x }<i> AND { ?sens rd ?y }<j> )
       THEN ?x <= ?y
```

Simulate STARQL



Outline

Landscape of MCSs with Stream Reasoning

Basic Concepts

- Multi-Context Systems (MCS) [Brewka and Eiter, 2007]
- managed Multi-Context Systems (mMCS) [Brewka et al., 2011]

Reactive MCS Family

- (old) reactive Multi-Context Systems [Brewka et al., 2014]
- evolving Multi-Context Systems (eMCS) [Gonçalves et al., 2014]
- reactive Multi-Context Systems (rMCS) [Brewka et al., 2018]

Focus on Reasoning on Streams

streaming Multi-Context Systems (sMCS) [Dao-Tran and Eiter, 2017]

Other Dynamic Extensions

dynamic managed Multi-Context Systems on timed Contexts (dmMCS) [Cabalar et al., 2017]

Other Stream Reasoning Approaches

LARS [Beck et al., 2015]

Logic-based framework for Analyzing Reasoning over Streams

- Rule-based concept
- Reasoning over Streams
- Novel Window operator ⊞ & modal logic operators ◇ and □
- Utilises FLP semantics of logic programs with a stream "memory" of the window size

STARQL [Özçep et al., 2013]

Streaming and Temporal ontology Access with a Reasoning-based Query Language

- Can be seen as C-SPARQL with intelligent reasoning
- Allows temporal and data-based partition of Streams

Reactive Logic Programming

EVOLP [Alferes et al., 2002]

- Logic programs which allow the use of assertions
- Every rule may be written as an assertion
- Assertions may be nested
- Programs are evaluated multiple times over time
- If an `assert`-predicate is true, it is considered as a rule further on

`oclingo` [Gebser et al., 2012]

- An ASP-solver
- Allows external atoms to appear during run-time
- Only communicates atoms, not rules
- But allows for fine grained AS-enumeration options