EMCL @ Institute of Information Systems

Thomas Eiter

Institute of Information Systems
Vienna University of Technology

eiter@kr.tuwien.ac.at

EMCL Student Workshop Vienna, February 21, 2012

Institute of Information Systems

- Distributed Systems Group (DSG)
- **Databases and Artificial Intelligence Group (DBAI)**
- **Knowledge Based Systems Group (KBS)**
- **Formal Methods in Systems Engineering (FORSYTE)**
- Parallel Computing

Databases and Artificial Intelligence Group (DBAI)

http://www.dbai.tuwien.ac.at/

R. Pichler, S. Woltran , G. Gottlob (U Oxford)

- Foundations of databases
- Semistructured data
- Advanced database systems
- Computational logic and complexity
- Knowledge Representation and Reasoning (e.g. logic-based argumentation systems)

Formal Methods in Systems Engineering

http://www.forsyte.tuwien.ac.at

H. Veith

- Formal Methods for Embedded Systems
- Model Checking and Constraint Solving
- Automata, Logic, and Complexity

Knowledge Based Systems Group (KBS)

http://www.kr.tuwien.ac.at/

U. Egly, T. Eiter, S. Szeider, H. Tompits

- Knowledge representation and reasoning
- Computational logic and complexity
- Declarative problem solving
- Discrete Reasoning Methods
- Intelligent agents
- Mobile robots
- Knowledge-based systems in engineering

Work @ TU Vienna.KBS

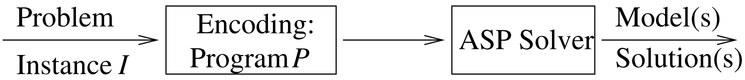
- There are several more specific research topics available in the groups
- Listing them all would be exhausting (visit the webpages!)
- In the KBS groups, some major topics are:
  - Answer Set Programming
  - Contextual Reasoning
  - Reasoning in Ontologies
  - SAT/QBF Solving, CSP
- Projects in these areas offer the opportunity of student projects/theses
- Limited funds for student grants are available

Topic 1: Answer Set Programming (ASP)

A recent *declarative problem solving method*

General idea

Reduce solving of a problem I to computing models of a logic program / SAT theory



1. *Encode* I as a (non-monotonic) logic program P, such that solutions of I are represented by models of P
2. *Compute* some model M of P, using an ASP solver
3. *Extract* some solution for I from M.

Example: Graph 3-Coloring

Color all nodes of a graph with colors r, g, b such that adjacent nodes have different color.

Problem specification P_{PS}

$$g(X) \vee r(X) \vee b(X) \leftarrow node(X) \quad \} \textbf{ Guess}$$

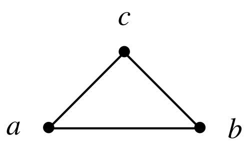
$$\left.\begin{array}{l} \leftarrow b(X), b(Y), edge(X,Y) \\ \leftarrow r(X), r(Y), edge(X,Y) \\ \leftarrow g(X), g(Y), edge(X,Y) \end{array}\right\} \textbf{ Check}$$

Data P_D: Graph $G = (V, E)$

$$P_D = \{node(v) \mid v \in V\} \cup \{edge(v,w) \mid (v,w) \in E\}.$$

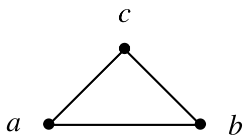
3-colorings $\rightleftharpoons$ models:
$v \in V$ has color $c \in \{r, g, b\}$ iff $c(v)$ is in the corr. model of $P_{PS} \cup P_D$.

Example: 3-Coloring (ctd.)

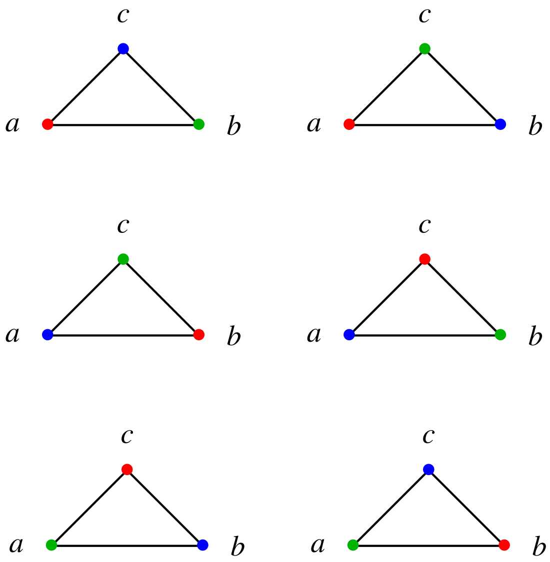


$$P_D = \{node(a), node(b),\\ node(c), edge(a,b),\\ edge(b,c), edge(a,c)\}$$

Example: 3-Coloring (ctd.)



$$P_D = \{node(a), node(b), \\ node(c), edge(a,b), \\ edge(b,c), edge(a,c)\}$$



ASP Applications

Problems in many domains, see

`http://www.kr.tuwien.ac.at/projects/WASP/report.html`

- configuration
- planning, routing
- diagnosis (E.g., Space shuttle reaction control)
- security analysis
- verification
- bioinformatics
- knowledge management
- combinatorics
- . . .

ASP Showcase:

`http://www.kr.tuwien.ac.at/projects/WASP/showcase.html`

The DLV System (TU Vienna / UNICAL)

http://www.dbai.tuwien.ac.at/proj/dlv/

- DLV is a state-of-the-art disjunctive answer set solver (ASP competitions 2007/09)
- Based on strong theoretical foundations
- Features non-monotonic negation and disjunction

$$works(X) :- component(X), \text{not}\, broken(X).$$
$$male(X) \vee female(X) :- person(X).$$

- Rich program syntax (⇒ **high expressiveness**)
- Front-ends for specific problems (diagnosis, planning, etc.).
- Many extensions: DLVHEX, DLV^{DB}, DLT, DLV-Complex, dl-programs, . . . ,

Ongoing Work

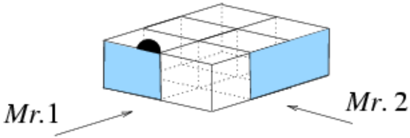
- Software Engineering for ASP
  Tools, debugging, methodologies (Hans Tompits, FWF)
- Modular logic programs
  Systems of logics programs / modular composition (FWF)
- Logic Programming with External Source Access (FWF)
- MyITS: Intelligent Travel Planning (FFG)
- Hybrid knowledge bases
  Combine ASP rules and ontologies (FP7 IP Ontorule, FWF)

Theory, prototypes, applications

With U Potsdam (T. Schaub), UNICAL (N.Leone, G.Ianni), international companies (ILOG/IBM;Ontoprise, etc), E. Erdem (Sabanci U), SIEMENS (A. Polleres), local industry,

Topic 2: Contextual Reasoning

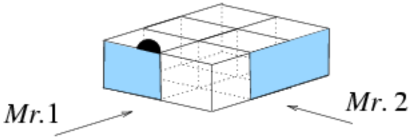
Magic Box



- J. McCarthy: How to interrelate contexts?

Topic 2: Contextual Reasoning

Magic Box

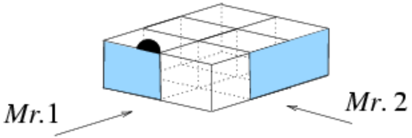


- J. McCarthy: How to interrelate contexts?
- Trento School (Giunchiglia, Serafini et al.)
  Bridge rules for information exchange

$$\begin{aligned} Mr.1: &\quad row(X) \leftarrow (Mr.2, sees\_row(X)) \\ Mr.2: &\quad col(Y) \leftarrow (Mr.1, sees\_col(Y)) \end{aligned}$$

Topic 2: Contextual Reasoning

Magic Box



- J. McCarthy: How to interrelate contexts?
- Trento School (Giunchiglia, Serafini et al.)
  Bridge rules for information exchange

$$\begin{aligned} Mr.1: &\quad row(X) \leftarrow (Mr.2, sees\_row(X)) \\ Mr.2: &\quad col(Y) \leftarrow (Mr.1, sees\_col(Y)) \end{aligned}$$

- Brewka & E_: *Nonmonotonic Multi Context Systems* (MCS)

Nonmonotonic Multi-Context Systems (MCS)

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

consists of contexts

$$C_i = (L_i, kb_i, br_i),\ i = 1, \ldots, n,$$

where

- each L_i is an (abstract) "logic,"
- each $kb_i \in \mathbf{KB}_i$ is a knowledge base in L_i, and
- each br_i is a set of bridge rules (possibly with negation)

- Captures many popular logics L_i, e.g. description logics, modal logics, temporal logics, default logics, logic programs
- Semantics in terms of *equilibria*, which are stable states $S = (S_1, \ldots, S_n)$ of M evaluating the kb_i and br_i

Example

Suppose a MCS $M = (C_1, C_2)$ has two contexts, expressing the individual views of a paper by its authors.

- C_1:
  - L_1 = Classical Logic
  - $kb_1 = \{ unhappy \supset revision \}$
  - $br_1 = \{ unhappy \leftarrow (2 : work) \}$
- C_2:
  - L_2 = Default Logic (R.Reiter)
  - $kb_2 = \{ good : accepted / accepted \}$
  - $br_2 = \{ work \leftarrow (1 : revision),$
    $good \leftarrow \text{not}(1 : unhappy) \}$

M has two equilibria:

- $E_1 = (Cn(\{unhappy, revision\}), Cn(\{work\}))$ and
- $E_2 = (Cn(\{unhappy \supset revision\}), Cn(\{good, accepted\}))$

MCS Aspects

- Fixpoint characterizations (under operational semantics)
- Relationship to game-theoretic concepts (e.g., Nash-equilibria of particular games, sometimes)
- A rich framework for interlinking heterogeneous knowledge systems
  - Databases, knowledge bases
  - Argumentation systems
- Potential for Applications in Social Aggregation

MCS – Ongoing work

- Project *Inconsistency Management for Knowledge-Integration Systems* (WWTF, M. Fink)
  - a general formalism and a suite of basic methods for inconsistency management in MCS,
  - algorithms for their practical realization.
- Distributed MCS
  - consistency
  - query answering
- Stream Processing
- Generalizations of MCS (e.g., Managed MCS)

With DERI Galway (M. Hauswirth, A. Polleres), U Leipzig (G.Brewka), UNICAL (N.Leone, G.Ianni), EPFL (CH. Koch)

Topic 3: Reasoning in Ontologies

- Formal ontologies serve for making conceptual models of domains (human anatomy, airplanes, products, ...)
- *Description Logics* are the premier logic-based formalism for ontology representation.
- They model *concepts* (classes of objects) and *roles* (binary relations between objects).
- A *DL knowledge base* comprises a *taxonomoy part* (*T-Box*) and *assertions* (*A-Box*, facts).

Example: Genealogy

$$\text{T-Box} = \left\{ \begin{array}{l} Person \equiv Female \sqcup Male, \\ Parent \equiv \exists hasChild.Person, \\ HasNoSons \equiv Parent \sqcap \forall hasChild.Female \end{array} \right\}$$

$$\text{A-Box} = \{Parent(Mary), hasChild(Tom, Jen), Female(Jen)\}$$

Semantics

- Semantically, many core DLs are decidable fragments of first-order predicate logic (FOL)
- The semantics of a KB may be given by a transformation to FOL

	Syntax	Semantics (FOL-translation)
negation	$\neg C$	$\neg C(x)$
conjunction	$C_1 \sqcap C_2$	$C_1(x) \wedge C_2(x)$
disjunction	$C_1 \sqcup C_2$	$C_1(x) \vee C_2(x)$
universal quant.	$\forall r.C$	$\forall y.r(x,y) \rightarrow C(y)$
existential quant.	$\exists r.C$	$\exists y.r(x,y) \wedge C(y)$

- Additional constructors, e.g.,

Qualified number restrictions ($\mathcal{Q}$)	$\leq n\, r.C$ $\geq n\, r.C$	$\exists^{\geq n} y.r(x,y) \wedge C(y)$ $\exists^{\leq n} y.r(x,y) \wedge C(y)$
Inverse roles ($\mathcal{I}$)	r^-	$r^-(x,y) \equiv r(y,x)$

- Optionally, axioms for roles, e.g.,

Role hierarchies ($\mathcal{H}$)	$r \sqsubseteq r'$	$\forall x,y.r(x,y) \rightarrow r'(x,y)$
Transitivity	Trans(r)	$\forall x,y,z.r(x,y) \wedge r(y,z) \rightarrow r(x,z)$

A large *family of DLs* exists: $\mathcal{ALC}$, $\mathcal{SH}$, $\mathcal{SHIQ}$, $\mathcal{SROIQ}$, ...

Reasoning Tasks

- Traditional reasoning tasks:
  - testing satisfiability: Is the KB logically consistent?
  - concept subsumption: Are all males persons?
  - instance checking: Is Jen a person?

Reasoning Tasks

- Traditional reasoning tasks:
  - testing satisfiability: Is the KB logically consistent?
  - concept subsumption: Are all males persons?
  - instance checking: Is Jen a person?
- Important new reasoning task: Conjunctive Query Answering (CQA)
  - well-known in databases
  - allows to *join pieces* of information

Example: Females (x) who have brothers

$$Female(x), hasChild(y, x), hasChild(y, z), Male(z)$$

Reasoning Tasks

- Traditional reasoning tasks:
  - testing satisfiability: Is the KB logically consistent?
  - concept subsumption: Are all males persons?
  - instance checking: Is Jen a person?
- Important new reasoning task: Conjunctive Query Answering (CQA)
  - well-known in databases
  - allows to *join pieces* of information

Example: Females (x) who have brothers

$$\exists y, z. Female(x), hasChild(y, x), hasChild(y, z), Male(z)$$

Reasoning Tasks

- Traditional reasoning tasks:
  - testing satisfiability: Is the KB logically consistent?
  - concept subsumption: Are all males persons?
  - instance checking: Is Jen a person?

- Important new reasoning task: Conjunctive Query Answering (CQA)

$$\text{T-Box} = \left\{ \begin{array}{l} \mathit{Person} \equiv \mathit{Female} \sqcup \mathit{Male}, \\ \mathit{Parent} \equiv \exists \mathit{hasChild}.\mathit{Person}, \\ \mathit{HasNoSons} \equiv \mathit{Parent} \sqcap \forall \mathit{hasChild}.\mathit{Female} \end{array} \right\}$$

$$\text{A-Box} = \{\mathit{Parent}(\mathit{Mary}), \mathit{hasChild}(\mathit{Tom}, \mathit{Jen}), \mathit{Female}(\mathit{Jen})\}$$

$$\exists y, z.\mathit{Female}(x), \mathit{hasChild}(y, x), \mathit{hasChild}(y, z), \mathit{Male}(z)$$

Reasoning Tasks

- Traditional reasoning tasks:
  - testing satisfiability: Is the KB logically consistent?
  - concept subsumption: Are all males persons?
  - instance checking: Is Jen a person?
- Important new reasoning task: Conjunctive Query Answering (CQA)
  - well-known in databases
  - allows to *join pieces* of information

Example: Females (x) who have brothers

$$Female(y), hasChild(y, x), hasChild(y, z), Male(z)$$

- Not (efficiently) reducible to traditional tasks in general
- **Problem:** develop *(worst-case) optimal algorithms* for CQA in relevant DLs
  Requires to characterize the *computational complexity* of CQA

Ongoing Work

- Reasoning in Hybrid Knowledge Bases (FWF P20840)
  - Combine ASP rules and ontologies
  - Conjunctive Query Answering (= Select-Project-Join)
  - System Prototypes
- Inconsistency Management in Hybrid KBs
- Recursive Queries over Semantically Enriched Data Repositories (Ortiz, FWF)
  - Queries extending SPJ with limited recursion

Theory, algorithms, prototypes (e.g., DREWEL, KAOS), applications

With U Bremen (C.Lutz), U Bolzano (D.Calvanese), U Oxford (G.Gottlob), KIT (S. Rudolph) etc

Logic

A *logic* L is a tuple $L = (\mathbf{KB}_L, \mathbf{BS}_L, \mathbf{ACC}_L)$, where

- $\mathbf{KB}_L$ is a set of well-formed knowledge bases, each being a set (of formulas)
- $\mathbf{BS}_L$ is a set of possible belief sets, each being a set (of formulas)
- $\mathbf{ACC}_L : \mathbf{KB}_L \rightarrow 2^{\mathbf{BS}_L}$ assigns each KB a set of acceptable belief sets

Thus, logic L caters for multiple extensions of a knowledge base.

Bridge Rules

A L_i-bridge rule over logics $L_1, \ldots, L_n$, $1 \leq i \leq n$, is of the form

$$s \leftarrow (r_1 : p_1), \ldots, (r_j : p_j), \mathrm{not}(r_{j+1} : p_{j+1}), \ldots, \mathrm{not}(r_m : p_m)$$

where $kb \cup \{s\} \in \mathbf{KB}_i$ for each $kb \in \mathbf{KB}_i$, each $r_k \in \{1, \ldots, n\}$, and each p_k is in some belief set of L_{r_k}.

Note: Such rules are akin to rules of normal logic programs!

Equilibrium Semantics

Belief State

A *belief state* is a sequence $S = (S_1, \dots, S_n)$ of belief sets S_i in L_i

Applicable Bridge Rules

For $M = (C_1, \dots, C_n)$ and belief state $S = (S_1, \dots, S_n)$, the bridge rule

$$s \leftarrow (r_1 : p_1), \dots, (r_j : p_j), \mathrm{not}(r_{j+1} : p_{j+1}), \dots, \mathrm{not}(r_m : p_m)$$

is *applicable in* S iff (1) $p_i \in S_{r_i}$, for $1 \leq i \leq j$, and (2) $p_k \notin S_{r_k}$, for $j < k \leq m$.

Equilibrium

A belief state $S = (S_1, \dots, S_n)$ of M is an equilibrium iff for all $i = 1, \dots, n$,

$$S_i \in \mathbf{ACC}_i(kb_i \cup \{head(r) \mid r \in br_i \text{ is applicable in } S\}) \; .$$

Note: Interpretable as Nash-equilibrium of an n-player game

Example

Simple Genealogy II

$$\text{T-Box} = \left\{ \begin{array}{l} Person \equiv Female \sqcup Male, \\ Parent \equiv \exists hasChild.Person, \\ HasNoSons \equiv Parent \sqcap \forall hasChild.Female \end{array} \right\}$$

$$\text{A-Box} = \{\exists hasChild.(\exists hasChild.male \sqcap \exists hasChild.female)(Mary)\}$$

Females (x) who have brothers

$$q = F(x), hC(y,x), hC(y,z), M(z)$$