FAKULTÄT FÜR !NFORMATIK

Faculty of !nformatics

Towards a General Argumentation System Based on Answer-set Programming

Sarah Alice Gaggl

Institute of Informationsystems, Vienna University of Technology

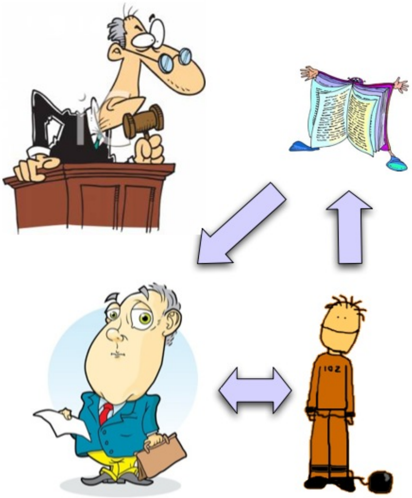
Edinburgh — July 21, 2010

FACULTY OF !NFORMATICS

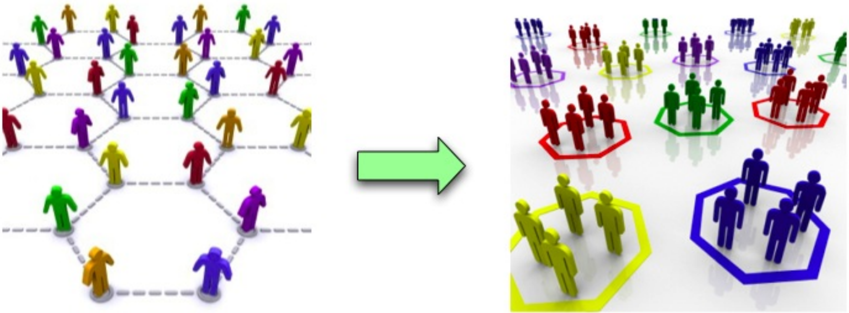
WWTF

Wiener Wissenschafts-, Forschungs- und Technologiefonds

Legal Reasoning



Social Networks



Decision Support

Argumentation Frameworks (AFs)

- AFs provide a formalism for a compact representation and evaluation of such scenarios.
- More complex semantics, especially in combination with an increasing amount of data, requires an automated computation of such solutions.
- Most of these problems are intractable, so implementing dedicated systems from the scratch is not the best idea.
- Logic Programming (LP), in particular Answer-set Programming (ASP), turned out to be adequate to solve problems associated to AFs.
- We use ASP to design the system ASPARTIX for the evaluation of several approaches how to deal with AFs.

Overview

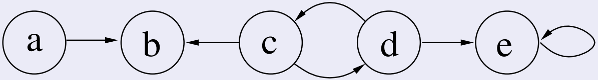
1. Introduction to Abstract Argumentation Frameworks
2. ASP Encoding
3. ASPARTIX - System Demonstration

- First introduced by Phan Minh Dung in 1995.
- AFs provide a formal way of dealing with conflicting knowledge.
- Represent arguments together with a binary attack relation.
- Conflicts are solved via semantics (admissible, preferred, stable).
- They can be represented as directed graphs.

More formally

An argumentation framework (AF) is a pair (A, R), where

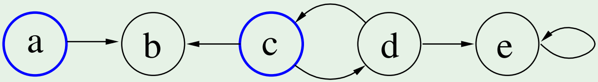
- A is a set of arguments
- $R \subseteq A \times A$ is a relation representing “attacks” (“defeats”)



Conflict-free

Let $F = (A, R)$ be an AF. A set $S \subseteq A$ is said to be conflict-free (in F), if there are no $a, b \in S$, such that $(a, b) \in R$. We denote the collection of sets which are conflict-free (in F) by $cf(F)$.

Example

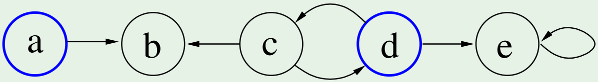


$cf(F) = \{\{a, c\},$

Conflict-free

Let $F = (A, R)$ be an AF. A set $S \subseteq A$ is said to be conflict-free (in F), if there are no $a, b \in S$, such that $(a, b) \in R$. We denote the collection of sets which are conflict-free (in F) by $cf(F)$.

Example

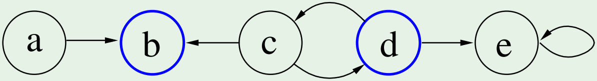


$cf(F) = \{\{a, c\}, \{a, d\},$

Conflict-free

Let $F = (A, R)$ be an AF. A set $S \subseteq A$ is said to be conflict-free (in F), if there are no $a, b \in S$, such that $(a, b) \in R$. We denote the collection of sets which are conflict-free (in F) by $cf(F)$.

Example



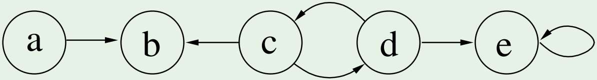
$cf(F) = \{\{a, c\}, \{a, d\}, \{b, d\},$

Semantics

Conflict-free

Let $F = (A, R)$ be an AF. A set $S \subseteq A$ is said to be conflict-free (in F), if there are no $a, b \in S$, such that $(a, b) \in R$. We denote the collection of sets which are conflict-free (in F) by $cf(F)$.

Example



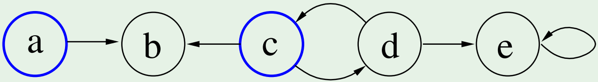
$$cf(F) = \{\{a, c\}, \{a, d\}, \{b, d\}, \{a\}, \{b\}, \{c\}, \{d\}, \emptyset\}$$

Stable Extension

Given an AF (A, R). A set $S \subseteq A$ is stable in F, if

- S is conflict-free in F
- for each $a \in A \setminus S$, there exists a $b \in S$, such that $(b, a) \in R$.

Example



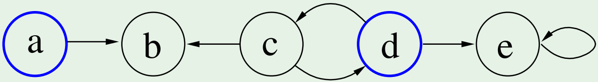
$stable(F) = \{\cancel{\{a, c\}},$

Stable Extension

Given an AF (A, R). A set $S \subseteq A$ is stable in F, if

- S is conflict-free in F
- for each $a \in A \setminus S$, there exists a $b \in S$, such that $(b, a) \in R$.

Example



$stable(F) = \{\cancel{\{a, c\}}, \{a, d\},$

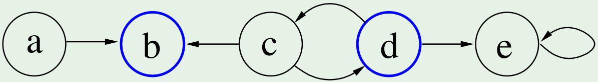
Semantics cont.

Stable Extension

Given an AF (A, R). A set $S \subseteq A$ is stable in F, if

- S is conflict-free in F
- for each $a \in A \setminus S$, there exists a $b \in S$, such that $(b, a) \in R$.

Example



$stable(F) = \{\cancel{\{a, c\}}, \{a, d\}, \cancel{\{b, d\}},$

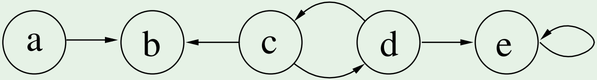
Semantics cont.

Stable Extension

Given an AF (A, R). A set $S \subseteq A$ is stable in F, if

- S is conflict-free in F
- for each $a \in A \setminus S$, there exists a $b \in S$, such that $(b, a) \in R$.

Example



$$\mathit{stable}(F) = \{\cancel{\{a, c\}}, \{a, d\}, \cancel{\{b, d\}}, \cancel{\{a\}}, \cancel{\{b\}}, \cancel{\{c\}}, \cancel{\{d\}}, \cancel{\emptyset}\}$$

Conflict-free Set

Given an AF (A, R).
A set $S \subseteq A$ is conflict-free in F, if, for each $a, b \in S$, $(a, b) \notin R$.

Encoding for $F = (A, R)$

$$\widehat{F} = \{\mathrm{arg}(a) \mid a \in A\} \cup \{\mathrm{att}(a, b) \mid (a, b) \in R\}$$

$$\pi_{cf} = \left\{ \begin{array}{lcl} \mathrm{in}(X) & \leftarrow & \mathit{not}\ \mathrm{out}(X), \mathrm{arg}(X) \\ \mathrm{out}(X) & \leftarrow & \mathit{not}\ \mathrm{in}(X), \mathrm{arg}(X) \\ & \leftarrow & \mathrm{in}(X), \mathrm{in}(Y), \mathrm{att}(X, Y) \end{array} \right\}$$

Result: For each AF F, $cf(F) \equiv \mathcal{AS}(\pi_{cf}(\widehat{F}))$

Stable Extension

Given an AF (A, R). A set $S \subseteq A$ is stable in F, if

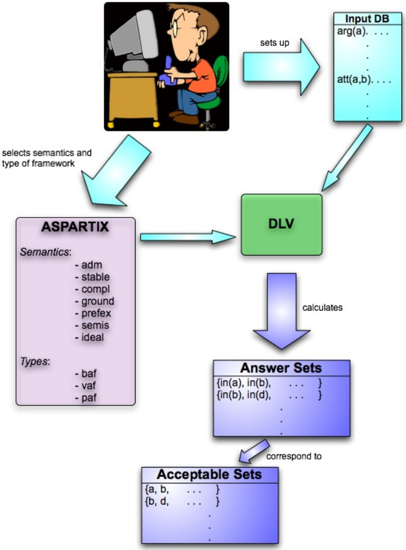
- S is conflict-free in F
- each $a \in A \setminus S$, there exists a $b \in S$, such that $(b, a) \in R$.

Encoding

$$\pi_{stable} = \left\{ \begin{array}{lcl} \mathrm{in}(X) & \leftarrow & \mathit{not}\ \mathrm{out}(X), \mathrm{arg}(X) \\ \mathrm{out}(X) & \leftarrow & \mathit{not}\ \mathrm{in}(X), \mathrm{arg}(X) \\ & \leftarrow & \mathrm{in}(X), \mathrm{in}(Y), \mathrm{att}(X, Y) \\ \mathrm{defeated}(X) & \leftarrow & \mathrm{in}(Y), \mathrm{att}(Y, X) \\ & \leftarrow & \mathrm{out}(X), \mathit{not}\ \mathrm{defeated}(X) \end{array} \right\}$$

Result: For each AF F, $\mathit{stable}(F) \equiv \mathcal{AS}(\pi_{stable}(\widehat{F}))$

ASPARTIX - System Description



ASPARTIX - System Description cont.

Semantics and types of AFs incorporated in ASPARTIX:

- admissible, complete, stable, preferred, grounded, ideal, stage, semi-stable and cf2;
- Preference-based AFs, Value-based AFs, Bipolar AFs, Dynamic AFs and AFs with Recursive Attacks.

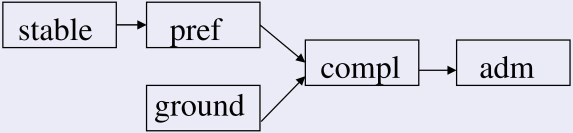
Conclusion

- AFs became very important in Artificial Intelligence. They provide a popular tool for modeling and evaluating conflicting knowledge.
- Problems associated to AFs are in general intractable, therefore we translate them to ASP.
- Web front-end of ASPARTIX is freely available.

AsPartiX

`http://rull.dbai.tuwien.ac.at:8080/ASPARTIX`

Relation between Semantics



Complexity

	stable	*adm*	*pref*	*comp*	*ground*
Cred	NP-c	NP-c	NP-c	NP-c	in P
Skept	coNP-c	(trivial)	Π_2^P-c	in P	in P

[Dimopoulos & Torres 96; Dunne & Bench-Capon 02; Coste-Marquis *et al.* 05]

Complexity cont.

Complexity of Argumentation

	stable	*adm*	*pref*	*comp*	*ground*
Cred	NP-c	NP-c	NP-c	NP-c	in P
Skept	coNP-c	(trivial)	Π_2^P-c	in P	in P

Recall: Data-Complexity of Datalog

	stratified programs	with negation	with neg. and disjunction
$\models_c$	P-c	NP-c	Σ_2^P-c
$\models_s$	P-c	coNP-c	Π_2^P-c

[Dantsin,Eiter,Gottlob,Voronkov 01]

Performance Tests

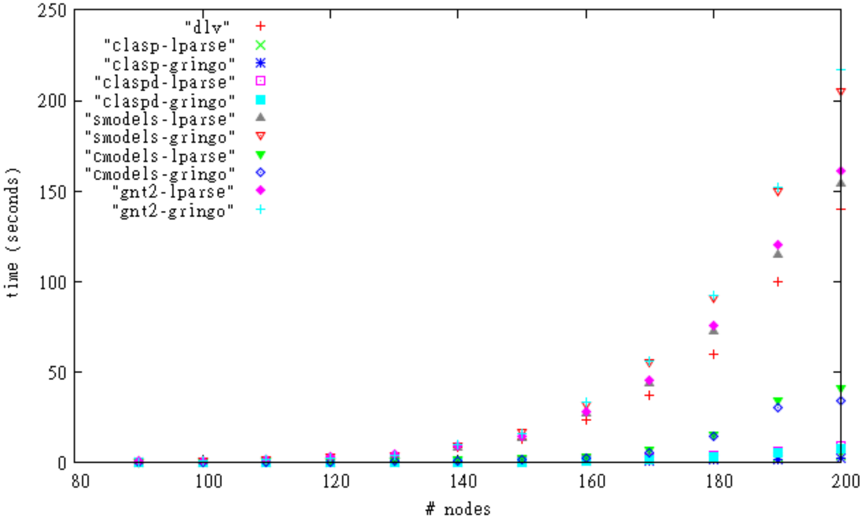
Tested Systems:

- Grounders: DLV, lparse, GrinGo
- Solvers: DLV, smodels, cmodels, clasp, claspD, gnt

Testing:

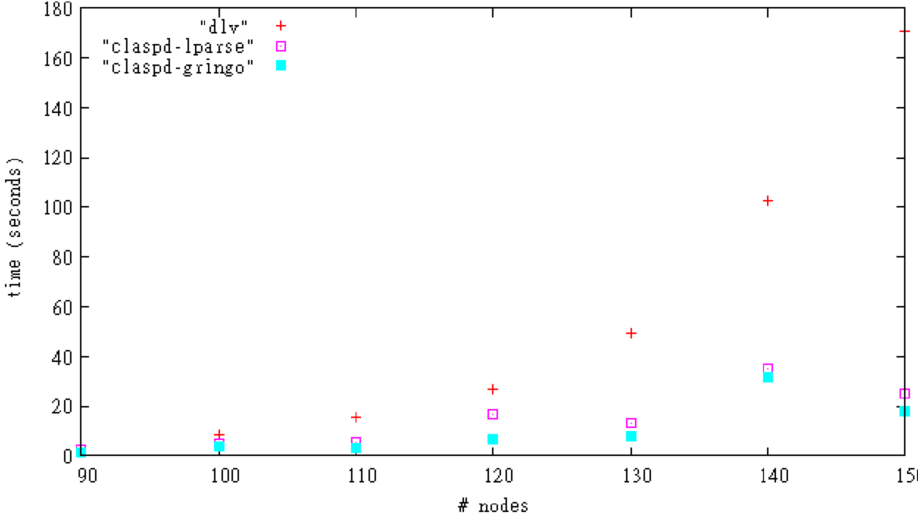
- Randomly generated AFs from 90 to 200 arguments with edge density from 10% to 30%.
- In total 21303 tests were performed.

Admissible Extensions:

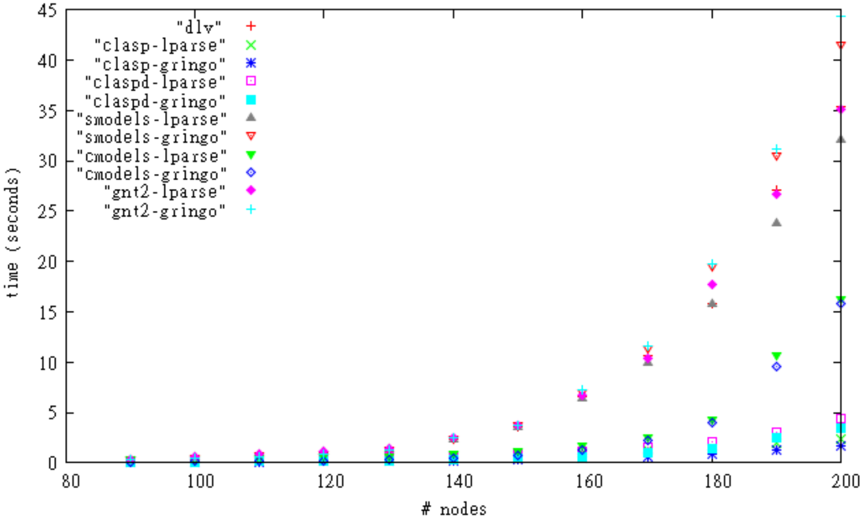


Test Results cont.

Preferred Extensions:



Stable Extensions:



FACULTY OF !NFORMATICS