# Data-Driven Logical Reasoning

Claudia d'Amato Volha Bryl, Luciano Serafini

November 11, 2012

$8^{th}$ International Workshop on Uncertainty Reasoning for the Semantic Web

$11^{th}$ ISWC, Boston (MA), USA.

# Heterogeneous Data

Heterogeneous resources of knowledge about the same domain:

**Non or simply structured data** (e.g., sensor data, signals, DB tracks, texts, bag of words, etc) containing (alpha)numeric feature based data

**Structured knowledge** (e.g., ontologies = T-box + A-box) describing the entity types, the relations and the objects of a particular domain.

# Heterogeneous Data

Heterogeneous resources of knowledge about the same domain:

**Non or simply structured data** (e.g., sensor data, signals, DB tracks, texts, bag of words, etc) containing (alpha)numeric feature based data

For instance a database of a company containing anagraphic data, salaries, evaluations, performances, and other relevant information about the employees.

**Structured knowledge** (e.g., ontologies = T-box + A-box) describing the entity types, the relations and the objects of a particular domain.

For instance the ontology for the organizational structure of a company describing, the roles, the activities, the responsibilities, etc. and the instantiation to the set of the company employees.

# Heterogeneous Inferences

Different type of data support different type of inference

**Non or simply structured data** enables the discovering of new knowledge as regularity patterns of data via **inductive reasoning**.

**Structured knowledge** allows to discover new knowledge via inferencing of classes or properties of objects based on logical **deductive reasoning**.

# Heterogeneous Inferences

Different type of data support different type of inference

**Non or simply structured data** enables the discovering of new knowledge as regularity patterns of data via **inductive reasoning**.

Usually older employees earn more than their younger colleagues

**Structured knowledge** allows to discover new knowledge via inferencing of classes or properties of objects based on logical **deductive reasoning**.

If a person is a project leader than he or she coordinates the work of all the people allocated to the project

# Combining heterogeneous Inferences

**Non or simply structured data** + **Structured knowledge** will enable the combination of knowledge induced from data and knowledge encoded in an ontology in a new form of mixed reasoning, that we call **data drive inference**.

Usually older employees earn more than their younger colleagues

+

If a person is a project leader than he or she coordinates the work of all the people allocated to the project

# Combining heterogeneous Inferences

**Non or simply structured data** + **Structured knowledge** will enable the combination of knowledge induced from data and knowledge encoded in an ontology in a new form of mixed reasoning, that we call **data drive inference**.

Usually older employees earn more than their younger colleagues

+

If a person is a project leader than he or she coordinates the work of all the people allocated to the project

=

Since project coordinators receive always the highest salary within a project team, the oldest person of project team, is most probably the project coordinator

# objectives

## Abstract objectives

1. To define a reference framework capable to represent quantitative data and logical knowledge in an integrated way
2. extend machine learning algorithms to support the induction of new knowledge from quantitative data integrated with logical knowledge
3. extend the logical reasoning algorithms to support logical inference in presence of knowledge induced from logical data integrated with logical knowledge

# objectives

## Concrete contributions (EKAW 2012)

1. define the notion of ontology integrated with a dataset, called grounded ontologies
2. introduce semantically enriched association rules to represent knowledge induced from grounded ontologies
3. define a simple algorithm for learning semantically enriched association rules from grounded ontologies.

# objectives

## Concrete contributions (EKAW 2012)

1. define the notion of ontology integrated with a dataset, called grounded ontologies
2. introduce semantically enriched association rules to represent knowledge induced from grounded ontologies
3. define a simple algorithm for learning semantically enriched association rules from grounded ontologies.

## Concrete contributions (URSW 2012)

4. define a new reasoning task, called the most plausible model that computes the most common model of an ontology w.r.t., a set of semantically enriched association rules
5. propose a first approximated tableaux algorithm to compute the most plausible model

# Grounded Ontology

- Dataset $\mathbf{D}$ is a non empty set of objects
  - $f_1, \ldots, f_n$ are $n$ feature functions defined on every element of $\mathbf{D}$, with $f_i : \mathbf{D} \to D_i$.

# Grounded Ontology

- Dataset **D** is a non empty set of objects
  - $f_1, \ldots, f_n$ are $n$ feature functions defined on every element of **D**, with $f_i : \mathbf{D} \to D_i$.

- Knowledge base $\mathcal{K}$ on an alphabet $\Sigma$, composed of three disjoint ets of symbols, $\Sigma_C$, $\Sigma_R$ and $\Sigma_I$, is a set $\mathcal{K}$ of DL inclusion axioms and DL assertions.

# Grounded Ontology

- Dataset $\mathbf{D}$ is a non empty set of objects
  - $f_1, \ldots, f_n$ are $n$ feature functions defined on every element of $\mathbf{D}$, with $f_i : \mathbf{D} \to D_i$.

- Knowledge base $\mathcal{K}$ on an alphabet $\Sigma$, composed of three disjoint ets of symbols, $\Sigma_C$, $\Sigma_R$ and $\Sigma_I$, is a set $\mathcal{K}$ of DL inclusion axioms and DL assertions.

- A grounding is a total function

$$g : \mathbf{D} \to \Sigma_I$$

# Grounded Ontology

- Dataset $\mathbf{D}$ is a non empty set of objects
  - $f_1, \ldots, f_n$ are $n$ feature functions defined on every element of $\mathbf{D}$, with $f_i : \mathbf{D} \to D_i$.

- Knowledge base $\mathcal{K}$ on an alphabet $\Sigma$, composed of three disjoint ets of symbols, $\Sigma_C$, $\Sigma_R$ and $\Sigma_I$, is a set $\mathcal{K}$ of DL inclusion axioms and DL assertions.

- A grounding is a total function

$$g : \mathbf{D} \to \Sigma_I$$

- for every $\mathbf{d} \in \mathbf{D}$ there is a constant $a \in \Sigma_I$ with $g(\mathbf{d}) = a$.

# Grounded Ontology

- Dataset $\mathbf{D}$ is a non empty set of objects
  - $f_1, \ldots, f_n$ are $n$ feature functions defined on every element of $\mathbf{D}$, with $f_i : \mathbf{D} \to D_i$.

- Knowledge base $\mathcal{K}$ on an alphabet $\Sigma$, composed of three disjoint ets of symbols, $\Sigma_C$, $\Sigma_R$ and $\Sigma_I$, is a set $\mathcal{K}$ of DL inclusion axioms and DL assertions.

- A grounding is a total function

$$g : \mathbf{D} \to \Sigma_I$$

- for every $\mathbf{d} \in \mathbf{D}$ there is a constant $a \in \Sigma_I$ with $g(\mathbf{d}) = a$.
- intuitively $g(\mathbf{d}) = a$ represents the fact that data $\mathbf{d}$ is an observation about the object $a$ of the knowledge base;

# Grounded Ontology

- Dataset $\mathbf{D}$ is a non empty set of objects
  - $f_1, \ldots, f_n$ are $n$ feature functions defined on every element of $\mathbf{D}$, with $f_i : \mathbf{D} \to D_i$.

- Knowledge base $\mathcal{K}$ on an alphabet $\Sigma$, composed of three disjoint ets of symbols, $\Sigma_C$, $\Sigma_R$ and $\Sigma_I$, is a set $\mathcal{K}$ of DL inclusion axioms and DL assertions.

- A grounding is a total function

$$g : \mathbf{D} \to \Sigma_I$$

- for every $\mathbf{d} \in \mathbf{D}$ there is a constant $a \in \Sigma_I$ with $g(\mathbf{d}) = a$.
- intuitively $g(\mathbf{d}) = a$ represents the fact that data $\mathbf{d}$ is an observation about the object $a$ of the knowledge base;
- there can be more than one observation for $a$ in $\mathbf{D}$. This implies that $g(\mathbf{d}) = a$ and $g(\mathbf{d}') = a$ for $\mathbf{d} \neq \mathbf{d}'$

# Grounded Ontology - Example

| ID | E_ID | salary | year |
|---|---|---|---|
| 001 | E01 | 30,000 | 2010 |
| 002 | E01 | 35,000 | 2011 |
| 003 | E01 | 40,000 | 2012 |
| 004 | E02 | 28,000 | 2011 |
| 005 | E02 | 33,000 | 2012 |
| 006 | E03 | 24,000 | 2012 |
| 007 | E04 | 25,000 | 2011 |
| 008 | E04 | 25,000 | 2012 |
| 009 | E05 | 40,000 | 2012 |

# Grounded Ontology - Example

| ID | E_ID | salary | year |
|---|---|---|---|
| 001 | E01 | 30,000 | 2010 |
| 002 | E01 | 35,000 | 2011 |
| 003 | E01 | 40,000 | 2012 |
| 004 | E02 | 28,000 | 2011 |
| 005 | E02 | 33,000 | 2012 |
| 006 | E03 | 24,000 | 2012 |
| 007 | E04 | 25,000 | 2011 |
| 008 | E04 | 25,000 | 2012 |
| 009 | E05 | 40,000 | 2012 |

$\top \sqsubseteq \mathit{Person} \sqcup \mathit{Prj}$

$\mathit{Person} \sqsubseteq \exists \mathit{leads}.\mathit{Prj} \sqcup \exists \mathit{worksFor}.\mathit{Prj}$

$\mathit{Prj} \sqsubseteq (=1)\mathit{leads}^{-} \qquad \mathit{Prj} \sqsubseteq \exists \mathit{worksFor}^{-}$

$\mathit{dom}(\mathit{worksFor}) \sqsubseteq \mathit{Person} \qquad \mathit{dom}(\mathit{leads}) \sqsubseteq \mathit{Person}$

---

$\mathit{leads}(\mathit{Alice}, P) \quad \mathit{worksFor}(\mathit{Bob}, P) \quad \mathit{worksFor}(\mathit{Chris}, P)$

$\mathit{leads}(\mathit{Bob}, Q) \quad \mathit{worksFor}(\mathit{Chris}, Q) \quad \mathit{worksFor}(\mathit{Dan}, Q)$

$\mathit{leads}(\mathit{Alice}, R) \quad \mathit{worksFor}(\mathit{Dan}, R)$

$\mathit{allDifferent}(\mathit{Alice}, \mathit{Bob}, \mathit{Chris}, \mathit{Dan})$

$\top(\mathit{Eva}), \quad \forall \mathit{leads}\bot(\mathit{Chris}) \quad \forall \mathit{leads}\bot(\mathit{Dan})$

# Grounded Ontology - Example

| ID | E_ID | salary | year |
|---|---|---|---|
| 001 | E01 | 30,000 | 2010 |
| 002 | E01 | 35,000 | 2011 |
| 003 | E01 | 40,000 | 2012 |
| 004 | E02 | 28,000 | 2011 |
| 005 | E02 | 33,000 | 2012 |
| 006 | E03 | 24,000 | 2012 |
| 007 | E04 | 25,000 | 2011 |
| 008 | E04 | 25,000 | 2012 |
| 009 | E05 | 40,000 | 2012 |

$\top \sqsubseteq Person \sqcup Prj$
$Person \sqsubseteq \exists leads.Prj \sqcup \exists worksFor.Prj$
$Prj \sqsubseteq (= 1) leads^{-}$  $Prj \sqsubseteq \exists worksFor^{-}$
$dom(worksFor) \sqsubseteq Person$  $dom(leads) \sqsubseteq Person$

$leads(Alice, P)$  $worksFor(Bob, P)$  $worksFor(Chris, P)$
$leads(Bob, Q)$  $worksFor(Chris, Q)$  $worksFor(Dan, Q)$
$leads(Alice, R)$  $worksFor(Dan, R)$
$allDifferent(Alice, Bob, Chris, Dan)$
$\top(Eva)$,  $\forall leads \bot(Chris)$  $\forall leads \bot(Dan)$

| | g |
|---|---|
| 001 | Alice |
| 002 | Alice |
| 003 | Alice |
| 004 | Bob |
| 005 | Bob |
| 006 | Chris |
| 007 | Dan |
| 008 | Dan |
| 009 | Eva |

# Grounding allow to join data with knowledge

Let $(\mathcal{K}, \mathbf{D}, g)$ be a grounded ontology, the semantic enrichment of $\mathbf{D}$ w.r.t., $\mathcal{K}$, denoted by $\mathbf{D}^{+\mathcal{K}}$ is dataset containing the same data as $\mathbf{D}$ with the following additional semantic attributes:

# Grounding allow to join data with knowledge

Let $(\mathcal{K}, \mathbf{D}, g)$ be a grounded ontology, the semantic enrichment of $\mathbf{D}$ w.r.t., $\mathcal{K}$, denoted by $\mathbf{D}^{+\mathcal{K}}$ is dataset containing the same data as $\mathbf{D}$ with the following additional semantic attributes:

**Conceptual attributes** for every primitive concept $C \in \Sigma_c$ the attribute $f_C$ is defined as follows:

$$f_C(\mathbf{d}) = \begin{cases} 1 & \text{if } \mathcal{K} \models C(g(\mathbf{d})) \\ 0 & \text{if } \mathcal{K} \models \neg C(g(\mathbf{d})) \\ \mathit{unknown} & \text{otherwise} \end{cases}$$

# Grounding allow to join data with knowledge

Let $(\mathcal{K}, \mathbf{D}, g)$ be a grounded ontology, the semantic enrichment of $\mathbf{D}$ w.r.t., $\mathcal{K}$, denoted by $\mathbf{D}^{+\mathcal{K}}$ is dataset containing the same data as $\mathbf{D}$ with the following additional semantic attributes:

**Conceptual attributes** for every primitive concept $C \in \Sigma_c$ the attribute $f_C$ is defined as follows:

$$f_C(\mathbf{d}) = \begin{cases} 1 & \text{if } \mathcal{K} \models C(g(\mathbf{d})) \\ 0 & \text{if } \mathcal{K} \models \neg C(g(\mathbf{d})) \\ \mathit{unknown} & \text{otherwise} \end{cases}$$

**Relational attributes** for every relation $R \in \Sigma_r$ the attribute $f_R$ is defined as follows:

$$f_R(\mathbf{d}) = \begin{cases} 1 & \text{if } \mathcal{K} \models \exists R(g(\mathbf{d})) \\ 0 & \text{if } \mathcal{K} \models \neg\exists R(g(\mathbf{d})) \\ \mathit{unknown} & \text{otherwise} \end{cases}$$

# Join data with knowledge - example

| ID | E_ID | salary | year | Person | Prj | leads | worksFor |
|---|---|---|---|---|---|---|---|
| 001 | E01 | 30,000 | 2010 | 1 | 0 | 1 | - |
| 002 | E01 | 35,000 | 2011 | 1 | 0 | 1 | - |
| 003 | E01 | 40,000 | 2012 | 1 | 0 | 1 | - |
| 004 | E02 | 28,000 | 2011 | 1 | 0 | 1 | 1 |
| 005 | E02 | 32,000 | 2012 | 1 | 0 | 1 | 1 |
| 006 | E03 | 24,000 | 2012 | 1 | 0 | 0 | 1 |
| 007 | E04 | 25,000 | 2011 | 1 | 0 | 0 | 1 |
| 008 | E04 | 25,000 | 2012 | 1 | 0 | 0 | 1 |
| 009 | E05 | 40,000 | 2012 | - | - | - | - |

# Association Rules

## Definition

Given a dataset **D** made by a set of attributes $\{A_1, \ldots, \mathcal{A}_n\}$

- an itemset of **D** is an expression

$$f_{i_1} = v_1 \wedge \cdots \wedge f_{i_k} = v_k \tag{1}$$

- the support of an itemset is the number of tuples (rows) in **D** that match the itemset.
- Association rule is an expression

$$f_{i_1} = v_1 \wedge \cdots \wedge f_{i_k} = v_k \Rightarrow f_{i_k+1} = v_{k+1} \wedge \cdots \wedge f_{i_n} = v_n \tag{2}$$

- The confidence of (2) is the fraction of cases in **D** that match the conclusion amongs the one that matches the premises.

$$conf(\theta \Rightarrow \varphi) = \frac{support(\theta \wedge \varphi)}{support(\theta)}$$

# Semantically Enriched Association Rules (SEAR)

## Semantically enriched association rules

Let $(\mathcal{K}, \mathbf{D}, g)$ be a grounded ontology A semantically enriched association rule is an association rule defined on the semantically enriched dataset $\mathbf{D}^{+\mathcal{K}}$.

# Learning association rules

- Association rules are learned by
  - finding the frequent itemsets w.r.t. a given support threshold,
  - extracting the rules from the frequent itemsets satisfying a given confidence threshold.
- The first subproblem is the most challenging/expensive
- We use the standard Apriori algorithm
  - key assumption: a set of variables is frequent only if all its subsets are frequent,
  - itemsets are built iteratively, incrementing the length at each step.

# Learing SEAR: example

| ID | E_ID | salary | year | Person | Prj | leads | worksFor |
|---|---|---|---|---|---|---|---|
| 001 | E01 | 30–40K | 2010 | 1 | 0 | 1 | - |
| 002 | E01 | 30–40K | 2011 | 1 | 0 | 1 | - |
| 003 | E01 | 30–40K | 2012 | 1 | 0 | 1 | - |
| 004 | E02 | 20–30K | 2011 | 1 | 0 | 1 | 1 |
| 005 | E02 | 30–40K | 2012 | 1 | 0 | 1 | 1 |
| 006 | E03 | 20–30K | 2012 | 1 | 0 | 0 | 1 |
| 007 | E04 | 20–30K | 2011 | 1 | 0 | 0 | 1 |
| 008 | E04 | 20–30K | 2012 | 1 | 0 | 0 | 1 |
| 009 | E05 | 30–40K | 2012 | - | - | 0 | - |

## Examples of SEARs

$$
\begin{array}{rcll}
f_{leads} = 1, f_{year} = 2011 & \Rightarrow & f_{Salary} = 30\text{–}40K & conf = 4/4 = 1.00 \\
 & \Rightarrow & f_{Person} = 1 & conf = 8/9 = 0.88 \\
 & \Rightarrow & f_{Project} = 0 & conf = 8/9 = 0.88 \\
f_{Salary} = 30\text{–}40K & \Rightarrow & f_{leads} = 1 & conf = 4/5 = 0.80 \\
f_{leads} = 1 & \Rightarrow & f_{Salary} = 30\text{–}40K & conf = 4/5 = 0.80 \\
f_{Salary} = 20\text{–}30K & \Rightarrow & f_{leads} = 0 & conf = 3/4 = 0.75 \\
f_{year} = 2012, \; f_{Salary} = 30\text{–}40K & \Rightarrow & f_{leads} = 1 & conf = 2/3 = 0.66
\end{array}
$$

# objectives

## Concrete contributions

1. defined the notion of ontology grounded on a set of data, also called grounded ontologies
2. introduced semantically enriched association rules to represent knowledge induced from grounded ontologies
3. defined a simple algorithm for learning semantically enriched association rules from grounded ontologies.

# objectives

## Concrete contributions

1. defined the notion of ontology grounded on a set of data, also called grounded ontologies
2. introduced semantically enriched association rules to represent knowledge induced from grounded ontologies
3. defined a simple algorithm for learning semantically enriched association rules from grounded ontologies.

## Concrete contributions

4. defined a new reasoning task, called the most plausible model that computes the most common model of an ontology w.r.t., a set of semantically enriched association rules
5. proposed a first approximated tableaux algorithm to compute the most plausible model

# The reasoning task

**Definition (Inference Problem)**

Given: $\mathbf{D}$, $\mathcal{K}$, the set $R$ of SEARs, an data item $\mathbf{d}_0$, a concept $C_0$, and the grounding $g(\mathbf{d}_0) = x_0$,

Determine: the model $\mathcal{I}_r$ for $\mathcal{K} \cup \{C_0(x_0)\}$ representing the most plausible model for $\mathcal{K} \cup \{C_0(x_0)\}$ w.r.t., $g$ and $R$.

# Plausibility ordering

## Definition (Plausibility ordering (first attempt))

Let $(\mathcal{K}, \{\mathbf{d}\}, g)$ be a grounded ontology:

$$
\begin{array}{rcl}
\mathcal{I}, \mathbf{d}, g \models f_i = a & \text{if} & f_i(\mathbf{d}) = a \\
\mathcal{I}, \mathbf{d}, g \models f_C = 1 & \text{if} & g(\mathbf{d})^{\mathcal{I}} \in C^{\mathcal{I}} \\
\mathcal{I}, \mathbf{d}, g \models f_C = 0 & \text{if} & g(\mathbf{d})^{\mathcal{I}} \notin C^{\mathcal{I}} \\
\mathcal{I}, \mathbf{d}, g \models f_R = 1 & \text{if} & g(\mathbf{d})^{\mathcal{I}} \in (\exists R.\top)^{\mathcal{I}} \\
\mathcal{I}, \mathbf{d}, g \models f_R = 0 & \text{if} & g(\mathbf{d})^{\mathcal{I}} \notin (\exists R.\top)^{\mathcal{I}} \\
\mathcal{I}, \mathbf{d}, g \models \phi_1 \wedge \cdots \wedge \phi_n & \text{if} & \mathcal{I}, \mathbf{d}, g \models \phi_i \text{ for } 1 \leq i \leq n \\
\mathcal{I}, \mathbf{d}, g \models \phi \Rightarrow \psi & \text{if} & \mathcal{I}, \mathbf{d}, g \not\models \phi \text{ or } \mathcal{I}, \mathbf{d}, g \models \psi
\end{array}
$$

# Plausibility ordering

## Definition (Plausibility ordering (first attempt))

Let $(\mathcal{K}, \{\mathbf{d}\}, g)$ be a grounded ontology:

$$
\begin{array}{rcl}
\mathcal{I}, \mathbf{d}, g \models f_i = a & \text{if} & f_i(\mathbf{d}) = a \\
\mathcal{I}, \mathbf{d}, g \models f_C = 1 & \text{if} & g(\mathbf{d})^{\mathcal{I}} \in C^{\mathcal{I}} \\
\mathcal{I}, \mathbf{d}, g \models f_C = 0 & \text{if} & g(\mathbf{d})^{\mathcal{I}} \notin C^{\mathcal{I}} \\
\mathcal{I}, \mathbf{d}, g \models f_R = 1 & \text{if} & g(\mathbf{d})^{\mathcal{I}} \in (\exists R.\top)^{\mathcal{I}} \\
\mathcal{I}, \mathbf{d}, g \models f_R = 0 & \text{if} & g(\mathbf{d})^{\mathcal{I}} \notin (\exists R.\top)^{\mathcal{I}} \\
\mathcal{I}, \mathbf{d}, g \models \phi_1 \wedge \cdots \wedge \phi_n & \text{if} & \mathcal{I}, \mathbf{d}, g \models \phi_i \text{ for } 1 \leq i \leq n \\
\mathcal{I}, \mathbf{d}, g \models \phi \Rightarrow \psi & \text{if} & \mathcal{I}, \mathbf{d}, g \not\models \phi \text{ or } \mathcal{I}, \mathbf{d}, g \models \psi
\end{array}
$$

$$
p(\mathcal{I}, \mathbf{d}, g, \alpha) = \begin{cases} 0 & \text{If } \mathcal{I}, \mathbf{d}, g \models \alpha \\ 2 * conf(\alpha) - 1 & \text{Otherwise} \end{cases}
$$

# Plausibility ordering

## Definition (Plausibility ordering (first attempt))

Let $(\mathcal{K}, \{\mathbf{d}\}, g)$ be a grounded ontology:

$$\begin{array}{rcl}
\mathcal{I}, \mathbf{d}, g \models f_i = a & \text{if} & f_i(\mathbf{d}) = a \\
\mathcal{I}, \mathbf{d}, g \models f_C = 1 & \text{if} & g(\mathbf{d})^{\mathcal{I}} \in C^{\mathcal{I}} \\
\mathcal{I}, \mathbf{d}, g \models f_C = 0 & \text{if} & g(\mathbf{d})^{\mathcal{I}} \notin C^{\mathcal{I}} \\
\mathcal{I}, \mathbf{d}, g \models f_R = 1 & \text{if} & g(\mathbf{d})^{\mathcal{I}} \in (\exists R.\top)^{\mathcal{I}} \\
\mathcal{I}, \mathbf{d}, g \models f_R = 0 & \text{if} & g(\mathbf{d})^{\mathcal{I}} \notin (\exists R.\top)^{\mathcal{I}} \\
\mathcal{I}, \mathbf{d}, g \models \phi_1 \wedge \cdots \wedge \phi_n & \text{if} & \mathcal{I}, \mathbf{d}, g \models \phi_i \text{ for } 1 \leq i \leq n \\
\mathcal{I}, \mathbf{d}, g \models \phi \Rightarrow \psi & \text{if} & \mathcal{I}, \mathbf{d}, g \not\models \phi \text{ or } \mathcal{I}, \mathbf{d}, g \models \psi
\end{array}$$

$$p(\mathcal{I}, \mathbf{d}, g, \alpha) = \begin{cases} 0 & \text{If } \mathcal{I}, \mathbf{d}, g \models \alpha \\ 2 * conf(\alpha) - 1 & \text{Otherwise} \end{cases}$$

$$\mathcal{I} \preceq \mathcal{J} \, iff \sum_{\alpha \in AR} p(\mathcal{I}, \mathbf{d}, g, \alpha) \geq \sum_{\alpha \in AR} p(\mathcal{J}, \mathbf{d}, g, \alpha)$$

# $\mathcal{ALC}$ Tableaux

## Tableaux

Let $C_0$ be an $\mathcal{ALC}$-concept in NNF. In order to test satisfiability of $C_0$, the ALC tableaux algorithm starts with $\mathcal{A}_0 := \{C_0(x_0)\}$, and applies the following rules:

| **Rule** | **Condition** | $\longrightarrow$ | **Effect** |
|---|---|---|---|
| $\rightarrow_\sqcap$ | $C \sqcap D(x) \in \mathcal{A}$ | $\longrightarrow$ | $\mathcal{A} := \mathcal{A} \cup \{C(x), D(x)\}$ |
| $\rightarrow_\sqcup$ | $C \sqcup D(x) \in \mathcal{A}$ | $\longrightarrow$ | $\mathcal{A} := \mathcal{A} \cup \{C(x)\}$ or $\mathcal{A} \cup \{D(x)\}$ |
| $\rightarrow_\exists$ | $\exists R.C(x) \in \mathcal{A}$ | $\longrightarrow$ | $\mathcal{A} := \mathcal{A} \cup \{R(x,y), C(y)\}$ |
| $\rightarrow_\forall$ | $\forall R.C(x), R(x,y) \in \mathcal{A}$ | $\longrightarrow$ | $\mathcal{A} := \mathcal{A} \cup \{C(y)\}$ |
| ... | ... | $\longrightarrow$ | ... |

# $\mathcal{ALC}$ Tableaux

## Tableaux

Let $C_0$ be an $\mathcal{ALC}$-concept in NNF. In order to test satisfiability of $C_0$, the ALC tableaux algorithm starts with $\mathcal{A}_0 := \{C_0(x_0)\}$, and applies the following rules:

| **Rule** | **Condition** | $\longrightarrow$ | **Effect** |
|---|---|---|---|
| $\rightarrow_\sqcap$ | $C \sqcap D(x) \in \mathcal{A}$ | $\longrightarrow$ | $\mathcal{A} := \mathcal{A} \cup \{C(x), D(x)\}$ |
| $\rightarrow_\sqcup$ | $C \sqcup D(x) \in \mathcal{A}$ | $\longrightarrow$ | $\mathcal{A} := \mathcal{A} \cup \{C(x)\}$ or $\mathcal{A} \cup \{D(x)\}$ |
| $\rightarrow_\exists$ | $\exists R.C(x) \in \mathcal{A}$ | $\longrightarrow$ | $\mathcal{A} := \mathcal{A} \cup \{R(x,y), C(y)\}$ |
| $\rightarrow_\forall$ | $\forall R.C(x), R(x,y) \in \mathcal{A}$ | $\longrightarrow$ | $\mathcal{A} := \mathcal{A} \cup \{C(y)\}$ |
| $\dots$ | $\dots$ | $\longrightarrow$ | $\dots$ |

## Indeterministic Choices

In applying $\rightarrow_\sqcup$ rule we have to do a choice, either we expand $\mathcal{A}$ with $C(x)$ or with $D(x)$. Depending on the choice we can generate a more or less plausible model.

# $\sqcup$-rule - a simple example

## Example (Simple)

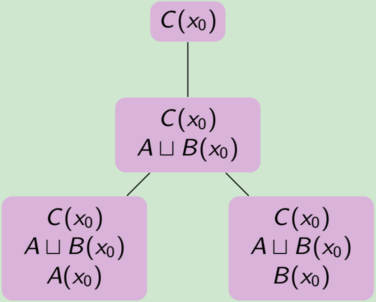
$C \sqsubseteq A \sqcup B$

$f_{Salary}(\mathbf{d}) = 27$

$f_{Salary} = 10\text{–}20 \Rightarrow A \quad conf = 0.77$

$f_{Salary} = 20\text{–}30 \Rightarrow B \quad conf = 0.66$

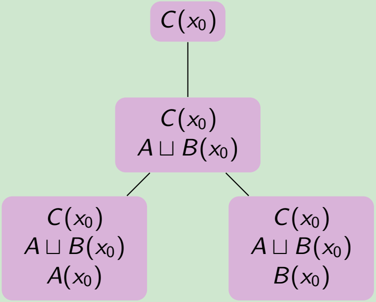
$g(\mathbf{d}) = x_0$



# $\sqcup$-rule - a simple example

## Example (Simple)

$C \sqsubseteq A \sqcup B$
$f_{Salary}(\mathbf{d}) = 27$
$f_{Salary} = 10\text{–}20 \Rightarrow A \quad conf = 0.77$
$f_{Salary} = 20\text{–}30 \Rightarrow B \quad conf = 0.66$
$g(\mathbf{d}) = x_0$




There are three possible models:

| | $\Delta^{\mathcal{I}}$ | $A^{\mathcal{I}}$ | $B^{\mathcal{I}}$ | $C^{\mathcal{I}}$ | $\sum_{\alpha \in AR} p(\mathcal{I}, \mathbf{d}, g, \alpha)$ |
|---|---|---|---|---|---|
| $\mathcal{I}_1$ | $\{x_0\}$ | $\{x_0\}$ | $\emptyset$ | $\{x_0\}$ | 0.32 |
| $\mathcal{I}_2$ | $\{x_0\}$ | $\emptyset$ | $\{x_0\}$ | $\{x_0\}$ | 0.00 |
| $\mathcal{I}_3$ | $\{x_0\}$ | $\{x_0\}$ | $\{x_0\}$ | $\{x_0\}$ | 0.00 |

The plausibility ordering is $\mathcal{I}_2 \prec \mathcal{I}_1$, $\mathcal{I}_3 \prec \mathcal{I}_1$

# A complete simple example

## Example (The most plausible model for Eva)

| <u>**ID**</u> | **E_ID** | **salary** | **year** | **Person** | **Prj** | **leads** | **worksFor** |
|---|---|---|---|---|---|---|---|
| <u>009</u> | E05 | 30–40K | 2012 | - | - | - | - |

$$\top(Eva)$$

# A complete simple example

## Example (The most plausible model for Eva)

| ID | E_ID | salary | year | Person | Prj | leads | worksFor |
|---|---|---|---|---|---|---|---|
| 009 | E05 | 30–40K | 2012 | - | - | - | - |

$$\top(Eva)$$

$\top \sqsubseteq Person \sqcup Prj$

$$Person \sqcup Prj(Eva)$$

# A complete simple example

## Example (The most plausible model for Eva)

| ID | E_ID | salary | year | Person | Prj | leads | worksFor |
|---|---|---|---|---|---|---|---|
| 009 | E05 | 30–40K | 2012 | - | - | - | - |

$$\top \sqsubseteq Person \sqcup Prj$$
$$\Rightarrow f_{Person} = 1[0.88]$$

$$\top(Eva)$$
$$Person \sqcup Prj(Eva)$$
$$Person(Eva)$$

# A complete simple example

## Example (The most plausible model for Eva)

| ID | E_ID | salary | year | Person | Prj | leads | worksFor |
|---|---|---|---|---|---|---|---|
| 009 | E05 | 30–40K | 2012 | - | - | - | - |

$$\top(Eva)$$

$\top \sqsubseteq Person \sqcup Prj$ $\qquad Person \sqcup Prj(Eva)$

$\Rightarrow f_{Person} = 1[0.88]$ $\qquad Person(Eva)$

$Person \sqsubseteq \exists worksFor.Prj \sqcup \exists leads.Prj$ $\qquad \exists worksFor.Prj \sqcup \exists leads.Prj$

# A complete simple example

## Example (The most plausible model for Eva)

| ID | E_ID | salary | year | Person | Prj | leads | worksFor |
|---|---|---|---|---|---|---|---|
| 009 | E05 | 30–40K | 2012 | - | - | - | - |

$$\top(Eva)$$

$\top \sqsubseteq Person \sqcup Prj$ $\qquad$ $Person \sqcup Prj(Eva)$

$\Rightarrow f_{Person} = 1[0.88]$ $\qquad$ $Person(Eva)$

$Person \sqsubseteq \exists worksFor.Prj \sqcup \exists leads.Prj$ $\qquad$ $\exists worksFor.Prj \sqcup \exists leads.Prj$

$f_{Salary}(Eva) = 30\text{–}40$

$f_{Salary} = 30\text{–}40 \Rightarrow f_{leads} = 1$ $\qquad$ $\exists leads.Prj(eva)$

# A complete simple example

## Example (The most plausible model for Eva)

| ID | E_ID | salary | year | Person | Prj | leads | worksFor |
|---|---|---|---|---|---|---|---|
| 009 | E05 | 30–40K | 2012 | - | - | - | - |

$$\top(Eva)$$

$\top \sqsubseteq Person \sqcup Prj$ $\qquad$ $Person \sqcup Prj(Eva)$

$\Rightarrow f_{Person} = 1[0.88]$ $\qquad$ $Person(Eva)$

$Person \sqsubseteq \exists worksFor.Prj \sqcup \exists leads.Prj$ $\qquad$ $\exists worksFor.Prj \sqcup \exists leads.Prj$

$f_{Salary}(Eva) = 30\text{–}40$

$f_{Salary} = 30\text{–}40 \Rightarrow f_{leads} = 1$ $\qquad$ $\exists leads.Prj(eva)$

$\rightarrow_{\exists}$ $\qquad$ $leads(eva, p),\ Prj(p)$

# Conclusions & Future Work

**Conclusions:**

- Proposed a framework for learning association rules from hybrid sources of information
- Preliminary ideas on exploitation of the learnt association rules during the deductive reasoning process

**Future works:**

- Experimental evaluation of the proposed preliminary methodology
- Extension and consolitation of the theoretical framework to include also relations between objects.