4-valued Semantics for Data Integration under the OWA: a Deductive Database Approach

Dominique Laurent

ETIS – CNRS-University Cergy Pontoise
FRANCE

Motivation

- Integrating data from data sources providing **explictly** true information and false information
- This leads to potential contradictions
- Our goal is not to eliminate contradictions, but rather to conduct reasoning based on them, considering that:
  - Data sources provide reliable information
  - The result is a database, with 'facts' and 'rules'
  - The Open World Assumption is made

Running Example

- Consider the atoms *a*, *b*, *c*, *d* and *e*
- Sources S_1 and S_2 are such that:
  - In S_1 : *a* and *b* are true, and *c* is false
  - In S_2 : *a* and *d* are true, and *b* and *c* are false
- In the integrated database
  - *a* is true and *c* is false (the sources agree)
  - *b* is contradictory (the sources disagree)
  - *d* is true (the sources do not disagree)
  - *e* is unknown (no information in the sources)

Running Example (cont'd)

- Consider the rules

  (1) $c \leftarrow a, \neg b$ (3) $c \leftarrow e$

  (2) $e \leftarrow d$ (4) $\neg e \leftarrow a, d$

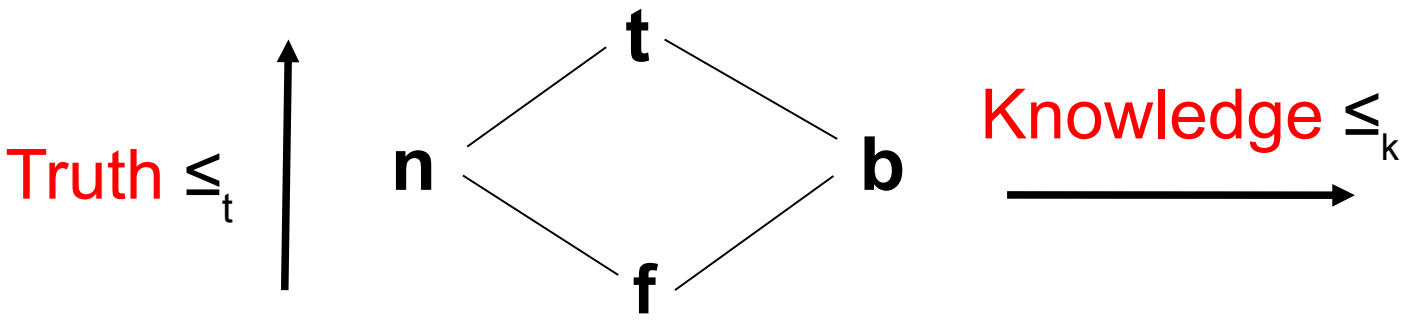
- Rules may generate inconsistencies
- Rules are applied based on OWA: if the body explictly holds, so does the head
  - Rule (2) applies because *d* is known to be true
  - Then (3) applies making *c* true...
  - What about (1)?

Background

- Belnap's four valued logic
  - True **t**
  - False **f**
  - Inconsistent **b**
  - Unknown **n**
- Designated values: **t** and **b**
  - A formula holds if its truth value is **t** or **b**

Background

- Belnap's four valued logic
  - True **t**
  - False **f**
  - Inconsistent **b**
  - Unknown **n**
- Designated values: **t** and **b**
  - A formula holds if its truth value is **t** or **b**
- Two orderings



Background

- ({**t**, **b**, **n**, **f**},$\leq_t$, $\leq_k$) is a distributive bi-lattice
- Basic Connectors
  - Disjunction: lub wrt $\leq_t$ **b** ∨ **n** gives **t**
  - Conjunction: glb wrt $\leq_t$ **b** ∧ **n** gives **f**
  - Negation wrt $\leq_t$: ¬**t** = **f**, ¬**f** = **t**, ¬**b** = **b**, ¬**n** = **n**
  - Weak-negation wrt $\leq_k$: –**t** = **b**, –**f** = **n**, –**b** = **t**, –**n** = **f**
- Additional connector (¬–¬–)
  - Complementation ~ ~**t** = **f**, ~**f** = **t**, ~**b** = **u**, ~**u** = **b**

Implication

- Defining $P \supset Q$ by $\neg P \vee Q$ does not work because $(P \wedge (P \supset Q)) \supset Q$ is not always valid
  - It is **f** when P is **b** and Q is **f**

$P \supset Q$	**t**	**b**	**n**	**f**
t	**t**	**b**	**n**	**f**
b	**t**	**b**	**t**	**b**
n	**t**	**t**	**n**	**n**
f	**t**	**t**	**t**	**t**

Should **not** be designated

Should be designated

Implication

- Defining $P \supset Q$ by $\neg P \vee Q$ does not work because $(P \wedge (P \supset Q)) \supset Q$ is not always valid
- This is not the case for
  - $P \Rightarrow Q$ defined by $\sim P \vee Q$

$P \Rightarrow Q$	t	b	n	f
t	t	b	n	f
b	t	t	n	n
n	t	b	t	b
f	t	t	t	t

Implication

- Defining $P \supset Q$ by $\neg P \vee Q$ does not work because $(P \wedge (P \supset Q)) \supset Q$ is not always valid
- This is not the case for
  - $P \Rightarrow Q$ defined by $\sim P \vee Q$
  - $P \rightarrow Q$ defined below

$P \Rightarrow Q$	t	b	n	f
t	t	b	n	f
b	t	t	n	n
n	t	b	t	b
f	t	t	t	t

$P \rightarrow Q$	t	b	n	f
t	t	b	n	f
b	t	b	n	f
n	t	t	t	t
f	t	t	t	t

$P \Rightarrow Q$ or $P \rightarrow Q$???

- $v(P \Rightarrow Q)$ and $v(P \rightarrow Q)$ are not designated iff $v(P)$ is designated and $v(Q)$ is not

$P \Rightarrow Q$ or $P \rightarrow Q$???

- $v(P \Rightarrow Q)$ and $v(P \rightarrow Q)$ are not designated iff $v(P)$ is designated and $v(Q)$ is not
- $v(P) \leq_t v(Q)$ iff $v(P \Rightarrow Q) = \mathbf{t}$

$P \Rightarrow Q$ or $P \rightarrow Q$???

- $v(P \Rightarrow Q)$ and $v(P \rightarrow Q)$ are not designated iff $v(P)$ is designated and $v(Q)$ is not
- $v(P) \leq_t v(Q)$ iff $v(P \Rightarrow Q) = \mathbf{t}$
- $v(P \rightarrow Q) = v(Q)$ iff $v(P)$ is designated
- None does satisfy the above two properties

$P \Rightarrow Q$ or $P \rightarrow Q$???

- $v(P \Rightarrow Q)$ and $v(P \rightarrow Q)$ are not designated iff $v(P)$ is designated and $v(Q)$ is not
- $v(P) \leq_t v(Q)$ iff $v(P \Rightarrow Q) = \mathbf{t}$
- $v(P \rightarrow Q) = v(Q)$ iff $v(P)$ is designated
- None does satisfy the above two properties, neither do they satisfy contraposition
  - $P \Rightarrow Q$ is *not* equivalent to $\neg Q \Rightarrow \neg P$
  - $P \rightarrow Q$ is *not* equivalent to $\neg Q \rightarrow \neg P$

$P \Rightarrow Q$ or $P \rightarrow Q$???

- $v(P \Rightarrow Q)$ and $v(P \rightarrow Q)$ are not designated iff $v(P)$ is designated and $v(Q)$ is not
- $v(P) \leq_t v(Q)$ iff $v(P \Rightarrow Q) = \mathbf{t}$
- $v(P \rightarrow Q) = v(Q)$ iff $v(P)$ is designated
- None does satisfy the above two properties, neither do they satisfy contraposition
  - $P \Rightarrow Q$ is *not* equivalent to $\neg Q \Rightarrow \neg P$
  - $P \rightarrow Q$ is *not* equivalent to $\neg Q \rightarrow \neg P$
- In fact, the choice is not so important...

Our Approach

- Valued pair (or v-pair): (F, **v**) such that
  - F is a fact in the underlying Herband Universe
  - **v** is a truth value in {**t**, **b**, **n**, **f**}
- A set E of v-pairs is a v-set
  E is consistent if E contains *at most* one v-pair involving a fact F
- Every consistent v-set defines a valuation v_E :
  - If (F, **v**) occurs in E then $v_E(F)$ = **v**
  - Otherwise, $v_E(F)$ = **n**

Our Approach

- A database *D* is a pair (*E*, *R*) such that
  - *E* is a v-set
  - *R* is a set of rules whose bodies and/or heads can contain negated atoms
- *D* is consistent if so is *E*
- If *D* = (*E*, *R*) is consistent then for every fact *F*
  - If *F* occurs in *E* then *F* has the truth value in the corresponding v-pair
  - Otherwise, *F* has the truth value **n**

Back to the Running Example

- $D = (E, R)$ where
  - $E = \{(a, \mathbf{t}), (b, \mathbf{b}), (c, \mathbf{f}), (d, \mathbf{t})\}$
  - R is the following set

    (1) $c \leftarrow a, \neg b$

    (2) $e \leftarrow d$

    (3) $c \leftarrow e$

    (4) $\neg e \leftarrow a, d$
- v_E is defined by
  - $v_E(a) = \mathbf{t}$; $v_E(b) = \mathbf{b}$; $v_E(c) = \mathbf{f}$; $v_E(d) = \mathbf{t}$; $v_E(e) = \mathbf{n}$
- (1), (2) and (3) do not hold based on v_E
  - $v_E(a, \neg b) = \mathbf{b}$, $v_E(c) = \mathbf{f}$ thus $v_E(\text{rule}(1)) = \mathbf{n}$ or $\mathbf{f}$

Back to the Running Example

- $D = (E, R)$ where

$P \Rightarrow Q$	t	b	n	f
t	t	b	n	f
b	t	t	n	n
n	t	b	t	b
f	t	t	t	t

$P \rightarrow Q$	t	b	n	f
t	t	b	n	f
b	t	b	n	f
n	t	t	t	t
f	t	t	t	t

- = **n**
- (1), (2) and (4) do not hold
  - $v_E(a, \neg b) = \mathbf{b}$, $v_E(c) = \mathbf{f}$ thus $v_E(\text{rule}(1)) = \mathbf{n}$ or **f**

Back to the Running Example

- $D = (E, R)$ where
  - $E = \{(a, \mathbf{t}), (b, \mathbf{b}), (c, \mathbf{f}), (d, \mathbf{t})\}$
  - R is the set of the following rules

    (1) $c \leftarrow a, \neg b$ (3) $c \leftarrow e$

    (2) $e \leftarrow d$ (4) $\neg e \leftarrow a, d$
- v_E is defined by
  - $v_E(a) = \mathbf{t}$; $v_E(b) = \mathbf{b}$; $v_E(c) = \mathbf{f}$; $v_E(d) = \mathbf{t}$; $v_E(e) = \mathbf{n}$
- (1), (2) and (4) do not hold based on v_E
  - $v_E(d) = \mathbf{t}$, $v_E(e) = \mathbf{n}$ thus $v_E(\text{rule}(2)) = \mathbf{n}$

Back to the Running Example

- $D = (E, P)$ where
-

$P \Rightarrow Q$	t	b	n	f
t	t	b	n	f
b	t	t	n	n
n	t	b	t	b
f	t	t	t	t

$P \rightarrow Q$	t	b	n	f
t	t	b	n	f
b	t	b	n	f
n	t	t	t	t
f	t	t	t	t

n

- (1), (2) and (4) do not hold based
  - $v_E(d)$ = **t**, v_E(e) = **n** thus v_E(rule(2)) = **n**

Back to the Running Example

- $D = (E, R)$ where
  - $E = \{(a, \mathbf{t}), (b, \mathbf{b}), (c, \mathbf{f}), (d, \mathbf{t})\}$
  - R is the set of the following rules

    (1) $c \leftarrow a, \neg b$ (3) $c \leftarrow e$

    (2) $e \leftarrow d$ (4) $\neg e \leftarrow a, d$
- v_E is defined by
  - $v_E(a) = \mathbf{t}$; $v_E(b) = \mathbf{b}$; $v_E(c) = \mathbf{f}$; $v_E(d) = \mathbf{t}$; $v_E(e) = \mathbf{n}$
- (1), (2) and (4) do not hold based on v_E
  - $v_E(a, d) = \mathbf{t}$, $v_E(\neg e) = \mathbf{n}$ thus $v_E(\text{rule}(4)) = \mathbf{n}$

Database Semantics

- A model *M* of *D* = (*E*, *R*) is a v-set such that
  - The content of *E* is preserved, i.e. $E \subseteq M$
  - The rules of *R* are valid
- Not possible unless rules have exceptions
  - An instantiated rule whose head involves a fact in *E* does **not** apply

Database Semantics

- A model M of $D = (E, R)$ is a v-set such that
  - The content of E is preserved, i.e. $E \subseteq M$
  - The rules of R are valid
- Not possible unless rules have exceptions
- For $E = \{(a, \mathbf{t}), (b, \mathbf{b}), (c, \mathbf{f}), (d, \mathbf{t})\}$ and R

  (1) $c \leftarrow a, \neg b$

  (2) $e \leftarrow d$

  (3) $c \leftarrow e$

  (4) $\neg e \leftarrow a, d$

  $M = \{(a, \mathbf{t}), (b, \mathbf{b}), (c, \mathbf{f}), (d, \mathbf{t}), (e, \mathbf{b})\}$ is a model
  - (1) and (3) do not apply
  - (2) and (4) are valid

Database Semantics

- (1)
- (3)

$P \Rightarrow Q$	t	b	n	f
t	t	b	n	f
b	t	t	n	n
n	t	b	t	b
f	t	t	t	t

$P \rightarrow Q$	t	b	n	f
t	t	b	n	f
b	t	b	n	f
n	t	t	t	t
f	t	t	t	t

(2) $e \leftarrow d$

(4) $\neg e \leftarrow a, d$

M = {(a, **t**), (b, **b**), (c, **f**), (d, **t**), (e, **b**)} is a model

- (1) and (3) do not apply
- (2) and (4) are valid

Computing the Database Semantics

- Given $D = (E, R)$, define $inst^E(R)$ as the set of all instantiated rules r such that:
  - r is an instantiation of a rule in R
  - The head of r does not occur in E
- Given a v-set S define: $\Gamma(S) = E \cup$

$\{(h, \mathbf{t}) \mid (\exists r \text{ in } inst^E(R))(h = head(r), v_S(body(r)) = \mathbf{t})\}$

$\{(h, \mathbf{b}) \mid (\exists r \text{ in } inst^E(R))(h = head(r), v_S(body(r)) = \mathbf{b})\}$

$\{(h, \mathbf{f}) \mid (\exists r \text{ in } inst^E(R))(h = \neg head(r), v_S(body(r)) = \mathbf{t})\}$

$\{(h, \mathbf{b}) \mid (\exists r \text{ in } inst^E(R))(h = \neg head(r), v_S(body(r)) = \mathbf{b})\}$

Example

- $D = (E, R)$ where $E = \emptyset$ and R

  (1) $c \leftarrow a, \neg b$

  (2) $e \leftarrow d$

  (3) $c \leftarrow e$

  (4) $\neg e \leftarrow a, d$

- $S = \{(a, \mathbf{t}), (b, \mathbf{b}), (d, \mathbf{t})\}$
- $\Gamma(S) = \{(c, \mathbf{b}), (e, \mathbf{t}), (e, \mathbf{f})\}$

because $v_s(body(4)) = \mathbf{t}$

because $v_s(body(1)) = \mathbf{b}$

because $v_s(body(2)) = \mathbf{t}$

Computing the Database Semantics

- Given $D = (E, R)$ and S, define $\Sigma(S)$ as

  $\Sigma(S) = \{(F, \mathbf{v}_k) \mid F$ occurs in $\Gamma(S)$ and

  $\mathbf{v}_k = lub_k\{\mathbf{v} \mid (F, \mathbf{v})$ in $\Gamma(S)\}\}$

- The following sequence is monotonic wrt $\leq_k$
  - $\Sigma_0 = E$
  - For $p > 0$, $\Sigma_{p+1} = \Sigma(\Sigma_p)$
- The semantics of $D = (E, R)$ is the limit of the sequence, denoted Σ^*

Example

- $D = (E, R)$; $E = \{(a, \mathbf{t}), (b, \mathbf{b}), (d, \mathbf{t})\}$ and R

  (1) $c \leftarrow a, \neg b$

  (2) $e \leftarrow d$

  (3) $c \leftarrow e$

  (4) $\neg e \leftarrow a, d$

- $inst^E(R) = R$ and $\Sigma_0 = \{(a, \mathbf{t}), (b, \mathbf{b}), (d, \mathbf{t})\}$

Example

- $D = (E, R)$; $E = \{(a, \mathbf{t}), (b, \mathbf{b}), (d, \mathbf{t})\}$ and R

  (1) $c \leftarrow a, \neg b$

  (2) $e \leftarrow d$

  (3) $c \leftarrow e$

  (4) $\neg e \leftarrow a, d$

- $inst^E(R) = R$ and $\Sigma_0 = \{(a, \mathbf{t}), (b, \mathbf{b}), (d, \mathbf{t})\}$
- Computation of Σ_1
  - $\Gamma(\Sigma_0) = \Sigma_0 \cup \{(c, \mathbf{b}), (e, \mathbf{t}), (e, \mathbf{f})\}$
  - $\Sigma_1 = \{(a, \mathbf{t}), (b, \mathbf{b}), (d, \mathbf{t}), (c, \mathbf{b}), (e, \mathbf{b})\}$

Example

- $D = (E, R)$; $E = \{(a, \mathbf{t}), (b, \mathbf{b}), (d, \mathbf{t})\}$ and R

  (1) $c \leftarrow a, \neg b$ (3) $c \leftarrow e$

  (2) $e \leftarrow d$ (4) $\neg e \leftarrow a, d$

- $inst^E(R) = R$ and $\Sigma_0 = \{(a, \mathbf{t}), (b, \mathbf{b}), (d, \mathbf{t})\}$
- Computation of Σ_1
  - $\Gamma(\Sigma_0) = \Sigma_0 \cup \{(c, \mathbf{b}), (e, \mathbf{t}), (e, \mathbf{f})\}$
  - $\Sigma_1 = \{(a, \mathbf{t}), (b, \mathbf{b}), (d, \mathbf{t}), (c, \mathbf{b}), (e, \mathbf{b})\}$
- $\Sigma_2 = \Sigma(\Sigma_1) = \Sigma_1$
- Thus Σ_1 is the semantics of D, i.e. $\Sigma^* = \Sigma_1$

Back to the Running Example

- $D = (E, R)$; $E = \{(a, \mathbf{t}), (b, \mathbf{b}), (c, \mathbf{f}), (d, \mathbf{t})\}$ and R

  (1) $c \leftarrow a, \neg b$ (3) $c \leftarrow e$

  (2) $e \leftarrow d$ (4) $\neg e \leftarrow a, d$

- $inst^E(R) = \{(2), (4)\}$ and $\Sigma_0 = E$
- Computation of Σ_1
  - $\Gamma(\Sigma_0) = \Sigma_0 \cup \{(e, \mathbf{t}), (e, \mathbf{f})\}$
  - $\Sigma_1 = \{(a, \mathbf{t}), (b, \mathbf{b}), (c, \mathbf{f}), (d, \mathbf{t}), (e, \mathbf{b})\}$
- $\Sigma_2 = \Sigma(\Sigma_1) = \Sigma_1$
- Thus $\Sigma^* = \Sigma_1$

Properties

- For every $D = (E, R)$, Σ^* is a model of D
- This model is minimal wrt set-inclusion

Properties

- For every $D = (E, R)$, Σ^* is a model of D
- This model is minimal wrt set-inclusion
- However
  - There are other minimal models wrt set-inclusion
  - This model is not minimal wrt truth ordering
  - This model is not minimal wrt knowledge ordering
- Conjecture:
  - All minimal models wrt set-inclusion contain the same valid facts

Future Work

- Ongoing investigations: characterizing the computed model, updates
- Further research:
  - Extend the relational algebra to this model
  - Study traditional constraints such as FDs, or TGDs
- Other future Issues
  - Implementation
  - Apply in real world applications