Probabilistic Data Integration and Data Exchange

Livia Predoiu

predoiu@ovgu.de

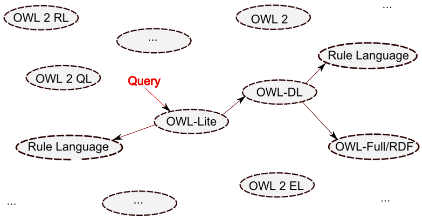
Outline

Sources of Uncertainty in Information Integration, Data Integration and Data Exchange:

- **Uncertain Schema Mappings**: creating precise mappings between data sources is not possible due to e.g. the domain complexity, scale of the data, . . .
- **Uncertain Data**: data is often extracted automatically from unstructured/semi-structred sources
- **Uncertain Queries**: keyword queries instead of structured queries $\rightarrow$ queries need to be translated into some structured form

Information Integration Challenges on the Semantic Web

- Knowledge in the Semantic Web is provided on independent peers
- Domains overlap, but no (global) reference ontology exists
- Mappings need to be created dynamically and automatically.
- Automatically created mappings are uncertain hypotheses (oversimplifying, erroneous)



Approach

- Uncertainty of the mapping hypotheses are modelled with probability theory.
- Mappings are represented as rules.

$\Rightarrow$ Integrated reasoning with deterministic ontologies (in DL) and uncertain mappings (in LP) in a logical framework integrating Description Logics (DL) and Logic Programming (LP) with an *extension for acounting for the probabilities in the mapping*

Advantages of using probability theory:

- rules of classical logics still hold (boolean truth values)
- uncertainty due to incomplete knowledge $\rightarrow$ uncertainty in an automatically created mapping interpreted as belief
- straight forward combination of the beliefs of several matchers (trust, mapping refinement)
- graphical models and well-known inference methods can be used for special kinds of distributions
- probabilistic information retrieval settings can be adjusted

Advantages of using mappings as rules:

- intuitive understanding of Instance Transformation and Instance Retrieval (set theory)
- Rule languages more appropriate for the inference task Instance Retrieval
- Description Logics KBs and Logic Programming KBs can be integrated (due to the interweaved integration of DL and LP used)

Integrated reasoning with ontologies and uncertain mappings provides

- more insight into the (un)certainty of the reasoning results
- better handling of the (un)certainty of mapping chains
- a natural ranking method over the reasoning results

The logical foundation

probabilistic extension of 2 formalisms that integrate DL and LP are appropriate:

- generalized dl-programs
  $\rightarrow$ generalized Bayesian dl-programs
- tightly coupled dl-programs
  $\rightarrow$ tightly coupled probabilistic dl-programs
  (2 semantics: answer set semantics and well-founded semantics)

Both tightly integrate a DL L and a LP P to an integrated knowledge base $KB = (L, P)$ and provide a probabilistic extension $KB = (L, P, C, \mu)$ and $KB = (L, P, \mu, Comb)$

generalized Bayesian dl-programs: Syntax

- A generalized Bayesian dl-program is a 4-tuple $KB = (L, P, \mu, Comb)$ where
    - L is a Description Logic knowledge base in the DLP fragment
    - P is a Datalog program
    - $\mu(r, v)$ is a probability function over all truth valuations w of the head atom associated with each rule r in $ground(P)$ and every truth valuation v of the body atoms of r
    - $Comb$ is a combining rule, which defines how rules of $r \in ground(P)$ with same head atom can be combined.

generalized Bayesian dl-programs: Semantics

- each generalized Bayesian dl-program $KB = (L, P, \mu, Comb)$ encodes the structure of a Bayesian Network *BN*
- Translation from *KB* to *BN*
  - (L, P) is translated into its Datalog equivalent $D = L' \cup P$
  - a ground atom *a* is active iff it belongs to the canonical model of *D*; $r \in ground(D)$ is active iff all its atoms are active
  - every active atom corresponds to a node in *BN*
  - μ is the conditional probability density for each active rule and is translated to arcs in *BN* encoding direct influence relations between the atoms involved in *r*
  - for at least 2 active rules with same head, the combining rule *Comb* generates a joint conditional distribution from the individual ones of the involved rules.

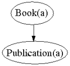
Example

$Report(a).published(a).Book(a).$

$Publication(x) \overset{(0.9,0.2)}{\leftarrow} Book(x).$

$Publication(x) \overset{(0.7,0.3,0.0,0.0)}{\leftarrow} Report(x), published(x, y).$

$Comb = Maximum$



	Publication(a)	Book(a)
0.9	t	t

	Publication(a)	Book(a)	Report(a)	published(a)
0.9	t	t	t	t
0.3	t	t	t	f
0.9	t	t	f	t
0.7	t	f	t	t
0.9	t	t	f	f
0.3	t	f	f	t
0.3	t	f	t	f
0.3	t	f	f	f

$$\forall X_1, \ldots, W_p \quad p_1(X_1, \ldots, X_n), \ldots, p_l(Y_1, \ldots, Y_k) | \\ p_{l+1}(Z_1, \ldots Z_m), \ldots p_o(W_1, \ldots, W_p)$$

Two types of queries:

- ground queries
- non-ground queries (information retrieval)

tightly coupled probabilistic dl-programs: Syntax and Semantics

Tightly coupled probabilistic dl-program $KB = (L, P, C, \mu)$:

- description logic knowledge base L (in $\mathcal{SHIF}$(D) or $\mathcal{SHOIN}$(D))),
- disjunctive program P with values of random variables $A \in C$ as "switches" in rule bodies,
- probability distribution μ over all joint instantiations B of the random variables $A \in C$.

A set of probability distributions over first-order models is specified: Every joint instantiation B of the random variables along with P specifies a set of first-order models of which the probabilities sum up to $\mu(B)$.

Description logic knowledge base L for an online store:

(1) $\mathit{Textbook} \sqsubseteq \mathit{Book}$; (2) $\mathit{PC} \sqcup \mathit{Laptop} \sqsubseteq \mathit{Electronics}$; $\mathit{PC} \sqsubseteq \neg\mathit{Laptop}$;
(3) $\mathit{Book} \sqcup \mathit{Electronics} \sqsubseteq \mathit{Product}$; $\mathit{Book} \sqsubseteq \neg\mathit{Electronics}$; (4) $\mathit{Sale} \sqsubseteq \mathit{Product}$;
(5) $\mathit{Product} \sqsubseteq\ \geqslant 1\ \mathit{related}$; (6) $\geqslant 1\ \mathit{related} \sqcup \geqslant 1\ \mathit{related}^- \sqsubseteq \mathit{Product}$;
(7) $\mathit{related} \sqsubseteq \mathit{related}^-$; $\mathit{related}^- \sqsubseteq \mathit{related}$;
(8) $\mathit{Textbook}(\mathit{tb\_ai})$; $\mathit{Textbook}(\mathit{tb\_lp})$; (9) $\mathit{related}(\mathit{tb\_ai}, \mathit{tb\_lp})$; (10) $\mathit{PC}(\mathit{pc\_ibm})$; $\mathit{PC}(\mathit{pc\_hp})$;
(11) $\mathit{related}(\mathit{pc\_ibm}, \mathit{pc\_hp})$; (12) $\mathit{provides}(\mathit{ibm}, \mathit{pc\_ibm})$; $\mathit{provides}(\mathit{hp}, \mathit{pc\_hp})$.

Disjunctive program P for an online store:

(1) $pc(pc_1)$; $pc(pc_2)$; $pc(obj_3) \vee laptop(obj_3)$;
(2) $brand\_new(pc_1)$; $brand\_new(obj_3)$;
(3) $vendor(dell, pc_1)$; $vendor(dell, pc_2)$;
(4) $avoid(X) \leftarrow camera(X), not\ sale(X)$;
(5) $sale(X) \leftarrow electronics(X), not\ brand\_new(X)$;
(6) $provider(V) \leftarrow vendor(V, X), product(X)$;
(7) $provider(V) \leftarrow provides(V, X), product(X)$;
(8) $similar(X, Y) \leftarrow related(X, Y)$;
(9) $similar(X, Z) \leftarrow similar(X, Y), similar(Y, Z)$;
(10) $similar(X, Y) \leftarrow similar(Y, X)$;
(11) $brand\_new(X) \vee high\_quality(X) \leftarrow expensive(X)$.

Syntax (deterministic tightly coupled dl-programs)

- Sets $\mathbf{A}$, $\mathbf{R}_A$, $\mathbf{R}_D$, $\mathbf{I}$, and $\mathbf{V}$ of atomic concepts, abstract roles, datatype roles, individuals, and data values, respectively.
- Finite sets Φ_p and Φ_c of predicate and constant symbols with: (i) Φ_p not necessarily disjoint to $\mathbf{A}$, $\mathbf{R}_A$, and $\mathbf{R}_D$, and (ii) $\Phi_c \subseteq \mathbf{I} \cup \mathbf{V}$.
- A tightly coupled disjunctive dl-program $KB = (L, P)$ consists of a description logic knowledge base L and a disjunctive program P.

Semantics (deterministic tightly coupled dl-programs)

- An interpretation I is any subset of the Herbrand base HB_Φ.
- I is a model of P is defined as usual.
- I is a model of L iff $L \cup I \cup \{\neg a \mid a \in HB_\Phi - I\}$ is satisfiable.
- I is a model of KB iff I is a model of both L and P.
- The Gelfond-Lifschitz reduct of $KB = (L, P)$ w.r.t. $I \subseteq HB_\Phi$, denoted KB^I, is defined as the disjunctive dl-program (L, P^I), where P^I is the standard Gelfond-Lifschitz reduct of P w.r.t. I.
- $I \subseteq HB_\Phi$ is an answer set of KB iff I is a minimal model of KB^I.
- KB is consistent iff it has an answer set.
- A ground atom $a \in HB_\Phi$ is a cautious (resp., brave) consequence of a disjunctive dl-program KB under the answer set semantics iff every (resp., some) answer set of KB satisfies a.

tightly coupled probabilistic dl-programs: Syntax and Semantics

Tightly coupled probabilistic dl-program $KB = (L, P, C, \mu)$:

- description logic knowledge base L (in $\mathcal{SHIF}$(D) or $\mathcal{SHOIN}$(D))),
- disjunctive program P with values of random variables $A \in C$ as "switches" in rule bodies,
- probability distribution μ over all joint instantiations B of the random variables $A \in C$.

A set of probability distributions over first-order models is specified: Every joint instantiation B of the random variables along with P specifies a set of first-order models of which the probabilities sum up to $\mu(B)$.

Example

Probabilistic rules in P along with the probability μ on the choice space C of a probabilistic dl-program $KB = (L, P, C, \mu)$:

- $avoid(X) \leftarrow Camera(X), not\ offer(X), avoid\_pos$;
- $offer(X) \leftarrow Electronics(X), not\ brand\_new(X), offer\_pos$;
- $buy(C, X) \leftarrow needs(C, X), view(X), not\ avoid(X), v\_buy\_pos$;
- $buy(C, X) \leftarrow needs(C, X), buy(C, Y), also\_buy(Y, X), a\_buy\_pos$.

μ: $avoid\_pos, avoid\_neg \mapsto 0.9, 0.1$; $offer\_pos, offer\_neg \mapsto 0.9, 0.1$; $v\_buy\_pos, v\_buy\_neg \mapsto 0.7, 0.3$; $a\_buy\_pos, a\_buy\_neg \mapsto 0.7, 0.3$.

$\{avoid\_pos, offer\_pos, v\_buy\_pos, a\_buy\_pos\} : 0.9 \times 0.9 \times 0.7 \times 0.7, \ldots$

Probabilistic query: $\exists(buy(john, ixus500))[L, U]$

$$\exists X_1, \ldots, W_p \quad p_1(X_1, \ldots, X_n), \ldots, p_l(Y_1, \ldots, Y_k) | \\ p_{l+1}(Z_1, \ldots Z_m), \ldots p_o(W_1, \ldots, W_p)[r, s]$$

Possible Queries:

- ground
- nonground (information retrieval)

Intuitively

- $L = O_1 \cup O_2$ encodes the ontologies
- P, μ encodes the mappings

Mappings:

- $Q(O_i)$ denotes the matchable elements of the ontology O_i
- Matching: Given two ontologies O and O', determine correspondences between $Q(O)$ and $Q(O')$.
- Correspondences are 5-tuples (id, e, e', r, n) such that
  - id is a unique identifier;
  - $e \in Q(O)$ and $e' \in Q(O')$;
  - $r \in R$ is a semantic relation (here: implication);
  - n is a degree of confidence in the correctness. (here: a probability according to a probability distribution)

Consistent correspondences are mappings.

Mappings in generalized bayesian dl-programs

(1) $O_1 : \mathit{Publication}(x) \stackrel{(0.9,0.2)}{\longleftarrow} O_2 : \mathit{Publication}(x)$;
(2) $O_1 : \mathit{Article}(x) \stackrel{(0.7,0.2)}{\longleftarrow} O_2 : \mathit{Paper}(x)$;
(3) $O_1 : \mathit{Person}(x) \stackrel{(0.9,0.2)}{\longleftarrow} O_2 : \mathit{Person}(x)$;
(4) $O_1 : \mathit{Collection}(x) \stackrel{(0.7,0.2)}{\longleftarrow} O_2 : \mathit{Proceedings}(x)$;
(5) $O_1 : \mathit{keyword}(x, y) \stackrel{(0.7,0.2)}{\longleftarrow} O_2 : \mathit{about}(x, y)$;
(6) $O_1 : \mathit{author}(y, x) \stackrel{(0.7,0.2)}{\longleftarrow} O_2 : \mathit{author}(x, y)$.

Mappings in tightly coupled probabilistic dl-programs

(1) $O_2 : \mathrm{Published}(X) \leftarrow O_1 : \mathrm{Publication}(X) \wedge \mathrm{not}\ O_1 : \mathrm{Unpublished}(X) \wedge \mathrm{hmatch}_1$.
(2) $O_2 : \mathrm{Publication}(X) \leftarrow O_1 : \mathrm{Published}(X) \wedge \mathrm{falcon}_1$.
(3) $O_2 : \mathrm{Publication}(X) \leftarrow O_1 : \mathrm{Unpublished}(X) \wedge \mathrm{falcon}_2$.

$C = \{\{\mathrm{hmatch}_1, \mathrm{not\_hmatch}_1\}, \{\mathrm{falcon}_1, \mathrm{not\_falcon}_1\}, \{\mathrm{falcon}_2, \mathrm{not\_falcon}_2\}\}$.

$\mu(\mathrm{hmatch}_1) = 0.72$, $\mu(\mathrm{hmatch}_2) = 0.71$, $\mu(\mathrm{falcon}_1) = 0.85$, $\mu(\mathrm{falcon}_2) = 0.92$.

Features

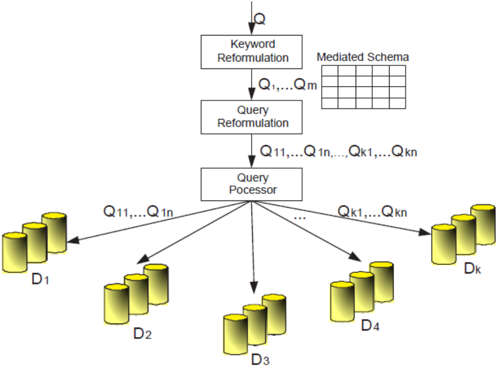
- Tight integration of mapping and ontology language
- Support for mappings refinement
- Support for repairing inconsistencies (tightly coupled dl-programs)
- Representation and combination of confidence
- Decidability and efficiency of instance reasoning (generalized bayesian dl-programs)

References:

- Andrea Cali, Thomas Lukasiewicz, Livia Predoiu and Heiner Stuckenschmidt. Tightly Coupled Probabilistic Description Logic Programs for the Semantic Web. Journal of Data Semantics 12, 2009
- Andrea Calì, Thomas Lukasiewicz, Livia Predoiu and Heiner Stuckenschmidt. Rule-Based Approaches for Representing Probabilistic Ontology Mappings. Uncertainty Reasoning for the Semantic Web I, 5327, Lecture Notes in Computer Science, Springer, 2008.
- Livia Predoiu and Heiner Stuckenschmidt. Probabilistic Extensions of Semantic Web Languages - A Survey. The Semantic Web for Knowledge and Data Management: Technologies and Practices, Idea Group Inc, 2008.
- Andrea Cali, Thomas Lukasiewicz, Livia Predoiu, Heiner Stuckenschmidt. Tightly Integrated Probabilistic Description Logic Programs for Representing Ontology Mappings. Proceedings of the International Symposium on Foundations of Information and Knowledge Systems, Pisa, Italy, 2008.
- Livia Predoiu. A Reasoner for Generalized Bayesian DL-Programs. Proceedings of the Fourth International Workshop on Uncertainty Reasoning for the Semantic Web, in conjunction with the ISWC, Karlsruhe, Germany, 2008.
- Andrea Cali, Thomas Lukasiewicz, Livia Predoiu, Heiner Stuckenschmidt. A Framework for Representing Ontology Mappings under Probabilities and Inconsistencies. In Proc. of the Workshop for Uncertainty Reasoning on the Semantic Web (URSW) in conjunction with the ISWC, Busan, Korea, 2007
- Thomas Lukasiewicz. A Novel Combination of Answer Set Programming with Description Logics for the Semantic Web. IEEE Transactions on Knowledge and Data Engineering (TKDE), 22(11), 1577-1592, November 2010.

Data Integration with Uncertainty (Dong, Halevy, Yu, 2007)

Architecture considered:



Example

- Source schema S = (pname, email-addr, permanent-addr, current-addr)
- Target schema T = (name, email, mailing-addr, home-addr, office-addr)
- Query: SELECT mailing-addr FROM T
- Query reformulations:
  - Q1: SELECT current-addr FROM S
  - Q2: SELECT permanent-addr FROM S
  - Q3: SELECT email-addr FROM S

Schema mappings

- relational data model, select-project-join (SPJ) Queries in SQL are considered
- schema contains a finite set of relations
- relation contains of a finite set of attributes ($R = \langle r_1, \ldots, r_n \rangle$)
- An *instance* D_R of R is a finite set of tuples

General GLAV mappings: $m : \forall x(\phi(x) \rightarrow \exists y \psi(x, y))$

Framework of Dong, Halevy and Yu make the following restrictions:

- only projection queries on a single table on each side of the mapping (schema matching)
- GLAV mappings where
  - ϕ (resp. ψ) is an atomic formula over S (resp. T)
  - no constants are included
  - each variable occurs at most once on each side of the mapping

mappings can be defined as *attribute correspondences* $C_{ij} = (s_i, t_j)$

schema mapping M = (S, T, m)

- $S \in \overline{S}$ is a source relation in the relational schema $\overline{S}$, $T \in \overline{T}$ is a target relation in the relational schema $\overline{T}$ and m a set of attribute correspondences between S and T

One-to-one relation mapping: each s_i and each t_j occurs in at most 1 correspondence in m

A schema mapping $\overline{M}$ is a set of one-to-one relation mappings between relations in $\overline{S}$ and $\overline{T}$ where every relation appears at most once.

probabilistic mapping (p-mapping) pM = (S, T, **m**)

- $S \in \overline{S}$ is a source relation in the relational schema $\overline{S}$, $T \in \overline{T}$ is a target relation in the relational schema $\overline{T}$
- **m** is a set $\{(m_1, Pr(m_1)), \ldots, (m_l, Pr(m_l))\}$ such that
  - for $i \in [1, l]$, m_i is a one-to-one mapping between S and T and $\forall i, j \in [1, l]$: $i \neq j \Rightarrow m_i \neq m_j$
  - $Pr(m_i) \in [0, 1]$ and $\sum_{i=1}^{l} Pr(m_i) = 1$

A schema p-mapping $\overline{pM}$ is a set of p-mappings between relations in $\overline{S}$ and $\overline{T}$ where every relation appears at most once in one p-mapping.

Example

- Source schema S = (pname, email-addr, permanent-addr, current-addr)
- Target schema T = (name, email, mailing-addr, home-addr, office-addr)

	Possible Mapping	Prob
m_1 =	{(pname, name), (email-addr, email), (current-addr, ***mailing-addr***), (permanent-addr, home-address)}	0.5
m_2 =	{(pname, name), (email-addr, email), (permanent-addr, ***mailing-addr***), (current-addr, home-address)}	0.4
m_3 =	{(pname, name), (email-addr, ***mailing-addr***), (current-addr, home-addr)}	0.1

D_S =

pname	email-addr	current-addr	permanent-addr
Alice	alice@	Mountain View	Sunnyvale
Bob	bob@	Sunnyvale	Sunnyvale

Query-Answer =

Tuple	Prob
('Sunnyvale')	0.9
('Mountain View')	0.5
('alice@')	0.1
('bob@')	0.1

Semantics of ordinary/deterministic mappings

- **Consistent Target Instance**: With M = (S, T, m) given, $D_T \in T$ is consistent with $D_S \in S$ and M if D_S and D_T satisfy m.
- **Certain Answer**: With M = (S, T, m), $Tar_M(D_S)$ being the set of all consistent target instances and Query Q over T given, a tuple t is a certain answer of Q w.r.t. D_S and M if $\forall D_T \in Tar_M(D_S) : t \in Q(D_T)$

Semantics of probabilistic mappings (by-table semantics vs. by-tuple semantics)

- **by-table semantics**
  - **by-table consistent target instance**: With pM = (S, T, **m**) given, $D_T \in T$ is by-table consistent with $D_S \in S$ and pM if there exists a mapping $m \in$ **m** s.t. D_S and D_T satisfy m.
  - **by-table answer**: With pM = (S, T, **m**), $Tar_m(D_S)$ being the set of all by-table consistent target instances, Query Q over T and t being a tuple given, $\overline{m(t)}$ is the subset of **m** s.t. $\forall m \in \overline{m(t)}$ and $\forall D_T \in Tar_m(D_S)$: $t \in Q(D_T)$. With $p = \sum_{m \in \overline{m(t)}} Pr(m)$, (t, p) is a by-table answer of Q w.r.t. D_S and pM if $p > 0$

Example by-table semantics

- Source schema S = (pname, email-addr, permanent-addr, current-addr)
- Target schema T = (name, email, mailing-addr, home-addr, office-addr)

	Possible Mapping	Prob
m_1 =	{(pname, name), (email-addr, email), (current-addr, **mailing-addr**), (permanent-addr, home-address)}	0.5
m_2 =	{(pname, name), (email-addr, email), (permanent-addr, **mailing-addr**), (current-addr, home-address)}	0.4
m_3 =	{(pname, name), (email-addr, **mailing-addr**), (current-addr, home-addr)}	0.1

D_S =

pname	email-addr	current-addr	permanent-addr
Alice	alice@	Mountain View	Sunnyvale
Bob	bob@	Sunnyvale	Sunnyvale

Query-Answer =

Tuple	Prob
('Sunnyvale')	0.9
('Mountain View')	0.5
('alice@')	0.1
('bob@')	0.1

By-table Query answering

- Algorithm
  - **Step 1**: Generate the possible reformulations Q'_1, ..., Q'_k of Q by considering every combination $(m^1, \ldots, m^l)$, m^i being one of the possible mappings in pM_i. The set of reformualtions is denoted by $Q'_1, \ldots, Q'_k$. The probability of a reformulation $Pr = Q' = (m^1, \ldots, m^l)$ is $\Pi^l_{i=1} Pr(m^i)$
  - **Step 2**: For each reformulation Q', retrieve each of the unique answers from the sources. For each answer obtained by $Q'_1 \cup \ldots \cup Q'_k$ the probability is obtained by summing up the probabilities
- Complexity results
  - With Q being an SPJ query and $\overline{pM}$ a schema p-mapping, answering Q w.r.t. $\overline{pM}$ is in PTIME in the size of the data and the mapping
  - With Q being an SPJ query with only equality conditions over $\overline{T}$ and pGM being a general p-mapping, computing $Q^{table}(D_S)$ w.r.t. pGM is in PTIME in the size of the data and the mapping.
    - general p-mappings are p-mappings that are extended to arbitraty GLAV mappings. A general p-mapping is a triple of the form $pGM = (\overline{S}, \overline{T}, \mathbf{gm})$ with $\mathbf{gm} = \{(gm_i, Pr(gm_i)) | i \in [1, n]\}$ s.t. for each $i \in [1, n]$, gm_i is a general GLAV mapping

Semantics of probabilistic mappings (by-table semantics vs. by-tuple semantics)

- **by-tuple semantics**
  - **by-tuple consistent instance**: With pM = (S, T, **m**) given, $D_T \in T$ is by-tuple consistent with $D_S \in S$ and pM if there exists a sequence $\langle m^1, \ldots, m^d \rangle$ s.t. $\forall i : 1 \leqslant i \leqslant d$:
    - $m^i \in$ **m** and
    - for the i^{th} tuple of D_S, t_i, there exists a target tuple $t_i' \in D_T$ s.t. t_i and t_i' satisfy m^i.
  - If there are l mappings in pM, there are l^d sequences of length d. $\text{seq}_d(pM)$ is the set of mapping sequences of length d generated from pM.
  - **by-tuple answer**: With pM = (S, T, **m**), $Tar_{seq_d}(D_S)$ being the set of all by-tuple consistent target instances with length d, Query Q over T and t being a tuple, $\overline{seq}(t)$ is the subset of $\mathbf{seq}_d$(pM) s.t $\forall seq \in \overline{seq}$ and $\forall D_T \in Tar_{seq}(D_S)$: $t \in Q(D_T)$. With $p = \sum_{seq \in \overline{seq}} Pr(seq)$, (t, p) is a by-tuple answer of Q w.r.t. D_S and pM if $p > 0$.

Example by-tuple semantics

- Source schema S = (pname, email-addr, permanent-addr, current-addr)
- Target schema T = (name, email, mailing-addr, home-addr, office-addr)

	Possible Mapping	Prob
m_1 =	{(pname, name), (email-addr, email), (current-addr, ***mailing-addr***), (permanent-addr, home-address)}	0.5
m_2 =	{(pname, name), (email-addr, email), (permanent-addr, ***mailing-addr***), (current-addr, home-address)}	0.4
m_3 =	{(pname, name), (email-addr, ***mailing-addr***), (current-addr, home-addr)}	0.1

D_S =

pname	email-addr	current-addr	permanent-addr
Alice	alice@	Mountain View	Sunnyvale
Bob	bob@	Sunnyvale	Sunnyvale

by-tuple consistent target instance:

mapping sequence: <m_2,m_3>

name	email	mailing-addr	home-addr	office-addr
Alice	alice@	Sunnyvale	Mountain View	office
Bob	email	bob@	Sunnyvale	office

Query-Answer =

Tuple	Prob
('Sunnyvale')	0.94
('Mountain View')	0.5
('alice@')	0.1
('bob@')	0.1

Example by-tuple semantics

- Source schema S = (pname, email-addr, permanent-addr, current-addr)
- Target schema T = (name, email, mailing-addr, home-addr, office-addr)

	Possible Mapping	Prob
$m_1 =$	{(pname, name), (email-addr, email), (current-addr, ***mailing-addr***), (permanent-addr, home-address)}	0.5
$m_2 =$	{(pname, name), (email-addr, email), (permanent-addr, ***mailing-addr***), (current-addr, home-address)}	0.4
$m_3 =$	{(pname, name), (email-addr, ***mailing-addr***), (current-addr, home-addr)}	0.1

$D_S =$

pname	email-addr	current-addr	permanent-addr
Alice	alice@	Mountain View	Sunnyvale
Bob	bob@	Sunnyvale	Sunnyvale

Query-Answer =

Tuple	Prob
! ('Sunnyvale')	0.94
('Mountain View')	0.5
('alice@')	0.1
('bob@')	0.1

←

Sequences:		
$m_1\ m_1$	0.25	+
$m_1\ m_2$	0.2	+
$m_2\ m_1$	0.2	+
~~$m_1\ m_3$~~	~~0.05~~	-
$m_3\ m_1$	0.05	+
$m_2\ m_2$	0.16	+
$m_2\ m_3$	0.04	+
$m_3\ m_2$	0.04	+
~~$m_3\ m_3$~~	~~0.01~~	-

By-tuple Query answering

- Note: We need to compute certain answers for every *mapping sequence* generated from *pM*
- General complexity results
  - With Q being an SPJ query and $\overline{pM}$ being a schema p-mapping, finding the probability for a by-tuple answer to Q w.r.t. $\overline{pM}$ is #P-complete w.r.t. data complexity and is in PTIME w.r.t. mapping complexity
  - Given an SPJ query and a schema p-mapping, returning all by-tuple answers without probabilities is in PTIME w.r.t. data complexity.

2 restricted cases with by-tuple query answering complexity in PTIME:

- **Queries with a single p-mapping subgoal**: With $\overline{pM}$ being a schema p-mapping and Q being an SPJ query, Q is a non-p-join-query w.r.t $\overline{pM}$ if at most one subgoal in the body of Q is the target of a p-mapping in $\overline{pM}$
- **projected p-join queries**: With $\overline{pM}$ being a schema p-mapping and Q being an SPJ query over the target of $\overline{pM}$, Q is a projected p-join query w.r.t $\overline{pM}$ if
  - at least 2 subgoals in the body of Q are targets of p-mappings in $\overline{pM}$
  - $\forall$ p-join predicates, the join attribute (or an attribute that is entailed to be equal by the predicates in Q) is returned in the SELECT clause
- Conjecture: no more cases with query answering in PTIME
- subgoals = tables in the FROM clause, each occurence of the same table is a different subgoal

Fagin, Kimelfeld, Kolaitis. Probabilistic Data Exchange. ICDT 2010.

- Conceptual Framework of Data Exchange in the context of uncertainty in the source data
- Generalization of the framework of (Dong, Halevy, Yu, 2007) for the by-table semantics

Preliminaries

We have

- fixed, countably infinite sets of constants (**Const**) and nulls (**Var**) with Const $\cap$ Var = $\emptyset$
- a Schema $\mathcal{R} = \langle R_1, \ldots, R_k \rangle$ consists of a finite sequence of distinct relation symbols R_i with fixed arity $r_i > 0$
- an instance $I = \langle R_1^I, \ldots, R_k^I$ (over $\mathcal{R}$) with $R_i^I \subset (\text{Const} \cup \text{Var})^{r_i}$
- R_i^I is the R_i*-Relation* of I, dom(I) is the set of all constants & nulls appearing in I
- a *ground* instance I does not contain nulls
- Inst($\mathcal{R}$) = class of all instances over $\mathcal{R}$, Inst^c($\mathcal{R}$) = class of all ground instances over $\mathcal{R}$
- K_1 and K_2 being instances over $\mathcal{R}$, a homomorphism $h : K_1 \rightarrow K_2$ is a mapping from dom(K_1) to dom(K_2) s.t.
  - $h(c) = c \forall c \in \text{dom}(K_1)$
  - $\forall$ facts $R(\mathbf{t})$ of K_1, $R(h(\mathbf{t})) \in \text{dom}(K_2)$
- $K_1 \rightarrow K_2$ denotes the existence of a homomorphism $h : K_1 \rightarrow K2$

Schema Mappings

- **source** schema $\mathbf{S} = \langle S_1, \ldots, S_n \rangle$ and **target** schema $\mathbf{T} = \langle T_1, \ldots, T_m \rangle$ not having any relation symbols in common
- $\langle \mathbf{S}, \mathbf{T} \rangle$ is the concatenation
- With I, J being instances of **S** and **T**: $K = \langle I, J \rangle \in \text{Inst}(\langle \mathbf{S}, \mathbf{T} \rangle)$ and $S_i^K = S_i^I$ and $T_j^K = T_j^J$ for $1 \leqslant i \leqslant n, 1 \leqslant j \leqslant m$
- Σ is a set of formulas expressing constraints over $\mathcal{R}$. With $I \in \text{Inst}(\mathcal{R})$ $\text{I} \models \Sigma$ denotes that I satisfies every formula of Σ
- Schema mappings are triples $(\mathbf{S}, \mathbf{T}, \Sigma)$ where the source schema **S** and the target schema **T** do not have any relation symbols in common and Σ is a set of formulas over $\langle \mathbf{S}, \mathbf{T} \rangle$, the *dependencys*. Furthermore
  - $I \in \text{Inst}^c(\mathbf{S})$ and $J \in \text{Inst}(\mathbf{T})$, J is a solution for I w.r.t Σ if $\langle I, J \rangle \models \Sigma$
  - A solution J for I w.r.t. Σ is universal if $J \rightarrow J' \quad \forall$ solutions J' of I w.r.t. Σ

Considered Probability Spaces (p-spaces)

Definitions and Notation

- finite or countably infinite space $\tilde{\mathcal{U}} = (\Omega(\tilde{\mathcal{U}}), p_{\tilde{\mathcal{U}}})$ with $\Omega(\tilde{\mathcal{U}})$ being a countable set and $p_{\tilde{\mathcal{U}}} : \tilde{\mathcal{U}} \rightarrow [0, 1]$ *satisfying* $\Sigma_{u \in \Omega(\tilde{\mathcal{U}})} p(u) = 1$
- $u \in \Omega(\tilde{\mathcal{U}})$ is a sample and $\Omega(\tilde{\mathcal{U}})$ is the sample space
- $\tilde{\mathcal{U}}$ is a p-space over $\Omega(\tilde{\mathcal{U}})$
- $\Omega_+(\tilde{U}) \subseteq \Omega(\tilde{U})$ is the **support** of $\tilde{\mathcal{U}}$ containing all $u \in \Omega(\tilde{\mathcal{U}})$ with $p(u) > 0$.
- $\tilde{U}$ is finite, if $\Omega_+(\tilde{\mathcal{U}})$ is finite
- An event is $X \in \Omega(\tilde{\mathcal{U}})$ with $\Pr_{\tilde{\mathcal{U}}} = \Sigma_{u \in X} p_{\tilde{\mathcal{U}}}(u)$
- $\mathcal{U}$ without the tilde sign denotes a random variable representing a sample of $\tilde{\mathcal{U}}$.
- an event is represented by a formula, e.g. $\varphi(U)$ is the same like $\{u \in \Omega(\tilde{U}) | \varphi(u)\}$
- $\tilde{\mathcal{U}}$ often used instead of $\Omega(\tilde{U})$
- With U and W being countable sets and $\tilde{\mathcal{P}}$ being a p-space over $U \times W$, $\tilde{\mathcal{P}} = (\Omega(\tilde{\mathcal{P}}), p_{\tilde{\mathcal{P}}})$ where $\Omega(\tilde{\mathcal{P}}) = U \times W$ and
  - the p-space $\tilde{\mathcal{U}}$ is the **left marginal** of $\tilde{\mathcal{P}}$ s.t. $\Omega(\tilde{\mathcal{U}}) = U$ and $\forall u \in U : p_{\tilde{\mathcal{U}}}(u) = \Sigma_{w \in W} p_{\tilde{\mathcal{P}}}(u, w)$
  - the p-space $\tilde{\mathcal{W}}$ is the **right marginal** $\tilde{\mathcal{P}}$ s.t. $\Omega(\tilde{\mathcal{W}}) = W$ and $\forall w \in W : p_{\tilde{\mathcal{W}}}(w) = \Sigma_{u \in U} p_{\tilde{\mathcal{P}}}(u, w)$

Exchanging probabilistic data

- Let $\mathcal{R}$ be a schema. A probabilistic database or probabilistic instance (over $\mathcal{R}$ is a p-space $\tilde{\mathcal{I}}$ over Inst($\mathcal{R}$).
- Let $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ be a mapping. A *source p-instance* is a ground p-instance $\tilde{\mathcal{I}}$ over **S** and a *target p-instance* is a p-instance $\tilde{\mathcal{J}}$ over **T**.
- **Example**:
  - **S**: *Researcher*(name, university), *RArea*(researcher, topic)
  - **T**: *UArea*(university, department, topic)
  - $\Sigma = \{\forall r, u, t(\text{Researcher}(r, u) \wedge \text{RArea}(r, t) \rightarrow \exists d UArea(u, d, t))\}$

Possible *Researcher* facts

r_e	*Researcher*(Emma, UCSD)
r_j	*Researcher*(John, UCSD)

Possible *RArea* facts

a_{eir}	*RArea*(Emma, IR)
a_{edb}	*RArea*(Emma, DB)
a_{jdb}	*RArea*(John, DB)
a_{jai}	*RArea*(John, AI)

Source p-instance $\tilde{\mathcal{I}}$

$I_1 = \{r_e, r_j, a_{eir}, a_{jdb}\}$	0.3
$I_2 = \{r_e, r_j, a_{eir}, a_{jai}\}$	0.3
$I_3 = \{r_e, r_j, a_{edb}, a_{jai}\}$	0.2
$I_4 = \{r_e, r_j, a_{edb}, a_{jdb}\}$	0.1
$I_5 = \{r_e, a_{edb}\}$	0.1

Possible *UArea* facts

u_{ir}	*UArea*(UCSD, $\perp_1$, IR)
u_{ai}	*UArea*(UCSD, $\perp_2$, AI)
u_{db}	*UArea*(UCSD, $\perp_3$, DB)

Probabilistic Match

- systematic way of extending a binary relationship between deterministic database instances into a binary relationship between p-spaces thereof
- based on the concept of joint (or bivariate) probability spaces with specified marginals [Morgenstern 1956, Frechet, 1951]
- **(Definition):** A **Probabilistic Match** of two p-spaces $\tilde{\mathcal{U}}$ and $\tilde{\mathcal{W}}$ w.r.t. a binary relation $R \subseteq \Omega(\tilde{\mathcal{U}}) \times \Omega(\tilde{\mathcal{W}})$ (for short an *R-match of* $\tilde{\mathcal{U}}$ *in* $\tilde{\mathcal{W}}$) is a p-space $\tilde{\mathcal{P}}$ over $\Omega(\tilde{\mathcal{U}}) \times \Omega(\tilde{\mathcal{W}})$ that satisfies the following 2 conditions
  - The left and right marginals of $\tilde{\mathcal{P}}$ are $\tilde{\mathcal{U}}$ and $\tilde{\mathcal{W}}$, respectively. I.e.
    - $\Sigma_{w \in \Omega(\tilde{\mathcal{W}})} p_{\tilde{\mathcal{P}}}(u, w) = p_{\tilde{\mathcal{U}}}(u) \quad \forall u \in \tilde{\mathcal{U}}$
    - $\Sigma_{u \in \Omega(\tilde{\mathcal{U}})} p_{\tilde{\mathcal{P}}}(u, w) = p_{\tilde{\mathcal{W}}}(w) \quad \forall w \in \tilde{\mathcal{W}}$
  - The support of $\tilde{\mathcal{P}}$ is contained in R, i.e. $Pr(\mathcal{P} \in R) = 1$

3 special cases of a probabilistic match are the following

1. In the product space of $\tilde{\mathcal{U}} \times \tilde{\mathcal{W}}$ where $R = \Omega(\tilde{\mathcal{U}}) \times \Omega(\tilde{\mathcal{W}})$ and the 2 coordinates are probabilistically independent (i.e. $p_{\tilde{\mathcal{U}} \times \tilde{\mathcal{W}}} = p_{\tilde{\mathcal{U}}}(u) \cdot p_{\tilde{\mathcal{W}}}(w) \forall u \in \tilde{\mathcal{U}}, w \in \tilde{\mathcal{W}}$
2. An R-match is left-trivial if $\forall u \in \Omega_+(\tilde{\mathcal{U}})$ there is exactly one $w \in \Omega(\tilde{\mathcal{W}}$ s.t. $p_{\tilde{\mathcal{P}}}(u, w) > 0$; equivalently $Pr_{\tilde{\mathcal{P}}}(u, w) = Pr_{\tilde{\mathcal{P}}}(u)$ whenever $Pr_{\tilde{\mathcal{P}}}(u, w) > 0$
3. An R-match is right-trivial if $\forall w \in \Omega_+(\tilde{\mathcal{W}})$ there is exactly one $u \in \Omega(\tilde{\mathcal{U}}$ s.t. $p_{\tilde{\mathcal{P}}}(u, w) > 0$; equivalently $Pr_{\tilde{\mathcal{P}}}(u, w) = Pr_{\tilde{\mathcal{P}}}(w)$ whenever $Pr_{\tilde{\mathcal{P}}}(u, w) > 0$

Possible *Researcher* facts

r_e	*Researcher*(Emma, UCSD)
r_j	*Researcher*(John, UCSD)

Source p-instance $\tilde{\mathcal{I}}$

$I_1 = \{r_e, r_j, a_{eir}, a_{jdb}\}$	0.3
$I_2 = \{r_e, r_j, a_{eir}, a_{jai}\}$	0.3
$I_3 = \{r_e, r_j, a_{edb}, a_{jai}\}$	0.2
$I_4 = \{r_e, r_j, a_{edb}, a_{jdb}\}$	0.1
$I_5 = \{r_e, a_{edb}\}$	0.1

Possible *RArea* facts

a_{eir}	*RArea*(Emma, IR)
a_{edb}	*RArea*(Emma, DB)
a_{jdb}	*RArea*(John, DB)
a_{jai}	*RArea*(John, AI)

Possible *UArea* facts

u_{ir}	*UArea*(UCSD, $\bot_1$, IR)
u_{ai}	*UArea*(UCSD, $\bot_2$, AI)
u_{db}	*UArea*(UCSD, $\bot_3$, DB)

Target p-instance $\tilde{\mathcal{J}}_1$

$J_1 = \{u_{ir}, u_{db}\}$	0.3
$J_2 = \{u_{ir}, u_{ai}\}$	0.3
$J_3 = \{u_{db}, u_{ai}\}$	0.2
$J_4 = \{u_{db}\}$	0.2

Target p-instance $\tilde{\mathcal{J}}_2$

$J_5 = \{u_{ir}, u_{db}\}$	0.35
$J_6 = \{u_{ir}, u_{ai}, u_{db}\}$	0.45
$J_7 = \{u_{ir}, u_{ai}\}$	0.2

p-Solution

- (**Definition**): Let $\mathcal{M}$ be a schema mapping and let $\tilde{\mathcal{I}}$ be a source p-instance. A ***p-solution*** *for* $\tilde{\mathcal{I}}$ *w.r.t* Σ is a target instance $\tilde{\mathcal{J}}$ s.t. there is a $\text{SOL}_{\mathcal{M}}$-match of $\tilde{\mathcal{I}}$ in $\tilde{\mathcal{J}}$
- $\text{SOL}_{\mathcal{M}}$ is an R-match with $R = (I, J) \in \text{Inst}^c(\mathbf{S} \times \text{Inst}(\mathbf{T}))$

Possible *Researcher* facts

r_e	*Researcher*(Emma, UCSD)
r_j	*Researcher*(John, UCSD)

Source p-instance $\tilde{\mathcal{I}}$

$I_1 = \{r_e, r_j, a_{eir}, a_{jdb}\}$	0.3
$I_2 = \{r_e, r_j, a_{eir}, a_{jai}\}$	0.3
$I_3 = \{r_e, r_j, a_{edb}, a_{jai}\}$	0.2
$I_4 = \{r_e, r_j, a_{edb}, a_{jdb}\}$	0.1
$I_5 = \{r_e, a_{edb}\}$	0.1

Possible *RArea* facts

a_{eir}	*RArea*(Emma, IR)
a_{edb}	*RArea*(Emma, DB)
a_{jdb}	*RArea*(John, DB)
a_{jai}	*RArea*(John, AI)

Possible *UArea* facts

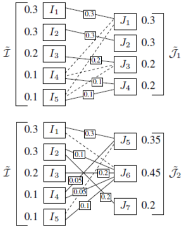
u_{ir}	*UArea*(UCSD, $\bot_1$, IR)
u_{ai}	*UArea*(UCSD, $\bot_2$, AI)
u_{db}	*UArea*(UCSD, $\bot_3$, DB)

Target p-instance $\tilde{\mathcal{J}}_1$

$J_1 = \{u_{ir}, u_{db}\}$	0.3
$J_2 = \{u_{ir}, u_{ai}\}$	0.3
$J_3 = \{u_{db}, u_{ai}\}$	0.2
$J_4 = \{u_{db}\}$	0.2

Target p-instance $\tilde{\mathcal{J}}_2$

$J_5 = \{u_{ir}, u_{db}\}$	0.35
$J_6 = \{u_{ir}, u_{ai}, u_{db}\}$	0.45
$J_7 = \{u_{ir}, u_{ai}\}$	0.2



Properties of a SOL$_{\mathcal{M}}$-match

- **Theorem**: Let $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ be a schema mapping. Let $\tilde{\mathcal{I}}$ be a source p-instance and let $\tilde{\mathcal{J}}$ be a target p-instance. The following are equivalent:
  - $\tilde{\mathcal{J}}$ is a p-solution (i.e. a SOL$_{\mathcal{M}}$-match of $\tilde{\mathcal{I}}$ in $\tilde{\mathcal{J}}$ exists)
  - $\forall E \subseteq \text{Inst}^c(\mathbf{S})$, $Pr_{\tilde{\mathcal{J}}}(\bigvee_{I \in E} \langle I, \mathcal{J} \rangle \models \Sigma) \geqslant Pr_{\tilde{\mathcal{I}}}(E)$
  - $\forall F \subseteq \text{Inst}(\mathbf{T})$, $Pr_{\tilde{\mathcal{I}}}(\bigvee_{J \in F} \langle \mathcal{I}, J \rangle \models \Sigma) \geqslant Pr_{\tilde{\mathcal{J}}}(F)$
- **Lemma**: Let $\tilde{\mathcal{U}}$ and $\tilde{\mathcal{W}}$ be two p-spaces and let $R \subseteq \Omega(\tilde{\mathcal{U}}) \times \Omega(\tilde{\mathcal{W}})$ be a binary relation. There exists an R-match of $\tilde{\mathcal{U}}$ in $\tilde{\mathcal{W}}$ iff $\forall$ events U of $\tilde{\mathcal{U}}$ it holds that $Pr_{\tilde{\mathcal{U}}}(U) \leqslant Pr_{\tilde{\mathcal{W}}}(\bigvee_{u \in U} R(u, \tilde{\mathcal{W}}))$

Universal p-solutions and query answering

- USOL$_{\mathcal{M}}$ is the relationship between pairs (I, J) of (ordinary) source and target instances, respectively, s.t. USOL$_{\mathcal{M}}$(I, J) holds iff J is a universal solution for I
- **Definition**: Let $\mathcal{M}$ be a schema mapping. Let $\tilde{\mathcal{I}}$ and $\tilde{\mathcal{J}}$ be source and target p-instances, respectively. $\tilde{\mathcal{J}}$ is a **universal p-solution** (for $\tilde{\mathcal{I}}$ w.r.t Σ) if there is a USOL$_{\mathcal{M}}$-match of $\tilde{\mathcal{I}}$ and $\tilde{\mathcal{J}}$

Possible *Researcher* facts

r_e	*Researcher*(Emma, UCSD)
r_j	*Researcher*(John, UCSD)

Possible *RArea* facts

a_{eir}	*RArea*(Emma, IR)
a_{edb}	*RArea*(Emma, DB)
a_{jdb}	*RArea*(John, DB)
a_{jai}	*RArea*(John, AI)

Source p-instance $\tilde{\mathcal{I}}$

$I_1 = \{r_e, r_j, a_{eir}, a_{jdb}\}$	0.3
$I_2 = \{r_e, r_j, a_{eir}, a_{jai}\}$	0.3
$I_3 = \{r_e, r_j, a_{edb}, a_{jai}\}$	0.2
$I_4 = \{r_e, r_j, a_{edb}, a_{jdb}\}$	0.1
$I_5 = \{r_e, a_{edb}\}$	0.1

Possible *UArea* facts

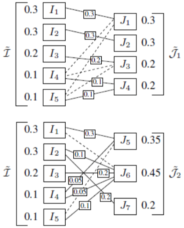
u_{ir}	*UArea*(UCSD, $\perp_1$, IR)
u_{ai}	*UArea*(UCSD, $\perp_2$, AI)
u_{db}	*UArea*(UCSD, $\perp_3$, DB)

Target p-instance $\tilde{\mathcal{J}}_1$

$J_1 = \{u_{ir}, u_{db}\}$	0.3
$J_2 = \{u_{ir}, u_{ai}\}$	0.3
$J_3 = \{u_{db}, u_{ai}\}$	0.2
$J_4 = \{u_{db}\}$	0.2

Target p-instance $\tilde{\mathcal{J}}_2$

$J_5 = \{u_{ir}, u_{db}\}$	0.35
$J_6 = \{u_{ir}, u_{ai}, u_{db}\}$	0.45
$J_7 = \{u_{ir}, u_{ai}\}$	0.2



Existence of a p-solution and a universal p-solution

- **Proposition** Let $\mathcal{M}$ be a schema mapping and let $\tilde{\mathcal{I}}$ be a source p-instance. A p-solution exists iff a solution exists $\forall I \in \Omega_+(\tilde{\mathcal{I}})$. Similarly, a universal p-solution exists iff a universal solution exists $\forall I \in \Omega_+(\tilde{\mathcal{I}})$.
- In the deterministic case, the notion of generality w.r.t. a universal solution is defined by means of a homomorphism (i.e. J_1 *generalizes* J_2 if $J_1 \rightarrow J_2$.

Generalizing the notion of homomorphism to p-instances:

- using the probabilistic match to extend the notion of homomorphism to p-instances: Let **T** be a schema. $\text{HOM}_{\mathbf{T}}$ then is the binary relation that includes all the pairs $(J_1, J_2) \in (\text{Inst}(\mathbf{T}))^2$ s.t. $J_1 \rightarrow J_2$. Consider two p-instances $\tilde{\mathcal{J}_\infty}$ and $\tilde{\mathcal{J}_\in}$ over **T**. $\tilde{\mathcal{J}_\infty} \xrightarrow{\text{mat}} \tilde{\mathcal{J}_\in}$ denotes that there is a $\text{HOM}_{\mathbf{T}}$-match of $\tilde{\mathcal{J}_\infty}$ in $\tilde{\mathcal{J}_\in}$
- **stochastic order** Let **T** be a schema. The existence of a homomorphism relationship can be viewed as a preorder over Inst(**T**) (c.f. the literature):
  - $J \preceq_{sp} J'$ is interpreted as $J \rightarrow J'$ (J is at most as specific as J'). The stochastic extension is $\tilde{\mathcal{J}_\infty} \xrightarrow{\preceq sp} \tilde{\mathcal{J}_\in}$ if $Pr(\mathcal{J}_\infty \rightarrow J) \geqslant Pr(\mathcal{J}_\in \rightarrow J)$ $\forall$ instances J over **T**
  - $J \preceq_{ge} J'$ is interpreted as $J' \rightarrow J$ (J is at most as general as J'). The stochastic extension is $\tilde{\mathcal{J}_\in} \xleftarrow{\preceq ge} \tilde{\mathcal{J}_\infty}$ if $Pr(J \rightarrow \mathcal{J}_\in) \geqslant Pr(J \rightarrow \mathcal{J}_\in)$ $\forall$ instances J over **T**

THEOREM 4.8. *Let $\mathcal{M}$ be a schema mapping. Let $\tilde{\mathcal{I}}$ be a source p-instance and let $\tilde{\mathcal{J}}$ be a p-solution. The following are equivalent.*

(1) *$\tilde{\mathcal{J}}$ is a universal p-solution (i.e., there is a $\text{USOL}_{\mathcal{M}}$-match of $\tilde{\mathcal{I}}$ in $\tilde{\mathcal{J}}$).*
(2) *$\tilde{\mathcal{J}} \xrightarrow{\text{mat}} \tilde{\mathcal{J}}'$ for all p-solutions $\tilde{\mathcal{J}}'$.*
(3) *$\tilde{\mathcal{J}} \xrightarrow{\preceq\text{sp}} \tilde{\mathcal{J}}'$ for all p-solutions $\tilde{\mathcal{J}}'$.*
(4) *$\tilde{\mathcal{J}} \xrightarrow{\succeq\text{ge}} \tilde{\mathcal{J}}'$ for all p-solutions $\tilde{\mathcal{J}}'$.*
(5) *Every $\text{SOL}_{\mathcal{M}}$-match of $\tilde{\mathcal{I}}$ in $\tilde{\mathcal{J}}$ is a $\text{USOL}_{\mathcal{M}}$-match.*

Conclusions:

- Information Integration on the Semantic Web by means of generalized bayesian dl-programs and tightly coupled dl-programs
- Data Integration with Uncertainty (by-table semantics and by-tuple semantics)
- Generalized Framework of Probabilistic Data Exchange
    - Generalization of Data Integration with Uncertainty based on by-table semantics

Outlook/Research questions:

- by-tuple semantics?
- more complex probability distributions?
- Certain Answers, tuple generating dependencies, . . . in the SW framework?