Data Integration with Uncertainty

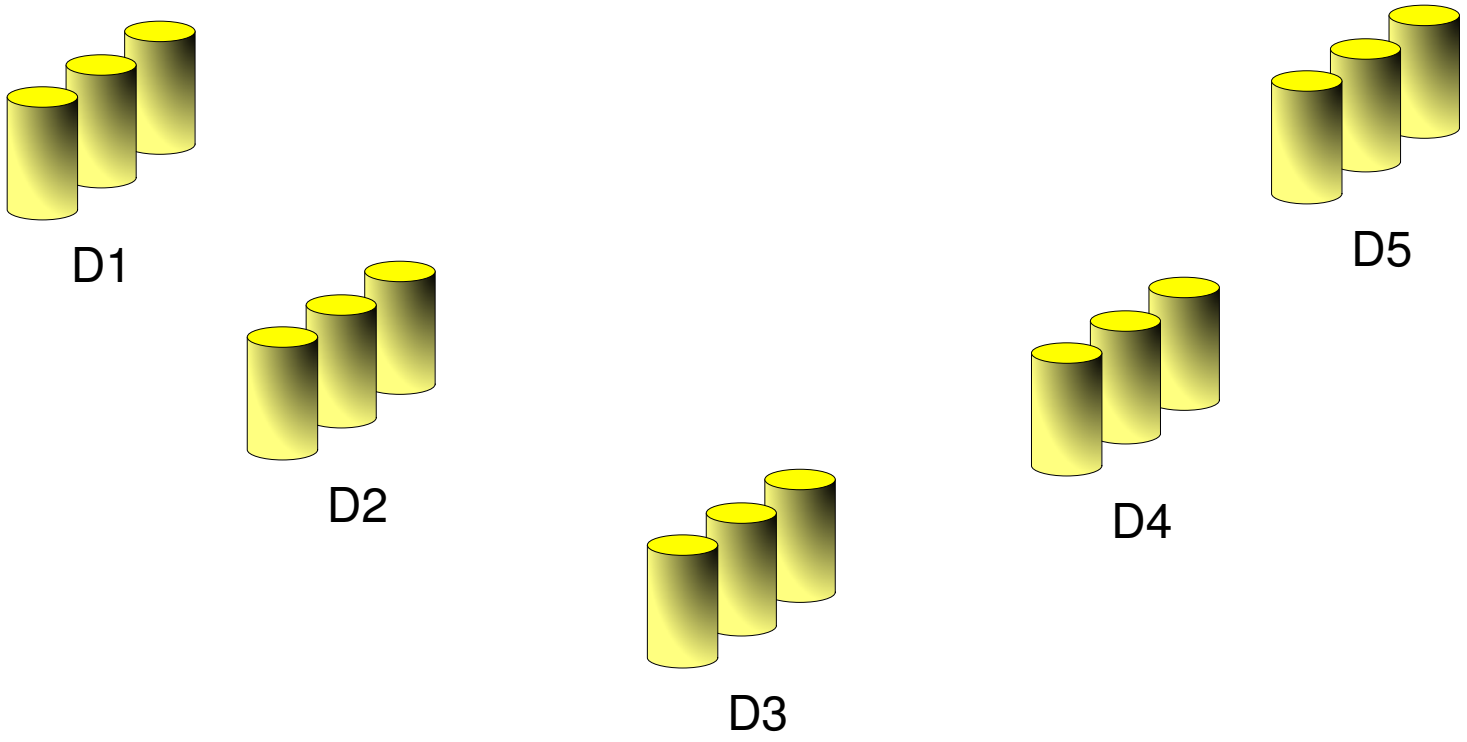
Xin (Luna) Dong
Univ. of Washington

Alon Halevy
Google Inc.

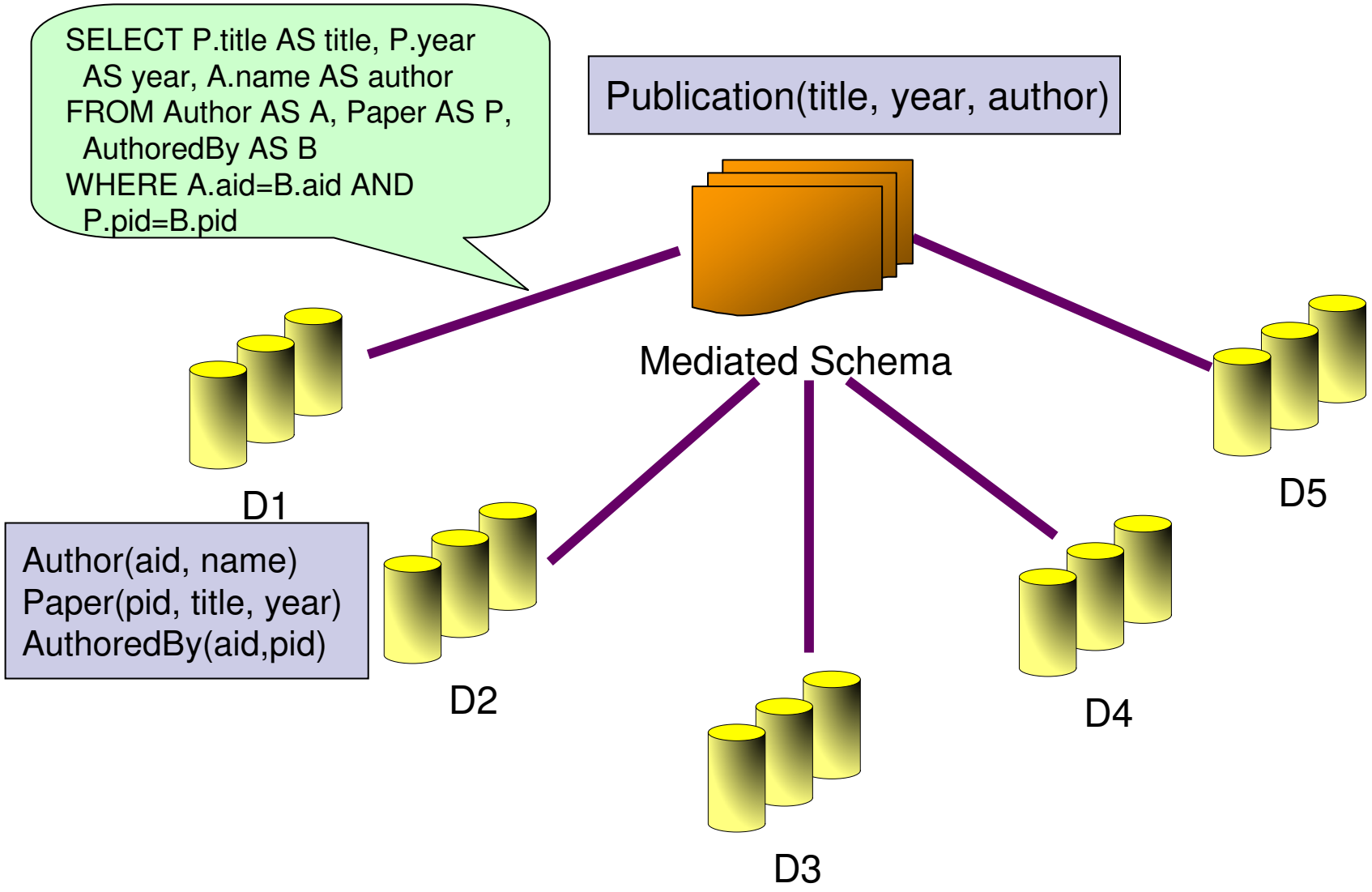
Cong Yu
Univ. of Michigan

@ VLDB 2007

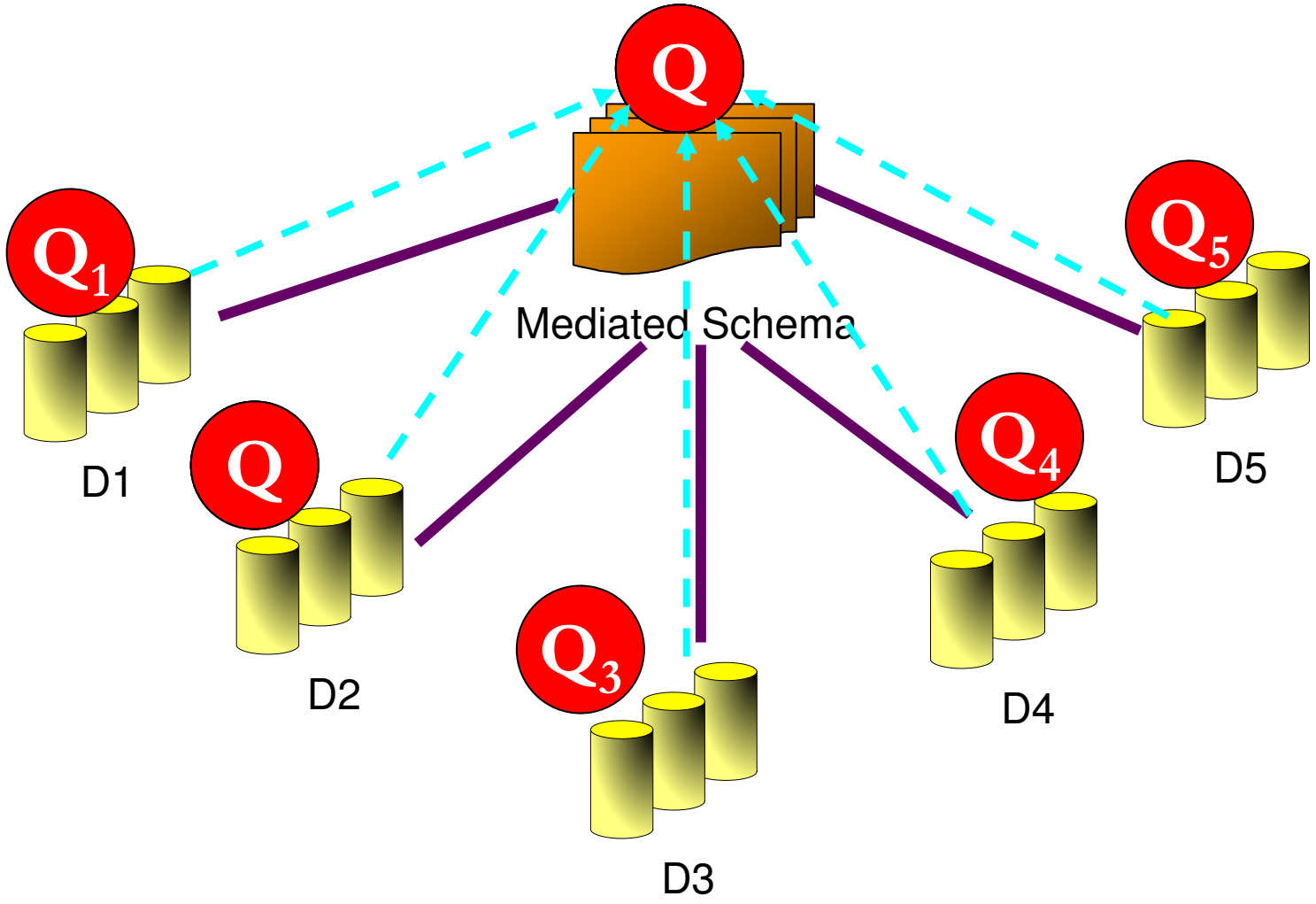
Many Applications Need to Manage Heterogeneous Data Sources



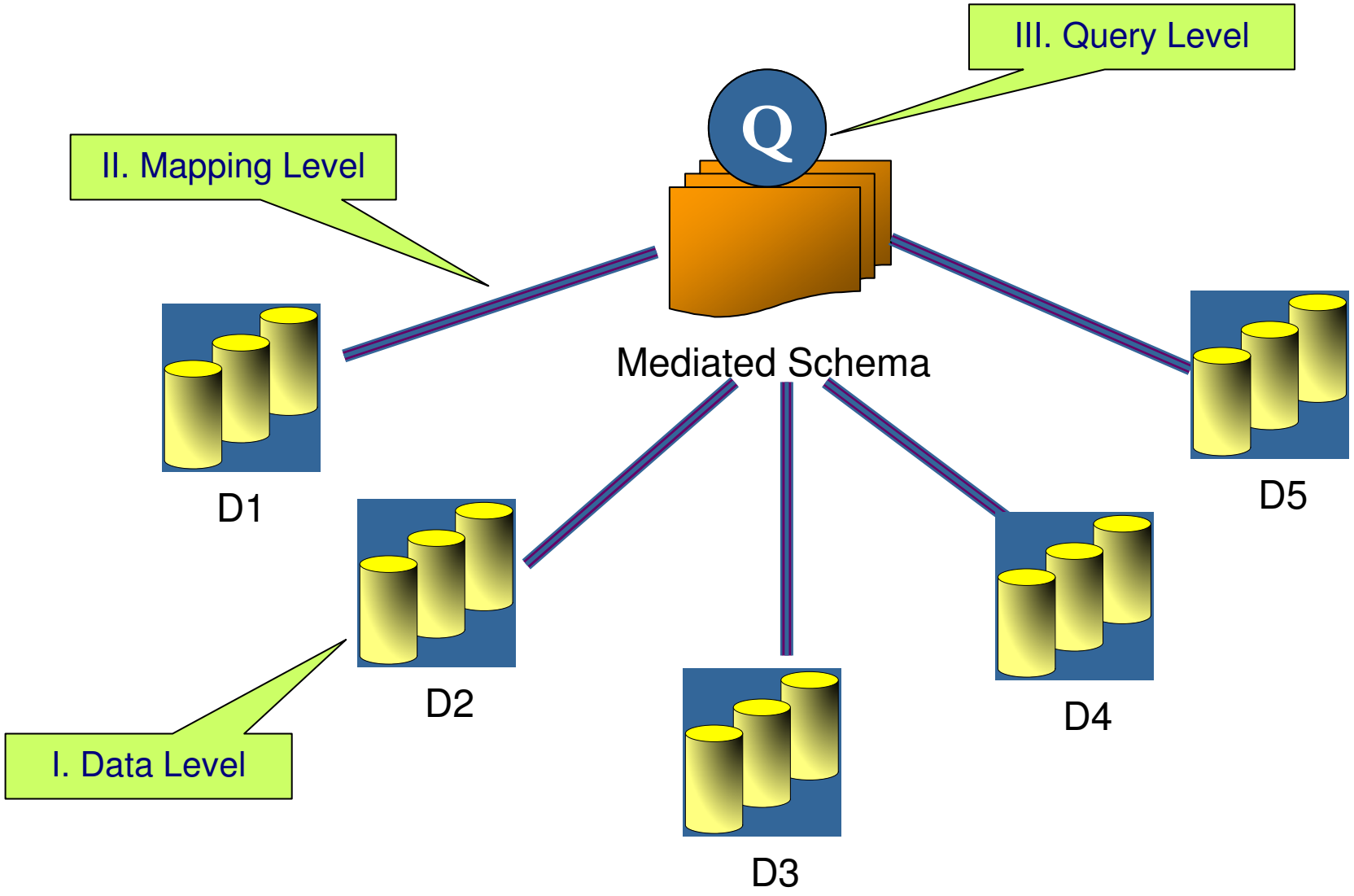
Traditional Data Integration Systems



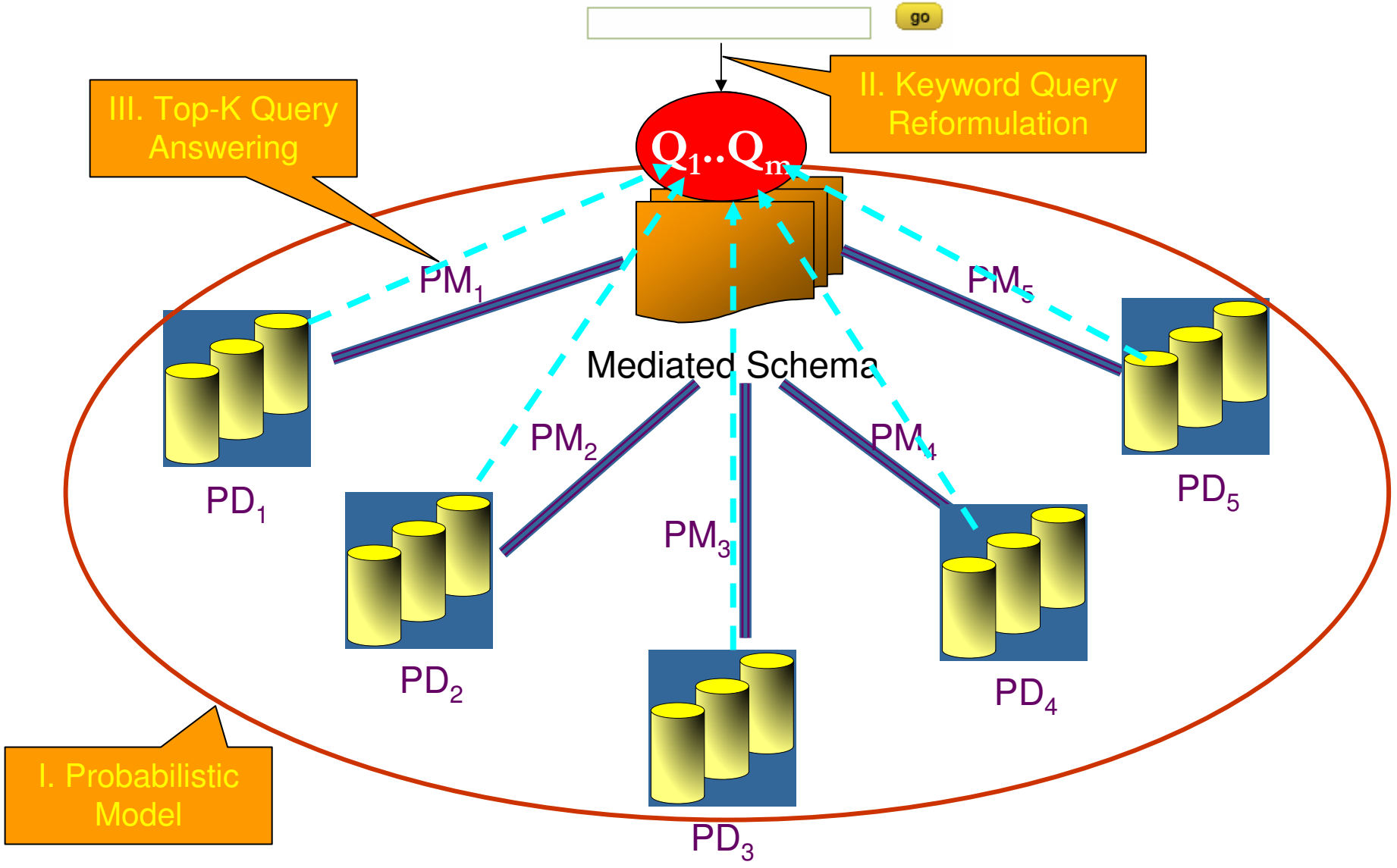
Querying on Traditional Data Integration Systems



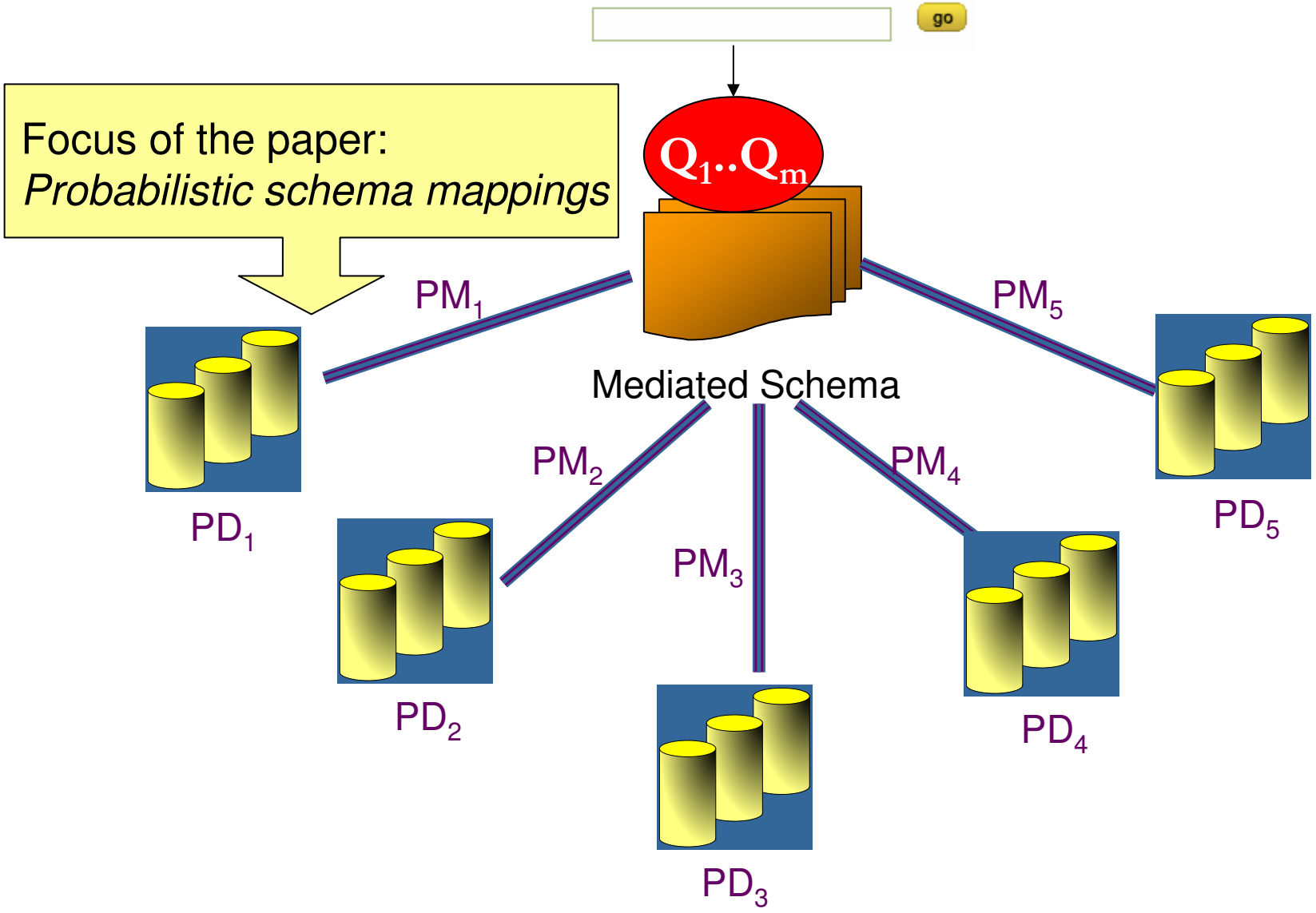
Uncertainty Can Occur at Three Levels in Data Integration Applications



Data Integration with Uncertainty

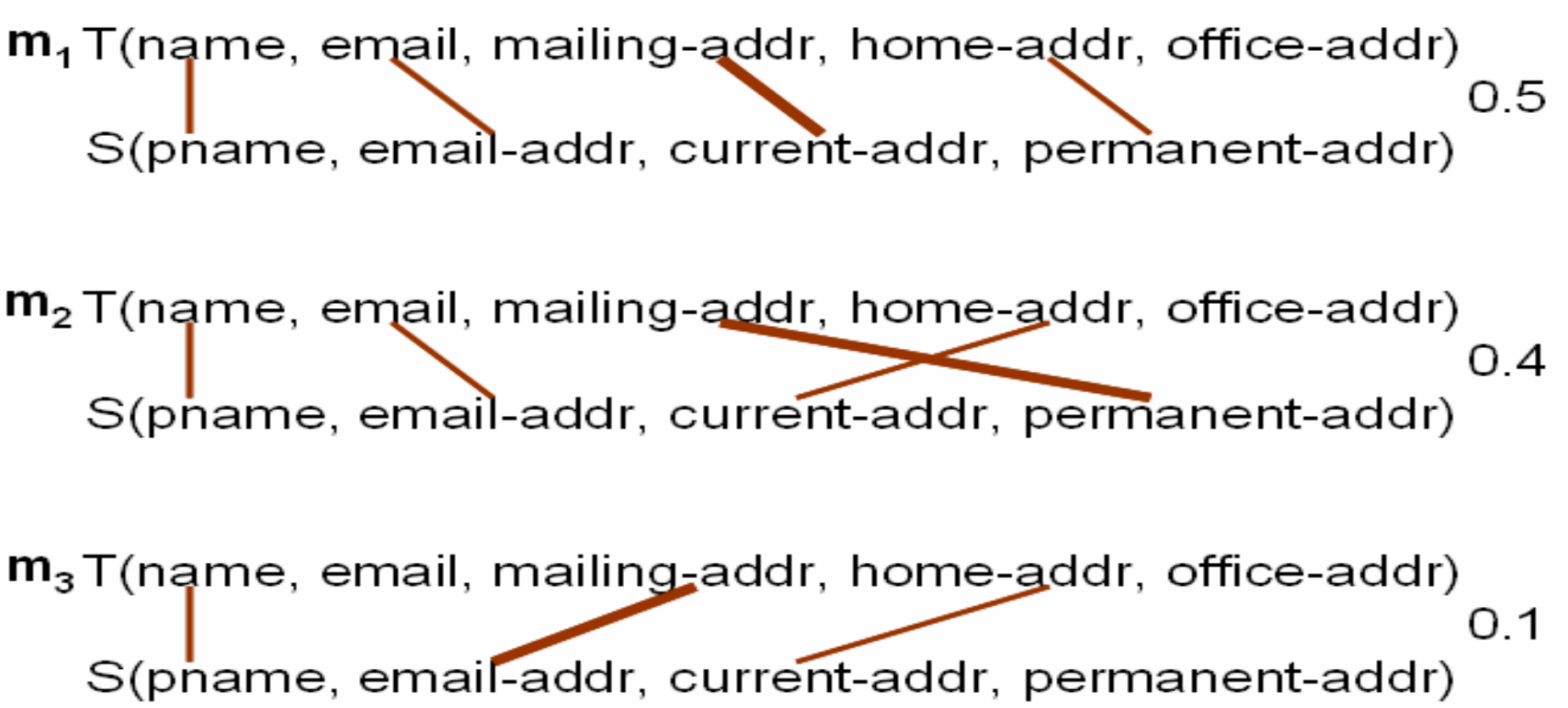


Data Integration with Uncertainty



Example Probabilistic Mappings

T(name, email, mailing-addr, home-addr, office-addr)



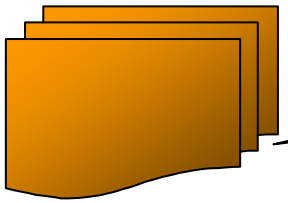
S

PName	Email-addr	Current-addr	Permanent-addr
Alice	alice@	Mountain View	Sunnyvale
Bob	bob@	Sunnyvale	Sunnyvale

Top-k Query Answering w.r.t. Probabilistic Mappings

T(name, email, mailing-addr, home-addr, office-addr)

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



Q: SELECT mailing-addr FROM T

Mediated Schema

m_1 T(name, email, mailing-addr, home-addr, office-addr)

S(pname, email-addr, current-addr, permanent-addr)

0.5

Q1: SELECT current-addr FROM S

m_2 T(name, email, mailing-addr, home-addr, office-addr)

S(pname, email-addr, current-addr, permanent-addr)

0.4

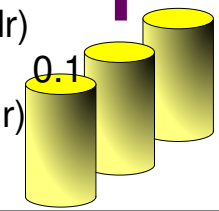
Q2: SELECT permanent-addr FROM S

m_3 T(name, email, mailing-addr, home-addr, office-addr)

S(pname, email-addr, current-addr, permanent-addr)

0.1

Q3: SELECT email-addr FROM S



S

PName	Email-addr	Current-addr	Permanent-addr
Alice	alice@	Mountain View	Sunnyvale
Bob	bob@	Sunnyvale	Sunnyvale

Contributions

- Definition of probabilistic mappings
  Semantics: by-table v.s. by-tuple
- Complexity of query answering with respect to probabilistic mappings

	By-table	By-tuple
Data Complexity	PTIME	#P-complete
Mapping Complexity	PTIME	PTIME

- Concise representations of probabilistic mappings
- More expressive extensions to probabilistic mappings

Outline

- ☑ Motivation and overview of our results
- ☞ Definition of probabilistic mappings
- Complexity of query answering

	By-table	By-tuple
Data Complexity	PTIME	#P-complete
Mapping Complexity	PTIME	PTIME

- Concise representations of probabilistic mappings
- More expressive extensions to probabilistic mappings

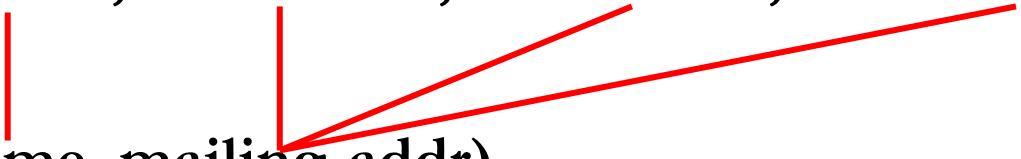
Schema Mapping

- S=(pname, email-addr, home-addr, office-addr)
- T=(name, mailing-addr)

- Mappings: one-to-one schema matching
- Queries: select-project-join queries

Probabilistic Mapping

- S=(pname, email-addr, home-addr, office-addr)
- T=(name, mailing-addr)



Possible Mapping	Probability
{(pname,name),(home-addr, mailing-addr)}	0.5
{(pname,name),(office-addr, mailing-addr)}	0.4
{(pname,name),(email-addr, mailing-addr)}	0.1

By-Table v.s. By-Tuple Semantics

By-Table v.s. By-Tuple Semantics

Possible Mapping	Probability
{(pname,name),(home-addr, mailing-addr)}	0.5
{(pname,name),(office-addr, mailing-addr)}	0.4
{(pname,name),(email-addr, mailing-addr)}	0.1

D_S=

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

D_T=

name	mailing-addr
Alice	Mountain View
Bob	Sunnyvale

$Pr(m_1)=0.5$

name	mailing-addr
Alice	Sunnyvale
Bob	Sunnyvale

$Pr(m_2)=0.4$

name	mailing-addr
Alice	alice@
Bob	bob@

$Pr(m_3)=0.1$

By-Table v.s. By-Tuple Semantics

Possible Mapping	Probability
{(pname,name),(home-addr, mailing-addr)}	0.5
{(pname,name),(office-addr, mailing-addr)}	0.4
{(pname,name),(email-addr, mailing-addr)}	0.1

Ds=

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

D_T=

name	mailing-addr
Alice	Mountain View
Bob	bob@

$Pr(<m_1,m_3>)=0.05$

name	mailing-addr
Alice	Sunnyvale
Bob	bob@

$Pr(<m_2,m_3>)=0.04$

name	mailing-addr
Alice	alice@
Bob	bob@

$Pr(<m_3,m_3>)=0.01$

…

By-Table Query Answering

D_S=

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

D_T=

name	mailing-addr
Alice	Mountain View
Bob	Sunnyvale

0.5

name	mailing-addr
Alice	Sunnyvale
Bob	Sunnyvale

0.4

name	mailing-addr
Alice	alice@
Bob	bob@

0.1

```
SELECT mailing-addr
FROM T
```

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

By-Tuple Query Answering

D_S=

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

D_T=

name	mailing-addr
Alice	Mountain View
Bob	bob@

0.05

name	mailing-addr
Alice	Sunnyvale
Bob	bob@

0.04

name	mailing-addr
Alice	alice@
Bob	bob@

0.01

…

```
SELECT mailing-addr
FROM T
```

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

Outline

- ☑ Motivation and overview of our results
- ☑ Definition of probabilistic mappings
- ☞ Complexity of query answering

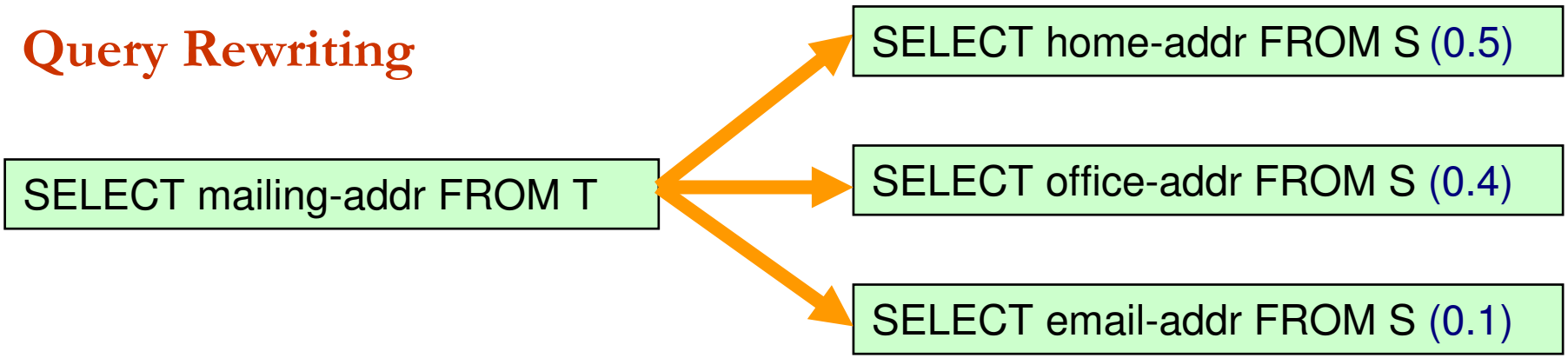
	By-table	By-tuple
Data Complexity	PTIME	#P-complete
Mapping Complexity	PTIME	PTIME

- Concise representations of probabilistic mappings
- More expressive extensions to probabilistic mappings

By-Table Query Answering

Possible Mapping	Probability
{(pname,name),(home-addr, mailing-addr)}	0.5
{(pname,name),(office-addr, mailing-addr)}	0.4
{(pname,name),(email-addr, mailing-addr)}	0.1

Query Rewriting



Theorem: Query answering in by-table semantics is in *PTIME* in the size of the data and the size of the mapping

By-Tuple Query Answering

D_S=

pname	email-addr	home-addr	office-addr
Alice	alice@	Mountain View	Sunnyvale
Bob	bob@	Sunnyvale	San Jose

Target Enumeration D_T=

SELECT mailing-addr
FROM T

name	mailing-addr
Alice	Mountain View
Bob	@bob

0.05

name	mailing-addr
Alice	Sunnyvale
Bob	@bob

0.04 •••

name	mailing-addr
Alice	alice@
Bob	bob@

0.01

name	mailing-addr
Alice	Sunnyvale
Bob	Sunnyvale

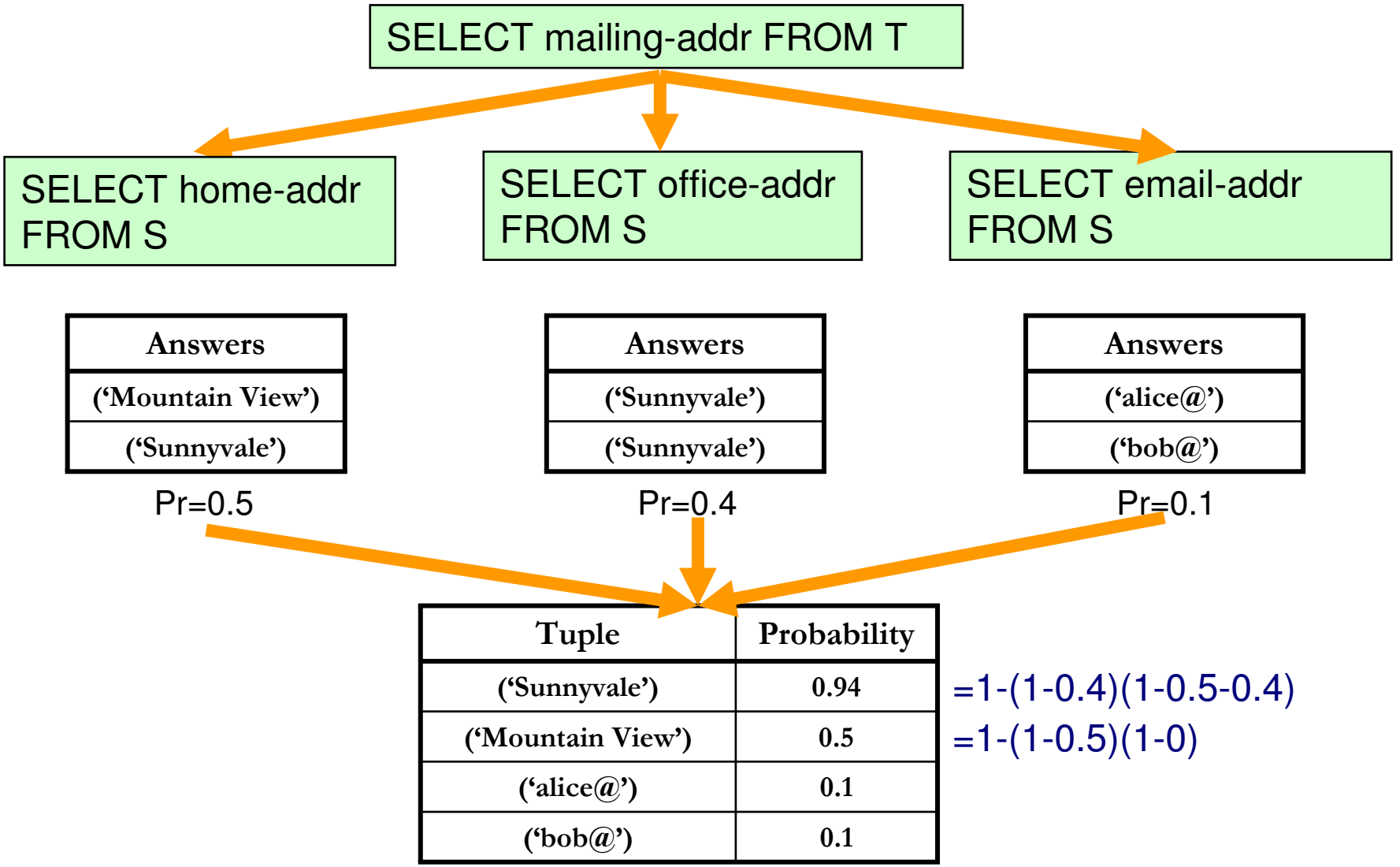
0.2

Theorem: Query answering in by-tuple semantics is *#P-complete* in the size of the data, and in *PTIME* in the size of the mapping

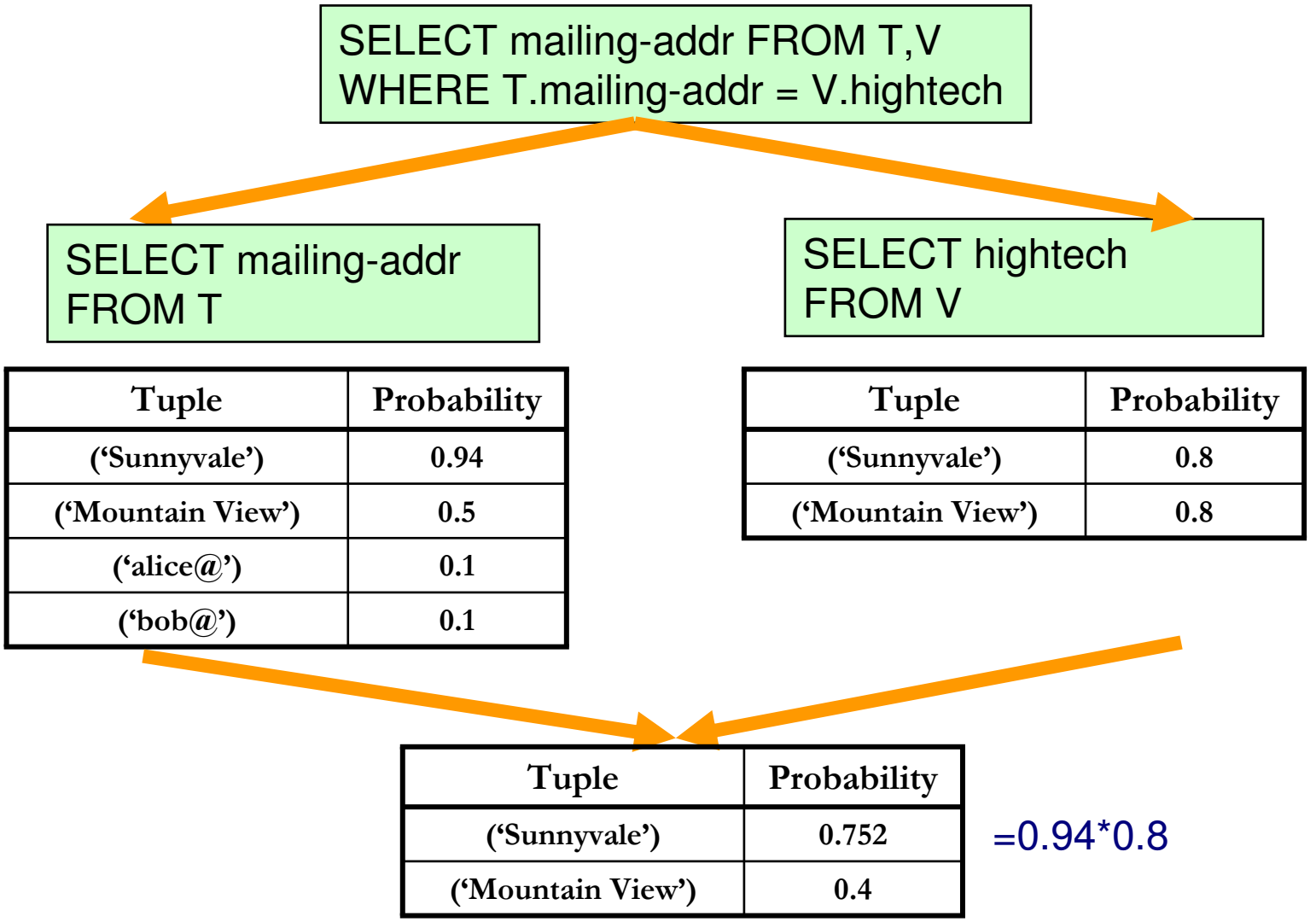
More on By-Tuple Query Answering

- The high complexity comes from computing probabilities
  - Theorem: Even computing the probability for one possible answer is #P-hard
  - Theorem: Computing all possible answers w/o probabilities is in PTIME
- In general query answering cannot be done by query rewriting
- There are two subsets of queries that can be answered in PTIME by query rewriting

PTIME for Queries with a Single P-Mapping Target



PTIME for Queries that Return Join Attributes



Query Rewriting Does Not Apply to Queries that Do NOT Return Join Attributes

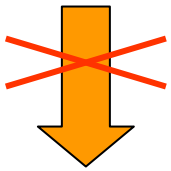
SELECT 'true' FROM T,V
WHERE T.mailing-addr = V.hightech

SELECT mailing-addr
FROM T

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

SELECT hightech
FROM V

Tuple	Probability
('Sunnyvale')	0.8
('Mountain View')	0.8



Tuple	Probability
('true')	0.864

≠0.94*0.8+0.5*0.8

Outline

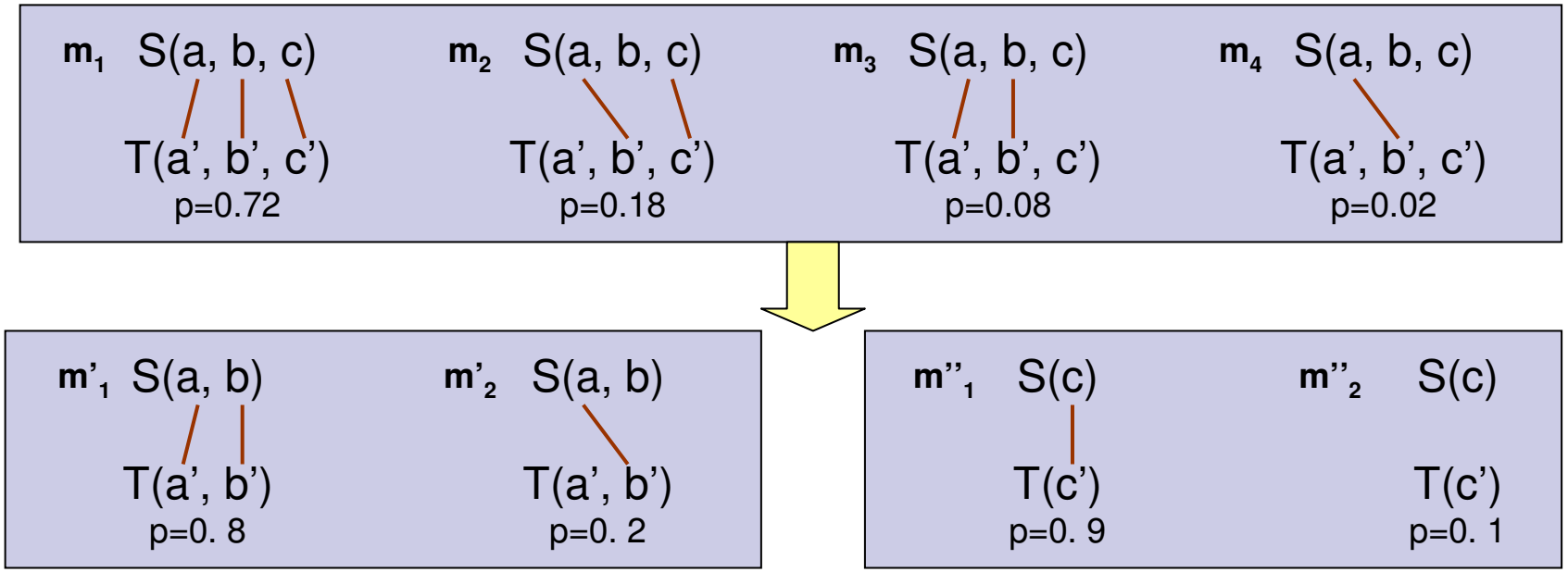
- ☑ Motivation and overview of our results
- ☑ Definition of probabilistic mappings
- ☑ Complexity of query answering

	By-table	By-tuple
Data Complexity	PTIME	#P-complete
Mapping Complexity	PTIME	PTIME

- ☞ Concise representations of probabilistic mappings
- ☐ More expressive extensions to probabilistic mappings

Group Probabilistic Mapping

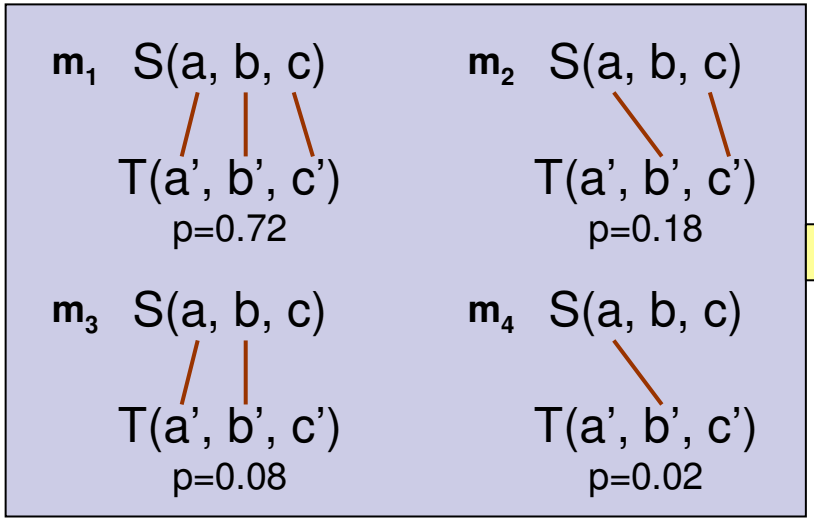
- Intuition: Divide a p-mapping into a set of independent mappings on disjoint attributes



- Theorem: Query answering is in PTIME in the size of the group probabilistic mapping

Probabilistic Correspondences

- Intuition: Compute marginal probabilities for attribute pairs



Attr Correspondence	Probability
(a, a')	0.8
(a, b')	0.2
(b, b')	0.8
(c, c')	0.9

- Theorem: Query answering is in PTIME in the size of the probabilistic correspondence
- Many-to-one relationship between probabilistic mappings and probabilistic correspondences
- We can use probabilistic correspondences to answer ONLY queries that contain no more than one attribute

Bayes Nets Representation

- Theorem: When we encode probabilistic mappings using a Bayes Net, query answering can be exponential in the size of the representation
- Example:

  $S(s_1, s_2, \ldots, s_n, s_1', s_2', \ldots, s_n')$

  $T(t_1, t_2, \ldots, t_n)$

  --$Pr(s_1, t_1) = Pr(s_1', t_1) = .5$

  --if s_1 maps to t_1, then for $2<=i<=n$, s_i maps to t_i with probability .9 and s_i' maps to t_i with probability .1; and the other way around

  Original representation: $O(2^n)$; Bayes representation: $O(n)$

Outline

- ☑ Motivation and overview of our approach
- ☑ Definition of probabilistic mappings
- ☑ Complexity of query answering

	By-table	By-tuple
Data Complexity	PTIME	#P-complete
Mapping Complexity	PTIME	PTIME

- ☑ Concise representations of probabilistic mappings
- ☞ More expressive extensions to probabilistic mappings

Extensions to More Expressive Mappings

- The complexity results for query answering carry over to three extensions to more expressive mappings
  - Complex mappings
    - E.g., address → street, city and state
  - GLAV mappings
    - E.g., Paper ⋈ Authorship ⋈ Author → Publication
  - Conditional mappings:
    - E.g., if age>65, Pr(home-addr → mailing-addr)=0.8
      if age<=65, Pr(home-addr → mailing-addr)=0.5

Related Work

- Using top-k schema mappings can increase the recall of query answering w/o sacrificing precision much [Magnani&Montesi, 2007]
- Using top-k schema mappings to improve schema mapping [Gal, 2006]
- Using probabilities to model overlaps between data sources [Florescu et al, 1997]
- Probabilistic Databases
  (Tutorial [Suciu&Dalvi, Sigmod'05])

Conclusions

- Contributions
  - Data integration with uncertainty
  - Probabilistic mappings and query answering in their presence

	By-table	By-tuple
Data Complexity	PTIME	#P-complete
Mapping Complexity	PTIME	PTIME

- Future work
  - Incorporate uncertainty on keyword reformulation and dirty data
  - Use the probabilistic model to improve schema mappings
  - Build a real system

Data Integration with Uncertainty

Xin (Luna) Dong
Univ. of Washington

Alon Halevy
Google Inc.

Cong Yu
Univ. of Michigan

@ VLDB 2007