Learning mappings and queries

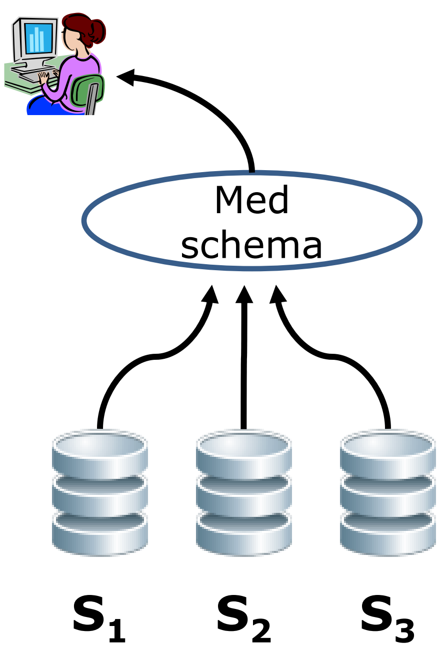
Marie Jacob
University Of Pennsylvania

DEIS 2010

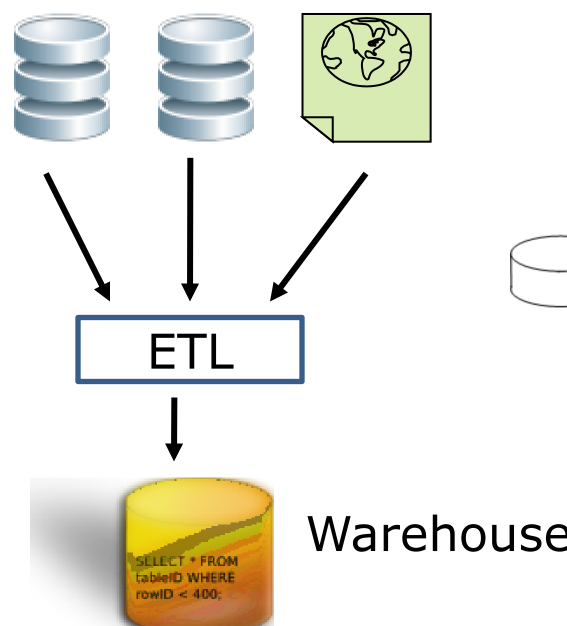
Schema mappings

- Denote relationships between schemas
- Relates source schema S and target schema T
- Defined in a query language like Datalog or first-order logic.

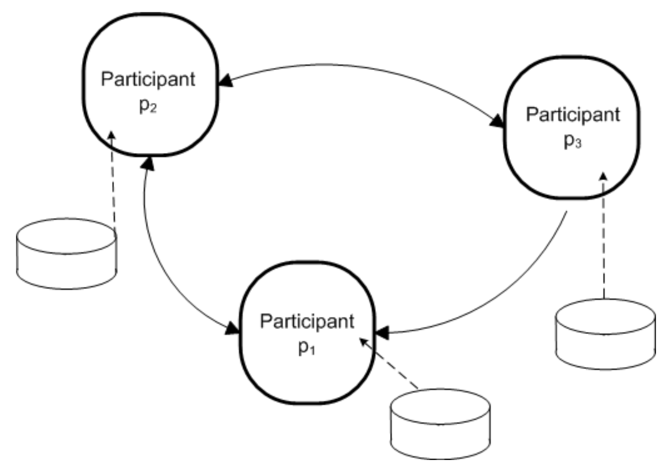
Data Integration



Data Warehousing



Data Sharing

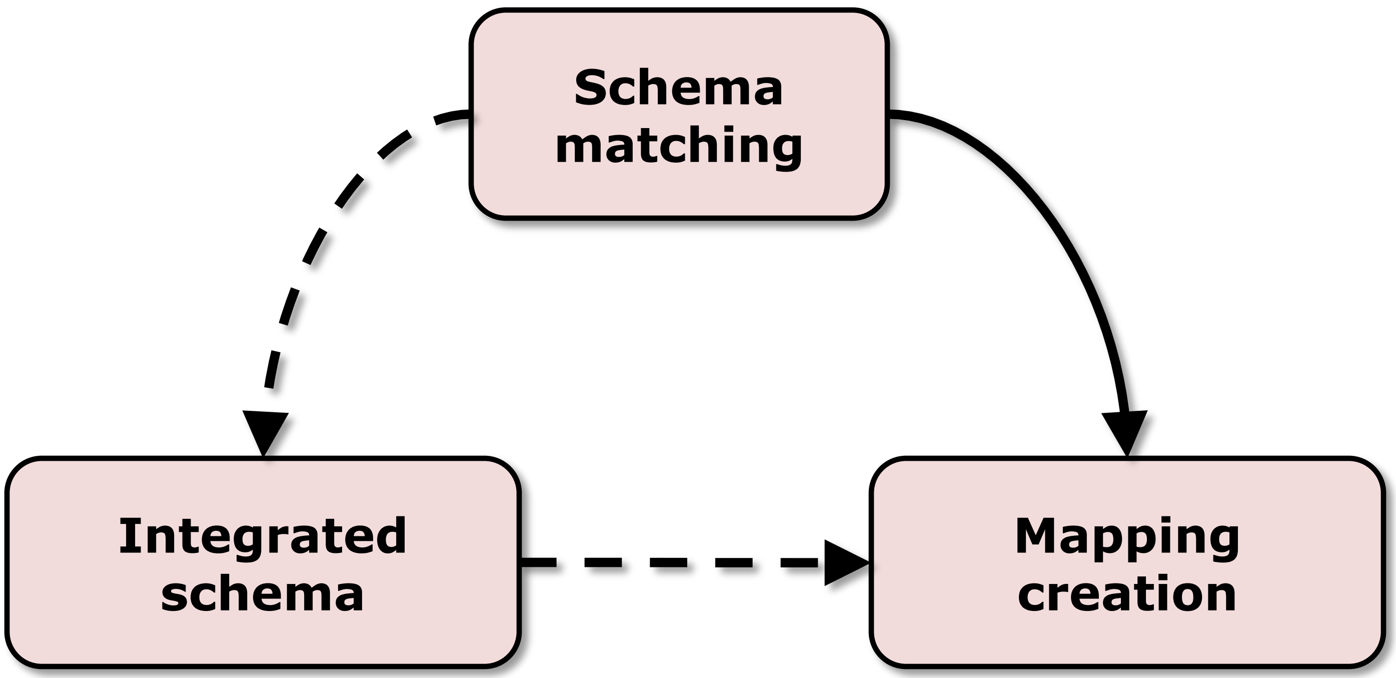


Informal problem

- Two variants:
  - Given schemas S and T, and instances I and J, find a set of s-t mappings that "naturally" translate S to T.
  - Given a set of schemas, S_1, S_2..., S_n, find an integrated schema that best reflects combination of all source schemas, and their corresponding s-t mappings.

Mapping Tasks

S-Match [Giunchigla et al, 2007]
iMap [Dhamankar et al, 2004]

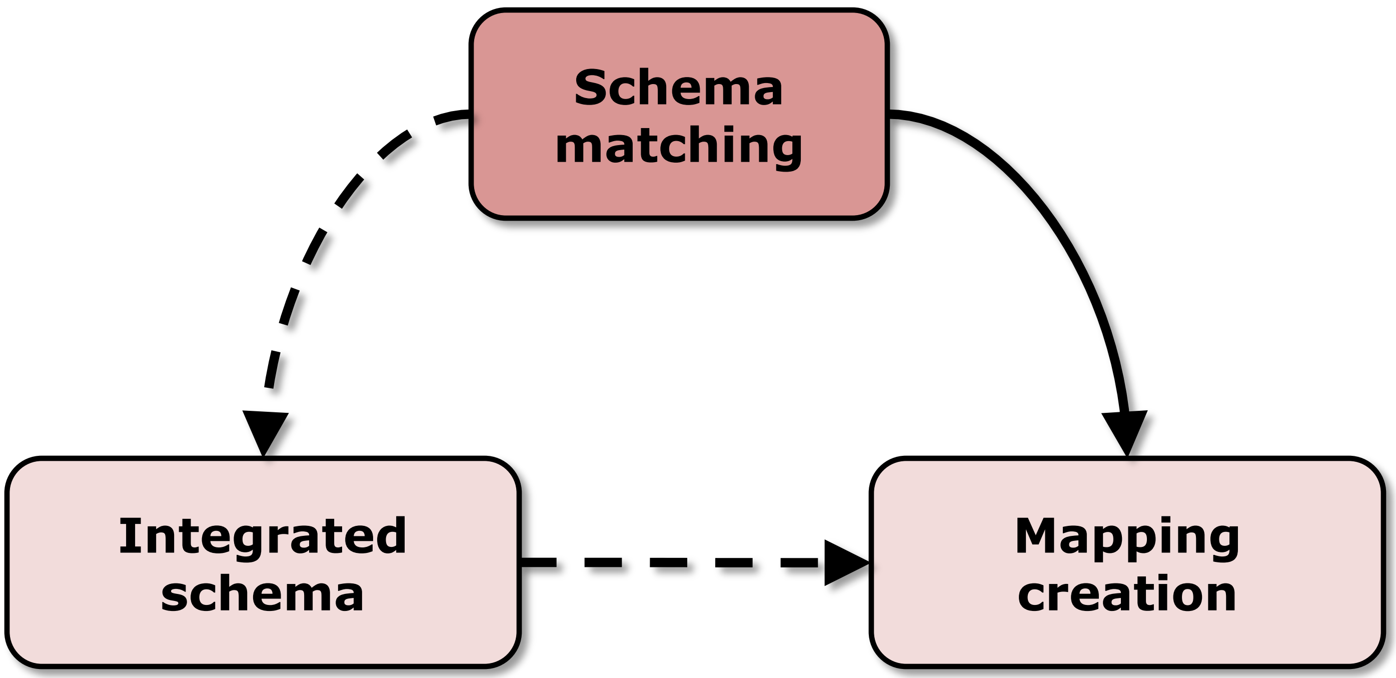


[Chiticariu et al, 2008]
[Das Sarma et al, 2008]

Clio [Miller et al, 2000]

Mapping Tasks

S-Match [Giunchigla et al, 2007]
iMap [Dhamankar et al, 2004]



[Chiticariu et al, 2008]
[Das Sarma et al, 2008]

Clio [Miller et al, 2000]

Schema matching

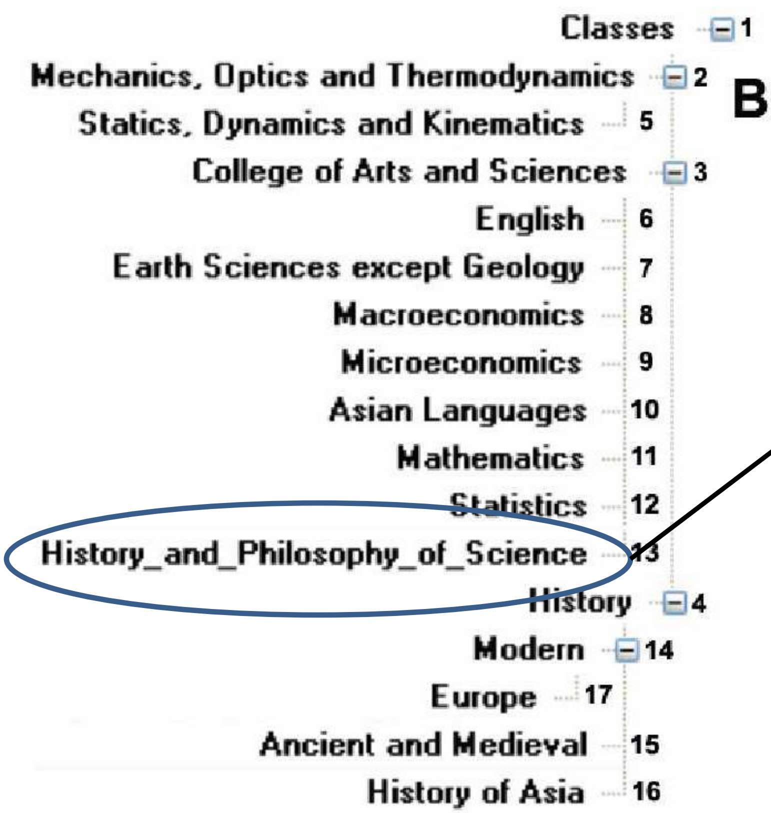
- Determine if two attributes relate to each other.
  - Is Employee(id) the same as Emp(eid)?
- Challenges:
  - Heterogeneity.
  - Types of relationships.
  - Complex matches.

S-Match [Giunchiglia et al, 2007]

- Matches elements in source and target tree-structured models (e.g.XML)
- Abstracts labels into high-level concepts, encoded in description logic.
- Label A has concept C_A
- Classifies pairwise concepts, C_A, C_B:
  - $C_A = C_B$ (equivalent)
  - $C_A \sqsubseteq C_B$ (less general)
  - $C_A \sqsupseteq C_B$ (more general)
  - $C_A \perp C_B$ (disjoint)

S-Match [Giunchiglia et al, 2007]

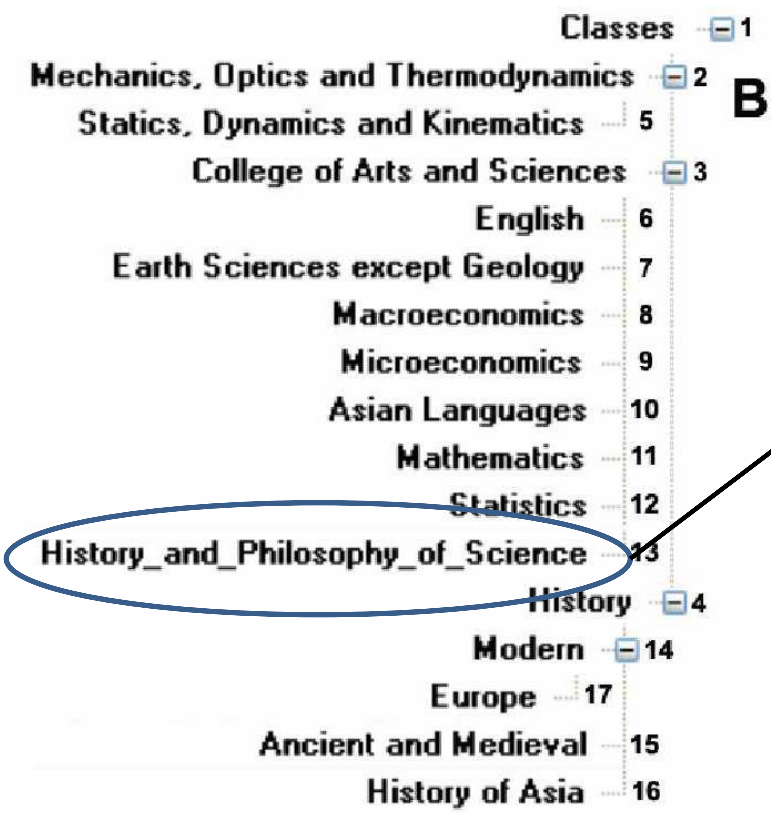
1. Compute concept of labels



$(C_{history} \sqcup C_{philosophy}) \sqcap C_{science}$

S-Match [Giunchiglia et al, 2007]

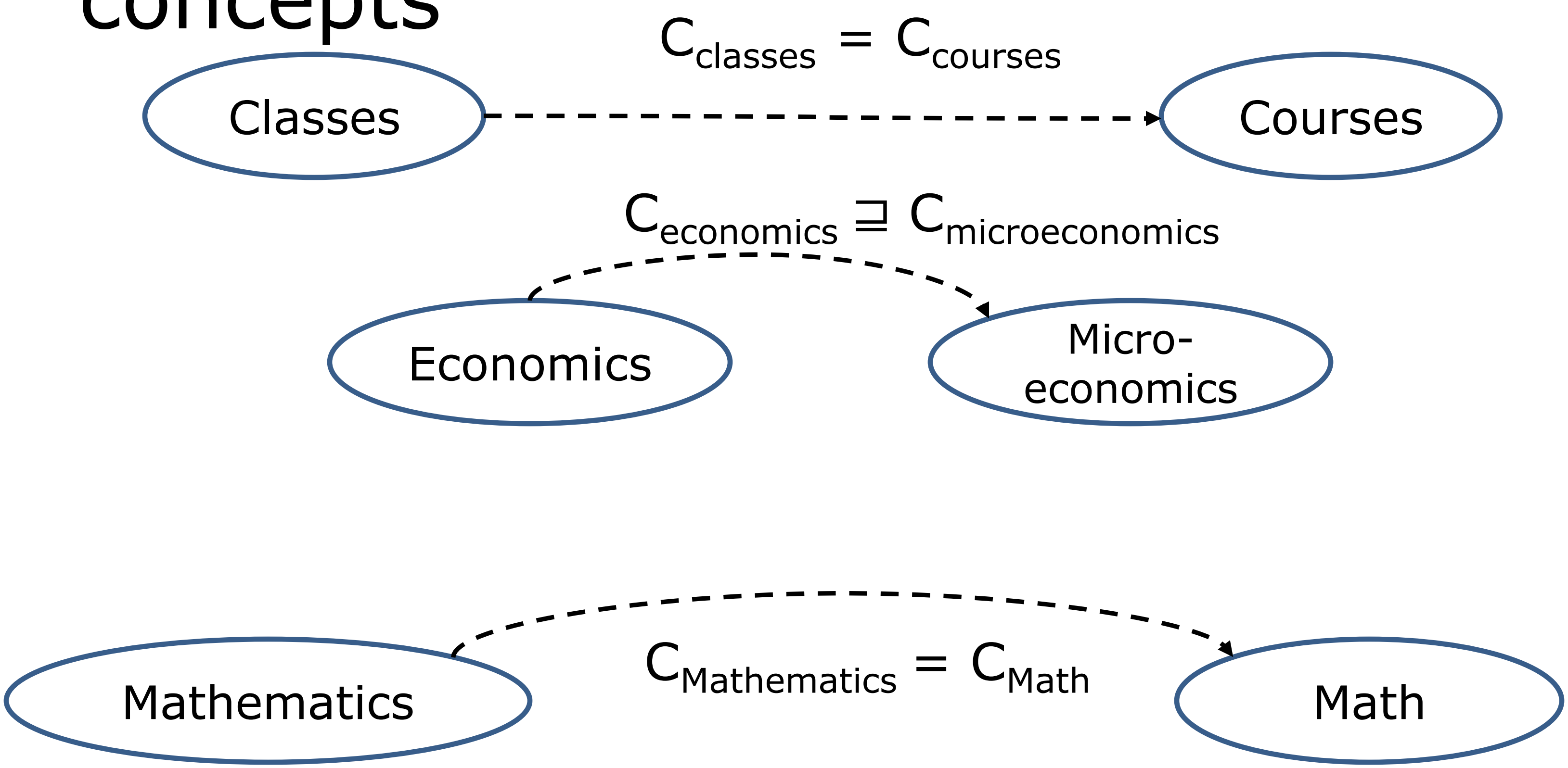
2. Compute concepts at nodes:



$$C_{classes}$$
$$\sqcap\ (C_{arts} \sqcup C_{science}) \sqcap C_{college}$$
$$\sqcap\ (C_{history} \sqcup C_{philosophy}) \sqcap C_{science}$$

S-Match [Giunchiglia et al, 2007]

3. Compute relations between atomic concepts



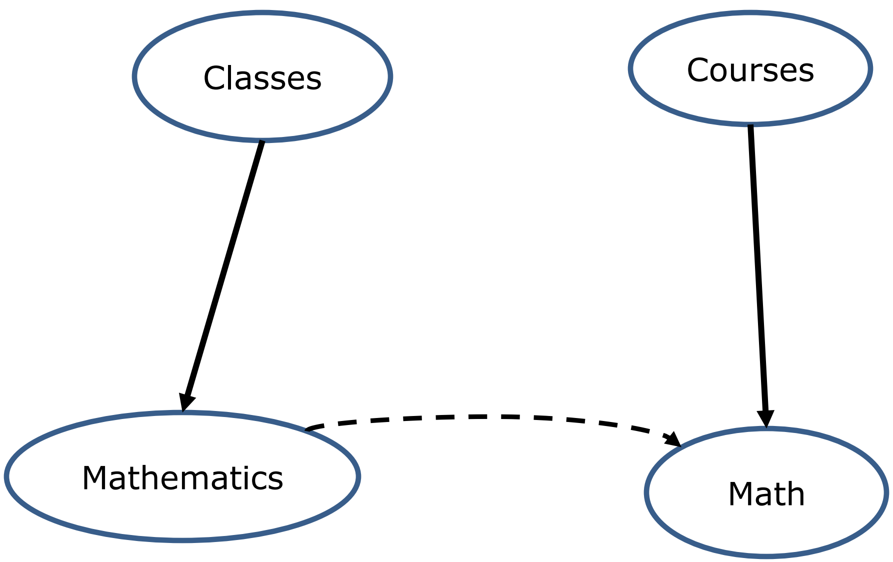
S-Match [Giunchiglia et al, 2007]

4. Compute relationships between nodes
   - Is $C_{classes} \sqcap C_{math}$ the same as $C_{courses} \sqcap C_{mathematics}$?
   - Construct logical implication formula axioms → **rel**(C_A, C_B)
   - If negation is unsatisfiable, **rel**(C_A, C_B) holds.

rel(C_A, C_B)	**Translation to prop. logic**
$C_A = C_B$	$C_A \Leftrightarrow C_B$
$C_A \sqsubseteq C_B$	$C_A \rightarrow C_B$
$C_A \sqsupseteq C_B$	$C_B \rightarrow C_A$
$C_A \perp C_B$	$\neg (C_A \wedge C_B)$

S-Match [Giunchiglia et al, 2007]

4. Compute relationships between nodes



Is $C_{classes} \sqcap C_{math}$ **=** $C_{courses} \sqcap C_{mathematics}$**?**

$(C_{classes} \Leftrightarrow C_{courses}) \wedge$
$(C_{Mathematics} \Leftrightarrow C_{Math})$
$\rightarrow$
$(C_{Classes} \wedge C_{Mathematics}) \Leftrightarrow$
$(C_{Courses} \wedge C_{Math})$?

Penn
UNIVERSITY of PENNSYLVANIA

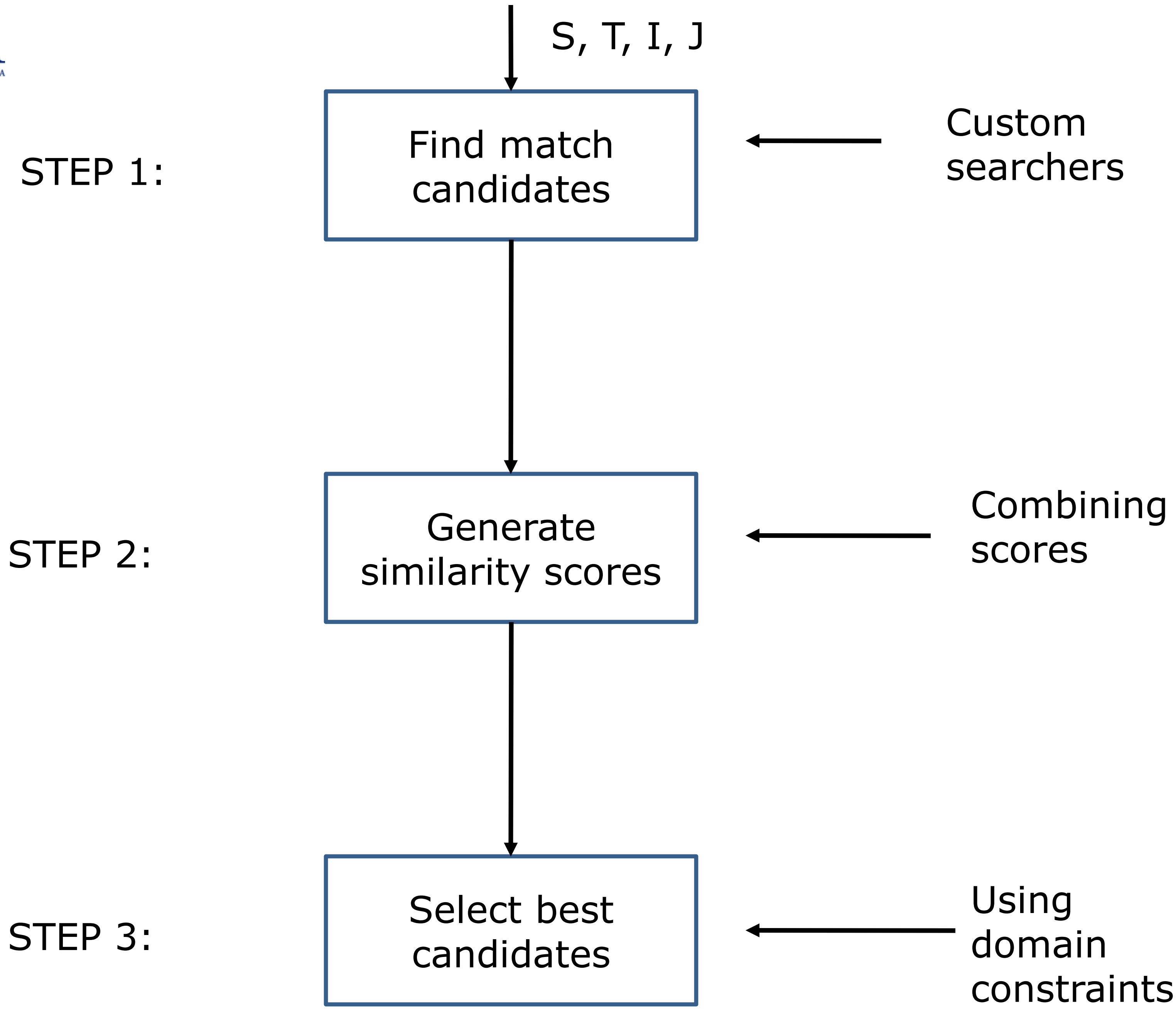
S-match

- Linguistic techniques a useful approach as attribute names/labels are described using natural language.
- Takes into account source structure.

- Would miss application-specific attribute namings (e.g. eid)
- Does not use type information

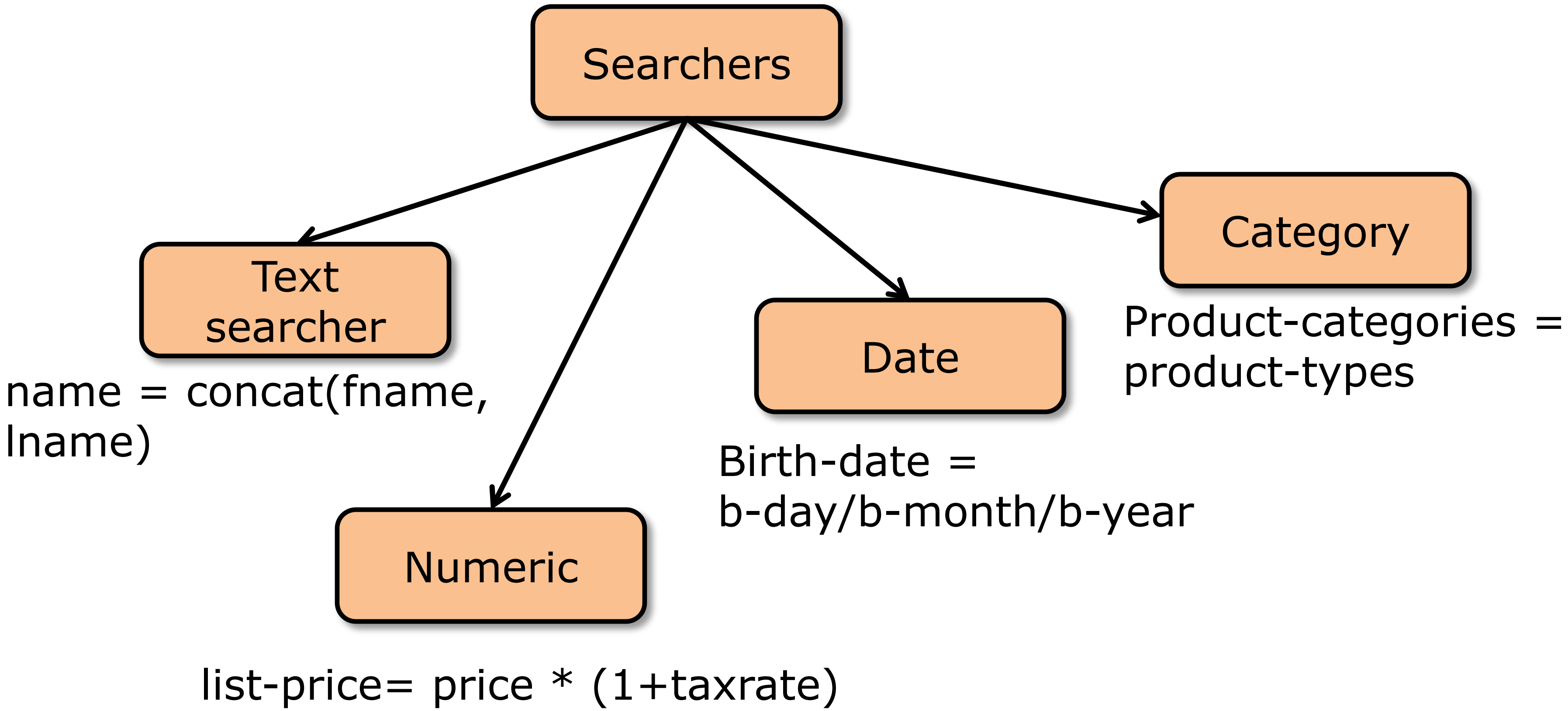
iMap [Dhanmankar, 04]

- System for determining complex matches between schemas.
  - Eg. concat(S.fname, S.lname)→ T.name
- Searches a space of possible matches:
  - Employing learning techniques
  - Employing domain knowledge
- Designed to be flexible, "plug-in" type architecture


S, T, I, J
STEP 1:
Find match candidates
Custom searchers
STEP 2:
Generate similarity scores
Combining scores
STEP 3:
Select best candidates
Using domain constraints

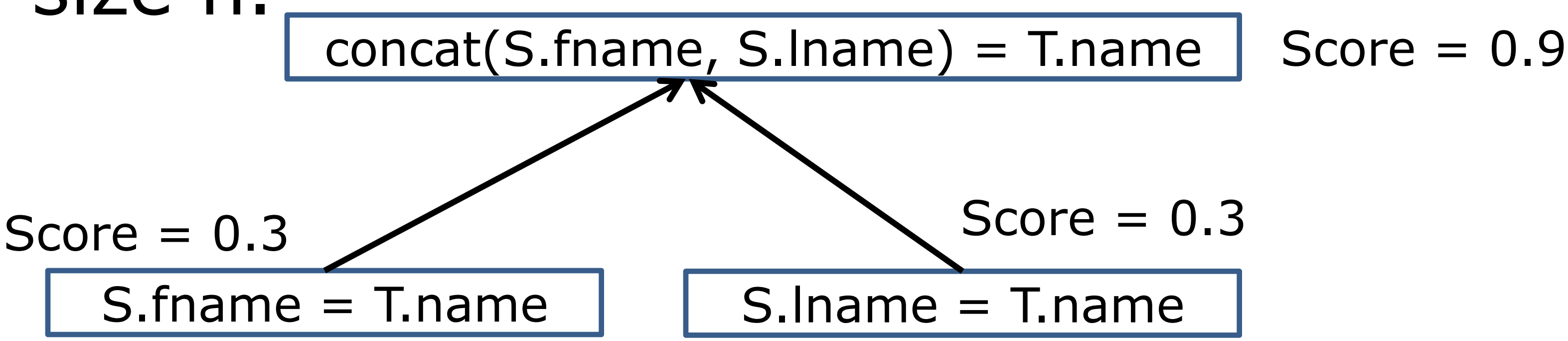
iMap [Dhanmankar, 04]

- Finding candidate matches:



iMap [Dhanmankar, 04]

- Searchers:
  - Search strategy: Keep only k-highest scoring candidates for each combination size n.



iMap[Dhanmankar, 04]

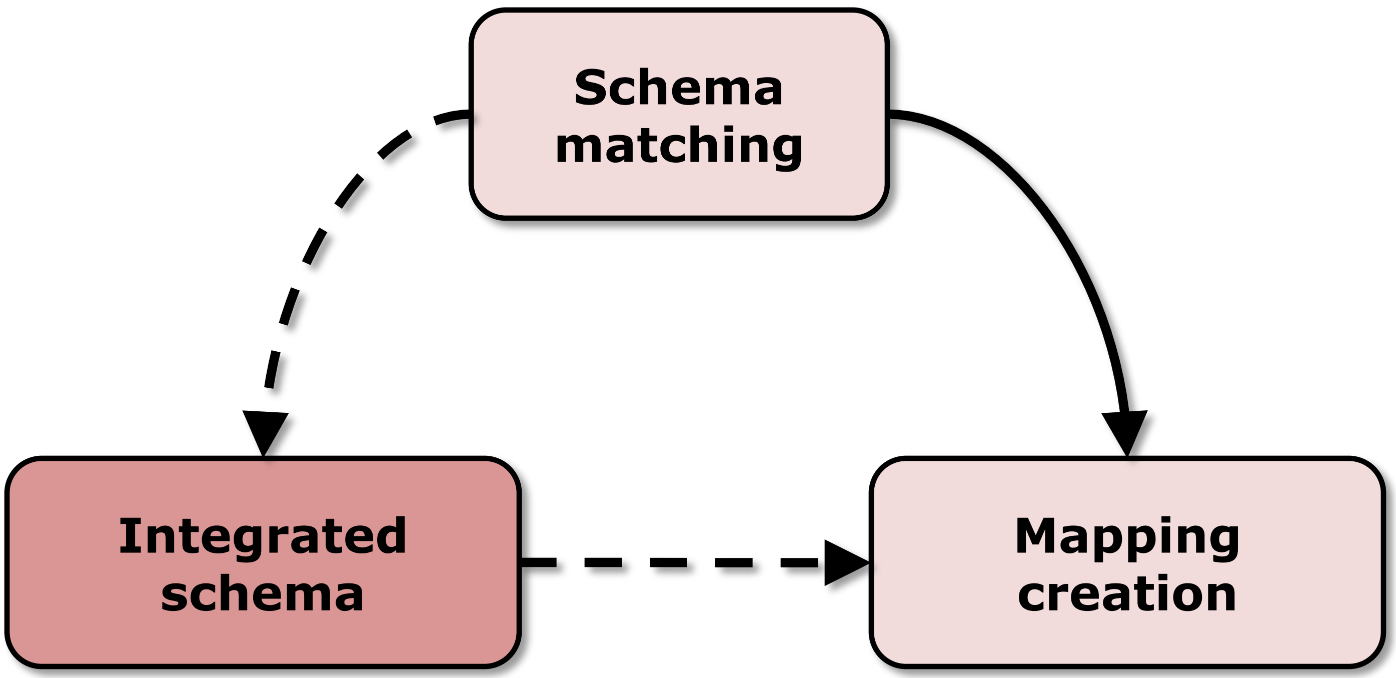
- Exploiting Domain Knowledge
  - Domain constraints (e.g name & email-address unrelated)
  - Overlap data: Test matches on overlapping data
  - External resources: thesaurus
- Generally higher accuracy when given domain knowledge.

Schema matching

- Discovering relationships between source and target attributes.
- Variety of work
  - Using instance-based approaches.
  - Using linguistic techniques.
  - Using structural constraints of schemas
- Survey on schema matching [Rahm, Bernstein, 2001]

Mapping Tasks

S-Match [Giunchigla et al, 2007]
iMap [Dhamankar et al, 2004]



[Chiticariu et al, 2008]
[Das Sarma et al, 2008]

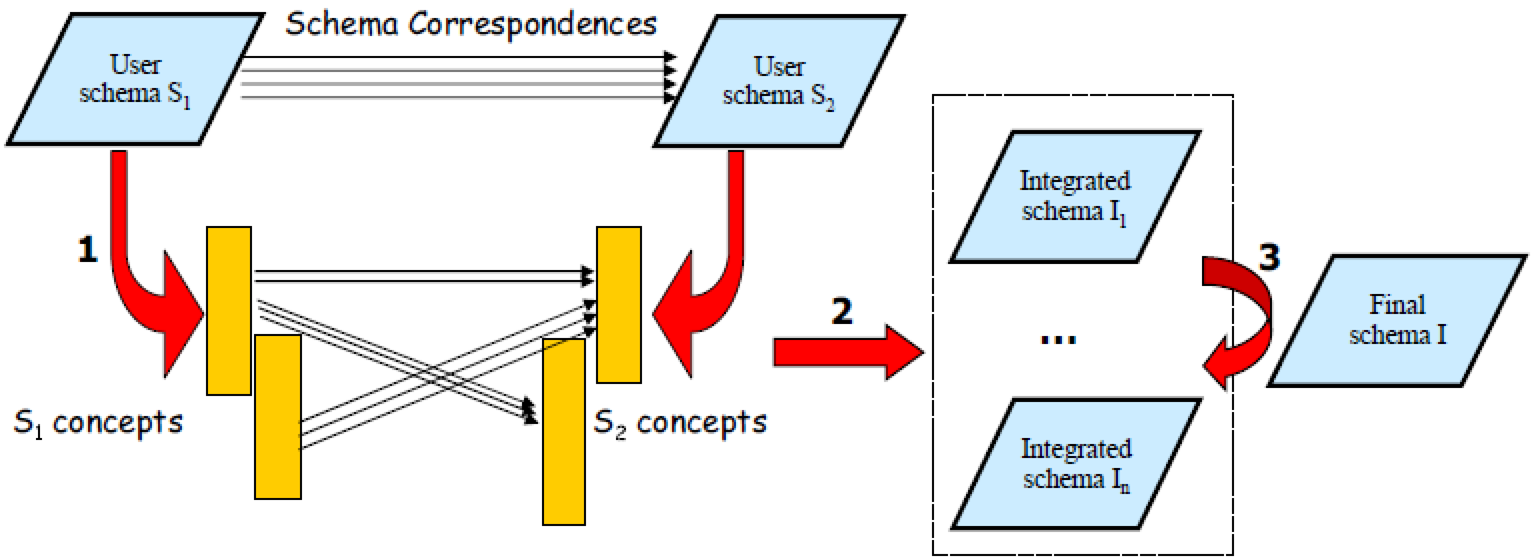
Clio [Miller et al, 2000]

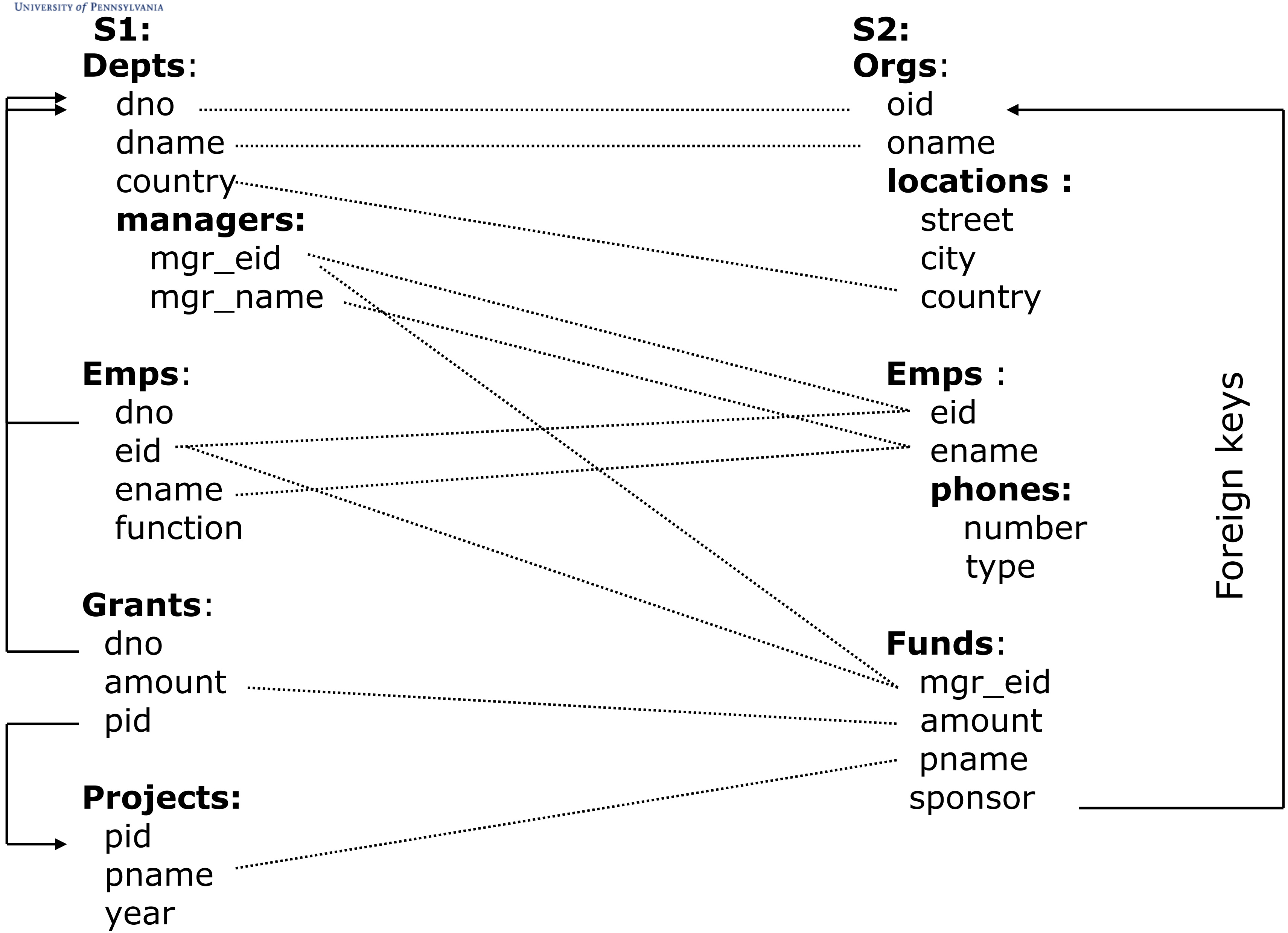
Integrated Schemas

- Given:
  - A set of source schemas $S_1, S_2..., S_n$
  - A set of pairs of source and target attributes (weighted correspondences)
- Find:
  - A unified target schema T best representing source schemas.

Integrated schemas [Chitacariu et al, 08]

- Interactive Generation of Integrated schemas



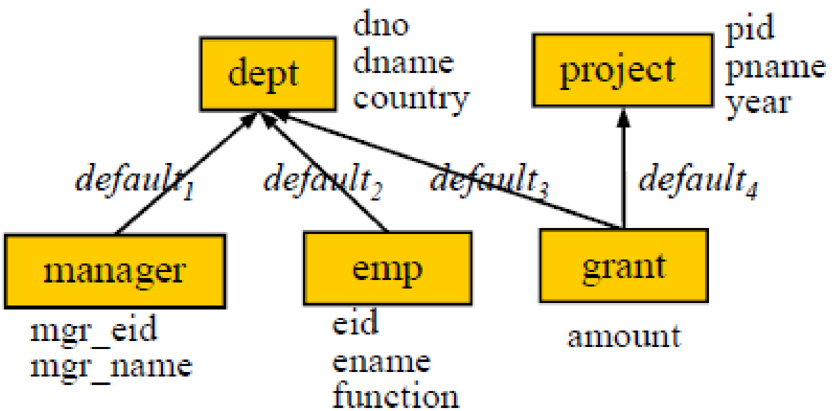

S1:
Depts:
dno
dname
country
managers:
mgr_eid
mgr_name
Emps:
dno
eid
ename
function
Grants:
dno
amount
pid
Projects:
pid
pname
year
S2:
Orgs:
oid
oname
locations :
street
city
country
Emps :
eid
ename
phones:
number
type
Funds:
mgr_eid
amount
pname
sponsor
Foreign keys

Concept graphs

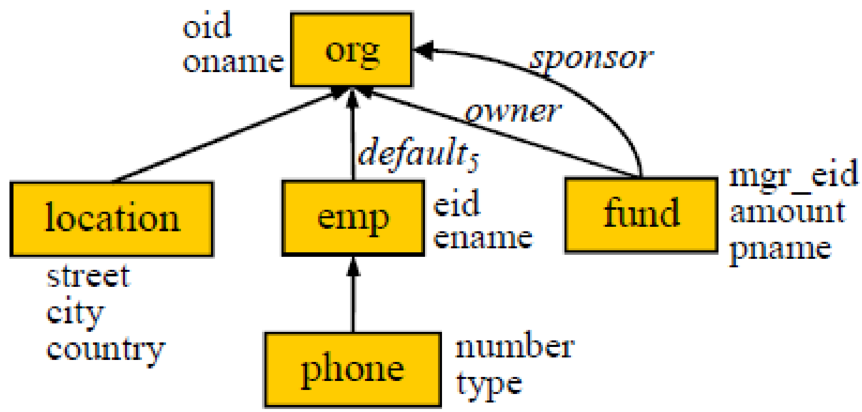
1: Construct concept graph

- each relation is a node
- each edge denotes parent-child or key-foreign key relationship

Schema S1:

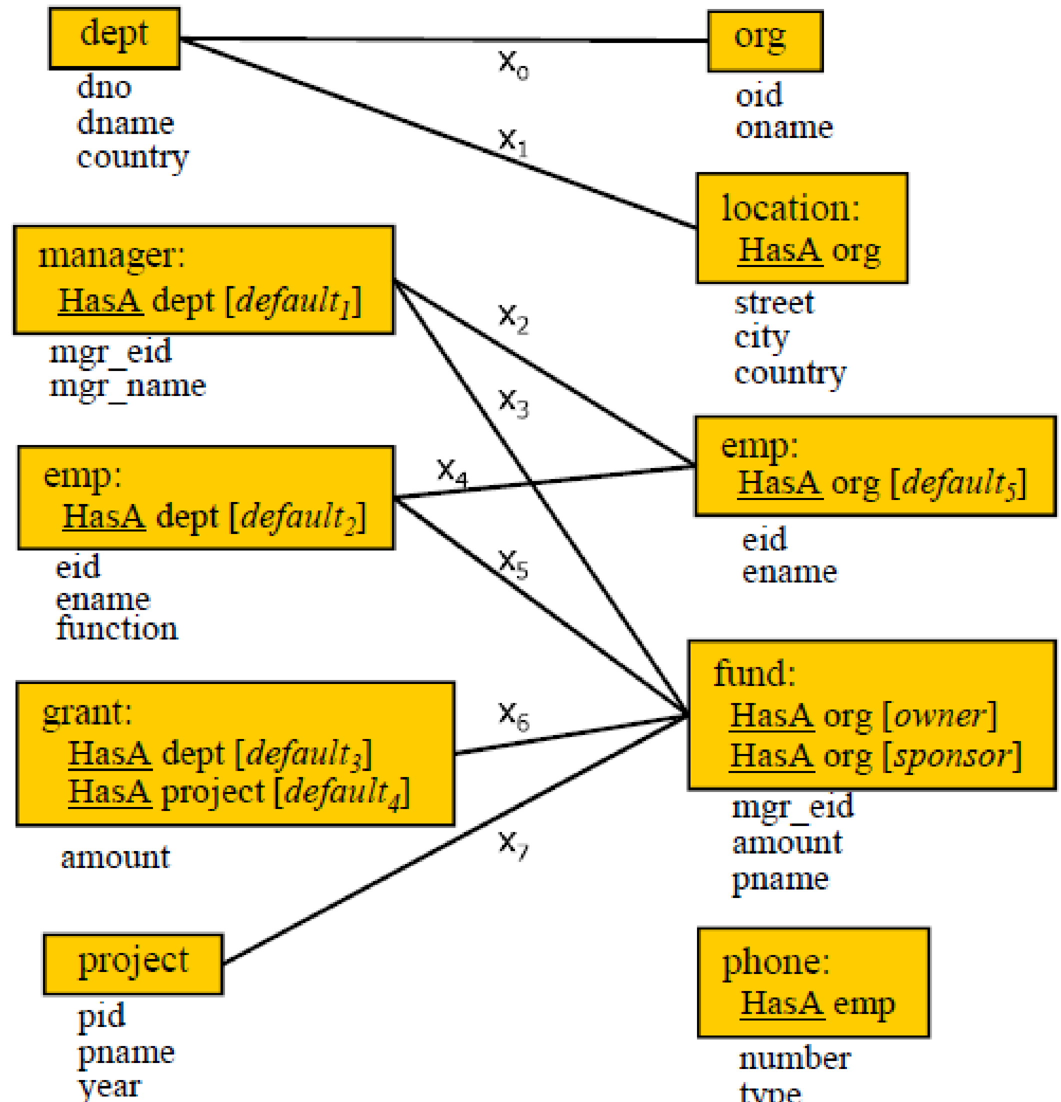


Schema S2:

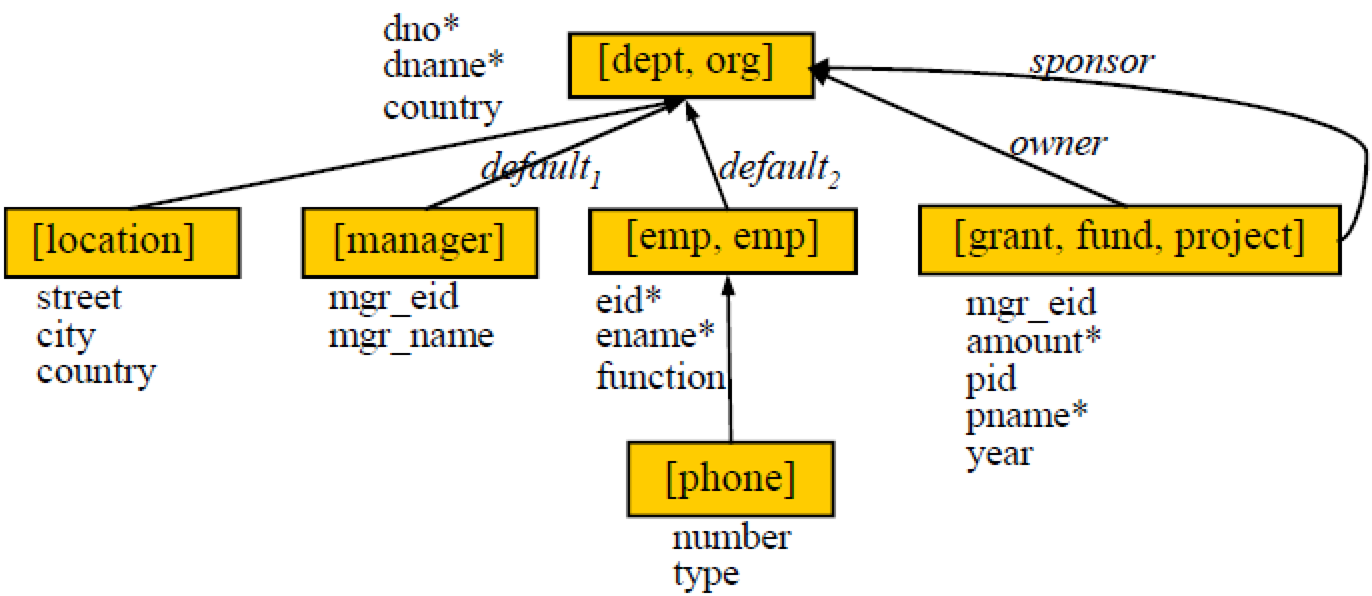


Matching graph

- Step 2: Form matching edges x_i
- Step 3: Find assignments of boolean variables x_i
- Step 4: For every edge x_i set to true, merge concepts



Integrated schema



Assignment: $x_1 = x_2 = x_3 = x_5 = 0,$
$x_0 = x_4 = x_6 = x_7 = 1$

Integrated schemas [Chitacariu et al, 08]

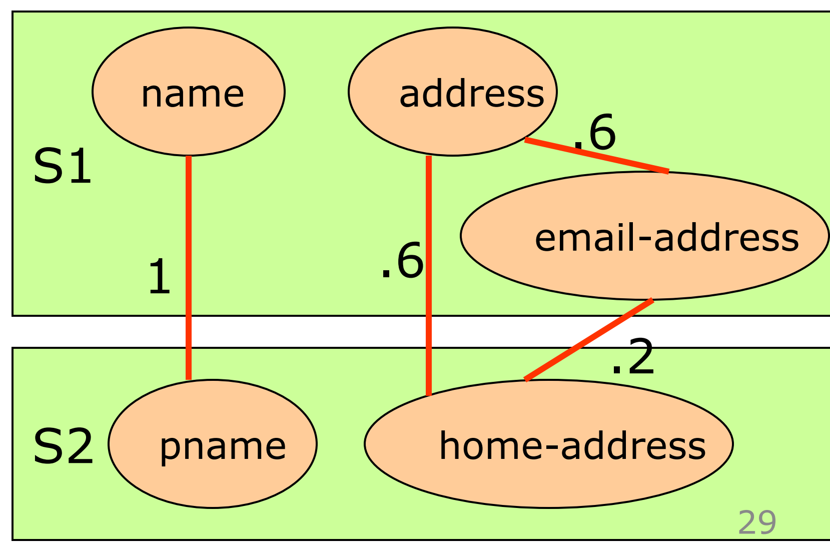
- Different assignments can lead to same schema
  - Add constraints to boolean variables
  - Find satisfying assignments for a set of Horn clauses
- Source-to-target mapping generation:
  - Use a variant of the chase in order to preserve source foreign key constraints in the target.

Integrated schemas [Das Sarma et al, 08]

- Key idea: build *probabilistic* schemas
  - Models uncertainty behind merging concepts
- Considers single relation source, target schemas
- Each attribute is a concept
- Attribute correspondences have weights

Integrated schemas [Das Sarma et al, 08]

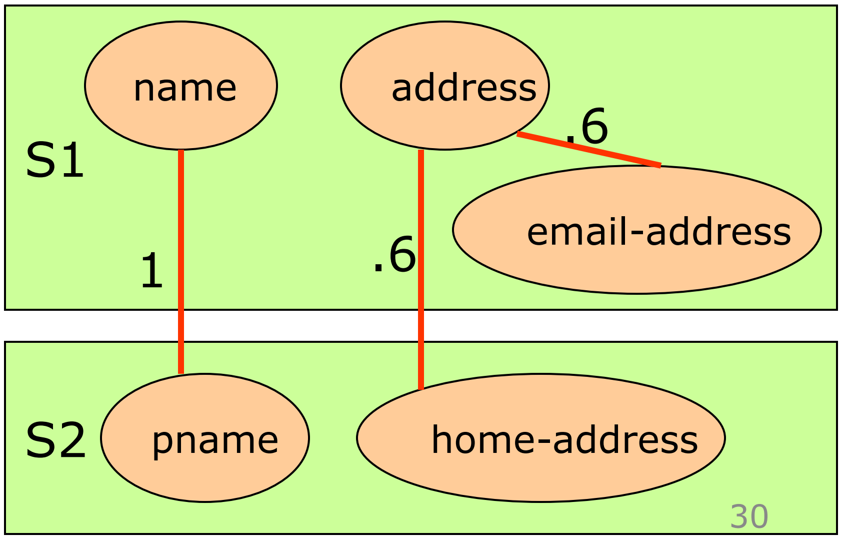
- Algorithm
  1. Construct weighted graph using correspondences
  2. Remove edges with weight below T
  3. Each connected component forms cluster



Integrated schemas [Das Sarma et al, 08]

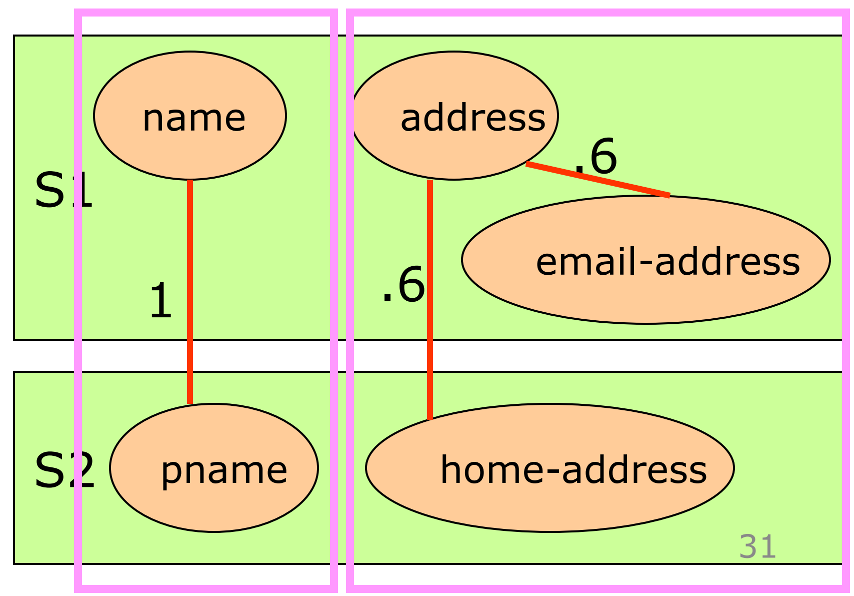
- Algorithm
  1. Construct weighted graph using correspondences
  2. Remove edges with weight below T
  3. Each connected component forms cluster

T= 0.2



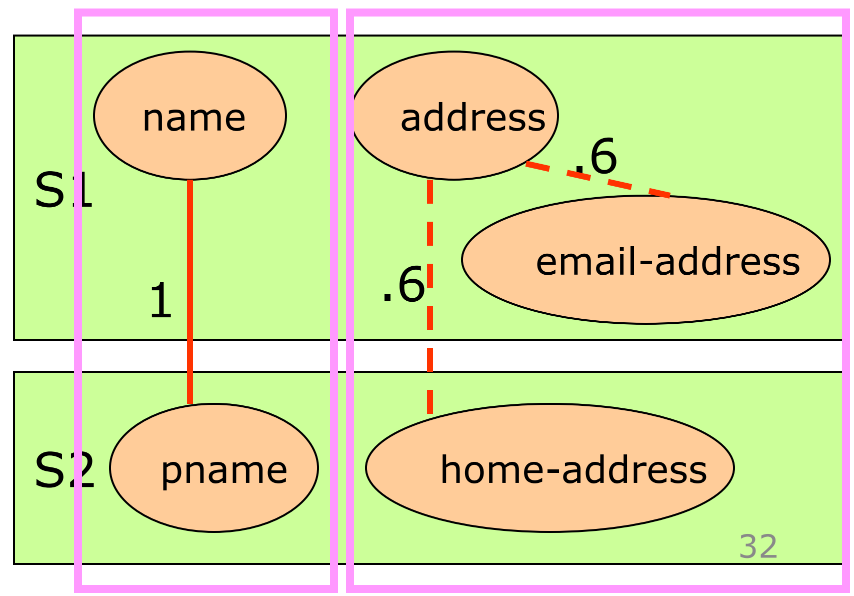
Integrated schemas [Das Sarma et al, 08]

- Algorithm
  1. Construct weighted graph using correspondences
  2. Remove edges with weight below T
  3. Each connected component forms cluster



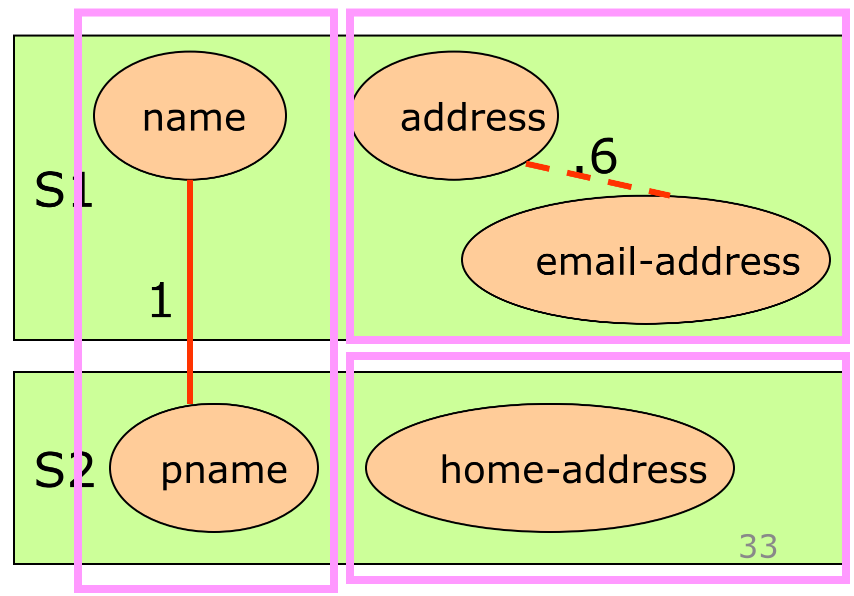
Integrated schemas [Das Sarma et al, 08]

- Partition edges into **certain** and **uncertain** edges
- Each uncertain edge
  with weight between T+ε and T-ε
- Create new schema by including/excluding uncertain edges.



Integrated schemas [Das Sarma et al, 08]

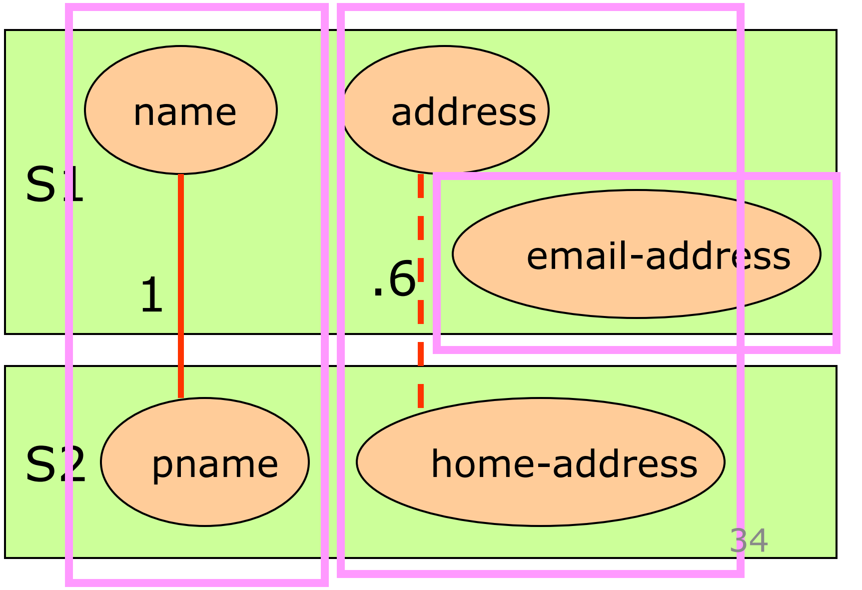
- Partition edges into **certain** and **uncertain** edges
- Each uncertain edge
  with weight between T+ε and T-ε
- Create new schema by including/excluding uncertain edges.



Integrated schemas [Das Sarma et al, 08]

- Partition edges into **certain** and **uncertain** edges
- Each uncertain edge

  with weight between T+ε and T-ε
- Create new schema by including/excluding uncertain edges.



Integrated schemas [Das Sarma et al, 08]

- Probability of schema M_i:

$$\Pr(M_i) = \frac{c_i}{\sum_{j=1}^{l} c_j}$$

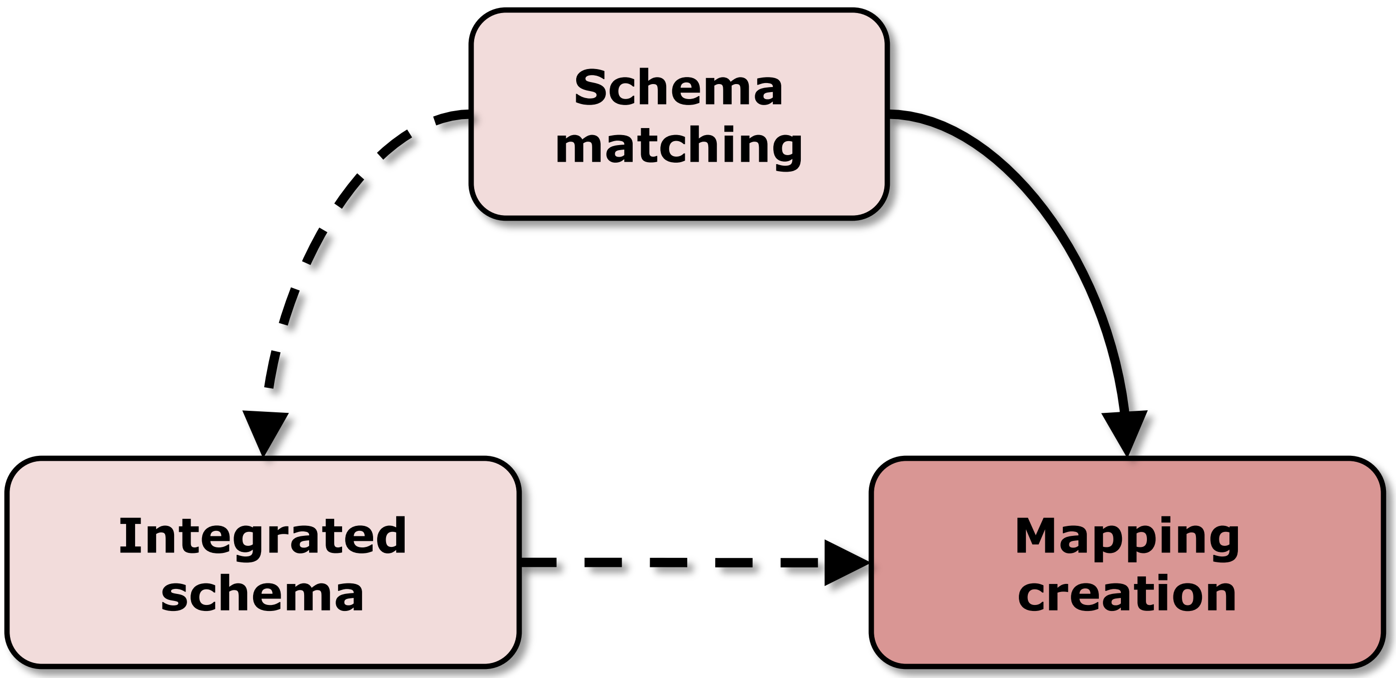
where c_i = number of sources *consistent* with M_i

- Source is consistent if no two distinct attributes are grouped together
  – Models uncertainty in grouping real-world concepts

- Also consider p-mappings: probablistic mappings.

Mapping Tasks

S-Match [Giunchigla et al, 2007]
iMap [Dhamankar et al, 2004]



[Chiticariu et al, 2008]
[Das Sarma et al, 2008]

Clio [Miller et al, 2000]

Mapping generation [Miller et al,2000]

- Goal:
  - to discover mappings between *independently created* source and target schemas
- Given:
  - Source schema S, single-table target schema T, set of value correspondences.
  - Value correspondence (f_i, p_i)

    where f_i is a function:

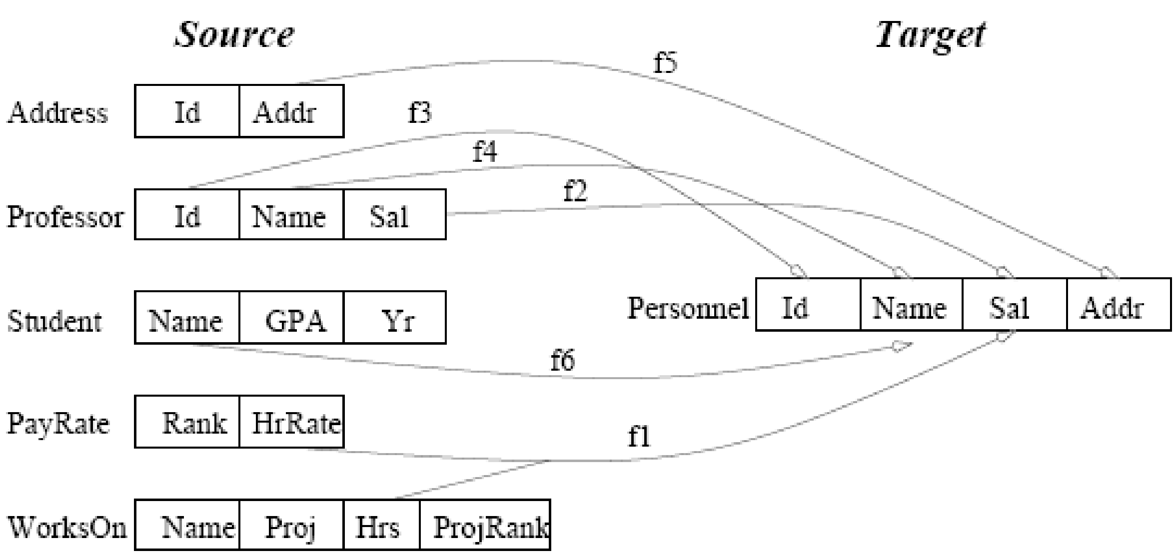
    $$f_i: dom(A_1) \times \cdots \times dom(A_q) \rightarrow dom(B)$$

    and pi is a predicate over the source attributes:

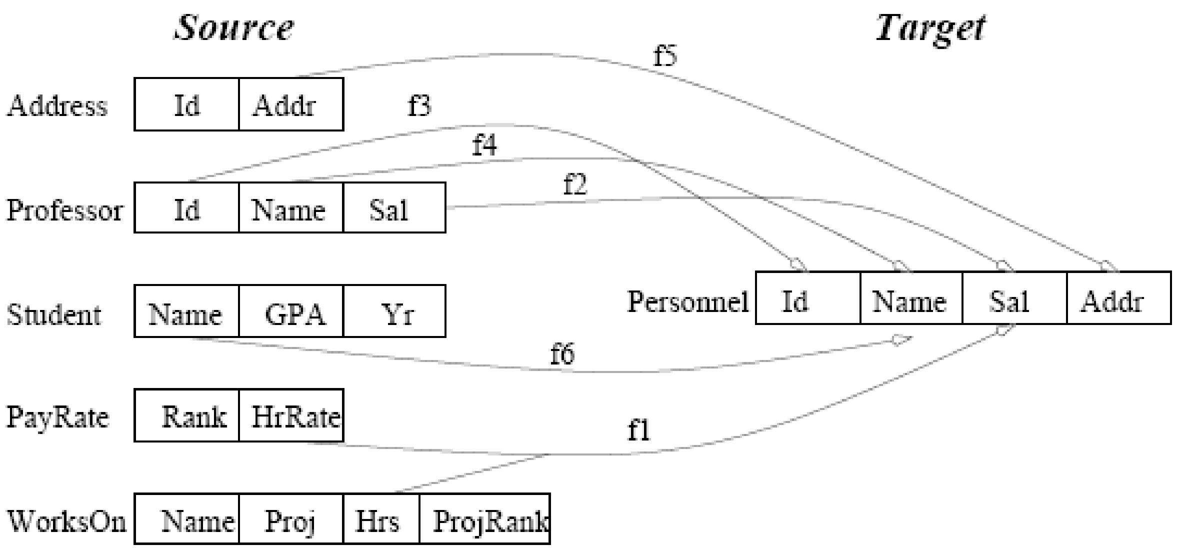
    $$p_i: dom(A_1) \times \cdots \times dom(A_q) \rightarrow boolean$$

Value Correspondences

- Example:
  - f1: **PayRate**(HrRate)***WorksOn**(Hrs) → **Personnel**(Sal)



Mapping generation



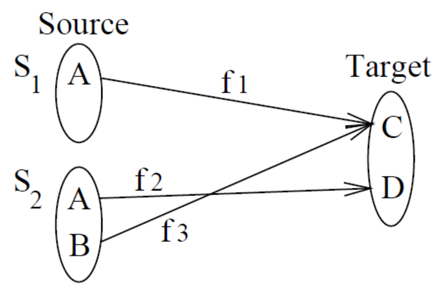
Sample mapping queries:

$$\{(i,n,s,a) \mid Professor(i,n,s) \wedge Address(i,a)\}$$

$$\{(i,n,s,a) \mid \exists r,h,y,x\ Student(n,g,y) \wedge PayRate(y,h) \wedge WorksOn(n,p,x,r) \wedge i = null \wedge a = null \wedge s = h * x\}$$

Algorithm

1. Input Value Correspondences



2. Group Correspondences into candidate sets:
   – At most one correspondence per target attribute for each candidate set

Candidate sets→ $\{\{f_1, f_2\}, \{f_2, f_3\}, \{f_1\}, \{f_2\}, \{f_3\}\}$

Algorithm

3. Prune candidate sets if they do not map to good queries
   – For set $\{f_1{:}S1.A \rightarrow T.C,\ f_2{:}S2.A \rightarrow T.D\}$ prune if no way to join S1 and S2
4. Select covers
   – Cover: Subset of candidate sets with each correspondence in at least one set
5. Rank covers
   – According to number of candidate sets

Conclusions

- Deriving mappings consists of several tasks:
  - Schema matching
  - Generation of Integrated schemas
  - Generation of mappings
- In general, lots of uncertainty
  - No way to exactly know semantic relationships
  - Tackle through probabilistic models
  - Learn from user feedback

Bibliography

- F. Giunchiglia , M.Yatskevich and P. Shvaiko. **Semantic Matching: Algorithms and Implementation**. Journal on Data Semantics. 2007.
- L. Chiticariu,P.G. Kolatis, and L. Popa: **Interactive generation of integrated schemas**. SIGMOD'08
- R. Dhamankar, Y. Lee, A. Doan, A.Y. Halevy, and P. Domingos. **iMAP: Discovering Complex Mappings between Database Schemas**. SIGMOD 2004.
- A. Das Sarma, X. Dong, and A.Y. Halevy: **Bootstrapping pay-as-you-go data integration systems**. SIGMOD'08.
- R.J. Miller, L.M. Haas, and M.A. Hernandez. **Schema Mapping as Query Discovery**. VLDB'00