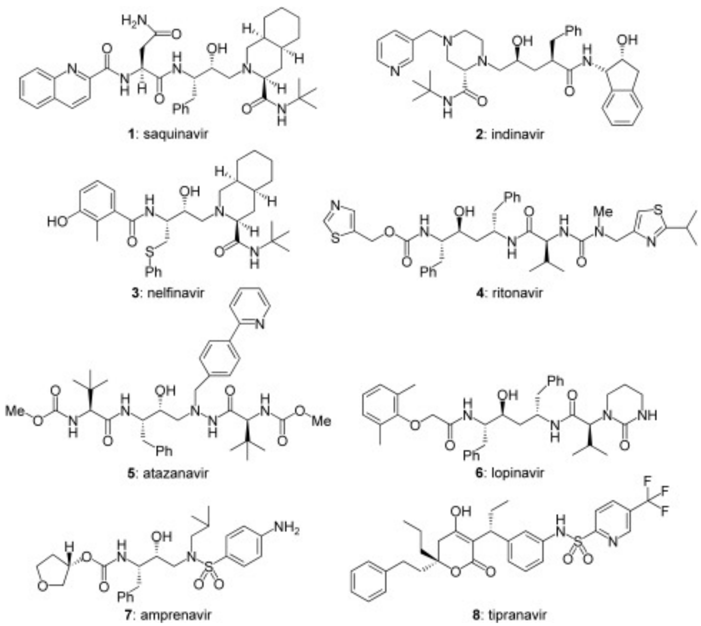
Schema Independent Relational Learning

Jose Picado, Arash Termehchy, Alan Fern, Parisa Ataei

Information and Data Management and Analytics (IDEA) Lab

Oregon State University

Design a drug to treat HIV



Oracle

What is the structure of compounds that have **anti-HIV** activity?

A compound has **anti-HIV** activity if it has the following substructure:

Oregon State University

Relational learning

compound

compId	atomId
c1	a1
c2	a10

atom

atomId	element
a1	N
a2	O

bond

atomId1	atomId2	type
a1	a2	single
a2	a3	single

Relational learning algorithm

```
anti-HIV(x) :-  compound(x,u), atom(u,N),
                compound(x,v), atom(v,O),
                compound(x,w), atom(w,N),
                bond(u,v,single), bond(v,w,single).
```

- Leverages the structure of the relational database
- Learns a Datalog definition

Training data:

anti-HIV

compId
c1
c3

no-anti-HIV

compId
c2
c4

Benefits of relational learning

compound	
compId	atomId
c1	a1
c2	a10

atom	
atomId	element
a1	N
a2	O

bond		
atomId1	atomId2	type
a1	a2	single
a2	a3	single

FOIL, Progol, …
Castor (new algorithm)

Existing algorithms

- ✓ Leverage the structure of data and learn over complex schemas with multiple tables
- ✓ Automatic feature extraction and selection
- ✓ Results are interpretable (Datalog)

anti-HIV(x) :- compound(x,u), atom(u,N),
compound(x,v), atom(v,O),
compound(x,w), atom(w,N),
bond(u,v,single), bond(v,w,single).

Schema 1

paperAuthor	
paperId	authorId
p1	mad
p1	bai
p2	soc
p2	man
p3	mad

author	
id	name
mad	Madden
sto	Stonebraker
soc	Socher
man	Manning
bai	Bailis

authorAffiliation	
id	affiliation
mad	MIT
sto	MIT
soc	Stanford
man	Stanford
bai	Stanford

paper	
id	title
p1	MacroBase: Priori…
p2	GloVe: Global Vect…

paperYear	
id	year
p1	2017
p2	2014

paperConf	
id	conf
p1	SIGMOD
p2	EMNLP

Which authors are **collaborators**?

collaborators	
person1	person2
Madden	Bailis
Socher	Manning
Madden	Stonebraker

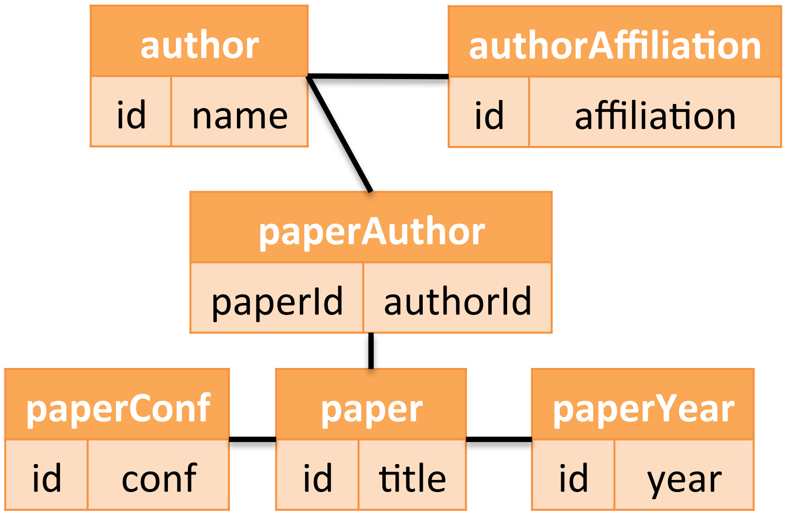
non-collaborators	
person1	person2
Madden	Socher
Manning	Bailis

FOIL learning algorithm

?

FOIL: relational learning algorithm

Schema 1



collaborators(x,y) :-

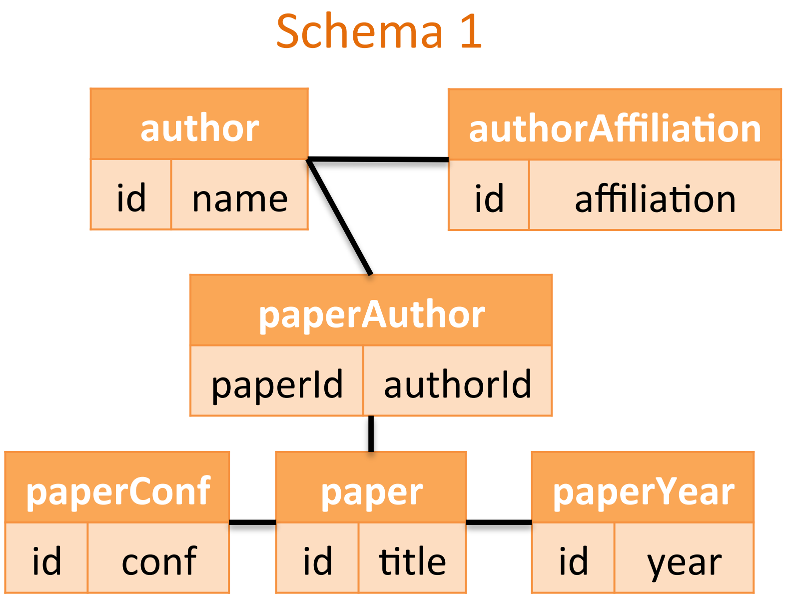
Scoring function **f: P - N**

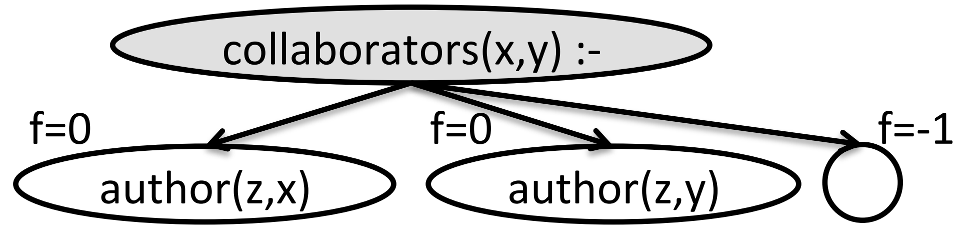
P: positive examples covered

N: negative examples covered

collaborators(x,y) :-
    true.

FOIL: relational learning algorithm





Scoring function **f: P - N**

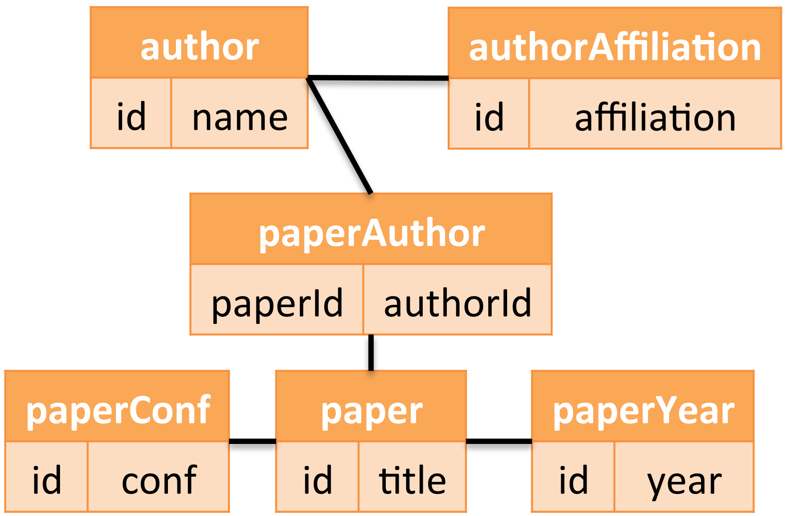
P: positive examples covered

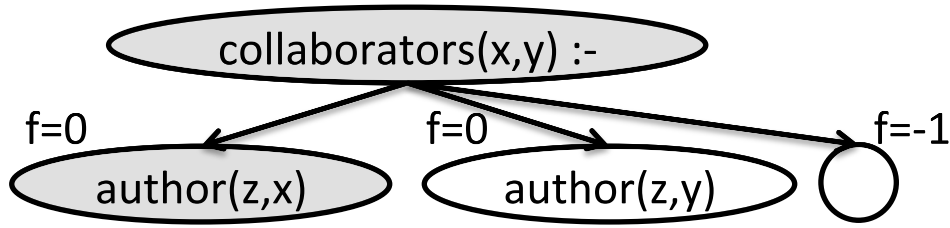
N: negative examples covered

collaborators(x,y) :-
    true.

FOIL: relational learning algorithm

Schema 1





Scoring function f: P - N

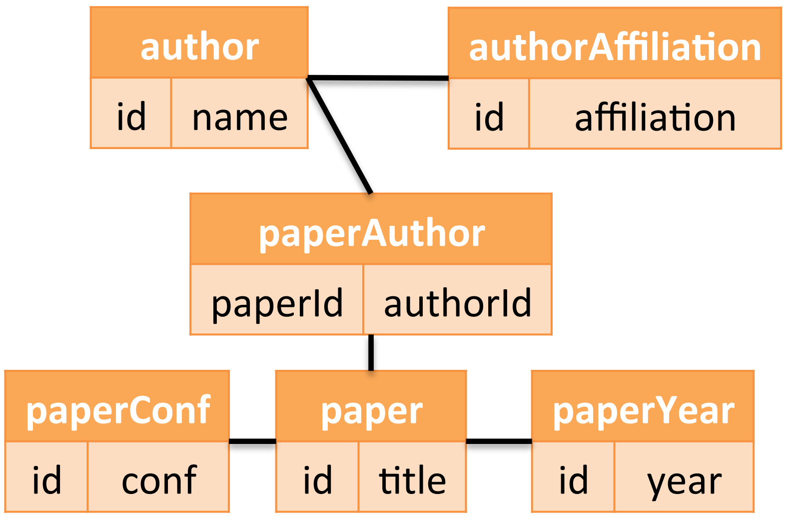
P: positive examples covered

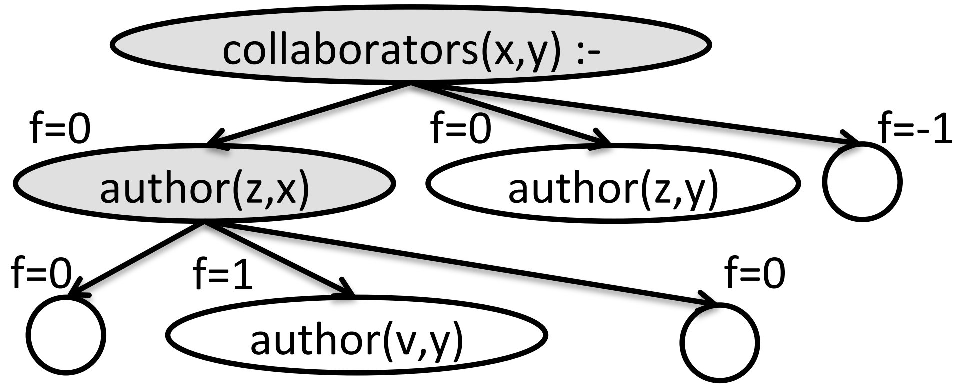
N: negative examples covered

collaborators(x,y) :-
    author(z,x).

FOIL: relational learning algorithm

Schema 1





Scoring function **f: P - N**

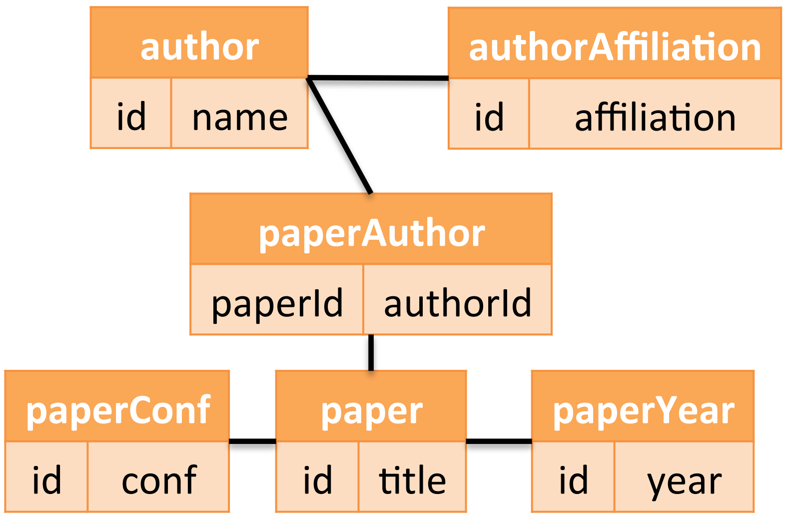
P: positive examples covered

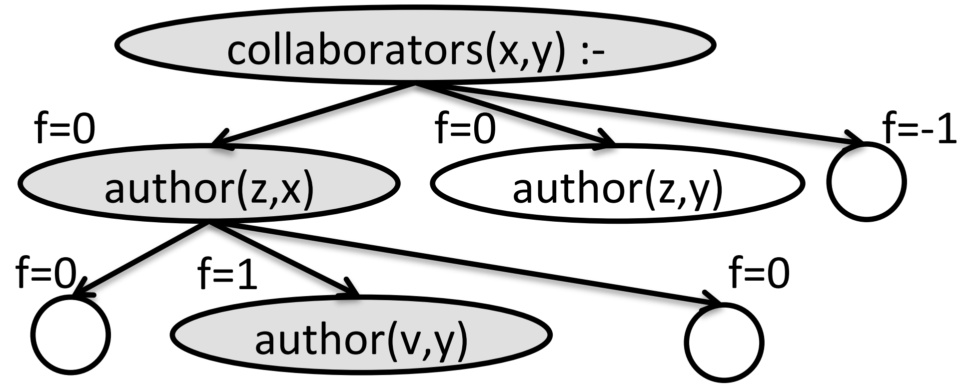
N: negative examples covered

collaborators(x,y) :-
    author(z,x).

FOIL: relational learning algorithm

Schema 1





Scoring function **f: P - N**

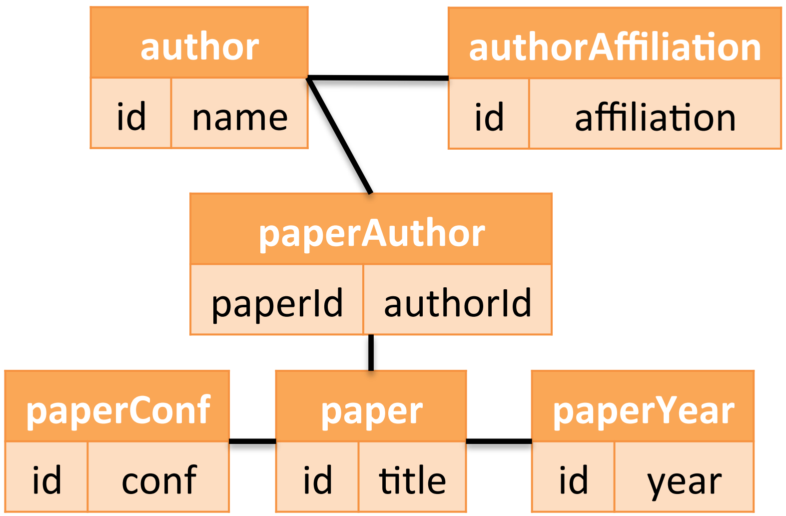
P: positive examples covered

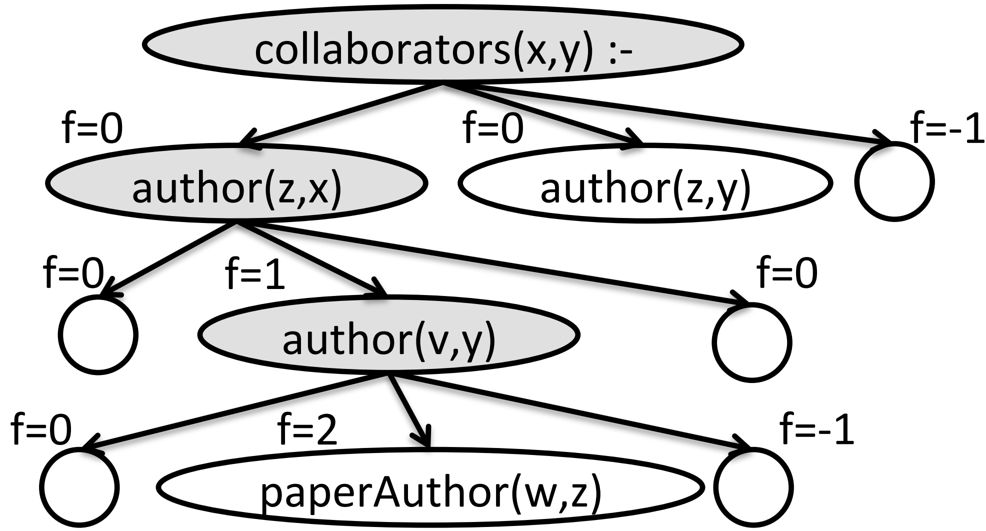
N: negative examples covered

collaborators(x,y) :-
    author(z,x), author(v,y).

FOIL: relational learning algorithm

Schema 1





Scoring function **f: P - N**

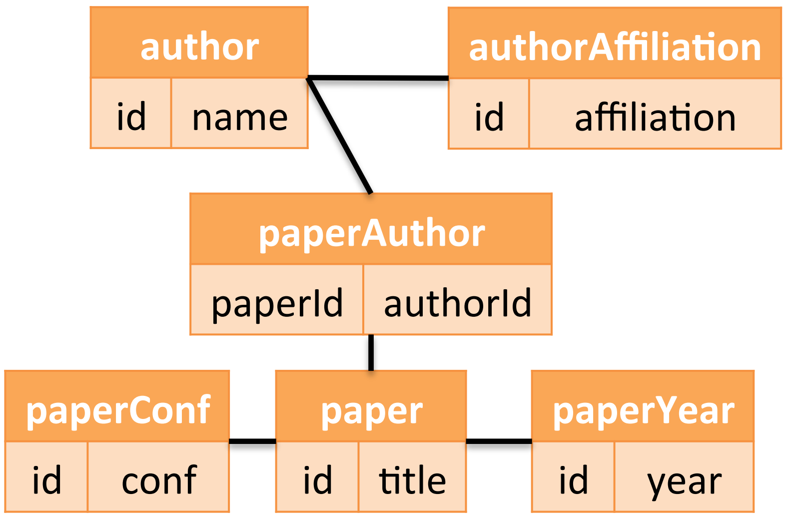
P: positive examples covered

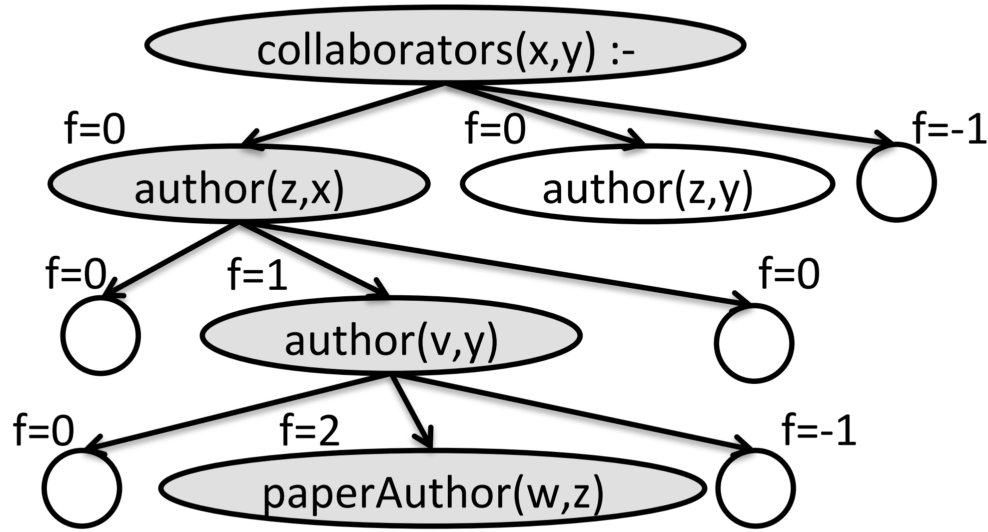
N: negative examples covered

collaborators(x,y) :-
    author(z,x), author(v,y).

FOIL: relational learning algorithm

Schema 1





Scoring function **f: P - N**

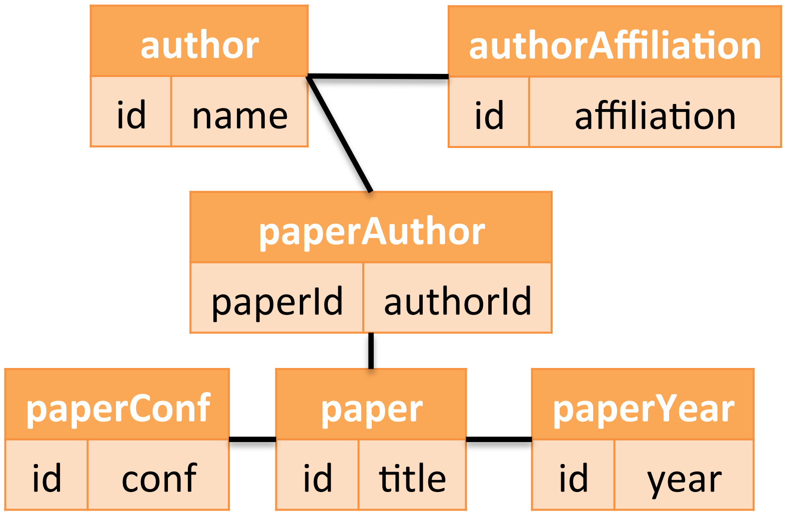
P: positive examples covered

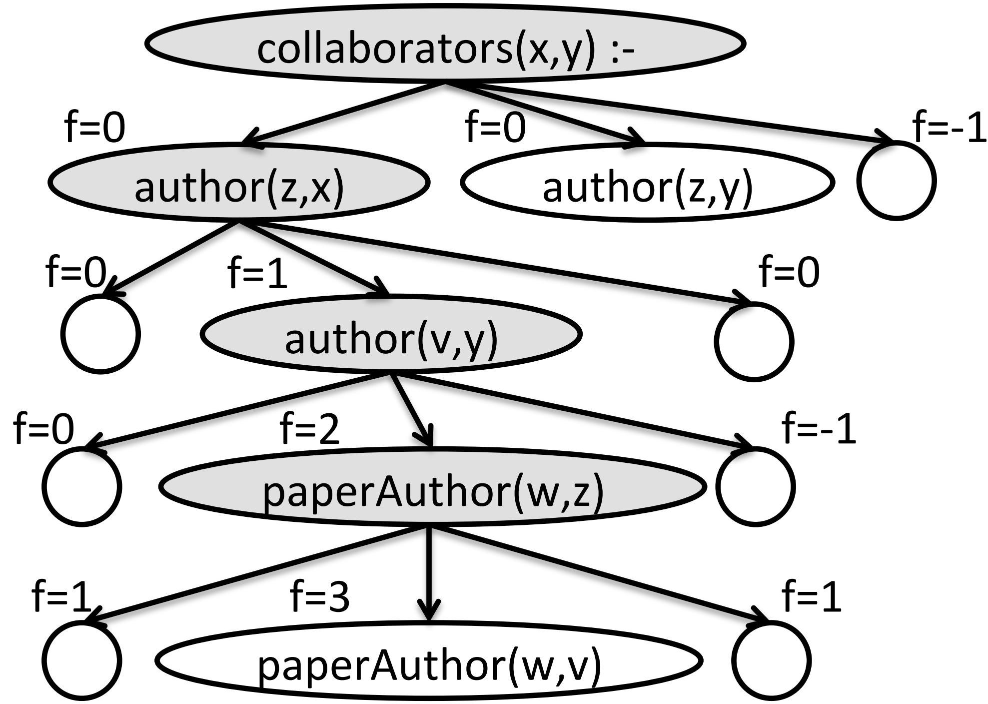
N: negative examples covered

collaborators(x,y) :-
    author(z,x), author(v,y),
    paperAuthor(w,z).

FOIL: relational learning algorithm

Schema 1





Scoring function **f: P - N**

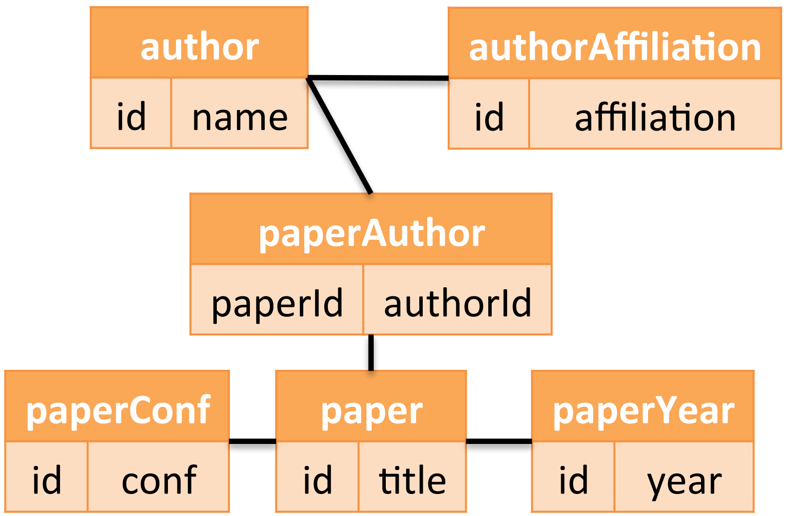
P: positive examples covered

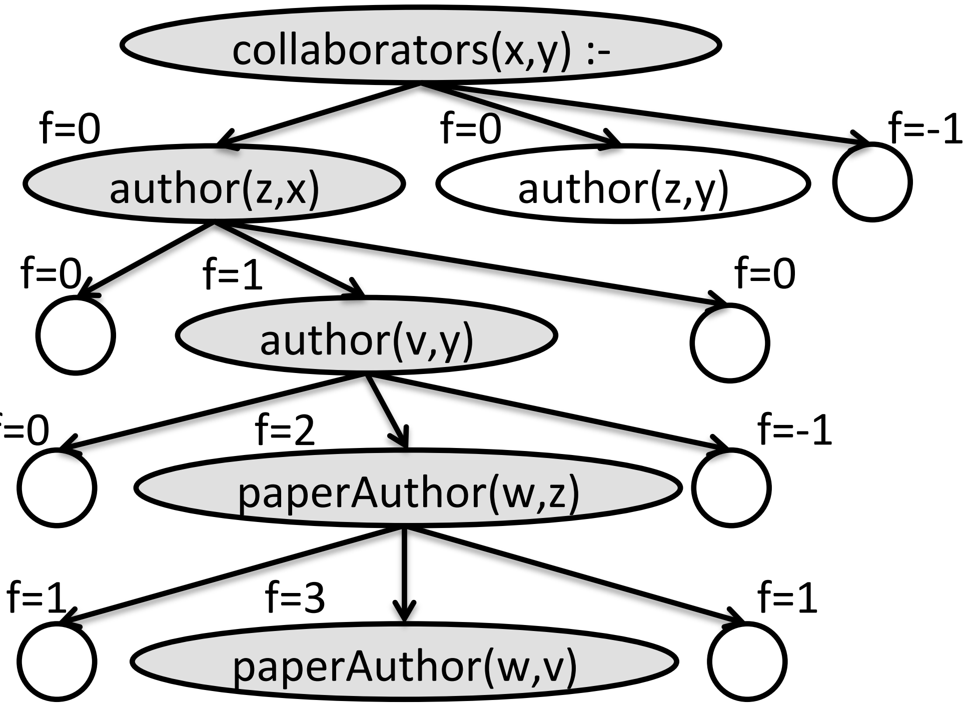
N: negative examples covered

collaborators(x,y) :-
  author(z,x), author(v,y),
  paperAuthor(w,z).

FOIL: relational learning algorithm

Schema 1





Scoring function **f: P - N**

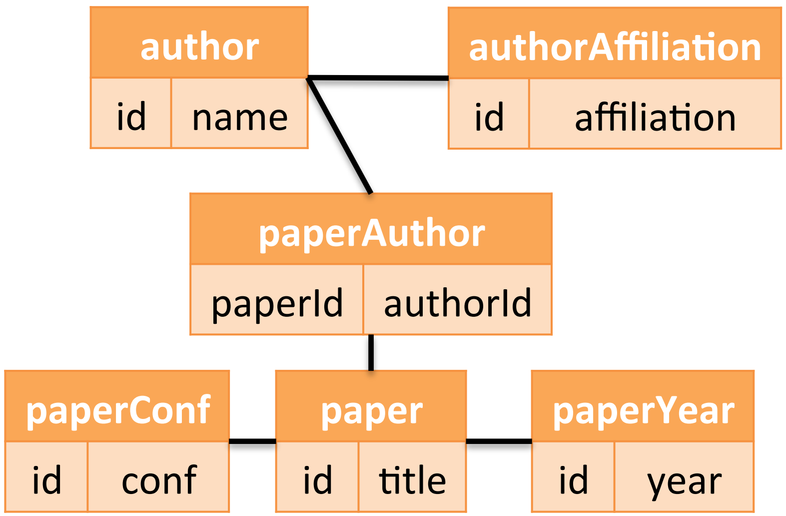
P: positive examples covered

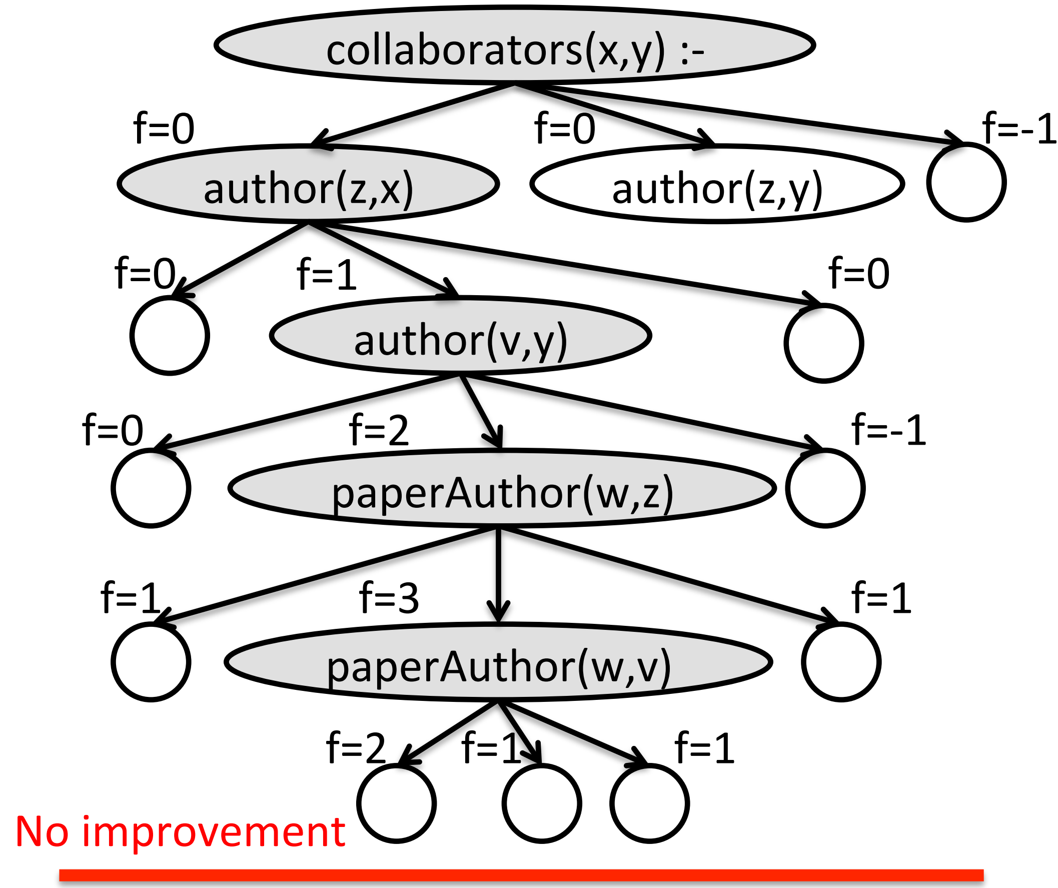
N: negative examples covered

collaborators(x,y) :-
    author(z,x), author(v,y),
    paperAuthor(w,z), paperAuthor(w,v).

FOIL: relational learning algorithm

Schema 1





Scoring function **f: P - N**

P: positive examples covered

N: negative examples covered

collaborators(x,y) :-
author(z,x), author(v,y),
paperAuthor(w,z), paperAuthor(w,v).

Schema 1

paperAuthor	
paperId	authorId
p1	mad
p1	bai
p2	soc
p2	man
p3	mad

author	
id	name
mad	Madden
sto	Stonebraker
soc	Socher
man	Manning
bai	Bailis

authorAffiliation	
id	affiliation
mad	MIT
sto	MIT
soc	Stanford
man	Stanford
bai	Stanford

paper	
id	title
p1	MacroBase: Priori…
p2	GloVe: Global Vect…

paperYear	
id	year
p1	2017
p2	2014

paperConf	
id	conf
p1	SIGMOD
p2	EMNLP

Which authors are **collaborators**?

collaborators	
person1	person2
Madden	Bailis
Socher	Manning
Madden	Stonebraker

non-collaborators	
person1	person2
Madden	Socher
Manning	Bailis

FOIL learning algorithm

f=3

collaborators(x,y) :-
author(z,x), author(v,y),
paperAuthor(w,z), paperAuthor(w,v).

Two people are collaborators if they are co-authors.

People represent the same data using different schemas

author

id	name
mad	Madden
sto	Stonebraker
soc	Socher
man	Manning
bai	Bailis

authorAffiliation

id	affiliation
mad	MIT
sto	MIT
soc	Stanford
man	Stanford
bai	Stanford

paper

id	title
p1	MacroBase: Priori...
p2	GloVe: Global Vect...

paperYear

id	year
p1	2017
p2	2014

paperConf

id	conf
p1	SIGMOD
p2	EMNLP

author

id	name	affiliation
mad	Madden	MIT
sto	Stonebraker	MIT
soc	Socher	Stanford
man	Manning	Stanford
bai	Bailis	Stanford

paper

id	title	year	conference
p1	MacroBase: Priori...	2017	SIGMOD
p2	GloVe: Global Vect...	2014	EMNLP

Composition
Denormalization
better performance

DBA

Schema 2

paperAuthor	
paperId	authorId
p1	mad
p1	bai
p2	soc
p2	man
p3	mad

author		
id	name	affiliation
mad	Madden	MIT
sto	Stonebraker	MIT
soc	Socher	Stanford
man	Manning	Stanford
bai	Bailis	Stanford

paper			
id	title	year	conference
p1	MacroBase: Priori…	2017	SIGMOD
p2	GloVe: Global Vect…	2014	EMNLP

Which authors are **collaborators**?

collaborators	
person1	person2
Madden	Bailis
Socher	Manning
Madden	Stonebraker

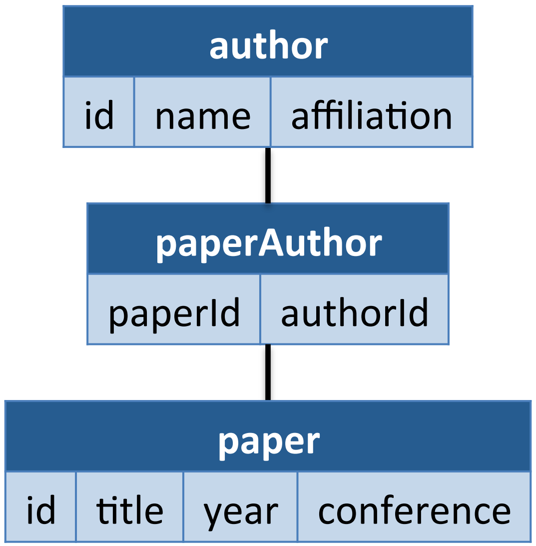
non-collaborators	
person1	person2
Madden	Socher
Manning	Bailis

FOIL learning algorithm

?

FOIL: relational learning algorithm

Schema 2



collaborators(x,y) :-

Scoring function **f: P - N**

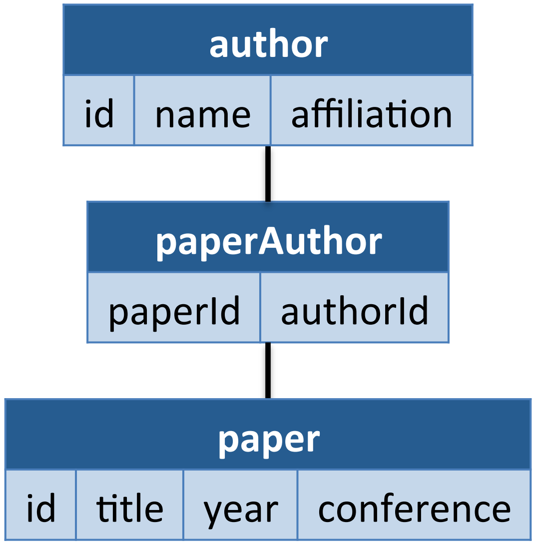
P: positive examples covered

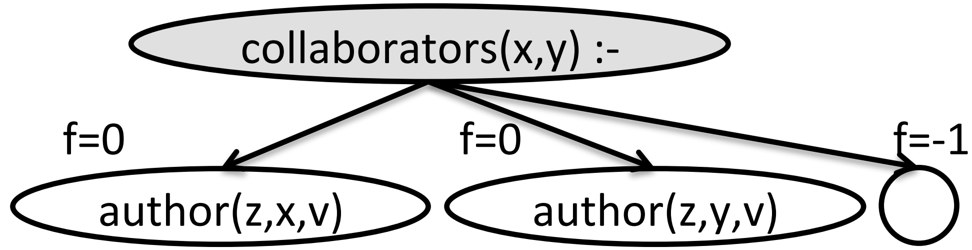
N: negative examples covered

collaborators(x,y) :-
    true.

FOIL: relational learning algorithm

Schema 2





Scoring function **f: P - N**

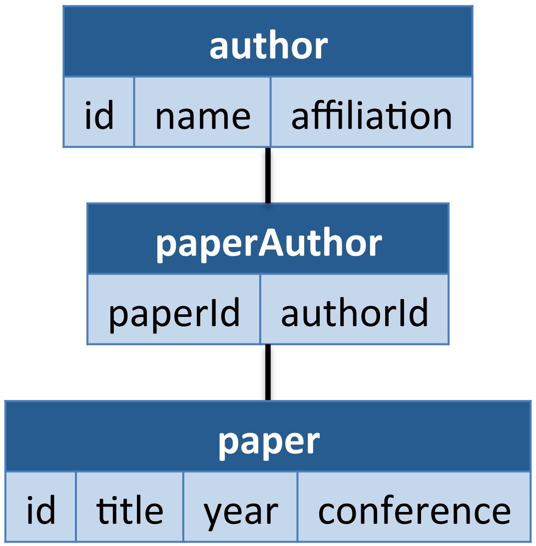
P: positive examples covered

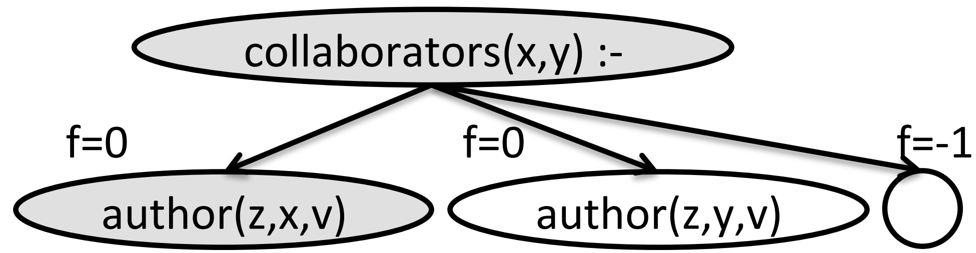
N: negative examples covered

collaborators(x,y) :-
    true.

FOIL: relational learning algorithm

Schema 2





Scoring function **f: P - N**

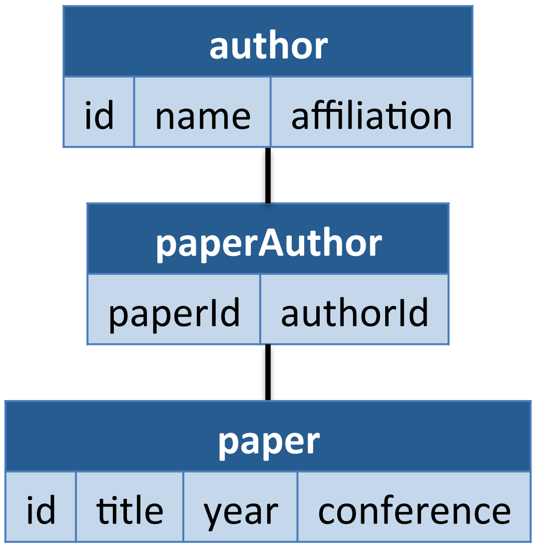
P: positive examples covered

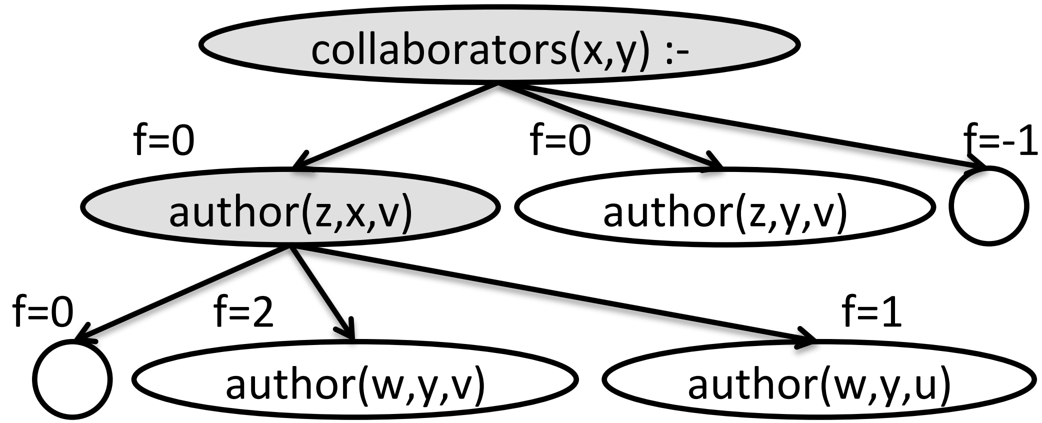
N: negative examples covered

collaborators(x,y) :-
    author(z,x,v).

FOIL: relational learning algorithm

Schema 2





Scoring function **f: P - N**

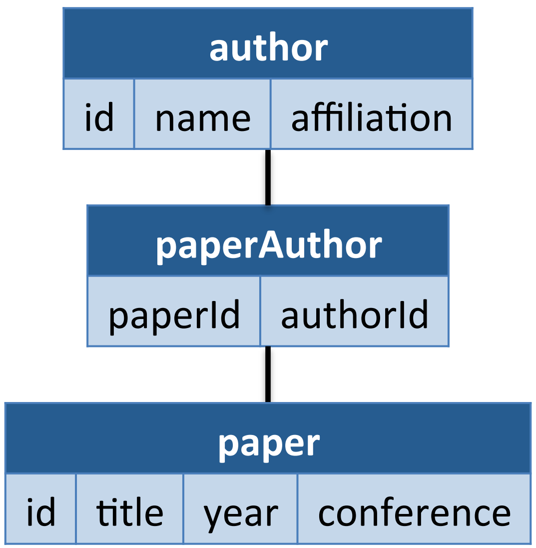
P: positive examples covered

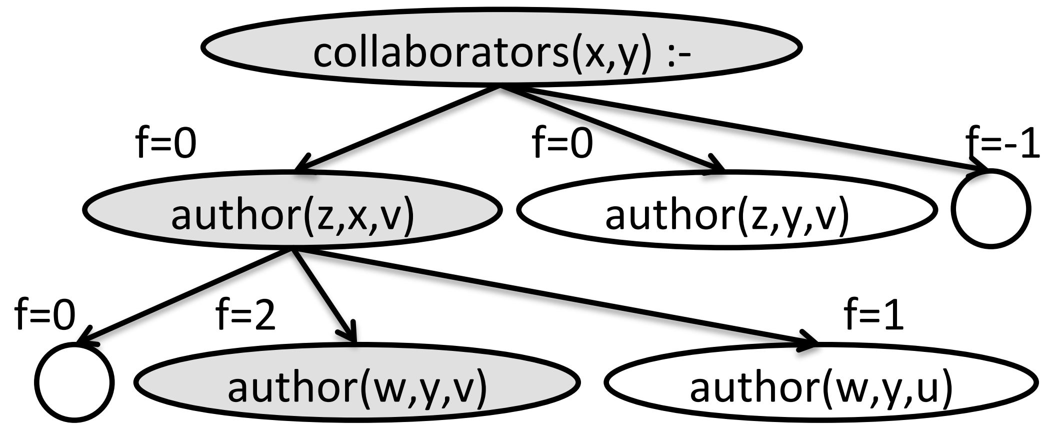
N: negative examples covered

collaborators(x,y) :-
    author(z,x,v).

FOIL: relational learning algorithm

Schema 2





Scoring function **f: P - N**

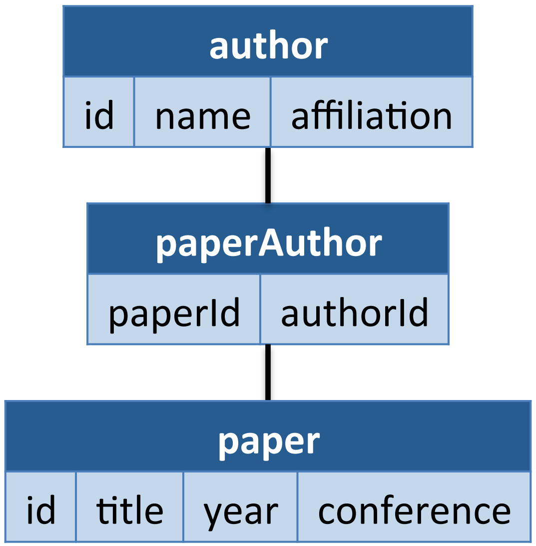
P: positive examples covered

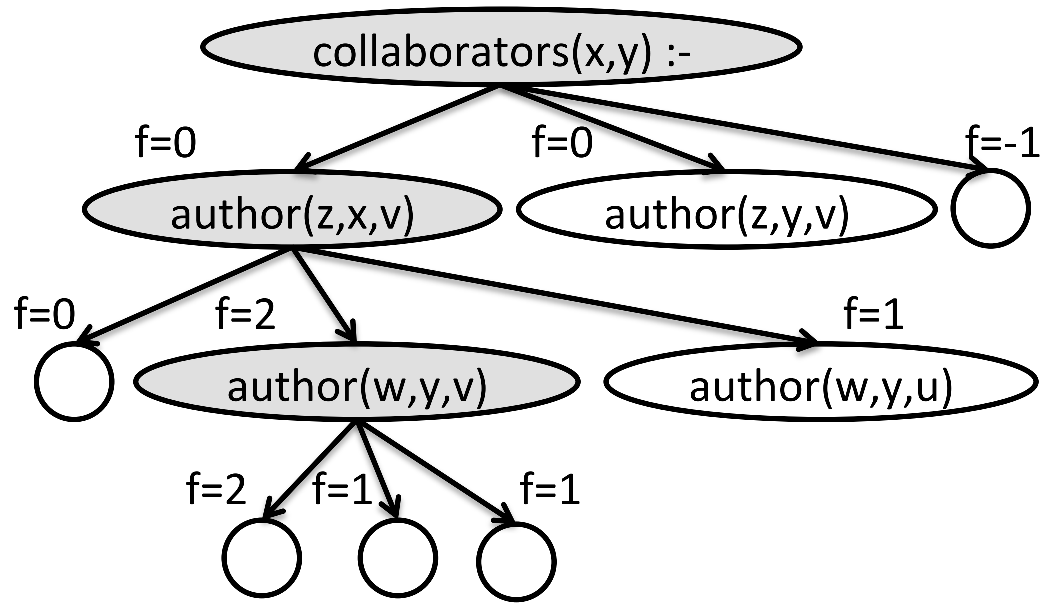
N: negative examples covered

collaborators(x,y) :-
    author(z,x,v), author(w,y,v).

FOIL: relational learning algorithm

Schema 2





No improvement

Scoring function **f: P - N**

P: positive examples covered

N: negative examples covered

collaborators(x,y) :-
    author(z,x,v), author(w,y,v).

Schema 2

paperAuthor	
paperId	authorId
p1	mad
p1	bai
p2	soc
p2	man
p3	mad

author		
id	name	affiliation
mad	Madden	MIT
sto	Stonebraker	MIT
soc	Socher	Stanford
man	Manning	Stanford
bai	Bailis	Stanford

paper			
id	title	year	conference
p1	MacroBase: Priori…	2017	SIGMOD
p2	GloVe: Global Vect…	2014	EMNLP

Which authors are **collaborators**?

collaborators	
person1	person2
Madden	Bailis
Socher	Manning
Madden	Stonebraker

non-collaborators	
person1	person2
Madden	Socher
Manning	Bailis

FOIL learning algorithm

f=2

collaborators(x,y) :-
    author(z,x,v), author(w,y,v).

Two people are collaborators if they work in the same institution.

Schema dependence: schema affects the learning outcomes

Schema 1

author	
id	name
mad	Madden
sto	Stonebraker
soc	Socher
man	Manning

authorAffiliation	
id	affiliation
mad	MIT
sto	MIT
soc	Stanford
man	Stanford

collaborators	
person1	person2
Madden	Bailis

non-collaborators	
person1	person2
Madden	Socher

Schema 2

author		
id	name	affiliation
mad	Madden	MIT
sto	Stonebraker	MIT
soc	Socher	Stanford
man	Manning	Stanford

FOIL learning algorithm

f=3

collaborators(x,y) :-
author(z,x), author(v,y),
paperAuthor (w,z), paperAuthor(w,v). ✓

Two people are collaborators if they are co-authors.

FOIL learning algorithm

f=2

collaborators(x,y) :-
author(z,x,v), author(w,y,v). ✗

Two people are collaborators if they work in the same institution.

Current solutions

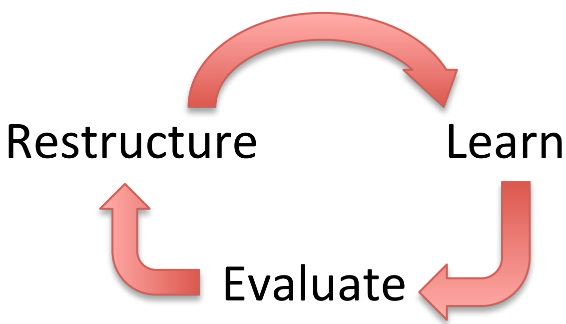
author		
id	name	affiliation
mad	Madden	MIT

→

author	
id	name
mad	Madden

authorAffiliation	
id	affiliation
mad	MIT

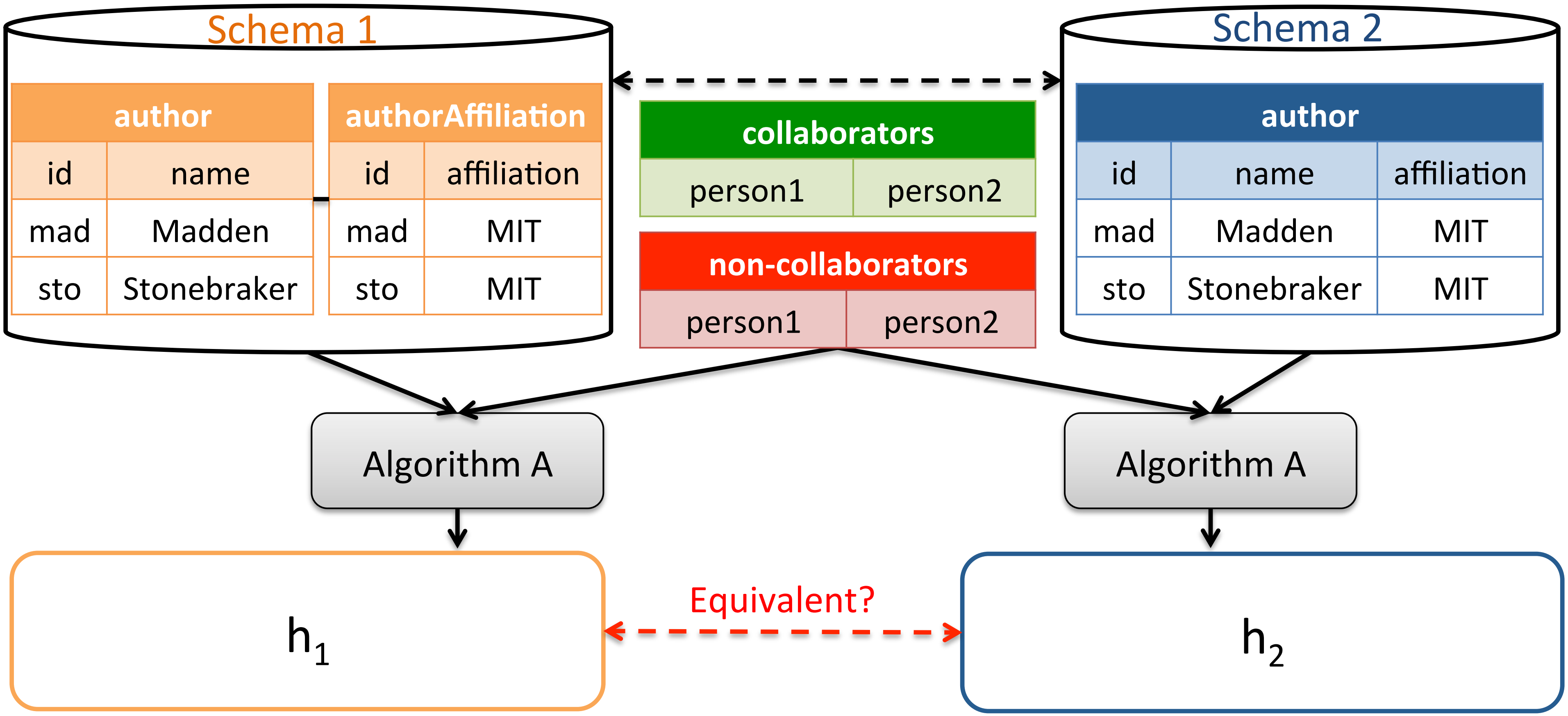
Users must restructure databases



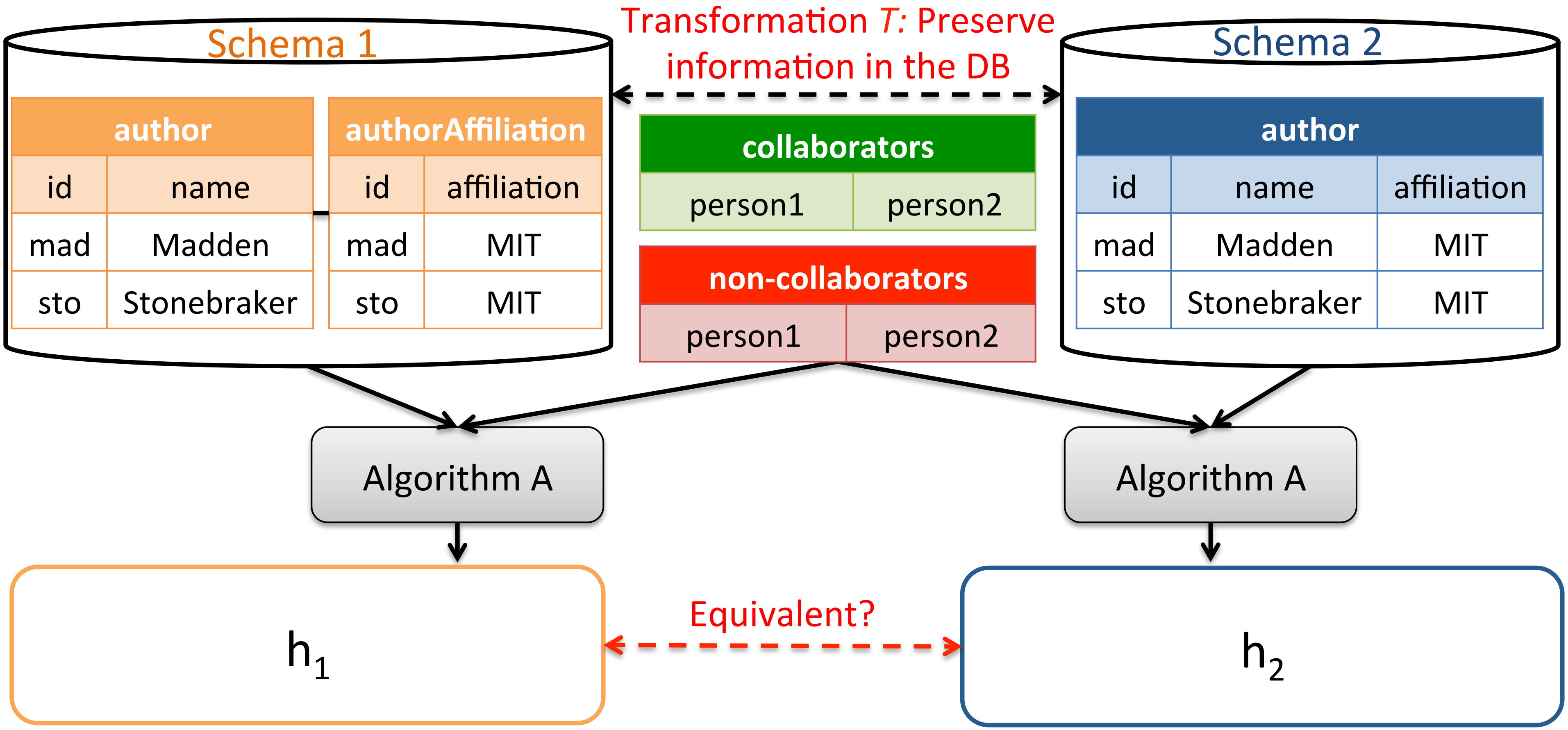
Which is the best schema?

Expert attention

Definition of schema independence



Definition of schema independence



Definition of schema independence

Transformation *T:* Preserve information in the DB

Schema 1

author	
id	name
mad	Madden
sto	Stonebraker

authorAffiliation	
id	affiliation
mad	MIT
sto	MIT

collaborators	
person1	person2

non-collaborators	
person1	person2

Schema 2

author		
id	name	affiliation
mad	Madden	MIT
sto	Stonebraker	MIT

Algorithm A

Algorithm A

f=3

h_1 = collaborators(x,y) :-
author(z,x), author(v,y),
paperAuthor(w,z), paperAuthor(w,v).

Equivalent through transformation *T*

f=3

h_2 = collaborators(x,y) :-
author(z,x,t), author(v,y,u),
paperAuthor(w,z), paperAuthor(w,v).

Algorithm A is **schema independent** under *T* iff
for all pairs of databases (*I, J*) and training examples *E*,
h_1 and h_2 are equivalent

We focus on schema independence under composition/decomposition

author		
id	name	affiliation
mad	Madden	MIT
sto	Stonebraker	MIT
soc	Socher	Stanford
man	Manning	Stanford

decomposition →

← composition

author	
id	name
mad	Madden
sto	Stonebraker
soc	Socher
man	Manning

authorAffiliation	
id	affiliation
mad	MIT
sto	MIT
soc	Stanford
man	Stanford

Inclusion dependencies (referential integrity constraints):
author[id] $\subseteq$ authorAffiliation[id]

- Most common schema transformations
- Used in normalization and denormalization
- We support combinations of compositions and decompositions

Current relational learning algorithms are NOT schema independent

Theorems:

- FOIL
- Progol
- ProGolem

are NOT schema independent under composition/decomposition

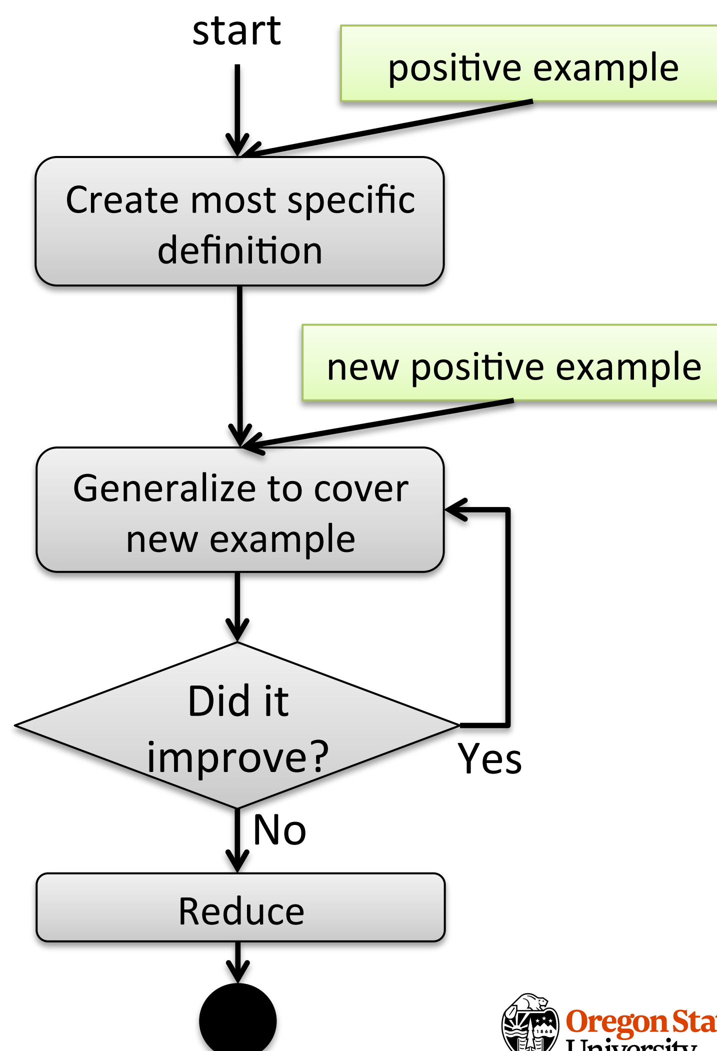
□

Reasons for schema dependence:

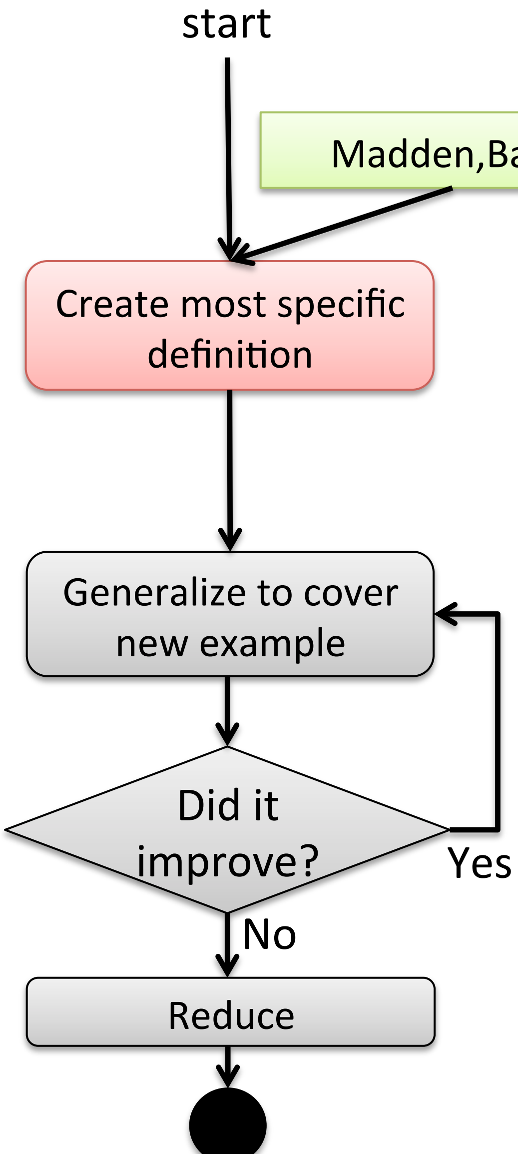
- Search process affected by schema
- Greedy search strategies

Our algorithm: **Castor**
schema independent algorithm

- Specific to general definitions
- Uses database constraints to achieve schema independence



Step 1: Create most specific definition



paperAuthor	
paperId	authorId
p1	mad
p1	bai

author	
id	name
mad	Madden
bai	Bailis

authorAffiliation	
id	affiliation
mad	MIT
bai	Stanford

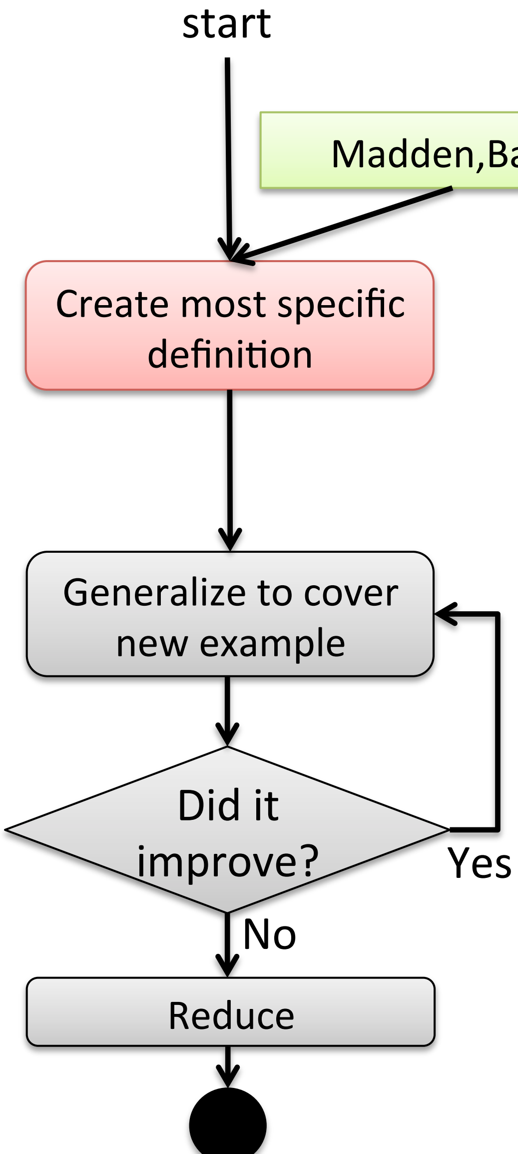
paper	
id	title
p1	MacroBase: Priori…
p2	GloVe: Global Vect…

paperYear	
id	year
p1	2017
p2	2014

paperConf	
id	conf
p1	SIGMOD
p2	EMNLP

collaborators(v_1,v_2) :-

Step 1: Create most specific definition



paperAuthor	
paperId	authorId
p1	mad
p1	bai

author	
id	name
mad	**Madden**
bai	**Bailis**

authorAffiliation	
id	affiliation
mad	MIT
bai	Stanford

paper	
id	title
p1	MacroBase: Priori…
p2	GloVe: Global Vect…

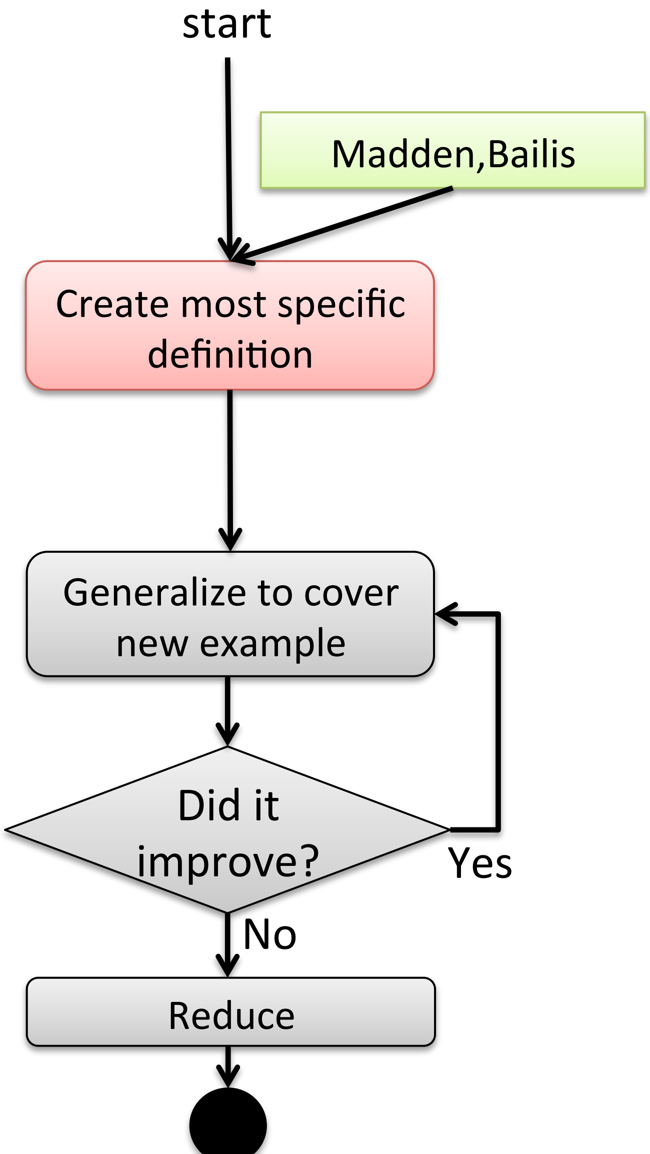
paperYear	
id	year
p1	2017
p2	2014

paperConf	
id	conf
p1	SIGMOD
p2	EMNLP

collaborators(v_1,v_2) :-
author(v_3,v_1), author(v_4,v_2).

Oregon State University

Step 1: Create most specific definition



paperAuthor	
paperId	authorId
p1	mad
p1	bai

author	
id	name
mad	Madden
bai	Bailis

authorAffiliation	
id	affiliation
mad	**MIT**
bai	**Stanford**

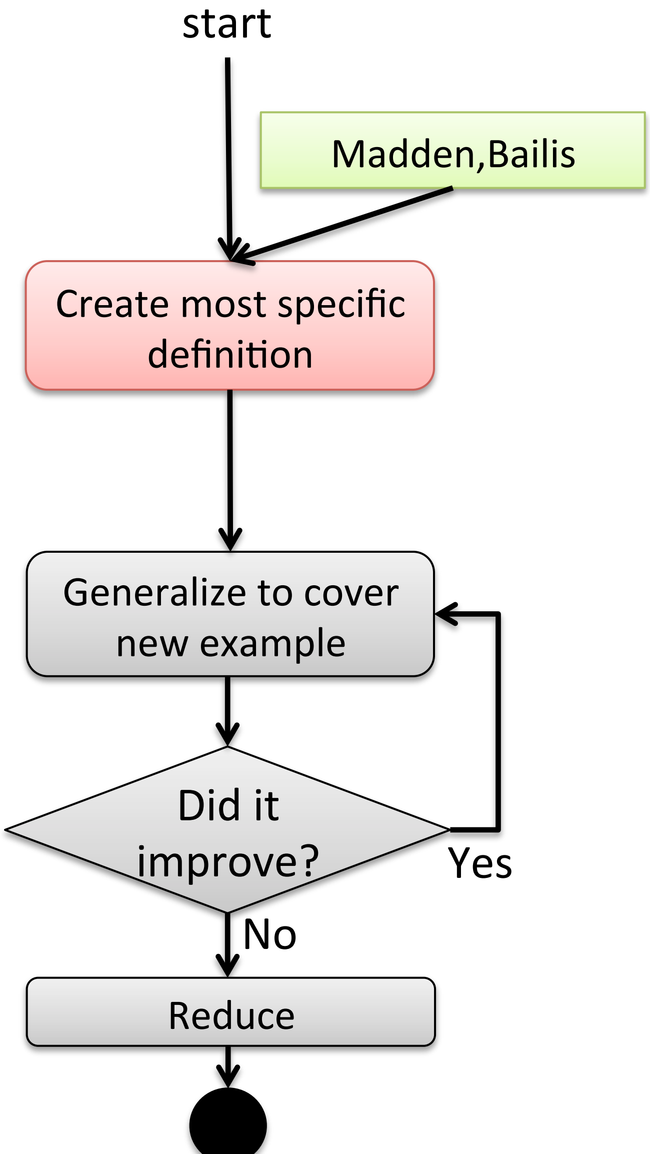
paper	
id	title
p1	MacroBase: Priori…
p2	GloVe: Global Vect…

paperYear	
id	year
p1	2017
p2	2014

paperConf	
id	conf
p1	SIGMOD
p2	EMNLP

collaborators(v_1,v_2) :-
author(v_3,v_1), author(v_4,v_2),
authorAffiliation(v_3,MIT), authorAffiliation(v_3,v_5),
authorAffiliation(v_4,Stanford), authorAffiliation(v_4,v_6).

Step 1: Create most specific definition



paperAuthor	
paperId	authorId
p1	mad
p1	bai

author	
id	name
mad	Madden
bai	Bailis

authorAffiliation	
id	affiliation
mad	MIT
bai	Stanford

paper	
id	title
p1	MacroBase: Priori…
p2	GloVe: Global Vect…

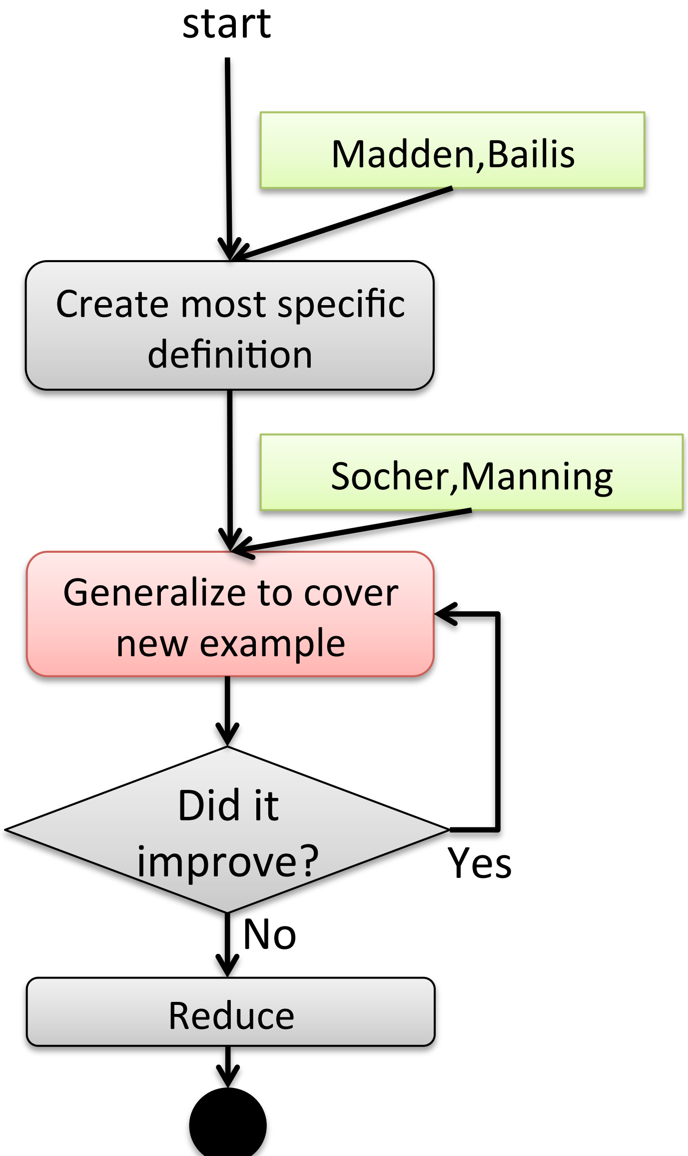
paperYear	
id	year
p1	2017
p2	2014

paperConf	
id	conf
p1	SIGMOD
p2	EMNLP

f = P – N = 1

collaborators(v_1,v_2) :-
author(v_3,v_1), author(v_4,v_2),
authorAffiliation(v_3,MIT), authorAffiliation(v_3,v_5),
authorAffiliation(v_4,Stanford), authorAffiliation(v_4,v_6),
paperAuthor(v_7,v_3), paperAuthor(v_7,v_4).

Step 2: Generalize definition



paperAuthor	
paperId	authorId
p2	soc
p2	man

author	
id	name
soc	Socher
man	Manning

authorAffiliation	
id	affiliation
soc	Stanford
man	Stanford

paper	
id	title
p2	GloVe: Global Vect…

paperYear	
id	year
p2	2014

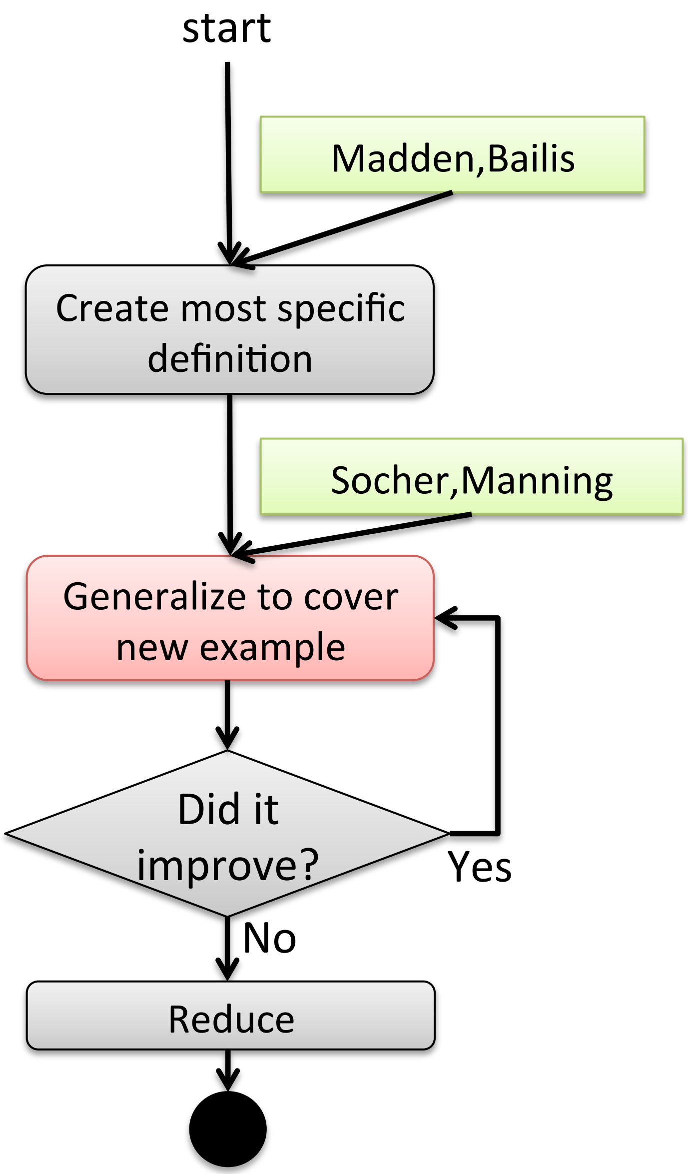
paperConf	
id	conf
p2	EMNLP

v_1 -> Socher
v_2 -> Manning

$f = P - N = 1$

collaborators(v_1,v_2) :-
author(v_3,v_1), author(v_4,v_2),
~~authorAffiliation(v_3,MIT)~~, authorAffiliation(v_3,v_5),
authorAffiliation(v_4,Stanford), authorAffiliation(v_4,v_6),
paperAuthor(v_7,v_3), paperAuthor(v_7,v_4).

Step 2: Generalize definition



paperAuthor

paperId	authorId
p2	soc
p2	man

author

id	name
soc	Socher
man	Manning

authorAffiliation

id	affiliation
soc	Stanford
man	Stanford

paper

id	title
p2	GloVe: Global Vect…

paperYear

id	year
p2	2014

paperConf

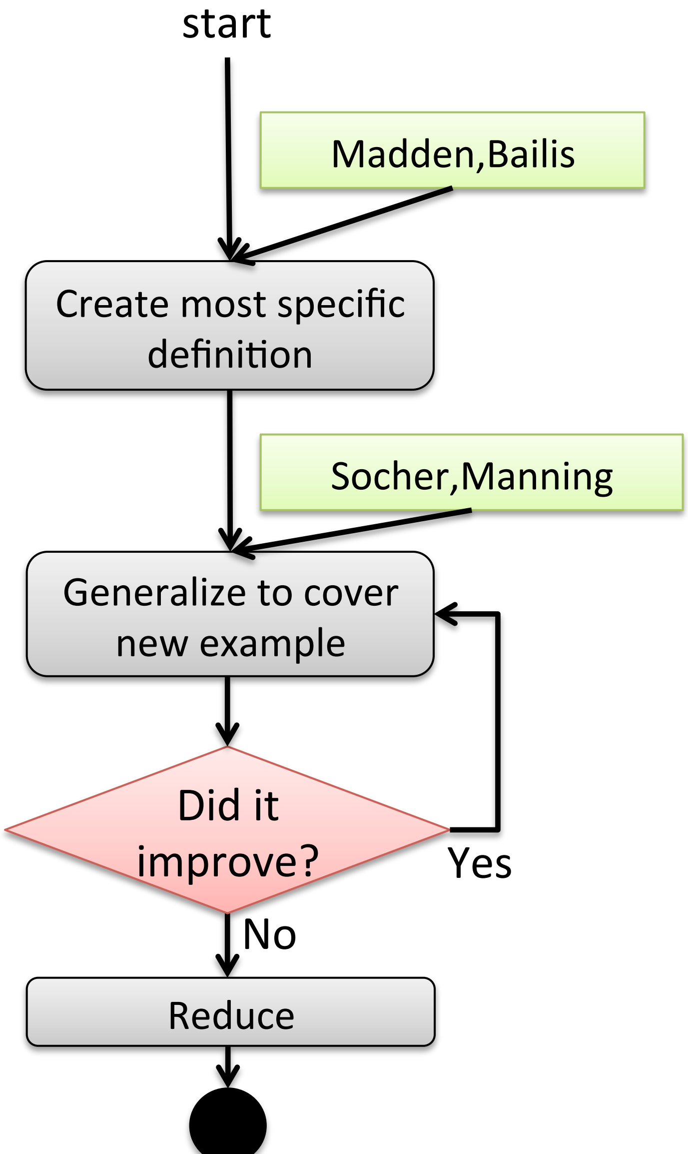
id	conf
p2	EMNLP

v_1 -> Socher
v_2 -> Manning

f = P – N = 2

collaborators(v_1,v_2) :-
author(v_3,v_1), author(v_4,v_2),
authorAffiliation(v_3,v_5),
authorAffiliation(v_4,Stanford), authorAffiliation(v_4,v_6),
paperAuthor(v_7,v_3), paperAuthor(v_7,v_4).

Step 2: Generalize definition



paperAuthor	
paperId	authorId
p2	soc
p2	man

author	
id	name
soc	Socher
man	Manning

authorAffiliation	
id	affiliation
soc	Stanford
man	Stanford

paper	
id	title
p2	GloVe: Global Vect…

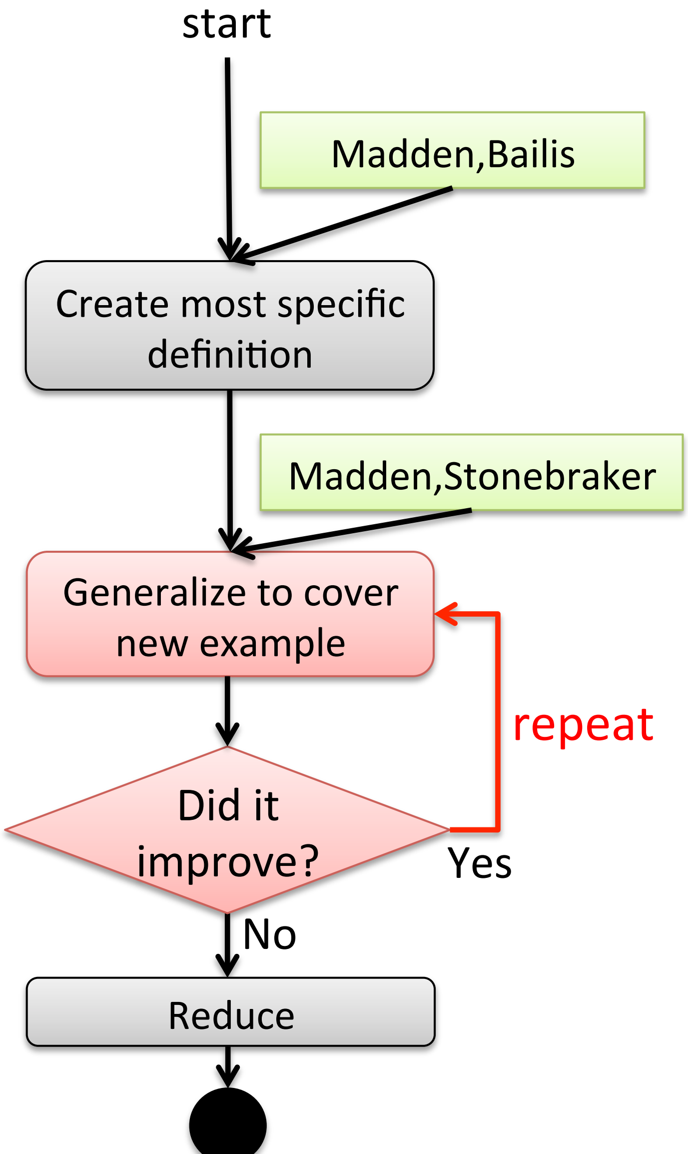
paperYear	
id	year
p2	2014

paperConf	
id	conf
p2	EMNLP

f = P – N = 2

collaborators(v_1,v_2) :-
author(v_3,v_1), author(v_4,v_2),
authorAffiliation(v_3,v_5),
authorAffiliation(v_4,Stanford), authorAffiliation(v_4,v_6),
paperAuthor(v_7,v_3), paperAuthor(v_7,v_4).

Step 2: Generalize definition



paperAuthor

paperId	authorId
p3	mad
p3	sto

author

id	name
mad	Madden
mad	Stonebraker

authorAffiliation

id	affiliation
mad	MIT
sto	MIT

paper

id	title
p3	The Data Civilizer…

paperYear

id	year
p3	2017

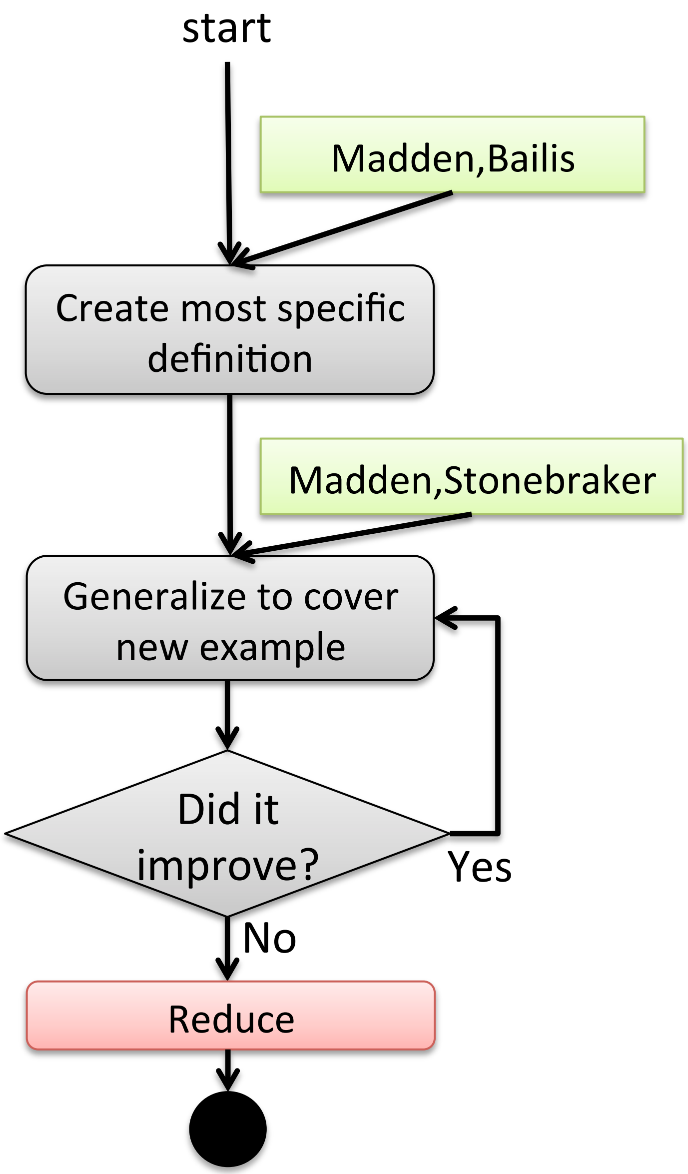
paperConf

id	conf
p3	CIDR

f = P – N = 3

collaborators(v_1,v_2) :-
author(v_3,v_1), author(v_4,v_2),
authorAffiliation(v_3,v_5), authorAffiliation(v_4,v_6),
paperAuthor(v_7,v_3), paperAuthor(v_7,v_4).

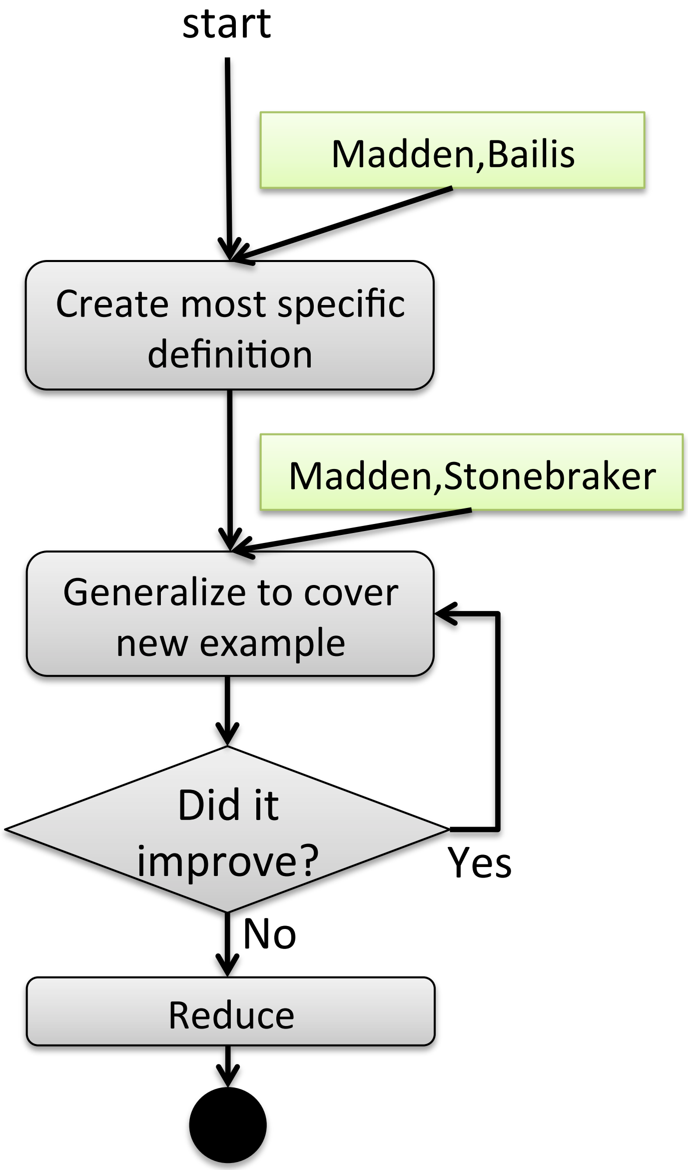
Step 3: Reduce definition



- Generalize even more to avoid overfitting
- Reduce definition using negative examples

Oregon State University

Learned definition



f = P – N = 3

collaborators(v_1,v_2) :-
author(v_3,v_1), author(v_4,v_2),
paperAuthor(v_7,v_3), paperAuthor(v_7,v_4).

Two people are collaborators if they are co-authors.

Castor achieves schema independence by using **database constraints**

author

id	name
mad	Madden
bai	Bailis

authorAffiliation

id	affiliation
mad	MIT
bai	Stanford

paperAuthor

paperId	authId
p3	mad
p3	sto

author[id]⊆authorAffiliation[id]
author[id]⊆paperAuthor[authId]

author

id	name	affiliation
mad	Madden	MIT
bai	Bailis	Stanford

paperAuthor

paperId	authId
p3	mad
p3	sto

author[id]⊆paperAuthor[authId]

Step 1: Create most specific definition using database constraints

author	
id	**name**
mad	Madden
bai	Bailis

authorAffiliation	
id	**affiliation**
mad	MIT
bai	Stanford

paperAuthor	
paperId	**authId**
p3	mad
p3	sto

author[id]$\subseteq$authorAffiliation[id]
author[id]$\subseteq$paperAuthor[authId]

collaborators(v_1, v_2) :-

Madden,Bailis

Create most specific definition

author		
id	**name**	**affiliation**
mad	Madden	MIT
bai	Bailis	Stanford

paperAuthor	
paperId	**authId**
p3	mad
p3	sto

author[id]$\subseteq$paperAuthor[authId]

collaborators(v_1, v_2) :-

Oregon State University

Step 1: Create most specific definition using database constraints

author	
id	name
mad	**Madden**
bai	**Bailis**

authorAffiliation	
id	affiliation
mad	MIT
bai	Stanford

paperAuthor	
paperId	authId
p3	mad
p3	sto

author[id]⊆authorAffiliation[id]
author[id]⊆paperAuthor[authId]

Madden,Bailis → Create most specific definition

author		
id	name	affiliation
mad	**Madden**	**MIT**
bai	**Bailis**	**Stanford**

paperAuthor	
paperId	authId
p3	mad
p3	sto

author[id]⊆paperAuthor[authId]

collaborators(v_1, v_2) :-
author(v_3,v_1), author(v_4,v_2).

collaborators(v_1, v_2) :-
author(v_3,v_1,MIT), author(v_4,v_2,Stanford).

Step 1: Create most specific definition using database constraints

author	
id	name
mad	Madden
bai	Bailis

authorAffiliation	
id	affiliation
mad	**MIT**
bai	**Stanford**

paperAuthor	
paperId	authId
p3	**mad**
p3	**sto**

author[id]⊆authorAffiliation[id]
author[id]⊆paperAuthor[authId]

Madden,Bailis

Create most specific definition

author		
id	name	affiliation
mad	Madden	MIT
bai	Bailis	Stanford

paperAuthor	
paperId	authId
p3	**mad**
p3	**sto**

author[id]⊆paperAuthor[authId]

collaborators(v_1, v_2) :-
author(v_3,v_1), author(v_4,v_2),
authorAffiliation(v_3,MIT), paperAuthor(v_3,v_5),
authorAffiliation(v_4,Stanford), paperAuthor(v_4,v_6).

$\neq$

collaborators(v_1, v_2) :-
author(v_3,v_1,MIT), author(v_4,v_2,Stanford),
paperAuthor(v_3,v_5), paperAuthor(v_4,v_6).

Ensures that the algorithm accesses the same information over all schemas

Step 2 and 3: Generalization and reduction using database constraints

author

id	name
mad	Madden
sto	Stonebraker

authorAffiliation

id	affiliation
mad	MIT
sto	MIT

paperAuthor

paperId	authId
p3	mad
p3	sto

author[id]⊆authorAffiliation[id]
author[id]⊆paperAuthor[authId]

collaborators(v_1, v_2) :-
author(v_3,v_1), authorAffiliation(v_3,MIT),
author(v_4,v_2), ~~authorAffiliation(v_4,Stanford)~~.

Madden,Stonebraker

Generalize to cover new example

author

id	name	affiliation
mad	Madden	MIT
sto	Stonebraker	MIT

paperAuthor

paperId	authId
p3	mad
p3	sto

author[id]⊆paperAuthor[authId]

collaborators(v_1, v_2) :-
author(v_3,v_1,MIT), ~~author(v_4,v_2,Stanford)~~.

Step 2 and 3: Generalization and reduction using database constraints

author	
id	name
mad	Madden
sto	Stonebraker

authorAffiliation	
id	affiliation
mad	MIT
sto	MIT

paperAuthor	
paperId	authId
p3	mad
p3	sto

author[id]⊆authorAffiliation[id]
author[id]⊆paperAuthor[authId]

collaborators(v_1, v_2) :-
author(v_3,v_1), authorAffiliation(v_3,MIT),
~~author(v_4,v_2)~~, ~~authorAffiliation(v_4,Stanford)~~.

Madden,Stonebraker

Generalize to cover new example

author		
id	name	affiliation
mad	Madden	MIT
sto	Stonebraker	MIT

paperAuthor	
paperId	authId
p3	mad
p3	sto

author[id]⊆paperAuthor[authId]

collaborators(v_1, v_2) :-
author(v_3,v_1,MIT), ~~author(v_4,v_2,Stanford)~~.

Oregon State University

Step 2 and 3: Generalization and reduction using database constraints

author	
id	name
mad	Madden
sto	Stonebraker

authorAffiliation	
id	affiliation
mad	MIT
sto	MIT

paperAuthor	
paperId	authId
p3	mad
p3	sto

author[id]⊆authorAffiliation[id]
author[id]⊆paperAuthor[authId]

Madden,Stonebraker

Generalize to cover new example

author		
id	name	affiliation
mad	Madden	MIT
sto	Stonebraker	MIT

paperAuthor	
paperId	authId
p3	mad
p3	sto

author[id]⊆paperAuthor[authId]

collaborators(v_1, v_2) :-
author(v_3,v_1), authorAffiliation(v_3,MIT).

≠

collaborators(v_1, v_2) :-
author(v_3,v_1,MIT).

Step 2 and 3: Generalization and reduction using database constraints

author	
id	name
mad	Madden
sto	Stonebraker

authorAffiliation	
id	affiliation
mad	MIT
sto	MIT

paperAuthor	
paperId	authId
p3	mad
p3	sto

author[id]$\subseteq$authorAffiliation[id]
author[id]$\subseteq$paperAuthor[authId]

Madden,Stonebraker

Generalize to cover new example

author		
id	name	affiliation
mad	Madden	MIT
sto	Stonebraker	MIT

paperAuthor	
paperId	authId
p3	mad
p3	sto

author[id]$\subseteq$paperAuthor[authId]

collaborators(v_1, v_2) :-
author(v_3,v_1), authorAffiliation(v_3,MIT).

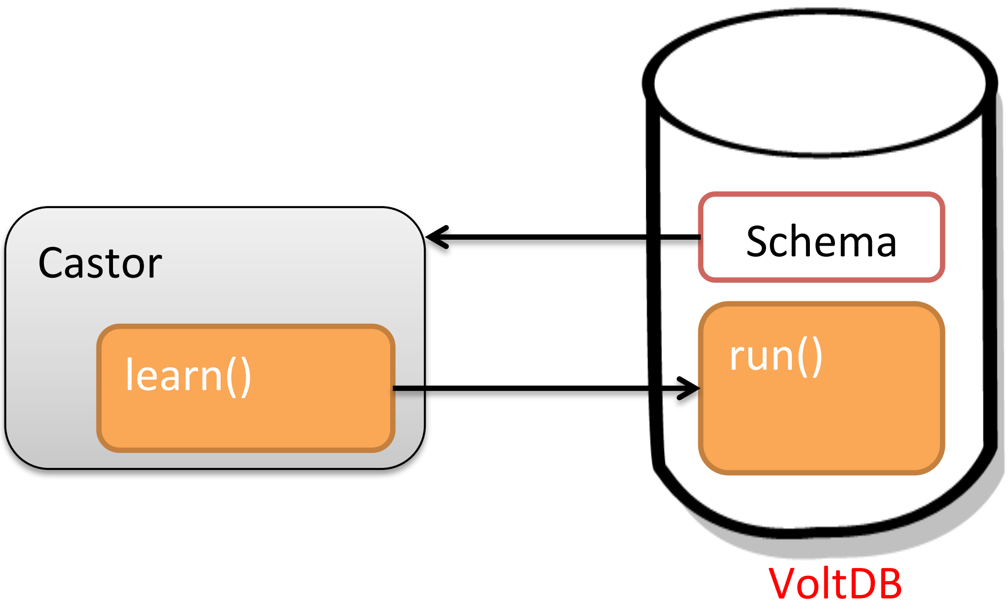
$\neq$

collaborators(v_1, v_2) :-
author(v_3,v_1,MIT).

Theorem: Castor is schema independent under composition / decomposition.

Techniques to achieve efficiency

1. Castor is implemented on top of the in-memory RDBMS VoltDB
   - Exploit RDBMS mechanisms
   - Part of the algorithm implemented in a stored procedure
2. Approximate and efficient definition minimization



Techniques to achieve efficiency

3. Castor efficiently checks whether a definition covers an example

Alternative approach:

Datalog:
collaborators(x,y) :-
    author(z,x), author(v,y), paperAuthor(w,z), paperAuthor(w,v).

SQL:
SELECT c.person1, c.person2
FROM collaborators c, author a1, author s2, paperAuthor pa1, paperAuthor pa2
WHERE c.person1 = a1.name AND c.person2 = a2.name AND a1.id = pa1.authorId
    AND a2.id = pa2.authorId AND pa1.id = pa2.id;

Castor's approach:

1. Compute most specific definition h_e for example e.
2. Definition h covers example e iff there is a substitution θ such that $h\theta \subseteq h_e$ (homomorphism).

✓ More efficient

Experimental results

- Database: UW-CSE – academic department
  - 9 relations, 2K tuples
  - 102 positive examples, 204 negative examples
- Target relation: advisedBy(student, professor)

Algorithm	Metric	Schema 1	Schema 2	Schema 3	Schema 4
FOIL	F1-score	0.49	0.49	0.54	0.61
	Time (s)	18.7	20.8	30.7	30.6
Progol	F1-score	**0.68**	0.61	0.53	0.38
	Time(s)	9.7	13.2	27.9	334.8
ProGolem	F1-score	**0.68**	**0.68**	0.60	0.61
	Time (s)	24.4	28.8	26.7	54.1
Castor	F1-score	**0.68**	**0.68**	**0.68**	**0.68**
	Time (s)	**7.2**	7.4	7.9	12.4

Experimental results

- Database: HIV – structure of chemical compounds
  - 80 relations, 14M tuples
  - 5K positive examples, 36K negative examples
- Target relation: anti-HIV(compound)

Algorithm	Metric	Schema 1	Schema 2
FOIL	F1-score	0.49	0.80
	Time (h)	3	**0.9**
Castor	F1-score	**0.83**	**0.83**
	Time(h)	3.5	1.9

Progol and ProGolem do not terminate after 5 days

Conclusions and future work

- Relational learning algorithms leverage the structure of data to learn Datalog definitions
- Schema independence is a desired property
- Current algorithms are not schema independent
- Castor is schema independent, accurate and efficient

- Future work:
  - Achieve schema independence over other transformations
  - Learn over different data sources

Thank you