Logical minimisation of metarules in meta-interpretive learning

Andrew Cropper and Stephen Muggleton

Outline

- Meta-interpretive learning
- minimisation of metarules
  - motivation
  - method
  - experiments
- related work
- conclusions and future work

Meta-interpretive learning

Prolog meta-interpreter

```
prove(true).

prove((Atom,Atoms)):-
 prove(Atom),
 prove(Atoms).

prove(Atom):-
 clause(Atom,Body),
 prove(Body).
```

MIL meta-interpreter

```
prove([],G,G).

prove([Atom|Atoms],G1,G2):-
 call(Atom),
 prove(Atoms,G1,G2).

prove([Atom|Atoms],G1,G2):-
 metarule(Name,Sub,(Atom:-Body)),
 abduce(Name,Sub,G1,G3),
 prove(Body,G3,G4).
 prove(Atoms,G4,G2).
```

Metarules

Name	Metarule	Instantiation
identity	P(X,Y) ← Q(X,Y)	loves(X,Y) ← married(X,Y)
inverse	P(X,Y) ← Q(Y,X)	child(X,Y) ← parent(Y,X)
chain	P(X,Y) ← Q(X,Z), R(Z,Y)	aunt(X,Y) ← sister(X,Z), parent(Z,Y)

P,Q,R are **existentially** quantified **higher-order** variables

X*,Y,Z* are **universally** quantified **first-order** variables

Chain metarule example

program

background

parent(ann, andrew) ←

sister(dorothy, ann) ←

goal

aunt(dorothy, andrew) ←

metarule

P(X,Y)←Q(X,Z),R(Z,Y)

proof outline

substitution

θ = {P/aunt, Q/sister, R/parent}

abduction store

chain(aunt,sister,parent) ←

clause

aunt(X,Y) ← sister(X,Z), parent(Z,Y)

Definitions

- Logic programs without function symbols are called **Datalog** programs
- $\mathbf{H^2_2}$ is a fragment of Datalog where each clause has at most two literals in the body and each literal is at most dyadic
- $\mathbf{H^2_{2\ chained}}$ is a fragment of Datalog where each clause has at most two literals in the body, each literal is dyadic, and every variable appears in exactly two literals

Motivation

Completeness

Incomplete without correct set of metarules, e.g. restricted to H^1_1 with the metarule P(X) ← Q(X)

Efficiency

Number of programs in H^2_2 of size *n* with *p* primitives and *m* metarules is $O(p^{3n}m^n)$

Encapsulation

Definition. **Atomic encapsulation**. Let A be higher-order or first-order atom of the form $P(t_1, .., t_n)$. We say that $enc(A) = m(P, t_1, .., t_n)$ is an encapsulation of A

Name	Metarule	Encapsulation
identity	P(X,Y) ← Q(X,Y)	m(P,X,Y) ← m(Q,X,Y)
inverse	P(X,Y) ← Q(Y,X)	m(P,X,Y) ← m(Q,Y,X)
chain	P(X,Y) ← Q(X,Z), R(Z,Y)	m(P,X,Y) ← m(Q,X,Z), m(R,Z,Y)

Minimisation of metarules in $H^2_{2\ chained}$

Maximal set
P(X,Y) ← Q(X,Y)
P(X,Y) ← Q(Y,X)
P(X,Y) ← Q(X,Z), R(Y,Z)
P(X,Y) ← Q(X,Z), R(Z,Y)
P(X,Y) ← Q(Y,X), R(X,Y)
P(X,Y) ← Q(Y,X), R(Y,X)
P(X,Y) ← Q(Y,Z), R(X,Z)
P(X,Y) ← Q(Y,Z), R(Z,X)
P(X,Y) ← Q(Z,X), R(Y,Z)
P(X,Y) ← Q(Z,X), R(Z,Y)
P(X,Y) ← Q(Z,Y), R(X,Z)
P(X,Y) ← Q(Z,Y), R(Z,X)

Plotkin's reduction algorithm →

Minimal set
P(X,Y) ← Q(Y,X) (*inverse*)
P(X,Y) ← Q(X,Z), R(Z,Y) (*H22 chain*)

Identity metarule from minimal set

$C = P(X,Y) \leftarrow Q(Y,X)$
(*inverse*)

$C' = P'(X',Y') \leftarrow Q'(Y',X')$
(*inverse*)

$\theta = \{P/Q', X/Y', Y/X'\}$

$P'(X',Y') \leftarrow Q(X',Y')$
(identity)

Left Euclidean metarule from minimal set

$$C = P(X,Y) \leftarrow Q(Y,X)$$

(inverse)

$$D = P'(X',Y') \leftarrow Q'(X',Z'),\ R'(Z',Y')$$

(H^2_2 chain)

$$\theta = \{P/R',\ X/Z', Y/Y'\}$$

$$P'(X',Y') \leftarrow Q'(X',Z'),\ Q(Y',Z')$$

(left Euclidean)

Minimisation of metarules in $H^2_{3\ chained}$

Maximal set
P(X,Y) ← Q(X,Z), R(Z,Y)
P(X,Y) ← Q(X,Z1), R(Z1,Z2), S(Z2,Y)
P(X,Y) ← Q(X,Z1), R(Z1,Z2), S(Z2,Z3), T(Z3,Y)

Plotkin's reduction algorithm

↓

Minimal set
P(X,Y) ← Q(Y,X) (*inverse*)
P(X,Y) ← Q(X,Z), R(Z,Y) (*H22 chain*)

H^2_3 chain metarule from minimal set

$C = P(X,Y) \leftarrow Q(X,Z), R(Z,Y)$

(H^2_2 chain)

$C' = P'(X',Y') \leftarrow Q'(X',Z'), R'(Z',Y')$

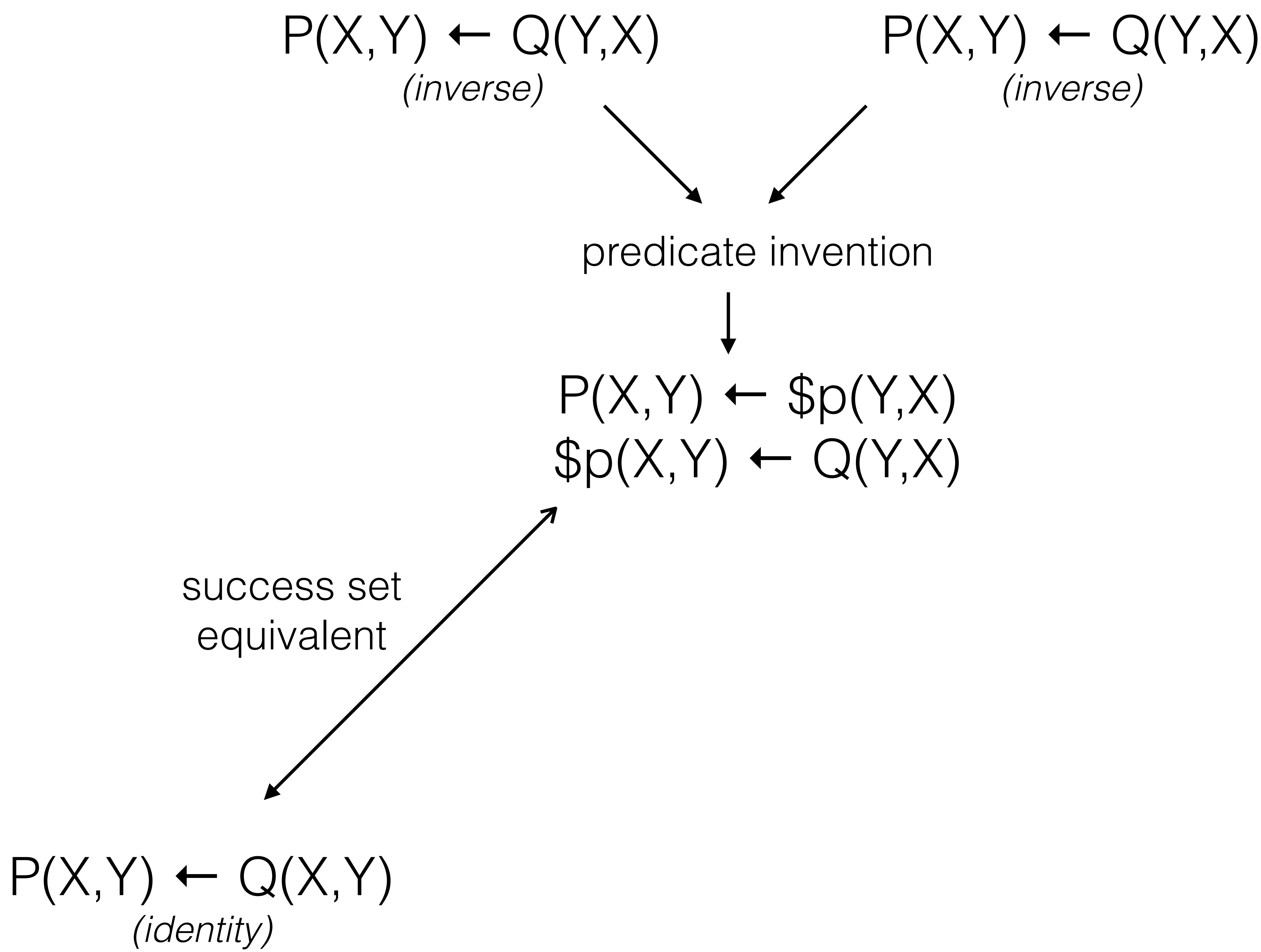
(H^2_2 chain)

$\theta = \{P/Q', X/X', Y/Z'\}$

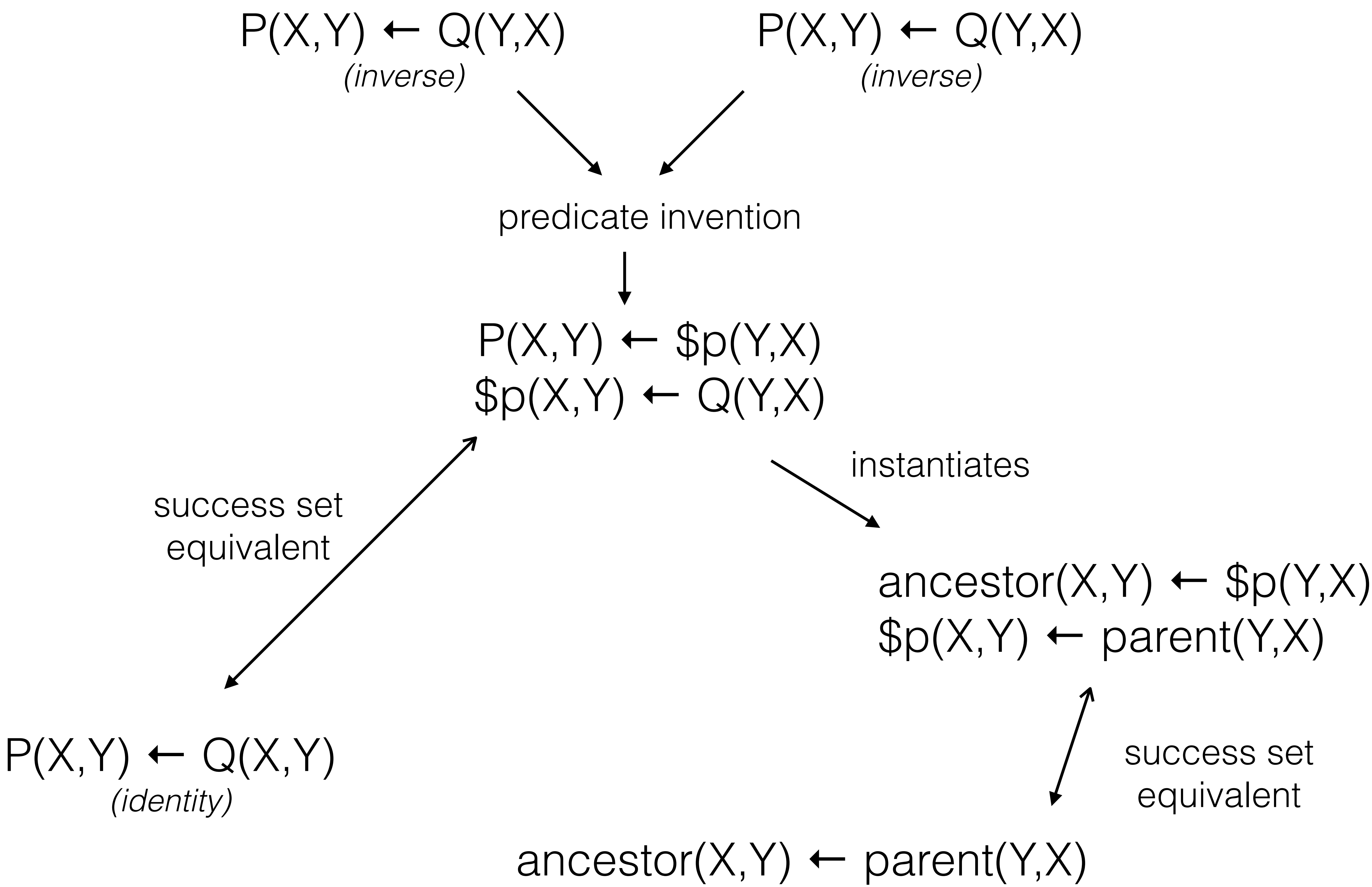
$P'(X',Y') \leftarrow Q(X',Z), R(Z,Z'), R'(Z',Y')$

(H^2_3 chain)

Identity metarule instantiation via predicate invention



Identity metarule instantiation via predicate invention



H^2_3 chain metarule instantiation

$P(X,Y) \leftarrow Q(X,Z), R(Z,Y)$

(H^2_2 chain)

$P(X,Y) \leftarrow Q(X,Z), R(Z,Y)$

(H^2_2 chain)

predicate invention

$P1(X,Y) \leftarrow \$p(X,Z), R1(Z,Y)$

$\$p(X,Y) \leftarrow Q2(X,Z), R2(Z,Y)$

success set equivalent

$P(X,Y) \leftarrow Q(X,Z1), R(Z1,Z2), S(Z2,Y)$

(H^2_3 chain)

H^2_3 chain metarule instantiation

P(X,Y) ← Q(X,Z), R(Z,Y)
(H^2_2 chain)

P(X,Y) ← Q(X,Z), R(Z,Y)
(H^2_2 chain)

predicate invention

P1(X,Y) ← $p(X,Z), R1(Z,Y)
$p(X,Y) ← Q2(X,Z), R2(Z,Y)

instantiates

success set equivalent

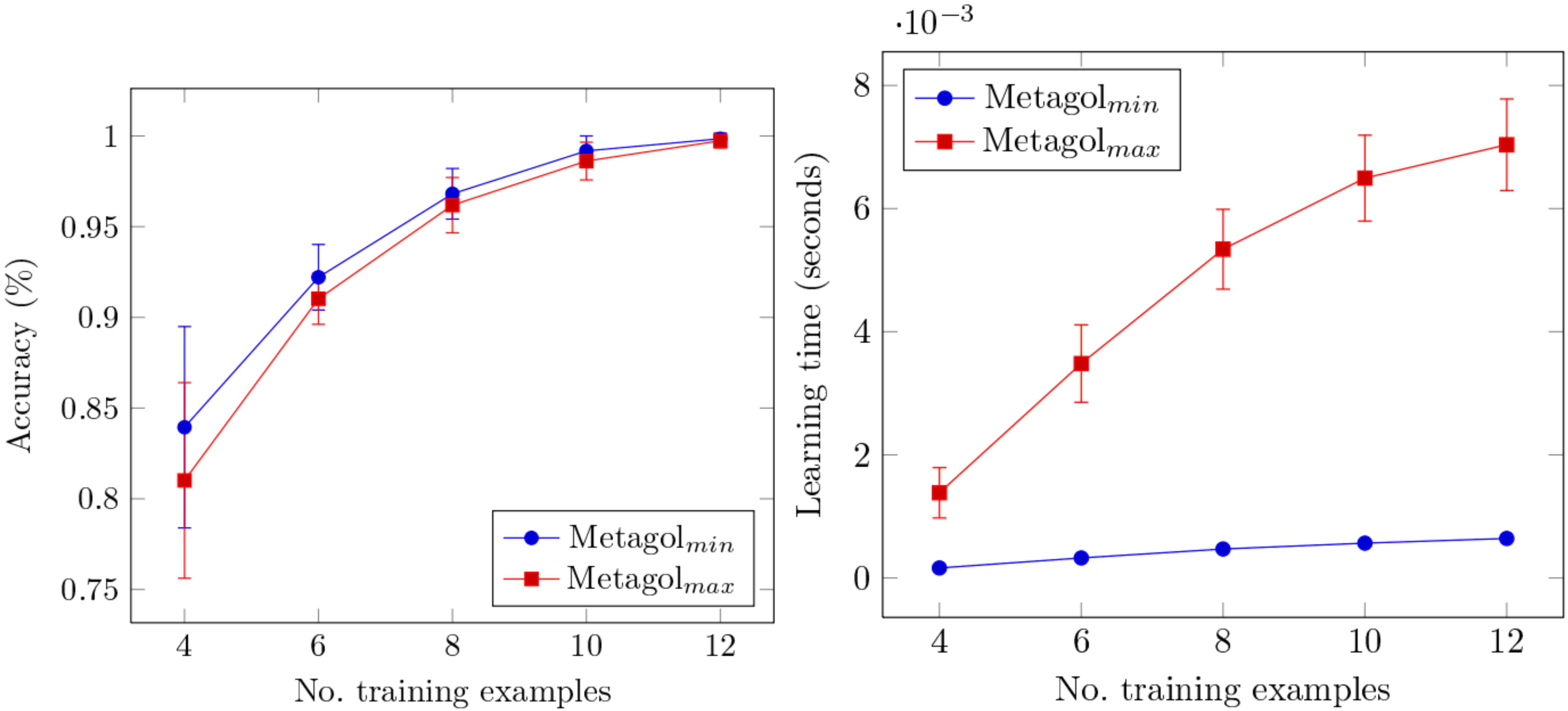
greatgrandparent(X,Y) ← $p(X,Z), parent(Z,Y)
$p(X,Y) ←parent(X,Z), parent(Z,Y)

P(X,Y) ← Q(X,Z1), R(Z1,Z2), S(Z2,Y)
(H^2_3 chain)

success set equivalent

greatgrandparent(X,Y) ← parent(X,Z1), parent(Z1,Z2), parent(Z2,Y)

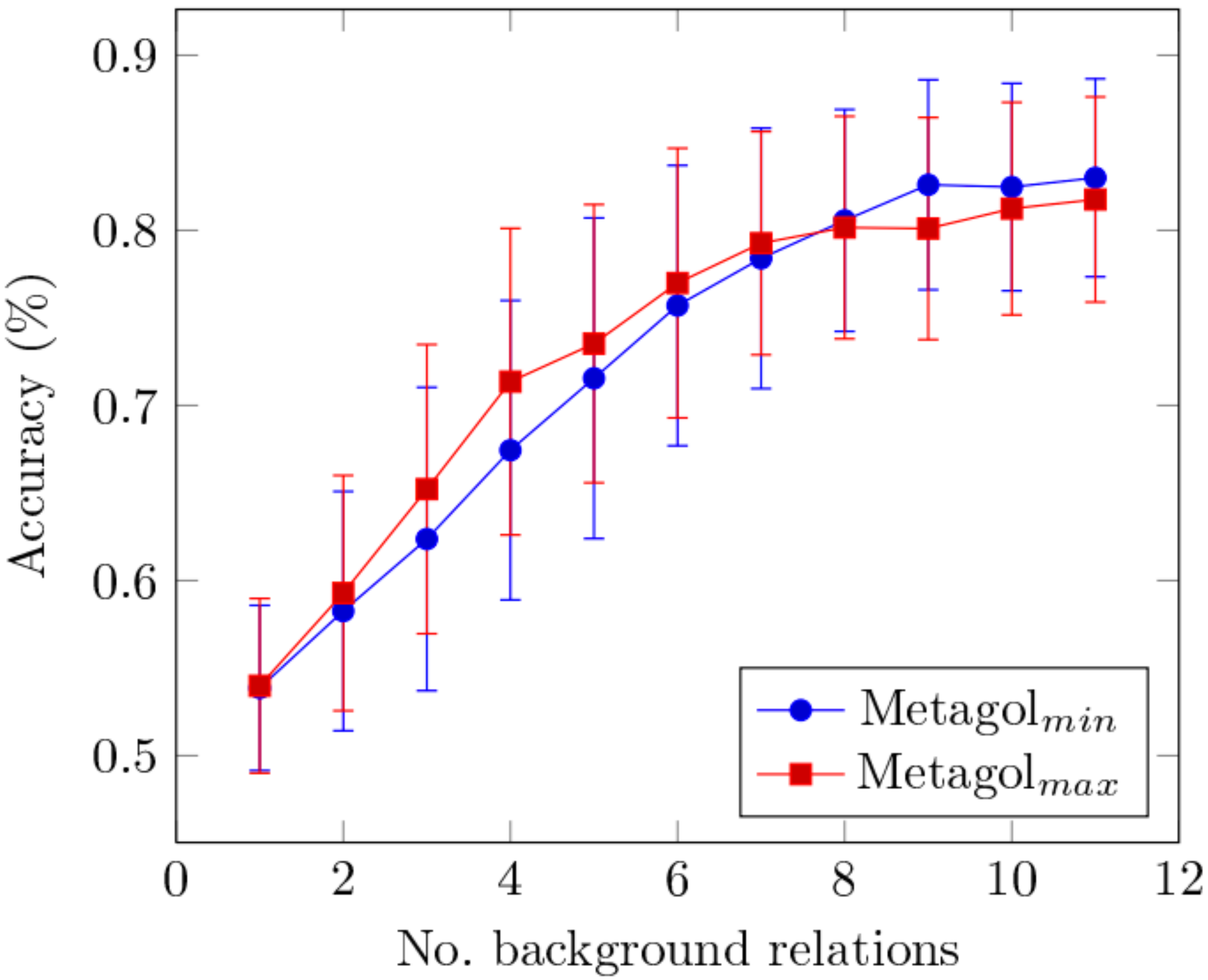
Kinship experiments - varying training data



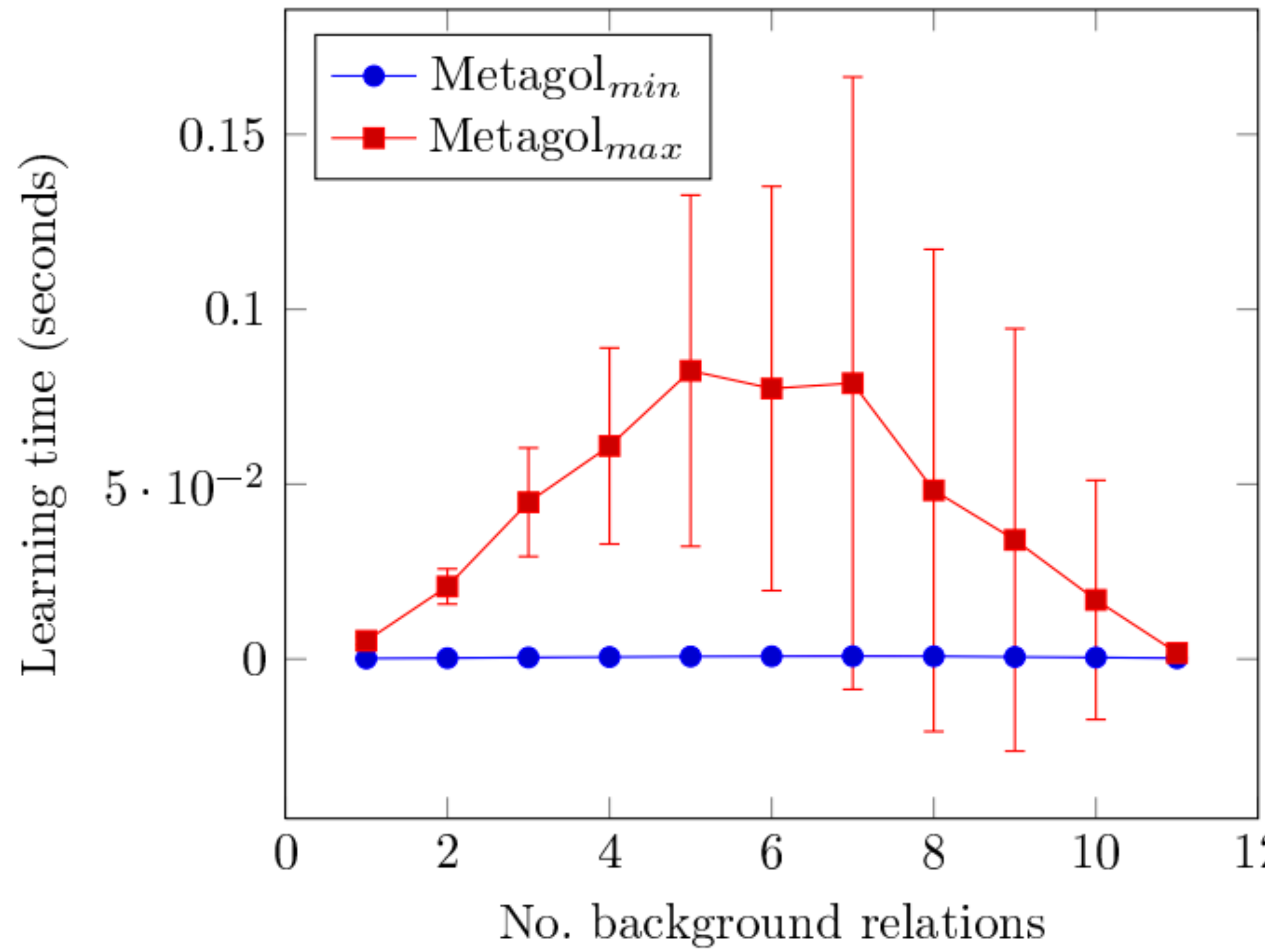
(a) Predictive Accuracies

(b) Learning Times

Kinship experiments - varying background relations

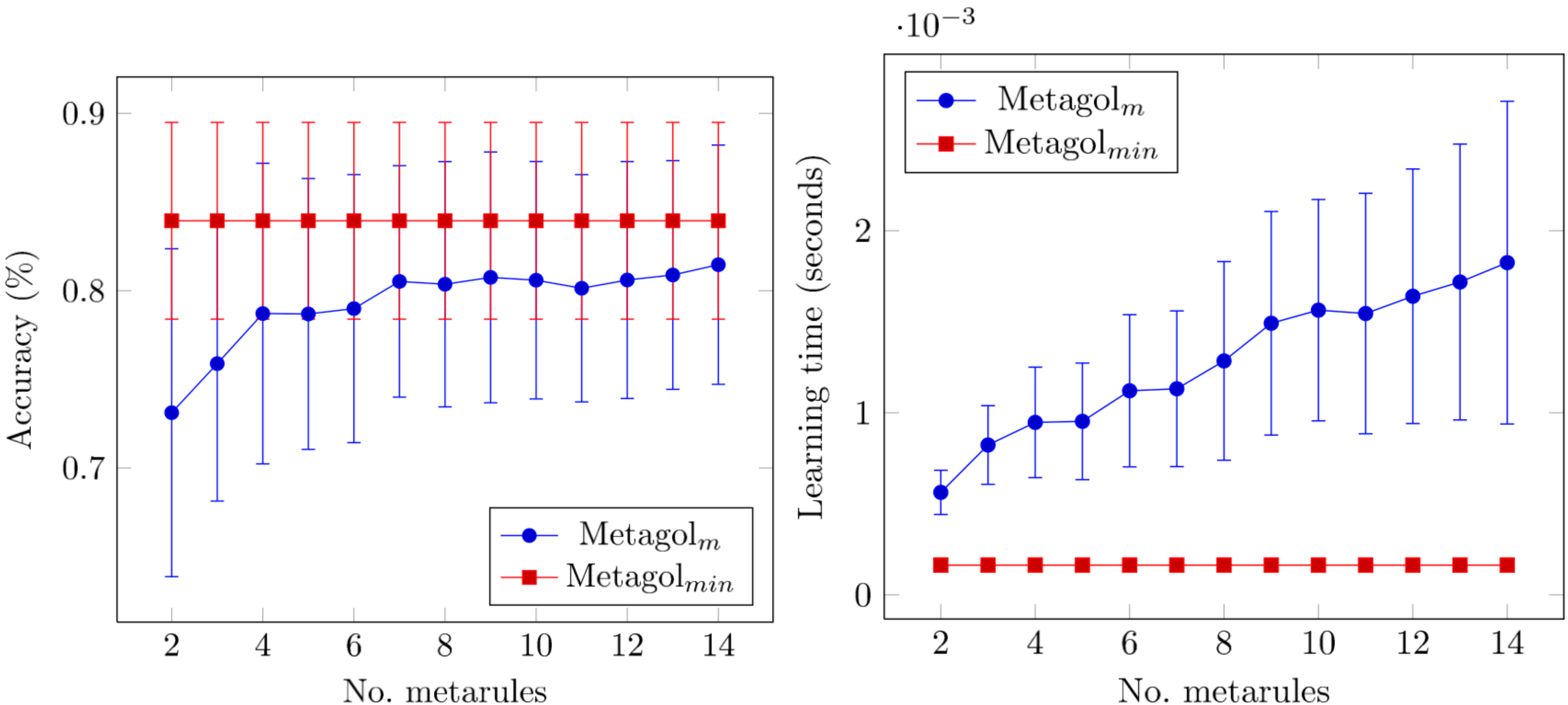


(a) Predictive Accuracies



(b) Learning Times

Kinship experiments - varying metarules



(a) Predictive Accuracies

(b) Learning Times

Related work

Meta-interpretive learning

- Meta-interpretive learning: application to grammatical inference [Muggleton et al, 2014]
- Meta-interpretive learning of higher-order dyadic datalog: Predicate invention [Muggleton & Lin, 2013]
- Bias reformulation for one-shot function induction [Lin et al, 2014]

ILP search

- Probabilistic search techniques: A study of two probabilistic methods for searching large spaces with ILP [Srinivasan, 2000]
- Query packs: Improving the efficiency of inductive logic programming through the use of query packs [Blockeel, et al, 2002]
- Special purpose hardware: Scalable acceleration of inductive logic programs [Muggleton, et al, 2001]

Refinement operators

- Algorithmic program debugging [Shapiro, 1983]
- Foundations of Inductive Logic Programming [Nienhuys-Cheng & Wolf, 1997]

Declarative bias

- Modes: Inverse entailment and Progol [Muggleton, 1995], The ALEPH manual [Srinivasan, 2001]
- Grammars: Grammatically biased learning: learning logic programs using an explicit antecedent description language [Cohen, 1994]

Conclusions and further work

Conclusions

- two metarules are complete and sufficient for generating all hypotheses in H^2_m*
- minimal set of metarules achieves higher predictive accuracies and lower learning times than the maximal set

Further work

- investigate the broader class of H^2_m
- minimise the metarules with respect to background clauses

Thank you

a.cropper13@imperial.ac.uk

s.muggleton@imperial.ac.uk