# Inductive Logic Programming
# Lecture 1.2
# Meta-Interpretive Learning of Grammars

Stephen Muggleton
Department of Computing
Imperial College, London

5th July, 2018

## Papers for this lecture

**Paper03:** S.H. Muggleton, D. Lin, N. Pahlavi, and A. Tamaddoni-Nezhad. Meta-interpretive learning: application to grammatical inference. Machine Learning, 94:25-49, 2014.

## Motivation

Logic Program (Kowalski, 1980)

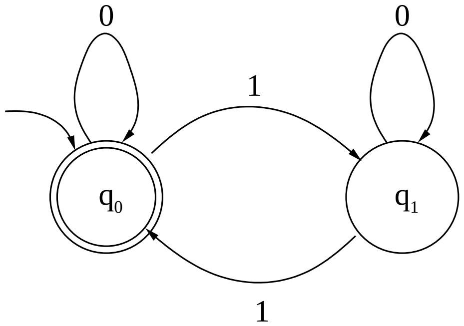
Inductive Logic Programming (Muggleton, 1991)

Machine Learn arbitrary programs

State-of-art ILP systems cannot learn grammars. Why?

Predicate Invention and Recursion (Muggleton et al, 2011)

## Parity example

| **Finite acceptor** | **Definite Clause Grammar (DCG)** | **Positive examples** |
|---|---|---|
|   | $q_0([],[]) \leftarrow$ | $\lambda$ |
| | $q_0([0\|A],B) \leftarrow q_0(A,B)$ | 0 |
| | $q_0([1\|A],B) \leftarrow q_1(A,B)$ | 11 |
| | $q_1([0\|A],B) \leftarrow q_1(A,B)$ | 00 |
| | $q_1([1\|A],B) \leftarrow q_0(A,B)$ | 101 |

DCG general form

$$
\begin{aligned}
Q([],[]) &\leftarrow \\
Q([C|x],y) &\leftarrow P(x,y)
\end{aligned}
$$

## Meta-Interpreter version of Parity

Predicate invention via higher-order abduction?

$$
\begin{aligned}
Q([],[]) &\leftarrow \\
Q([C|x],y) &\leftarrow P(x,y)
\end{aligned}
$$

| **Meta-Interpreter (Regular)** | **Ground facts** |
|---|---|
| $parse(S) \leftarrow parse(q0, S, [])$. | $acceptor(q0) \leftarrow$ |
| $parse(Q, [], []) \leftarrow acceptor(Q)$. | $delta1(q0, 0, q0) \leftarrow$ |
| $parse(Q, [C\|X], Y) \leftarrow$ | $delta1(q0, 1, q1) \leftarrow$ |
| $\quad delta1(Q, C, P)$, | $delta1(q1, 0, q1) \leftarrow$ |
| $\quad parse(P, X, Y)$. | $delta1(q1, 1, q0) \leftarrow$ |

**Meta-Interpretive Learning (MIL) setting**

| | |
|---|---|
| Input | $\langle B, E\rangle$ where $B = \langle B_M, B_A\rangle$,<br>$B_M$ is Meta-Interpreter and $B_A$ is Atomic background<br>$E = \langle E^+, E^-\rangle$ are positive and negative examples |
| Output | $H \in \mathcal{H}_{B,E}$ where $H$ is higher-order existentially-quantified Datalog atoms such that $B, H \models E^+$ and $B, E^-$ consistent |
| Inverse Entailment | $B, \neg E^+ \models \neg H$<br>where $\neg E^+, \neg H$ are Universally-quantified denials |

## MIL examples for Parity

| $E^+$ | $\neg E^+$ | $E^-$ |
|---|---|---|
| $parse([]) \leftarrow$ | $\leftarrow \quad parse([]),$ | $\leftarrow parse([1])$ |
| $parse([1,1]) \leftarrow$ | $parse([1,1]),$ | $\leftarrow parse([0,1])$ |
| $parse([0,1,1]) \leftarrow$ | $parse([0,1,1]),$ | $\leftarrow parse([1,0])$ |
| $parse([1,0,1]) \leftarrow$ | $parse([1,0,1]),$ | $\leftarrow parse([0,0,1])$ |
| $parse([1,1,0]) \leftarrow$ | $parse([1,1,0]).$ | $\leftarrow parse([1,1,1])$ |

## Hypothesis Ordering and properties

**Definition ($\succeq_{B,E}$ relation in MIL)** Within the MIL setting we say that $H \succeq_{B,E} H'$ in the case that $H, H' \in \mathcal{H}_{B,E}$ and $\neg H' \succeq_{\theta} \neg H$.

**Proposition (Lattice)** $\langle \mathcal{H}_{B,E}, \succeq_{B,E} \rangle$ forms a lattice.

**Proposition (Unique $\top$)** There exists $\top \in \mathcal{H}_{B,E}$ such that $\top \succeq_{B,E} H$ for each $H \in \mathcal{H}_{B,E}$ and $\top$ is unique up to renaming of Skolem constants.

**Proposition (Unique $\bot$)** For finite $\mathcal{H}_{B,E}$ there exists $\bot$ such that $H \succeq_{B,E} \bot$ for each $H \in \mathcal{H}_{B,E}$ and $\bot$ is unique up to renaming of Skolem constants.

## Meta-Interpreter for Context-Free Grammars

$$parse(S) \leftarrow start(Q), parse(Q, S, []).$$

$$parse(Q, X, X) \leftarrow acceptor(Q).$$

$$parse(Q, [C|X], Y) \leftarrow delta1(Q, C, P), parse(P, X, Y).$$

$$parse(Q, X, Y) \leftarrow delta2(Q, P, C), parse(P, X, [C|Y]).$$

$$parse(Q, X, Y) \leftarrow delta3(Q, P, R), parse(P, X, Z), parse(R, Z, Y).$$

## **Metagol$_R$ implementation in Prolog**

```
parse(S,G1,G2) :- parse(s(0),S,[],G1,G2).

parse(Q,X,X,G1,G2) :- abduce(acceptor(Q),G1,G2).
parse(Q,[C|X],Y,G1,G2) :- skolem(P), abduce(delta1(Q,C,P),G1,G3),
        parse(P,X,Y,G3,G2).

abduce(X,G,G) :- member(X,G).
abduce(X,G,[X|G]) :- not(member(X,G)).

skolem(s(0)). skolem(s(1)). ...
```

**Prolog query and answer for Metagol$_R$**

Query

```
:- parse([],[],G1), parse([0],G1,G2), parse([0,0],G2,G3),    % Positive
       parse([1,1],G3,G4), parse([0,0,0],G4,G5),
       parse([0,1,1],G5,G6), parse([1,0,1],G6,G),
       not(parse([1],G,G)), not(parse([0,1],G,G)).            % Negative
```

Answer

```
G = [delta1(s(1),0,s(1)), delta1(s(1),1,s(0)),
       delta1(s(0),1,s(1)), delta1(s(0),0,s(0)),
       acceptor(s(0))]
```

**Experiments**

**Null Hypothesis 1.1** Metagol$_R$ cannot learn randomly chosen Regular languages.

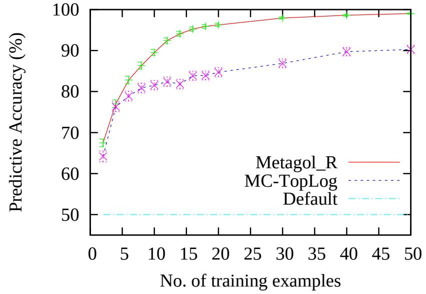
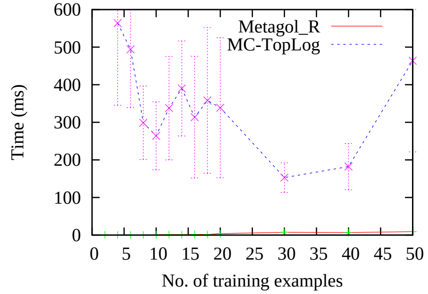
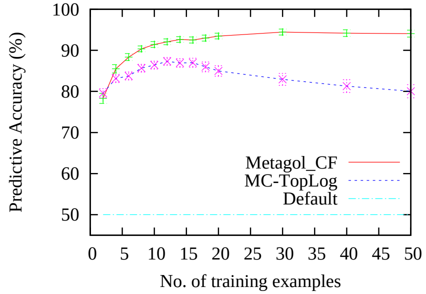
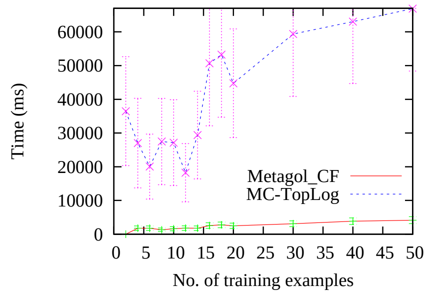
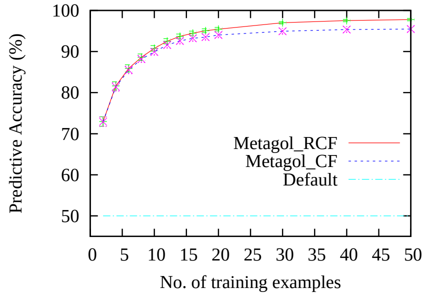
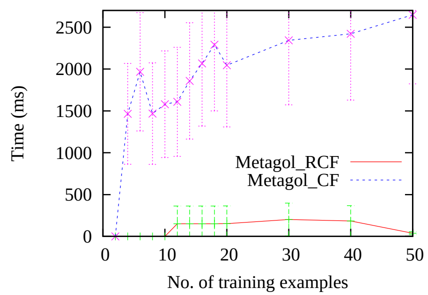
**Null Hypothesis 1.2** Metagol$_R$ cannot outperform a state-of-the-art ILP system on learning randomly chosen Regular languages.

**Null Hypothesis 2.1** Metagol$_{CF}$ cannot learn randomly chosen Context-Free languages.

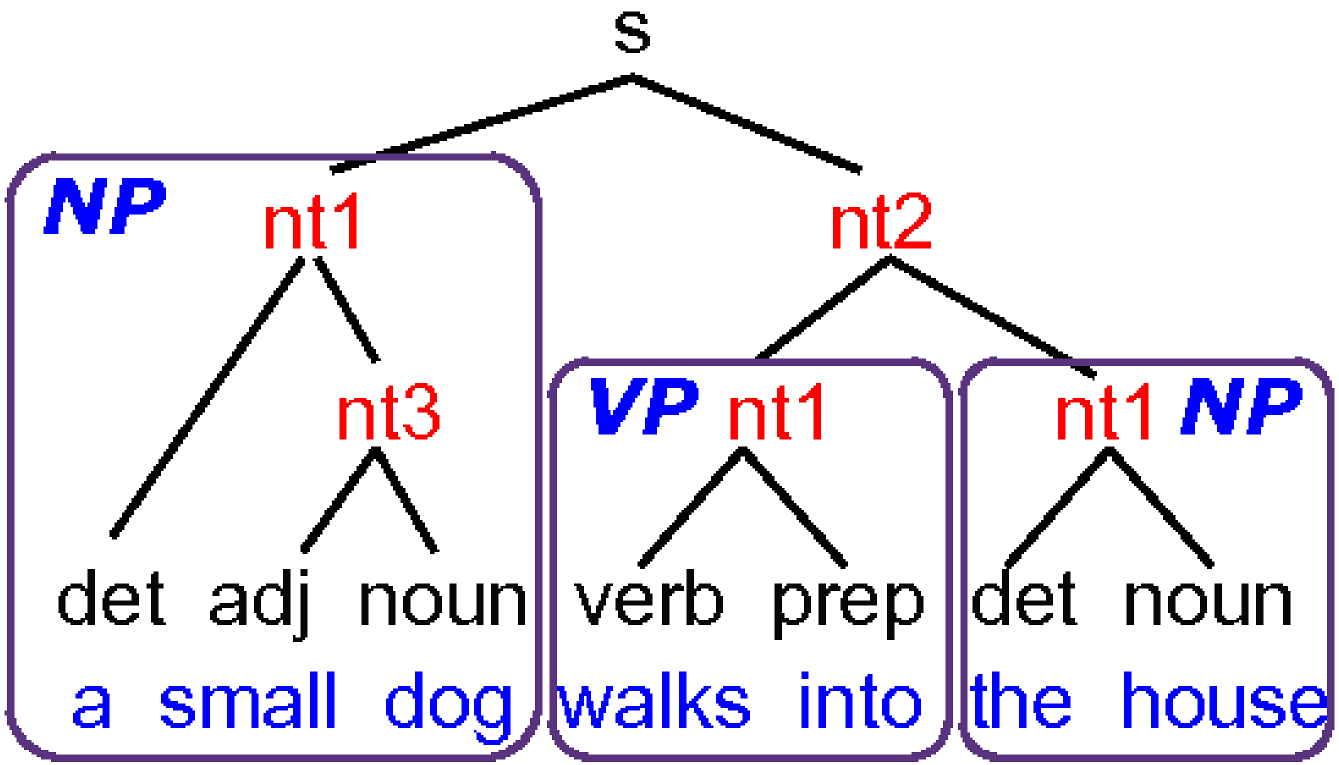
**Null Hypothesis 2.2** Metagol$_{CF}$ cannot outperform a state-of-the-art ILP system on learning randomly chosen Context-Free languages.

**Null Hypothesis 3** Metagol$_{RCF}$ cannot improve performance by changing representation from Regular to Context-Free languages.

# Results

<table>
<tr><th>Null</th><th>Predictive accuracy</th><th>Running time</th></tr>
<tr><td>1</td><td>
</td><td>
</td></tr>
<tr><td>2</td><td>
</td><td>
</td></tr>
<tr><td>3</td><td>
</td><td>
</td></tr>
</table>

**MIL invention in natural grammars**



## Related work

**Meta-level abduction** Propositional predicate invention by abduction (Inoue et al, 2009/2010) applied to biochemical networks and cello playing.

**Grammatical inference** Learning of grammars from examples (Higuera, 2010).

**Automata learning** Studied since 1950s (Moore, 1956). Provably efficient algorithms for deterministic finite state acceptors.

**Learnability** Both regular and CF learnable in the limit from positive/negative presentations. Regular languages PAC-learnable, but CF not known to be.

**Context-free** Efficient and complete approach (Sakakibara, 2002). Only heuristic approaches (Langley and Stromsten, 2000) for learning CF from positive and negative examples.

## Summary and limitations

**Meta-Interpretive Learning** Theory, implementation and experiments.

**Grammar application** MIL outperforms MC-TopLog in accuracy and speed.

**Predicate invention and recursion** MIL ease in implementing predicate invention through Higher-order Datalog constructs.

**Declarative bias** Learn bias for MIL hypothesis space in first-order logic.

**Further work** Answer Set Programming (ASP), chart parsing, natural grammars, non-empty ground background knowledge, meta-interpreters for other first-order fragments such as Monadic and Dyadic logic.