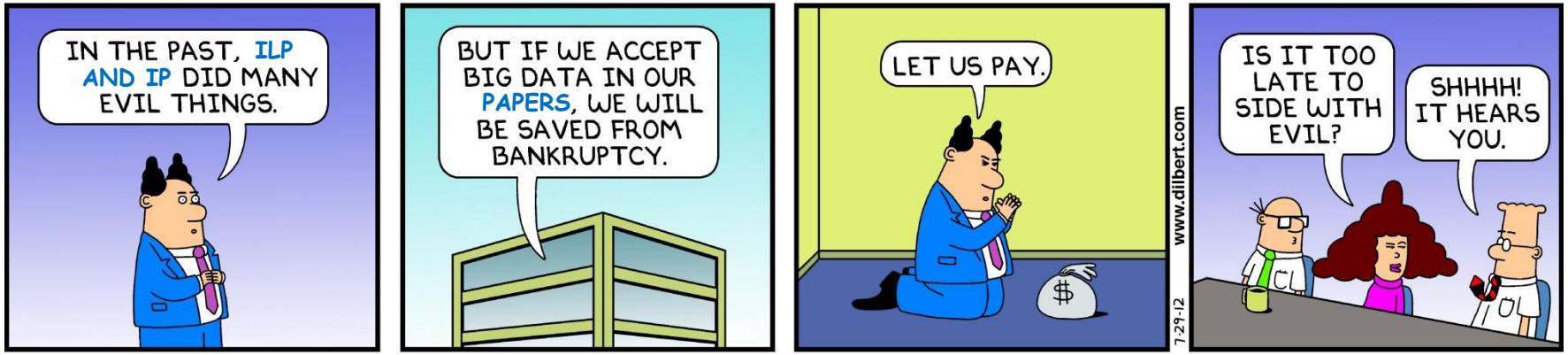
SMALL but DEEP: What can we learn from inductive programming?

José Hernández-Orallo
Dep. de Sistemes Informàtics i Computació,
Universitat Politècnica de València
jorallo@dsic.upv.es

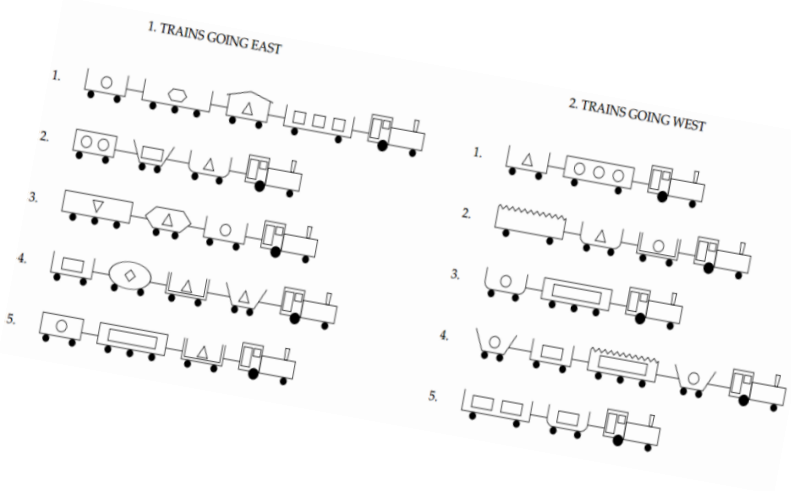
"Approaches and Applications of Inductive Programming"
Dagstuhl Seminar, December 8-11, 2013
http://www.dagstuhl.de/13502/Materials/.

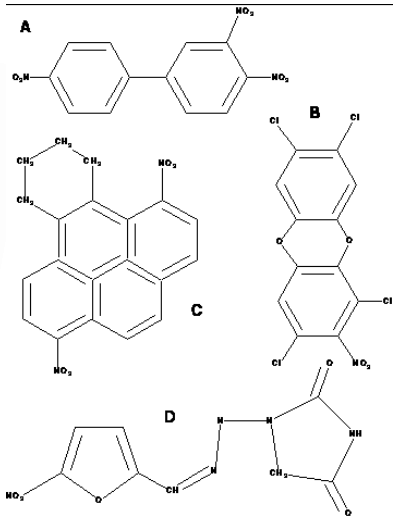
WHO SAID BIG DATA…

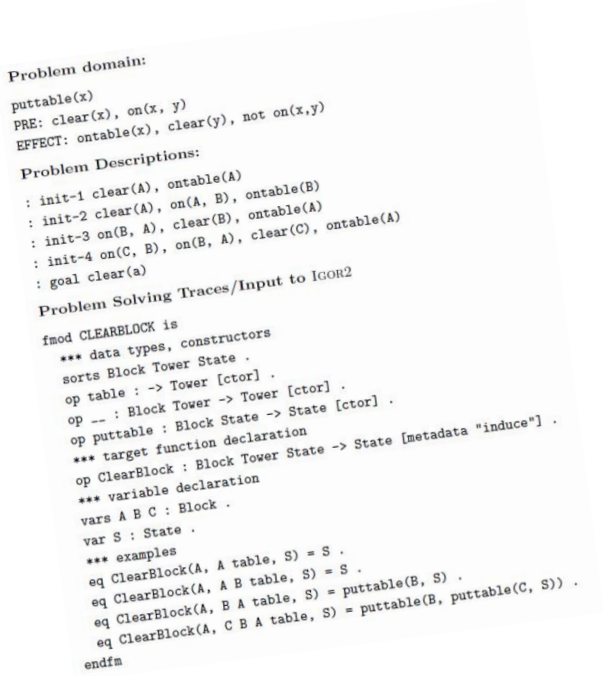


Facetiously adapted from **dilbert.com**

DEEP DATA…







Problem domain:

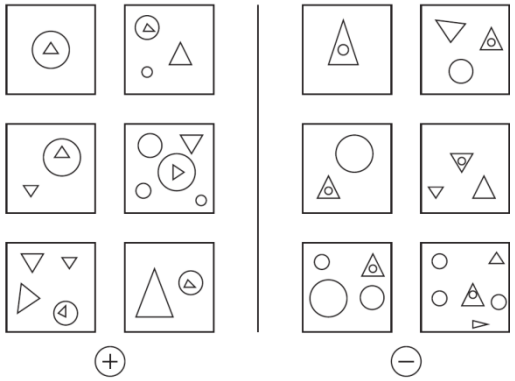
```
puttable(x)
PRE: clear(x), on(x, y)
EFFECT: ontable(x), clear(y), not on(x,y)
```

Problem Descriptions:

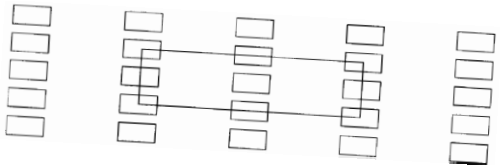
```
: init-1 clear(A), ontable(A)
: init-2 clear(A), on(A, B), ontable(B)
: init-3 on(B, A), clear(B), ontable(A)
: init-4 on(C, B), on(B, A), clear(C), ontable(A)
: goal clear(a)
```

Problem Solving Traces/Input to IGOR2

```
fmod CLEARBLOCK is
  *** data types, constructors
  sorts Block Tower State .
  op table : -> Tower [ctor] .
  op __ : Block Tower -> Tower [ctor] .
  op puttable : Block State -> State [ctor] .
  *** target function declaration
  op ClearBlock : Block Tower State -> State [metadata "induce"] .
  *** variable declaration
  vars A B C : Block .
  var S : State .
  *** examples
  eq ClearBlock(A, A table, S) = S .
  eq ClearBlock(A, A B table, S) = S .
  eq ClearBlock(A, B A table, S) = puttable(B, S) .
  eq ClearBlock(A, C B A table, S) = puttable(B, puttable(C, S)) .
endfm
```



qsort([3,4,1,8,5,6])=[1,3,4,5,6,8]



- Images from {Michie-etal1994trains} {Srinivasan-etal1994mutagenesis} {Schmid-etal2008analytical} (Olson1995incremental) (De raedt2011-encyclopedia)

DEEP KNOWLEDGE!

$$qsort([],[]) \leftarrow$$
$$qsort([X|T],S) \leftarrow part(X,T,L1,L2), qsort(L1,S1), qsort(L2,S2), app(S1,[X|S2],S)$$

prod(s(X0), X1) = sum(prod(X0, X1), X1)
prod(0, X0) = 0

delOdds(L,R) ← L=[], R=[]
delOdds(L,R) ← L=[HL|TL], odd(HL), delOdds(TL,TR), R=TR
delOdds(L,R) ← L=[HL|TL], ¬odd(HL), delOdds(TL,TR), R=[HL|TR]

$$\boldsymbol{B}_m \forall x.((group\_likes\ x) = \text{if } ((\wedge_3\ \blacklozenge \boldsymbol{B}_a likes\ \boldsymbol{B}_b likes\ \boldsymbol{B}_c likes)\ x) \text{ then } \top \text{ else if } ((\wedge_3\ \boldsymbol{B}_c likes\ (\vee_2\ \boldsymbol{B}_a likes\ \boldsymbol{B}_b likes)\ top)\ x) \text{ then } \top \text{ else } \bot).$$

1.0: likes(X,Y):- friendof(X,Y).
0.8: likes(X,Y):- friendof(X,Z), likes(Z,Y).
0.5: friendof(john,mary).
0.5: friendof(mary,pedro).
0.5: friendof(mary,tom).
0.5: friendof(pedro,tom).

```
% Bush syntactic model:
12 :: imperative_sentence(verb_phrase(verb,noun_phrase(noun)))
13 :: affirmative_sentence(noun_phrase(noun),verb_phrase(verb))
 3 :: imperative_sentence(verb_phrase(verb,noun_phrase(noun),
      prop_pred(prep,noun_phrase(det,noun))))
 3 :: imperative_sentence(verb_phrase(verb,noun_phrase(noun),
      prop_pred(prep,noun_phrase(noun))))
 4 :: affirmative_sentence(connector,
      affirmative_sentence(noun_phrase(noun),
      verb_phrase(verb(aux,verb,verb),noun_phrase(noun))))
 4 :: affirmative_sentence(noun_phrase(noun),
      verb_phrase(verb(aux,aux,verb)))
 4 :: imperative_sentence(verb_phrase(verb))
 7 :: affirmative_sentence(noun_phrase(noun),
      verb_phrase(verb,noun_phrase(det,noun)))
...
```

- Examples from {Muggleton-Deraedt1994ilp} {Ferri-etal2001incremental} {Flener-Yilmaz1999inductive} {Castillo2012stochastic} {Deraedt-etal2007problog} {Lloyd-Ng2007modal}

WHAT'S DISTINCTIVE ABOUT INDUCTIVE PROGRAMMING?

- Many characteristics have been mentioned
  - Examples include relations between objects.
  - Features are non-scalar.
  - Patterns are constructive, rather than flat.
  - Use of variables or constructor terms (or both).
  - Use of recursion.
  - Models can be comprehensible.

but not all of them are found in every particular IP approach.

WHAT'S DISTINCTIVE ABOUT INDUCTIVE PROGRAMMING?

- Many different declarative languages have been used:
  - Logical, functional, functional logic, hybrid…
  - Unconditional – Conditional – Constraints…
  - Propositional – First-order – Higher-order…
  - Probabilistic – Stochastic – Bayesian….
  - Modal, Action, ..

One possible inclusive characterisation:
Inductive inference using ***declarative*** *languages that are (nearly)* ***Turing-complete***

WHAT MAKES AN INDUCTIVE PROBLEM HARD?

- Problem classes or problem instances?

  I refer to problem instances

- Elements:
  - Data *D*
  - Target model(s) or hypothesis *h*.
  - Hypothesis space *H*
  - Background Knowledge *B (possibly a component of H)*.

*Are they **big** and **deep**?*

  - Not a meaningful question in isolation (*D* can be large and complex but *h* can still be trivial).

WHAT MAKES AN INDUCTIVE PROBLEM HARD?

- It is insightful to think about *B* and *H* together (bias).
  - B has a dual effect.
    - If it does not contain key auxiliary concepts, the problem becomes very difficult as the concepts need to be **invented**, *but*
    - If it contains too many auxiliary concepts, the problem is now how the appropriate auxiliary concepts are **chosen**.

$$O((|h| \cdot |B|)^{|h|})$$

    - No way unless *h* is *syntactically* very small.

WHAT IS COGNITIVELY DIFFICULT FOR HUMANS?

- Cognitive science has identified some issues:
  - Hypothesis size
    - Humans can only process small hypotheses at a time, including a small set of concepts ("The magical number seven, plus or minus two" (Miller 1956)).
      - This number varies with people.
  - Data size.
    - Humans are not good at big data.
      - Except for perception mechanisms: e.g., vision, speech, music, etc.).
  - Knowledge size.
    - Humans are good at using their knowledge appropriately
      - Difficulty depends on how unrelated or non-contextual the solution is w.r.t. previous knowledge.

WHAT IS COGNITIVELY DIFFICULT FOR HUMANS?

- A different view of ***scalability*** for inductive inference.

Scalability on the size of knowledge

- In humans, fluid vs. crystallised intelligence are distinguished.
  - Fluid intelligence is affected by scalability on *D*, *H* and *h*.
  - Crystallised intelligence is *also* affected by scalability on *B*.

BIG DEEP KNOWLEDGE

- The construction and contextual application of (large repositories of) background knowledge or function libraries for inductive problems is a key issue:
  - Incrementality (data {Kietz-Wrobel1992controlling}{Ferri-etal2001incremental} and knowledge {Olsson1995incremental} {Solomonoff2002incremental} {Schmidhuber2004optimal}, {Henderson2010incremental}), repeat learning {Khan-etal-1998repeat}.
  - Learning to learn, meta-learning, incremental self-improvement {Schmidhuber-etal1997incremental}
  - Function and predicate invention {Muggleton-Buntine1992invention} {Olsson1999invent} {Henderson-Muggleton2012invention}
  - Constructive induction {Muggleton1987Duce} and constructive reinforcement learning {HernandezOrallo2000constructive}.
  - Meta-knowledge {Cabral-etal2005metaknow}{Mccreath-Sharma1995metaknow}, declarative bias {Bridewell-Todorovski2008declarativebias}, transfer learning, relational reinforcement learning , beliefs and modality {Lloyd-Ng2007modal}.

(continues)

BIG DEEP KNOWLEDGE

(continues)

- (Interactive) Theory Revision {RichardsMooney1991revision}{Deraedt1992theoryrevision}, Theory completion {Muggleton2000completion}.
- Integration with abduction {Flach-Kakas2000abduction} and deduction.
- Representation, knowledge level change and learning {Dietterich1986knowledgelevel},
- Knowledge dependency and redundancy, what to keep explicitly and implicitly (inductive and deductive gains) {HernandezOrallo2000InferenceGain},
- Expert systems / knowledge-based systems
- …

- Oldies (but goldies) in Inductive Logic Programming (ILP), Inductive Programming (IP), program synthesis, AI, cognitive science and other areas.
  - Many also have recently had a new revival.

WHY IS INDUCTIVE PROGRAMMING GOOD FOR THIS?

- If we use the human brain as a reference, especially in cognitive science…

Shouldn't neural networks be the preferred paradigm?

  - This is precisely the bet for "deep learning", where neural networks and other kinds of connectionist systems *self-organise*.
- How can we supervise and understand this knowledge?
  - Natural language: wait until these systems develop natural language to interview them?
  - Artificial language: choose expressive, comprehensible languages → Inductive programming.

WHY IS INDUCTIVE PROGRAMMING GOOD FOR THIS?

- Inductive programming as a **means** for developing cognitive systems and intelligent agents:
  - We can understand many pieces (or all) of *H*, *h*, *D* and *B*.
  - This can hold in the short, mid and long terms for *B*.
  - We can provide start-up knowledge B_0.
  - We can revise and fix their knowledge.
  - We have tools to combine different sources of knowledge.
  - Agents can exchange knowledge.

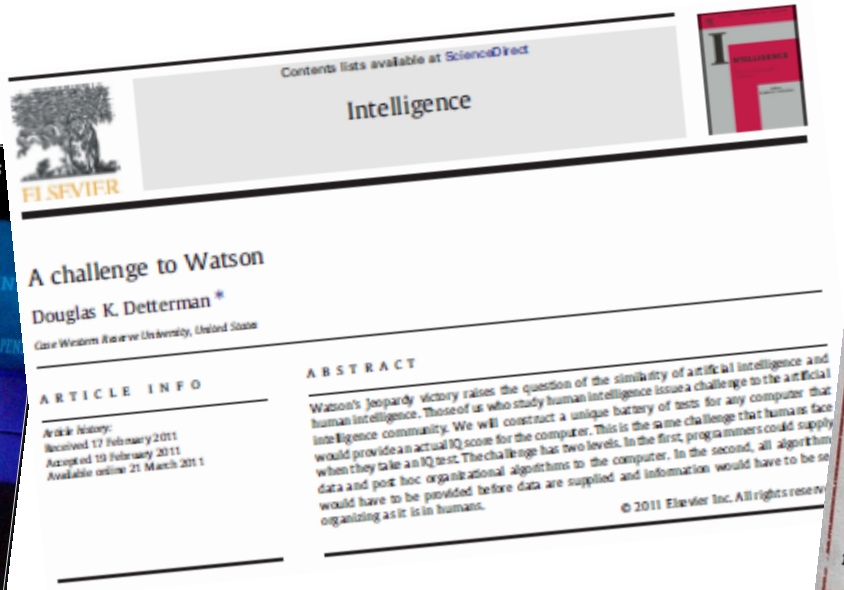
- Many applications require this understanding (scientific discovery, multi-agent systems, software engineering, engineering modelling…)

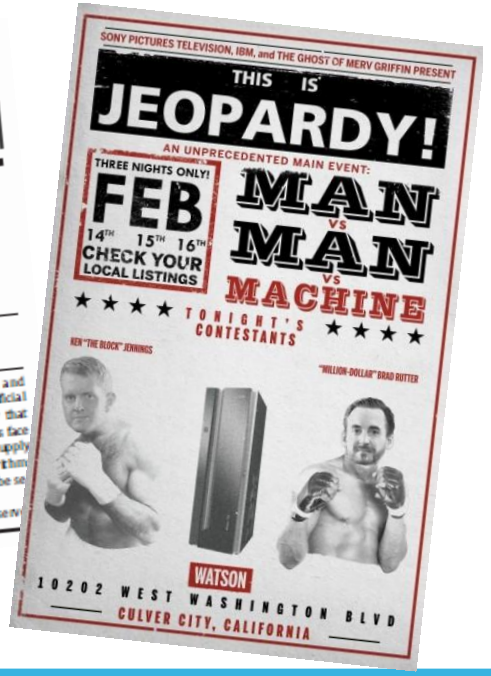
WHY IS INDUCTIVE PROGRAMMING GOOD FOR THIS?

- Inductive programming as an experimental tool for cognitive science:
  - Reasons (in contrast to other approaches):
    - IP hypotheses are usually related to the solutions that humans would find for the same problem.
    - We can understand the solutions reached by the system.
    - We can see explicitly what the system has been given (*B*).
  - Applications:
    - Explore problem difficulty and significance.
    - Explore different solutions.
    - Compare to human solutions, performance and variance.
  - Several examples (common cognitive tasks {Schmid-etal2008analytical}, cognitive "Tutors" {Matsuda-etal2006cognitivetutors}, number or symbol series tasks {Burghardt2005generalization} {Siebers-Schmid2012numbers}.

A COGNITIVE STUDY USING IP: IQ TEST PROBLEMS

- Let's use IQ tests for the evaluation of AI systems!
  - This has been suggested several times in the past.
  - Detterman, editor of the *Intelligence Journal*, made this suggestion serious and explicit: "A challenge to Watson (2011)" {Detterman2011}
    - As a response to specific domain tests and landmarks (such as Watson).





A COGNITIVE STUDY USING IP: IQ TEST PROBLEMS

- Hold on!
  - {Sanghi-Dowe2003} implemented a small program (in Perl) which could score relatively well on many IQ tests.
    - A 3rd year student project
    - Less than 1000 lines of code

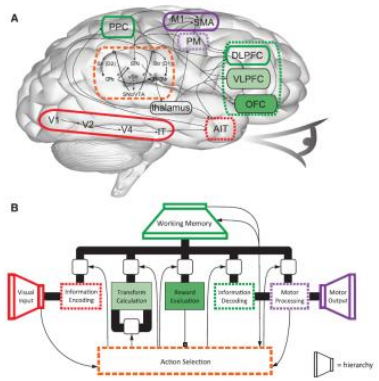
Test	I.Q. Score	Human Average
A.C.E. I.Q. Test	108	100
Eysenck Test 1	107.5	90-110
Eysenck Test 2	107.5	90-110
Eysenck Test 3	101	90-110
Eysenck Test 4	103.25	90-110
Eysenck Test 5	107.5	90-110
Eysenck Test 6	95	90-110
Eysenck Test 7	112.5	90-110
Eysenck Test 8	110	90-110
I.Q. Test Labs	59	80-120
Testedich.de:I.Q. Test	84	100
I.Q. Test from Norway	60	100
Average	96.27	92-108

This made the point unequivocally:
*this program is **not intelligent***

  - 2012 rejoinder to Detterman: "IQ tests are not for machines" {Dowe-HernandezOrallo2012}.

A COGNITIVE STUDY USING IP: IQ TEST PROBLEMS

- Many non-IP approaches attempt exercises in IQ tests.
  - Just an example {Eliasmith-etal2012}



Science AAAS

A Large-Scale Model of the Functioning Brain
Chris Eliasmith *et al.*
Science **338**, 1202 (2012);
DOI: 10.1126/science.1225266

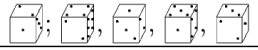
  - Uses exercises like those in IQ tests to evaluate the model brain.
    - *But the problems (without the letter recognition phase) do not look very challenging for an IP system*

Table S2: Example input/output mappings for each task. Randomly selected handwritten digits not used during training are used in the first two tasks. If Spaun does not know the proper response, it will write a dash (see, e.g., serial working memory).

Task Name	Input Examples	Typical Output
Image recognition	A1 [3 ?	3
Copy drawing	A0 [4 ?	4
Reinforcement learning	A2 ? 1 ? 0 ? 0...	0 0 1
Serial working memory	A3 [3 2 4 3 5] ? A3 [4 2 5 3 4 2] ?	3 2 4 3 5 4 2 5 – 4 2
Counting	A4 [3] [2] ? A4 [0] [7] ?	5 7
Question answering	A5 [4 3 8 6] [K] [4] ? A5 [4 3 8 6] [P] [4] ?	1 6
Rapid variable creation	A6 [3312] [12][3392] [92] [3362] [62] [3342] ? A6 [342] [2][345] [5] [347] [7] [340] ?	4 2 0
Fluid reasoning	A7 [5] [55] [555] [2] [22] [222] [1] [11] ? A7 [4] [3] [2] [8] [7] [6] [4] [3] ?	1 1 1 2

A COGNITIVE STUDY USING IP: IQ TEST PROBLEMS

- IP approaches to solve IQ tests are much more explicit.
  - These are the solutions found by {Burghardt2005generalization}:

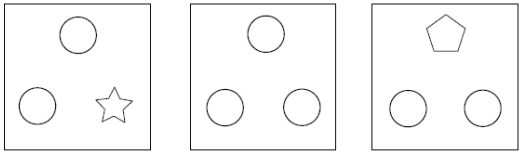
Theory	Series	Law	No	Time
$+,*$	$0;1,4,9$	$v_p * v_p$	1.	2797
$+,*$	$0;2,4,6$	$s(s(v_1))$	1.	3429
$+,*$	$0;2,4,6$	$v_p + v_p$	3.	3429
$+,*$	$1,1;2,3,5$	$v_1 + v_2$	1.	857
$+,*,if,ev$	$0,1;2,1,4,1$	$if(ev(v_p), v_p, 1)$	13.	13913
$+,*,if,ev$	$0,0,1;1,0,0,1,1$	$ev(v_2)$	1.	61571
$+,*,if,ev$	$0,0;1,0,0,1$	$ev(v_1 + v_2)$	1.	8573
$+,*,if,ev$	$0;1,3,7$	$s(v_1 + v_1)$	1.	3714
$+,*,if,ev$	$1,2,2,3,3,3,4;4,4,4$	—		8143
$cube,if,ev$		$rg(if(ev(v_p), v_1, \boxed{\cdot\cdot\cdot}))$	1.	14713
$cube,if,ev$		$cl(if(ev(v_p), up(v_1), dn(v_1)))$	1.	604234

We can learn much more from here about whether this is significant progress, and about the types of solutions and problem difficulty, than from non-IP approaches.

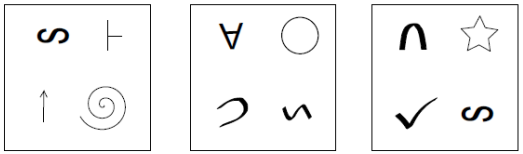
A COGNITIVE STUDY USING IP: IQ TEST PROBLEMS

- We also had a go with gErl {MartinezPlumed-etal2013gErl}:
  - Odd-one-out problems

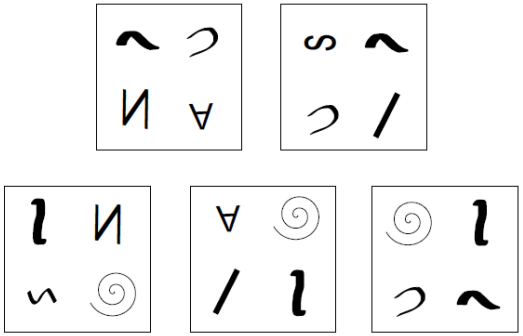
Item # 3 - Easy



Item # 23 - Medium



Item # 34 - Difficult



Problem presentation {Ruiz2011oddoneout}:
1 exercise → 1 example

Item	Set1	Set 2	Set 3	Set 4	Set 5
1	AAA	AAA	ABB		
2	AAA	AAA	BCD		
3	AAA	AAB	AAC		
4	AAA	ABB	ABB		
5	AAA	BBB	ABC		
6	AAA	BCD	EFG		
7	AAA	BBC	CCB		
8	AAB	AAB	ABC		
9	AAB	AAC	DEF		
10	AAB	ABB	EFG		
11	ABC	ABC	ABD		
12	AAB	ABB	ABC		
13	ABC	ADE	FGH		
14	AAAA	BBDE	CCFG		
15	AAAA	AABB	AACC		
16	AAAD	BBEF	CCGH		
17	AABB	AABB	ABCD		
18	AABC	AACD	ABCD		
19	AAAB	BBBD	CCCE		
20	ABCD	ABCD	ABCE		
21	ABCD	ABCE	ABFG		
22	AABC	BBAC	CCAF		
23	ABCD	AEFG	HIJK		
24	AAAA	AAAA	BBBB	BBBB	CCCC
25	AAAD	AAAE	BBBF	BBBG	CCCH
26	AABB	BBCC	AADD	DDCC	EEFF
27	AAEF	BBGH	CCIJ	DDKL	ABCD
28	AAAE	BBBF	CCGH	DDIJ	ABCD
29	AAAE	BBBF	CCGH	DDIJ	AABB
30	AAAB	BBBF	CCGH	DDIJ	AABB
31	AABB	BBCC	AADD	DDCC	AAEE
32	ABCD	BCDE	CDEF	DEFG	FGAB
33	ACDE	AFGH	BIJK	BLMN	OPQR
34	ABEF	ABGH	CDEG	CDFH	ABCD
35	ACDE	AFGH	BIJK	BLMN	ABOP

Table 4: 35 examples of R-ASCM abstract representation (solutions in grey) from Ruiz [54].

A COGNITIVE STUDY USING IP: IQ TEST PROBLEMS

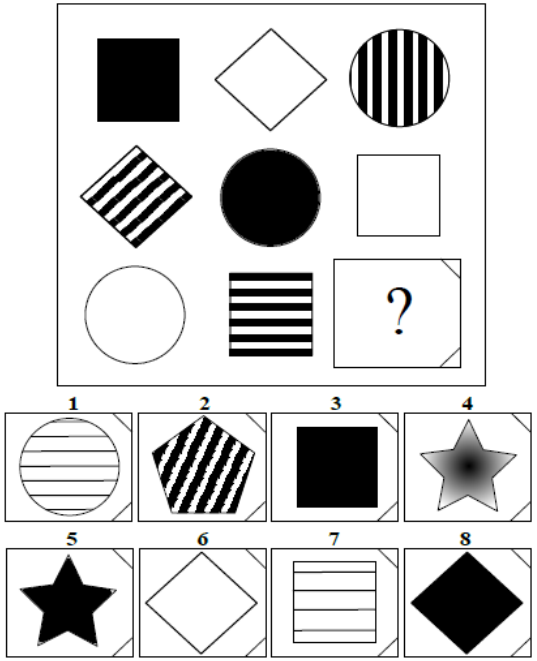
- We also had a go with gErl {MartinezPlumed-etal2013gErl}:
  - Odd-one-out problems. gErl results

Approach	Example	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	TOTAL
gErl	$ooo(V_{lists}) \rightarrow distinct(map(hamming, V_{lists}))$	✓	✓	✗	✓	✓	✗	✓	✓	✓	✓	✓	✗	✓	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	**28**
	$ooo(V_{lists}) \rightarrow distinct(map(diffObj, V_{lists}))$	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✗	✓	✗	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	✗	✗	✗	✗	✗	✗	✗	17
Ruiz	$1^{st} step$	✓	✓	✗	✓	✓	✗	✓	✓	✓	✓	✓	✗	✓	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	**28**
	$2^{nd} step$	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	5

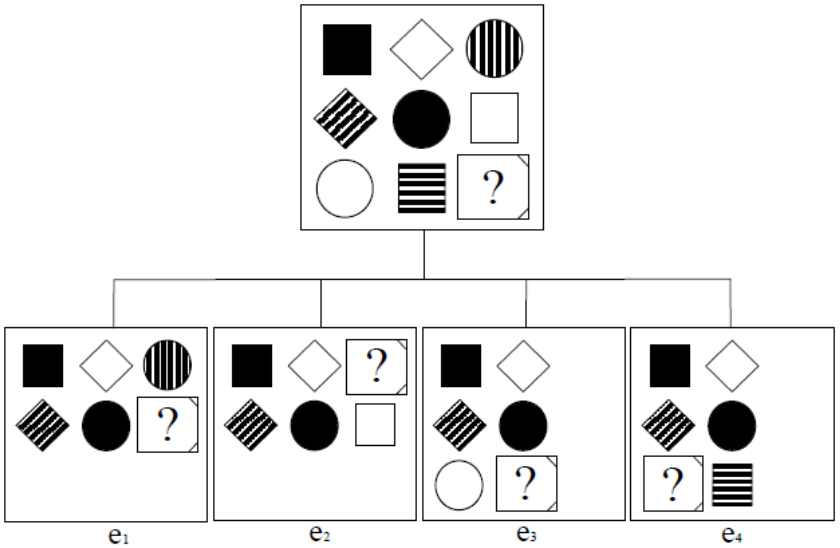
- Results are better than other specialised systems, but how does gErl do it?
  - It uses the same R-ASCM representation.
  - It uses two *ad-hoc* functions from background knowledge: *hamming* and *diffObj*.
    - *It is not very difficult in this way*.
    - It doesn't match the way humans solve these problems

A COGNITIVE STUDY USING IP: IQ TEST PROBLEMS

- We also had a go with gErl {MartinezPlumed-etal2013gErl}:
  - Raven's Progressive Matrices (RPM)



Problem presentation
1 exercise → several examples



A COGNITIVE STUDY USING IP: IQ TEST PROBLEMS

- We also had a go with gErl {MartinezPlumed-etal2013gErl}:
  - Raven's Progressive Matrices (RPM). gErl results

Id	Solution	Steps	E^+	o
25	$ravenPM(V) \rightarrow [\langle identity(shape), none, none, none, none\rangle]$	37	3	2
26	$ravenPM(V) \rightarrow [\langle identity(shape), progressive(size), none, none, none\rangle]$	99	4	3
27	$ravenPM(V) \rightarrow [\langle identity(shape), progressive(size), none, none, none\rangle]$	99	4	3
28	$ravenPM(V) \rightarrow [\langle identity(shape), progressive(size), none, none, none\rangle]$	111	4	3
29	$ravenPM(V) \rightarrow [\langle identity(shape), none, progressive(quantity), progressive(position), none\rangle]$	131	4	4
30	$ravenPM(V) \rightarrow [\langle identity(shape), progressive(size), none, none, none\rangle]$	88	4	3
31	$ravenPM(V) \rightarrow [\langle identity(shape), none, none, progressive(position), none\rangle]$	81	4	3
32	$ravenPM(V) \rightarrow [\langle identity(shape), none, progressive(quantity), none, none\rangle]$	79	4	3
33	$ravenPM(V) \rightarrow [\langle identity(shape), none, none, progressive(position), none\rangle]$	91	4	3
34	$ravenPM(V) \rightarrow [\langle identity(shape), none, none, progressive(position), none\rangle]$	91	4	3
35	$ravenPM(V) \rightarrow [\langle identity(shape), none, progressive(quantity), none, none\rangle]$	81	4	3
36	$ravenPM(V) \rightarrow [\langle identity(shape), none, none, progressive(position), none\rangle]$	83	4	3
37	$ravenPM(V) \rightarrow [\langle identity(shape), none, none, none, identity(type)\rangle]$	75	4	3
38	$ravenPM(V) \rightarrow [\langle distrib3val(shape), none, none, none, none\rangle]$	69	4	2
39	$ravenPM(V) \rightarrow [\langle distrib3val(shape), none, none, none, none\rangle]$	71	4	2
40	$ravenPM(V) \rightarrow [\langle identity(shape), none, none, none, distrib3val(type)\rangle]$	94	6	3
41	$ravenPM(V) \rightarrow [\langle identity(shape), none, none, none, distrib3val(type)\rangle]$	96	6	3
42	$ravenPM(V) \rightarrow [\langle identity(shape), none, none, none, distrib3val(type)\rangle]$	93	6	3
43	$ravenPM(V) \rightarrow [\langle distrib3val(shape), none, none, none, distrib3val(type)\rangle]$	106	6	3
44	$ravenPM(V) \rightarrow [\langle distrib3val(shape), none, none, none, distrib3val(type)\rangle]$	91	6	3
45	$ravenPM(V) \rightarrow [\langle distrib3val(shape), none, none, none, distrib3val(type)\rangle]$	104	6	3
46	$ravenPM(V) \rightarrow [\langle identity(shape), none, none, none, distrib3val(type)\rangle]$	93	6	3
47	$ravenPM(V) \rightarrow [\langle identity(shape), none, distrib3val(quantity), none, distrib3val(type)\rangle]$	146	6	4
48	$ravenPM(V) \rightarrow [\langle distrib3val(shape), none, none, none, distrib3val(type)\rangle]$	106	6	3
49	$ravenPM(V) \rightarrow [\langle addition(shape), none, none, none, none\rangle]$	61	4	2
50	$ravenPM(V) \rightarrow [\langle addition(shape), none, none, none, none\rangle]$	55	4	2
51	$ravenPM(V) \rightarrow [\langle addition(shape), none, none, none, identity(type)\rangle]$	99	6	3
52	$ravenPM(V) \rightarrow [\langle distrib2val(shape), none, none, none, none\rangle]$	63	4	2
53	$ravenPM(V) \rightarrow [\langle distrib2val(shape), none, none, none, none\rangle]$	60	4	2
54	$ravenPM(V) \rightarrow [\langle distrib2val(shape), none, none, none, none\rangle]$	61	4	2
55	$ravenPM(V) \rightarrow [\langle distrib2val(shape), none, none, none, none\rangle]$	77	4	2
56	$ravenPM(V) \rightarrow [\langle distrib2val(shape), none, none, none, none\rangle]$	99	4	3
57	$ravenPM(V) \rightarrow [\langle distrib2val(shape), none, none, none, distrib3val(type)\rangle]$	10	4	3
58	$ravenPM(V) \rightarrow [\langle distrib2val(shape), none, none, none, none\rangle]$	60	4	2
59	$ravenPM(V) \rightarrow [\langle distrib2val(shape), none, none, none, none\rangle]$	65	4	2

Table 7: Solutions returned and steps needed by gErl to learn the Raven's Standard Progressive Matrix [57].

- How does gErl do it?
  - It uses functions from the background knowledge: *identity, progressive, distrib3val, distrib2val, addition, ...*
  - *More challenging*
  - *More similar to human solutions.*

A COGNITIVE STUDY USING IP: IQ TEST PROBLEMS

- We also had a go with gErl {MartinezPlumed-etal2013gErl}:
  - Thurstone letter series completion problems

1. cdcdcdcd_
2. aaabbbcccdd_
3. atbataatbat_
4. abmcdmefmghm_
5. defgefghfghi_
6. qxapxbqxa_
7. aducuaeuabuafua_
8. mabmbcmcdm_
9. urtustuttu_
10. abyabxabwab_
11. rscdstdetuef_
12. npaoqapraqsa_
13. wxaxybyzczadab_
14. jkqrklrslmst_
15. pononmnmlmlk_

Problem presentation
1 exercise → several examples

$$e_1 : thurstone(\text{“}cdcdcdcd\text{”}) \rightarrow \text{“}c\text{”} \begin{cases} e_{1.1} : thurstone(\text{“}cd\text{”}) \rightarrow \text{“}c\text{”} \\ e_{1.2} : thurstone(\text{“}cdc\text{”}) \rightarrow \text{“}d\text{”} \\ e_{1.3} : thurstone(\text{“}cdcd\text{”}) \rightarrow \text{“}c\text{”} \\ e_{1.4} : thurstone(\text{“}cdcdc\text{”}) \rightarrow \text{“}d\text{”} \\ e_{1.5} : thurstone(\text{“}cdcdcd\text{”}) \rightarrow \text{“}c\text{”} \\ e_{1.6} : thurstone(\text{“}cdcdcdc\text{”}) \rightarrow \text{“}d\text{”} \\ e_{1.7} : thurstone(\text{“}cdcdcdcd\text{”}) \rightarrow \text{“}c\text{”} \end{cases}$$

A COGNITIVE STUDY USING IP: IQ TEST PROBLEMS

- We also had a go with gErl {MartinezPlumed-etal2013gErl}:
  - Thurstone letter series completion problems. gErl results

Id	Problem	Solution	Steps	E^+	o
1.	cdcdcdcd_	$thurstone(V) \rightarrow last(init(V))$	42	5	3
2.	aaabbbcccdd_	$thurstone(V) \rightarrow next(init(init(V)))$	91	9	4
3.	atbataatbat_	$thurstone(V) \rightarrow last(init(init(init(init(init(V))))))$	131	7	7
4.	abmcdmefmghm_	$thurstone(V)$ when $length(V)$ mod $3 = 0 \rightarrow last(init(init(V)))$ $thurstone(V)$ when $length(V)$ mod $3 \neq 0 \rightarrow next(init(V))$	174	8	5 4
5.	defgefghfghi_	$thurstone(V) \rightarrow next(init(init(init(V))))$	105	8	5
6.	qxapxbqxa_	$thurstone(V) \rightarrow last(init(init(init(init(init(V))))))$	129	7	7
7.	aducuaeuabuafua_				11
8.	mabmbcmcdm_	$thurstone(V)$ when $length(V)$ mod $3 = 0 \rightarrow last(init(init(V)))$ $thurstone(V)$ when $length(V)$ mod $3 \neq 0 \rightarrow next(init(init(V)))$	165	8	5 5
9.	turtustuttu_	$thurstone(V)$ when $length(V)$ mod $3 = 0 \rightarrow next(init(init(V)))$ $thurstone(V)$ when $length(V)$ mod $3 \neq 0 \rightarrow last(init(init(V)))$	192	9	5 5
10.	abyabxabwab_	$thurstone(V)$ when $length(V)$ mod $3 = 0 \rightarrow previous(init(init(V)))$ $thurstone(V)$ when $length(V)$ mod $3 \neq 0 \rightarrow last(init(init(V)))$	182	9	5 5
11.	rscdstdetuef_	$thurstone(V) \rightarrow next(init(init(init(init(V)))))$	154	9	6
12.	npaoqapraqsa_	$thurstone(V)$ when $length(V)$ mod $3 = 0 \rightarrow last(init(init(V)))$ $thurstone(V)$ when $length(V)$ mod $3 \neq 0 \rightarrow next(init(init(V)))$	141	8	5 5
13.	wxaxybyzczadab_	$thurstone(V) \rightarrow next(init(init(V)))$	99	9	4
14.	jkqrklrslmst_	$thurstone(V) \rightarrow next(init(init(init(V))))$	103	9	5
15.	pononmnmlmlk_	$thurstone(V) \rightarrow previous(init(init(V)))$	112	9	5

- How does gErl do it?
  - It uses functions from the background knowledge: *last, next, init, length, mod, previous …*
  - *Even more challenging and very close to the patterns humans find.*

A COGNITIVE STUDY USING IP: IQ TEST PROBLEMS

- What crucial things are there?
  - Background knowledge.
  - Data representation (structures)
  - Data presentation (examples per problem instance)

- What do we learn from these problems?
  - Does difficulty correlate with humans and other systems?
  - Do the hypotheses match those identified by humans?

DISCUSSION

- General systems are difficult to compare when background knowledge is used.
  - Many different kinds of problems have been used in ILP, IP, AI, ML and program synthesis.
    - We only have informal assessments of their difficulty.
  - Some systems are able to solve them from scratch, others with background knowledge.
  - Some systems are able to solve just one type of problems, others are more general.

We need **libraries** of problems (featuring deep data and deep background knowledge), to really know what the challenges are, when there is real progress, etc.

DISCUSSION

- Some suggestions:
  - Real problems: e.g. program synthesis, ILP problems, AI problems and IQ tests provide some starting collection.
    - Systematic assessment of their properties à la {Macia-etal2013datasets} {Macia-Bernado2013uciplus}.
    - Cognitive difficulty assessment (relative to humans).
  - Artificial problems:
    - Theoretical account of difficulty (e.g., using AIT, {Hernandez-Orallo2000a,2000b},

$$
\begin{array}{lll}
k = 9 & : \text{a, d, g, j, } \ldots & \text{Answer : m} \\
k = 12 & : \text{a, a, z, c, y, e, x, } \ldots & \text{Answer : g} \\
k = 14 & : \text{c, a, b, d, b, c, c, e, c, d, } \ldots & \text{Answer : d}
\end{array}
$$

    - Enriching existing approaches for flat (attribute-value) artificial dataset generation {Rios-etal2008artificial} {Marzukhi-etal2013artificial}.

DISCUSSION

- It gets more interesting (and difficult) when induction is made from small deep data and especially with **big deep knowledge**.
  - More needs to be done on knowledge acquisition and reuse.

Let's counter the research bias towards ***disposable*** learning.

THANK YOU!

- Special thanks to:
  - M.J. Ramírez-Quintana and C. Ferri for joint work on inductive logic programming, the FLIP system and its incremental version.
  - F. Martínez-Plumed for his results with gErl over these IQ test problems.
  - V. Estruch, J. Bedoya and H. Castillo for some other joint work in ILP and IP.
  - D. Dowe for joint work on universal psychometrics and meaning/usefulness of IQ tests.
  - The seminar organisers: Sumit, Emanuel and Ute!

COPY&PASTE REFERENCES


```
techreport{Michie-etal1994trains,
 title={To the international computing community: A new East-West challenge},
 author={Michie, Donald and Muggleton, Stephen and Page, David and Srinivasan, Ashwin},
 year={1994},
 institution={Technical report, Oxford University Computing laboratory, Oxford, UK}
}

@inproceedings{Srinivasan-etal1994mutagenesis,
 title={Mutagenesis: ILP experiments in a non-determinate biological domain},
 author={Srinivasan, Ashwin and Muggleton, Stephen and King, Ross D and Sternberg, Micheal JE},
 booktitle={Proceedings of the 4th international workshop on inductive logic programming},
 volume={237},
 pages={217--232},
 year={1994},
 organization={Citeseer}
}

@inproceedings{Schmid-etal2008analytical,
 title={Analytical inductive programming as a cognitive rule acquisition devise},
 author={Schmid, Ute and Hofmann, Martin and Kitzelmann, Emanuel},
 booktitle={Proceedings of the Second Conference on Artificial General Intelligence},
 pages={162--167},
 year={2008}
}

@article{Olsson1995incremental,
 title={Inductive functional programming using incremental program transformation},
 author={Olsson, Roland},
 journal={Artificial intelligence},
 volume={74},
 number={1},
 pages={55--81},
 year={1995},
 publisher={Elsevier}
}

@chapter{Deraedt-2011encyclopedia,
 title={Inductive logic programming},
 author={De Raedt, Luc},
 booktitle={Sammut, Claude and Webb, Geoffrey I "Encyclopedia of machine learning"},
 pages={529--534},
 year={2011},
 publisher={Springer-Verlag}
}

@article{Muggleton-Deraedt1994ilp,
 title={Inductive logic programming: Theory and methods},
 author={Muggleton, Stephen and De Raedt, Luc},
 journal={The Journal of Logic Programming},
 volume={19},
 pages={629--679},
 year={1994},
 publisher={Elsevier}
}

@incollection{Ferri-etal2001incremental,
 title={Incremental learning of functional logic programs},
 author={Ferri-Ram{\'i}rez, Cesar and Hern{\'a}ndez-Orallo, Jos{\'e} and Ram{\'i}rez-Quintana, M Jos{\'e}},
 booktitle={Functional and Logic Programming},
 pages={233--247},
 year={2001},
 publisher={Springer}
}

@article{Flener-Yilmaz1999inductive,
 title={Inductive synthesis of recursive logic programs: Achievements and prospects},
 author={Flener, Pierre and Y{\i}lmaz, Serap},
 journal={The Journal of Logic Programming},
 volume={41},
 number={2},
 pages={141--195},
 year={1999},
 publisher={Elsevier}
}

@article{Castillo2012stochastic,
 title={An MML-based tool for evaluating the complexity of (stochastic) logic theories},
 author={Castillo-Andreu, H{\'e}ctor},
 journal={arXiv preprint arXiv:1210.5974},
 year={2012}
}

@inproceedings{Deraedt-etal2007problog,
 title={ProbLog: A Probabilistic Prolog and Its Application in Link Discovery.},
 author={De Raedt, Luc and Kimmig, Angelika and Toivonen, Hannu},
 booktitle={IJCAI},
 volume={7},
 pages={2462--2467},
 year={2007}
}

@incollection{Lloyd-Ng2007modal,
 title={Learning modal theories},
 author={Lloyd, John W and Ng, Kee Siong},
 booktitle={Inductive Logic Programming},
 pages={320--334},
 year={2007},
 publisher={Springer}
}

@article{Schmid-Kitzelman2011inductive,
 title={Inductive rule learning on the knowledge level},
 author={Schmid, Ute and Kitzelmann, Emanuel},
 journal={Cognitive Systems Research},
 volume={12},
 number={3},
 pages={237--248},
 year={2011},
 publisher={Elsevier}
}

@inproceedings{solomonoff2002incremental,
 title={Progress in incremental machine learning},
 author={Solomonoff, Ray J},
 booktitle={NIPS Workshop on Universal Learning Algorithms and Optimal Search},
 pages={211--256},
 year={2002}
}

@article{Schmidhuber2004optimal,
 title={Optimal ordered problem solver},
 author={Schmidhuber, J{\"u}rgen},
 journal={Machine Learning},
 volume={54},
 number={3},
 pages={211--254},
 year={2004},
 publisher={Springer}
}

@inproceedings{Henderson2010incremental,
 title = {Incremental Learning In Inductive Programming},
 author = {Henderson, R.},
 booktitle = {Approaches and Applications of Inductive Programming},
 publisher = {Springer},
 series = {LNCS},
 pages = {74--92},
 volume = {5812},
 year = {2010},
 url = {http://www.doc.ic.ac.uk/~rjh09/publications/henderson10.pdf}
}
```

COPY&PASTE REFERENCES


```
@incollection{Khan-etal-1998repeat, title={Repeat learning using predicate invention},
 author={Khan, Khalid and Muggleton, Stephen and Parson, Rupert},
 booktitle={Inductive Logic Programming},
 pages={165--174},
 year={1998},
 publisher={Springer}
}

@article{Schmidhuber-etal1997incremental,
 title={Shifting inductive bias with success-story algorithm, adaptive Levin search, and incremental self-improvement},
 author={Schmidhuber, J{\"u}rgen and Zhao, Jieyu and Wiering, Marco},
 journal={Machine Learning},
 volume={28},
 number={1},
 pages={105--130},
 year={1997},
 publisher={Springer}
}

@article{Muggleton-Buntine1992invention,
 title={Machine invention of first-order predicates by inverting resolution},
 author={Muggleton, Stephen and Buntine, Wray},
 journal={Inductive logic programming},
 pages={261--280},
 year={1992},
 publisher={Academic Press, Orlando, Fla}
}

@incollection{Olsson1999invent,
 title={How to invent functions},
 author={Olsson, J Roland},
 booktitle={Genetic Programming},
 pages={232--243},
 year={1999},
 publisher={Springer}
}

@inproceedings{Henderson-Muggleton2012,
 title = {Automatic Invention of Functional Abstractions},
 author = {Henderson, R. J. and Muggleton, S. H.},
 booktitle = {Latest Advances in Inductive Logic Programming},
 publisher = {Imperial College Press},
 year = {2012},
 note = {To appear},
 url = {http://www.doc.ic.ac.uk/~rjh09/publications/henderson12.pdf}
}

@inproceedings{Muggleton1987Duce,
 title={Duce, An Oracle-based Approach to Constructive Induction},
 author={Muggleton, Stephen},
 booktitle={IJCAI},
 pages={287--292},
 year={1987},
 organization={Citeseer}
}

@article{HernandezOrallo2000constructive,
 title={Constructive reinforcement learning},
 author={Hern{\'a}ndez-Orallo, Jos{\'e}},
 journal={International Journal of Intelligent Systems},
 volume={15},
 number={3},
 pages={241--264},
 year={2000},
 publisher={Citeseer}
}

@incollection{Cabral-etal2005metaknow,
 title={Converting semantic meta-knowledge into inductive bias},
 author={Cabral, John and Kahlert, Robert C and Matuszek, Cynthia and Witbrock, Michael and Summers, Brett},
 booktitle={Inductive Logic Programming},
 pages={38--50},
 year={2005},
 publisher={Springer}
}

@inproceedings{Mccreath-Sharma1995metaknow,
 title={Extraction of meta-knowledge to restrict the hypothesis space for ILP systems},
 author={McCreath, Eric and Sharma, Arun},
 booktitle={AI-CONFERENCE},
 pages={75--82},
 year={1995},
 organization={Citeseer}
}

@incollection{Bridewell-Todorovski2008declarativebias,
 title={Learning declarative bias},
 author={Bridewell, Will and Todorovski, Ljup{\v{c}}o},
 booktitle={Inductive Logic Programming},
 pages={63--77},
 year={2008},
 publisher={Springer}
}

@article{Deraedt1992theoryrevision,
 title={Interactive theory revision: an inductive logic programming approach},
 author={De Raedt, Luc},
 journal={status: published},
 year={1992},
 publisher={Academic Press}
}

@incollection{Muggleton2000completion,
 title={Theory completion using inverse entailment},
 author={Muggleton, Stephen H and Bryant, Christopher H},
 booktitle={Inductive Logic Programming},
 pages={130--146},
 year={2000},
 publisher={Springer}
}

@book{Flach-Kakas2000abduction,
 title={Abduction and Induction: Essays on their relation and integration},
 author={Flach, Peter A and Kakas, Antonis C},
 volume={18},
 year={2000},
 publisher={Springer}
}

@article{Dietterich1986knowledgelevel,
 title={Learning at the knowledge level},
 author={Dietterich, Thomas G},
 journal={Machine Learning},
 volume={1},
 number={3},
 pages={287--315},
 year={1986},
 publisher={Springer}
}

@article{HernandezOrallo2000InferenceGain,
 title={Computational measures of information gain and reinforcement in inference processes [1] The complete dissertation in English is available at http://www. dsic. upv. es/\~{} jorallo/tesi/.},
 author={Hernandez-Orallo, Jose},
 journal={AI Communications},
 volume={13},
 number={1},
 pages={49--50},
 year={2000},
 publisher={IOS Press}
}
```

COPY&PASTE REFERENCES

@article{Matsuda-etal2006cognitivetutors,
title={Applying Machine Learning to Cognitive Modeling for Cognitive Tutors},
author={Matsuda, Noboru and Cohen, William W and Sewall, Jonathan and Koedinger, Kenneth R},
year={2006}
}

@incollection{Siebers-Schmid2012numbers,
title={Semi-analytic natural number series induction},
author={Siebers, Michael and Schmid, Ute},
booktitle={KI 2012: Advances in Artificial Intelligence},
pages={249--252},
year={2012},
publisher={Springer}
}

@article{Burghardt2005generalization,
title={E-generalization using grammars},
author={Burghardt, Jochen},
journal={Artificial intelligence},
volume={165},
number={1},
pages={1--35},
year={2005},
publisher={Elsevier}
}

@article{eliasmith2012large,
title={A large-scale model of the functioning brain},
author={Eliasmith, Chris and Stewart, Terrence C and Choo, Xuan and Bekolay, Trevor and DeWolf, Travis and Tang, Charlie and Rasmussen, Daniel},
journal={science},
volume={338},
number={6111},
pages={1202--1205},
year={2012},
publisher={American Association for the Advancement of Science}
}

@inproceedings{Sanghi-Dowe2003,
title={A computer program capable of passing IQ tests},
author={Sanghi, Pritika and Dowe, David L},
booktitle={Proc. 4th ICCS International Conference on Cognitive Science (ICCS'03)},
pages={570--575},
year={2003}
}

@article{Dowe-HernandezOrallo2012,
title={IQ tests are not for machines, yet},
author={Dowe, David L and Hern{\'a}ndez-Orallo, Jos{\'e}},
journal={Intelligence},
volume={40},
number={2},
pages={77--81},
year={2012},
publisher={Elsevier}
}

@article{MartinezPlumed-etal2013gErl,
title={On the definition of a general learning system with user-defined operators},
author={Mart{\'i}nez-Plumed, Fernando and Ferri, C{\`e}sar and Hern{\'a}ndez-Orallo, Jos{\'e} and Ram{\'i}rez-Quintana, Mar{\'i}a-Jos{\'e}},
journal={arXiv preprint arXiv:1311.4235},
year={2013}
}

@article{HernandezOrallo2000b,
title={On the computational measurement of intelligence factors},
author={Hern{\'a}ndez-Orallo, Jos{\'e} and i Computacio, Dcpartamcnt de Sistcmcs Informatics},
journal={NIST SPECIAL PUBLICATION SP},
pages={72--79},
year={2001},
publisher={NATIONAL INSTIUTE OF STANDARDS \& TECHNOLOGY}
}

@article{HernandezOrallo2000a,
title={Beyond the Turing test},
author={Hernandez-Orallo, Jose},
journal={Journal of Logic, Language and Information},
volume={9},
number={4},
pages={447--466},
year={2000},
publisher={Springer}
}

@inproceedings{Rios-etal2008artificial,
title={Artificial Data Sets based on Knowledge Generators: Analysis of Learning Algorithms Efficiency},
author={Rios-Boutin, Joaquin and Orriols-Puig, Albert and Garrell-Guiu, J-M},
booktitle={Hybrid Intelligent Systems, 2008. HIS'08. Eighth International Conference on},
pages={873--878},
year={2008},
organization={IEEE}
}

@inproceedings{Marzukhi-etal2013artificial,
title={Adaptive artificial datasets through learning classifier systems for classification tasks},
author={Marzukhi, Syahaneim and Browne, Will N and Zhang, Mengjie},
booktitle={Proceeding of the fifteenth annual conference companion on Genetic and evolutionary computation conference companion},
pages={1243--1250},
year={2013},
organization={ACM}
}

@article{Macia-etal2013datasets,
title={Learner excellence biased by data set selection: A case for data characterisation and artificial data sets},
author={Maci{\`a}, N{\'u}ria and Bernad{\'o}-Mansilla, Ester and Orriols-Puig, Albert and Kam Ho, Tin},
journal={Pattern Recognition},
year={2012},
publisher={Elsevier}
}

@article{Macia-Bernado2013uciplus,
title={Towards UCI+: A mindful repository design},
author={Maci{\`a}, N{\'u}ria and Bernad{\'o}-Mansilla, Ester},
journal={Information Sciences},
year={2013},
publisher={Elsevier}
}

@inproceedings{Kietz-Wrobel1992controlling,
title={Controlling the complexity of learning in logic through syntactic and task-oriented models},
author={Kietz, J{\"o}rg-Uwe and Wrobel, Stefan},
booktitle={Inductive logic programming},
year={1992}
}

@inproceedings{RichardsMooney1991revision,
title={First-order theory revision},
author={Richards, Bradley L and Mooney, Raymond J},
booktitle={ML}, pages={447--451},
year={1991}
}

@article{detterman2011challenge, title={A challenge to Watson}, author={Detterman, Douglas K}, journal={Intelligence}, volume={39}, number={2}, pages={77--78}, year={2011}, publisher={Elsevier} }

@article{Ruiz2011oddoneout, title={Building and solving odd-one-out classification problems: A systematic approach}, author={Ruiz, Philippe E}, journal={Intelligence}, volume={39}, number={5}, pages={342--350}, year={2011}, publisher={Elsevier} }