ASCoL: A Tool For Improving Automatic Planning Domain Model Acquisition

Rabia Jilani
University of Huddersfield, UK

Authors: Rabia Jilani, Andrew Crampton, Diane Kitchin and Mauro Vallati

Introduction

- Intelligent agents performing in the real-world use models of the world to plot their actions

NASA Mars Rover

Machine tool calibration

Robotics

- Hand-coded... time consuming, error prone
- Autonomous systems –LOCM, ARMS, Opmaker,...
- Challenges: Input to output ratio

Limitations of Learning Systems

- LOCM (ICAPS 2009 by Cresswell, McCluskey and West) :
  - Learns domain models from logged sequences of actions (plans) only
  - No need for information like predicate specification or state information
  - Declare Static facts manually

sequence_task(15, [board(c12, l2), sail(l2, l1), debark(c12, l1), board(c1, l1), sail(l1, l3), debark(c1, l3), board(c11, l3), sail(l3, l4), debark(c11, l4), board(c3, l4), sail(l4, l1), debark(c3, l1), board(c2, l1), sail(l1, l3), debark(c2, l3), board(c24, l3), sail(l3, l4), debark(c24, l4), board(c9, l4), sail(l4, l2), debark(c9, l2), board(c15, l2), sail(l2, l0), debark(c15, l0), board(c4, l0), sail(l0, l4), debark(c4, l4), board(c10, l4), sail(l4, l0), debark(c10, l0), board(c6, l0), sail(l0, l4), debark(c6, l4), board(c14, l4), sail(l4, l1), debark(c14, l1), board(c8, l1), sail(l1, l3), debark(c8, l3), board(c25, l3), sail(l3, l0), debark(c25, l0), board(c13, l0), sail(l0, l2), debark(c13, l2), board(c17, l2), sail(l2, l0), debark(c17, l0), board(c21, l0), sail(l0, l1), debark(c21, l1), board(c16, l1), sail(l1, l4), debark(c16, l4), board(c18, l4), sail(l4, l0), debark(c18, l0), board(c26, l0), sail(l0, l3), debark(c26, l3)], _, _). ....

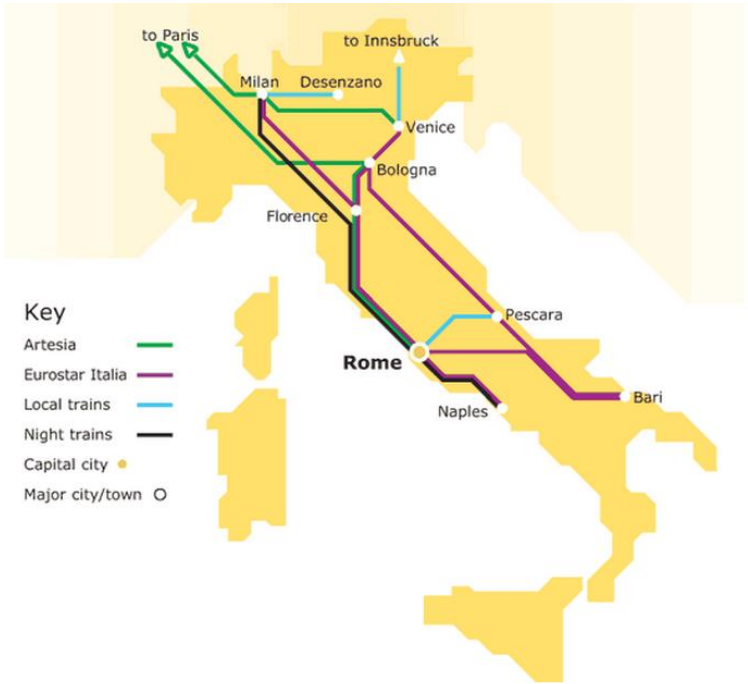
"The only exception to this is the option to specify a "static" precondition, necessary in some domains which require static knowledge."

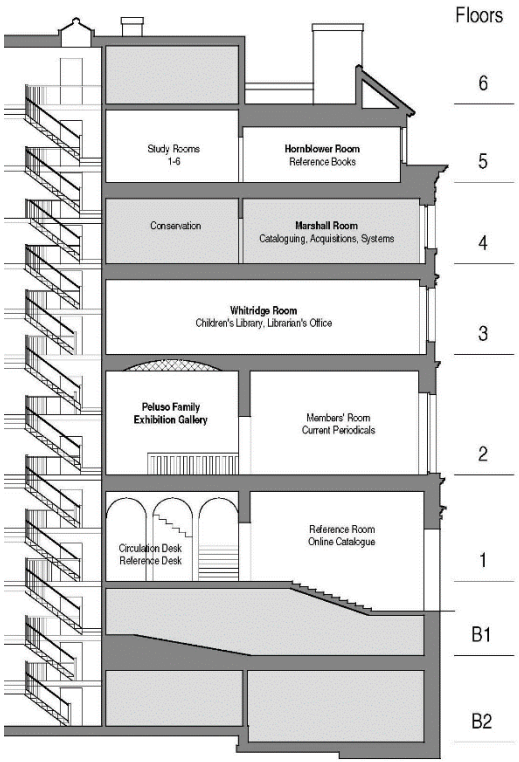
(Cresswell et al., 2013)

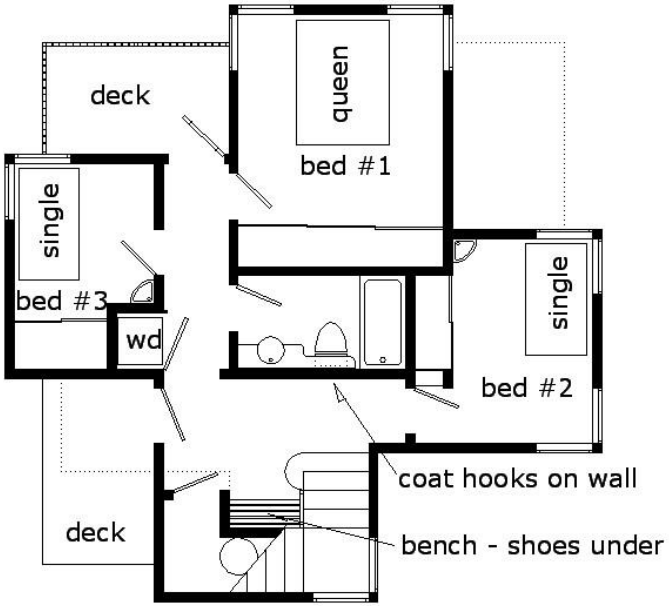
Static Knowledge – What is it?

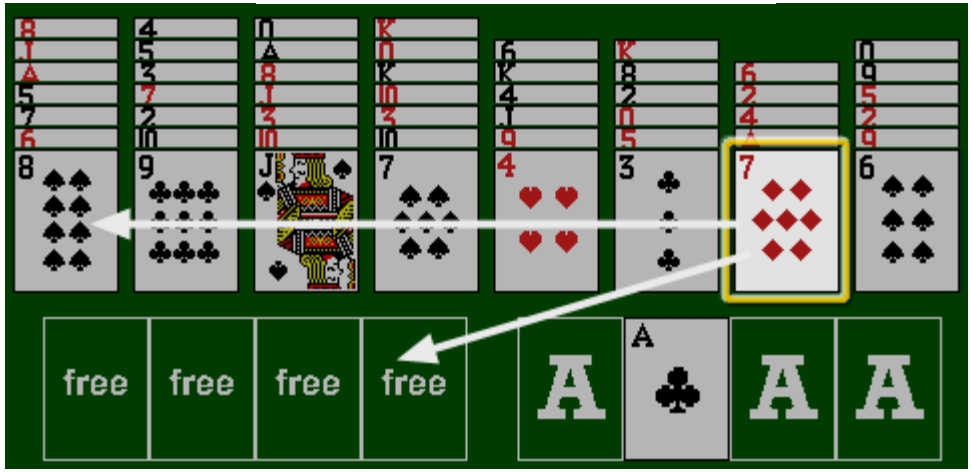
Relationships/properties that never change in the world and are implicit in the domain model but would not be directly expressed in plan traces

- Appear only in preconditions only
- Restricts values of variables
- E.g. layout of roads, level of floors, fixed relationships between specific cards in *freecell, ...*


to Paris
to Innsbruck
Milan
Desenzano
Venice
Bologna
Florence
Pescara
Rome
Bari
Naples
Key
Artesia
Eurostar Italia
Local trains
Night trains
Capital city
Major city/town


Floors
6
5
4
3
2
1
B1
B2
Study Rooms 1-6
Hornblower Room Reference Books
Conservation
Marshall Room Cataloguing, Acquisitions, Systems
Whitridge Room Children's Library, Librarian's Office
Peluso Family Exhibition Gallery
Members' Room Current Periodicals
Circulation Desk Reference Desk
Reference Room Online Catalogue


deck
queen
bed #1
single
bed #3
wd
single
bed #2
coat hooks on wall
deck
bench - shoes under


free
free
free
free

ASCoL (Automated Static Constraint Learner)

- Exploits graph analysis for automatically identifying static relations to enhance planning domain models by observing a set of training plan traces

- We enhance the output domain model of the LOCM system to capture static domain constraints from the same set of input training plans as used by LOCM

ASCoL -The learning problem

Input plan traces contain tacit knowledge about constraints validation/acquisition

...,sendtofree(club8,diamond4,n2,n1),sendtofree(spade9,heart7,n1,n0),move(heart7,diamonda,spade8),sendtohome(diamonda,spade3,diamond,n1,diamond0,n0),move(diamond4,diamond9,club5),colfromfreecell(club8,diamond9,n0,n1),move(diamond7,heart4,club8),sendtofree(heart6,cluba,n1,n0),sendtohome(cluba,club2,club,n1,club0,n0),sendtohome(club2,club9,club,n2,cluba,n1),move(spade3,heart5,diamond4),move(spade4,diamond3,heart5),move(diamond3,club4,spade4),colfromfreecell(heart8,club9,n0,n1),sendtofree(club4,club6,n1,n0),move(club6,club3,heart7),sendtohome_b(club3,club,n3,club2,n2,n0,n1),homefromfreecell(club4,club,n4,club3,n3,n0,n1),sendtonewcol(spade3,diamond4,n1,n0),...

Static Preconditions (Constraints)

Successor_card(heart9, spade8)

...

Every domain has separate static background

Input : a tuple (P, T), **P** = plan traces ,**T** = types of action arguments in P

Output : **R** = constraint repository

Methodology

1. Input Action Sequence

...,sendtofree(club8,diamond4,n2,n1),sendtofree(spade9,heart7,n1,n0),move(heart7,diamonda,spade8),sendtohome(diamonda,spade3,diamond,n1,diamond0,n0),move(diamond4,diamond9,club5),colfromfreecell(club8,diamond9,n0,n1),move(diamond7,heart4,club8),sendtofree(heart6,cluba,n1,n0),sendtohome(cluba,club2,club,n1,club0,n0),sendtohome(club2,club9,club,n2,cluba,n1),move(spade3,heart5,diamond4),move(spade4,diamond3,heart5),move(diamond3,club4,spade4),colfromfreecell(heart8,club9,n0,n1),sendtofree(club4,club6,n1,n0),move(club6,club3,heart7),sendtohome_b(club3,club,n3,club2,n2,n0,n1),homefromfreecell(club4,club,n4,club3,n3,n0,n1),sendtonewcol(spade3,diamond4,n1,n0),...

$\mathbf{A_i(O_{i1},...,O_{ij})}$ **for i = 1,...,N**

2. An action arguments' pairing based on same-type rule

sendtofree-b(cardX **-card** N1 N0 N3 N4 **-num**)

Pair1 (N1, N0), Pair2 (N1, N3), Pair3 (N1, N4),
Pair4 (N0, N3), Pair5 (N0, N4), Pair6 (N3, N4).

3. Multiple instances of an action in input

...
sendtofree-b (cardX N1 N0 N3 N4)
sendtofree-b (cardY N2 N1 N2 N3)
...

4. Graph edges in *Conn* set for *pair1*

Conn = {...(N1,N0), (N2,N1)...}

5. The directed graph of *pair1*

...N2 → N1 → N0...

Evaluation

number of operators **(# Operators**), total number of static relations (**# SR**) are presented, number of identified static relationships (**Learnt SR**), number of additional static relations provided (**Additional SR**) , number of plans (**#Plans**), average number of actions per plan (**A/P**), **CPU-time** in milliseconds

Domain	# Operators	# SR	Learnt SR	Additional SR	# Plans	Avg. A/P	CPU-time (ms)
TPP	4	7	7	0	7	28	171
Zenotravel	5	4	6	2	4	24	109
Miconic	4	2	2	0	1	177	143
Storage	5	5	5	0	24	15	175
Freecell	10	19	13	0	20	60	320
Hanoi	1	0	1	1	1	60	140
Logistics	6	0	1	1	3	12	98
Driverlog	6	2	2	0	3	12	35
Mprime	4	7	7	0	10	30	190
Spanner	3	1	1	0	1	8	144
Gripper	3	0	1	1	1	14	10
Ferry	3	1	2	1	1	18	130
Barman	12	3	3	0	1	150	158
Gold-miner	7	3	1	0	13	20	128
Trucks	4	3	3	0	6	25	158

- Input plans generated using **Metric-FF planner** on randomly generated problems
- Implemented in Java, and run on a **Core 2 Duo/8GB** processor

Thank you