Knowledge Representation and Reasoning Automate Warehouse

Name: Mukesh Kumar Rohil
Email: rohil@pilani.bits-pilani.ac.in

Name: Raju Singh
Email: p20200106@pilani.bits-pilani.ac.in

Abstract

The report details out programming approach for automating warehouse using *Answer Set Programming.*

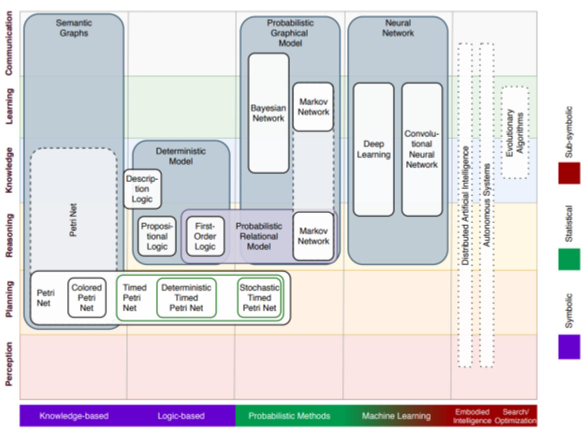
Problem Statement

In warehouses, there are several processes which can be automated in order to reduce the dependency on the human and introduce robots which are specifically programmed to work continuously with any supervision. One such scenario could be pickup-drop from station to station for order fulfilment purposes. Robots, specifically programmed using Answer Set Programming could freely move between different section of the warehouse based on predefined instruction. For the matter of simplicity, we can consider warehouse to be a form of matrix, n*n. In this matrix, robots can roam around, however, the movement itself is controlled. Each robot is program to run through a definitive path, schedule the delivery, detect collision and self-recover.

<u>Conditional Exceptions</u>: Time is a critical factor in the warehouse. So, the time between pick and delivery is calculated for each of the robots. To enable it, we must not enter any situation where robots could collide with

detection component must be executed, and the probability of collision must converge towards zero.

Overview of knowledge representation system.



Project Background

In any supply chain management framework, warehousing and warehouse management is a key component. A knowledge based systems, powered with an efficient programming base and algorithms would help solve key warehousing automation problem.

Answer Set Programming language is designed to provide knowledge representation based approach. It provides a robust methodologies to solve the problem. While coding the algorithm, one would be able to key in the constrains, collision detection, source and destination.

Approach to solve the problem.

The problem can be solved using :

- *Material collection process.*
- *Verification process*.

First step would be to create rules which is based on behaviour driven development methodologies (*BDD*). In BDD, we start with generating patterns for robots behaviours, break it down into simple rules, code rules in *ASP*, and execute in *CLINGO*. As the satisfiability rules in place, next starts the integration specificiation for these rules in iterative manner. Integration helps so that each rule can run in its entirety, without breaking the expected output. The programmed robot can move to adjacent cells, performs all necessary actions.

Main results and analysis.

A n*n (4*4) cells is created. The cell or grid includes product, shelves, pickup and delivery stations and order. The goal is of the order 2,2,0,m.

```
init(object(node,'n'),value(at, pair('x','y')))
where, the term 'n' denotes a grid cell at
coordinates positive integers 'x' and 'y'.


 init(object(highway,'h'),value(at, pair('x','y')))
 where, the term 'h' denotes a highway with coordinates
 at 'x' and 'y'.


 init(object(pickingStation,'p'),value(at, pair('x','y')))
 where, the term 'p' denotes a picking station with
 coordinates at 'x' and 'y'.

 init(object(robot,'r'),value(at, pair('x','y')))
 where, the term 'r' denotes a robot, initially
 located at coordinates 'x' and 'y'.

 init(object(shelf,'s'),value(at, pair('x','y')))
 where, the term 's' denoted a shelf, initially
 located at coordinates 'x' and 'y'.
```

Robots are initially placed at a unique location in idle state. Inputs are provided to the robots in form of order, shelves locations, pick station, delivery stations, routs to follows, detect collision etc. With the given satisfiability constraints, the code can be executed as:

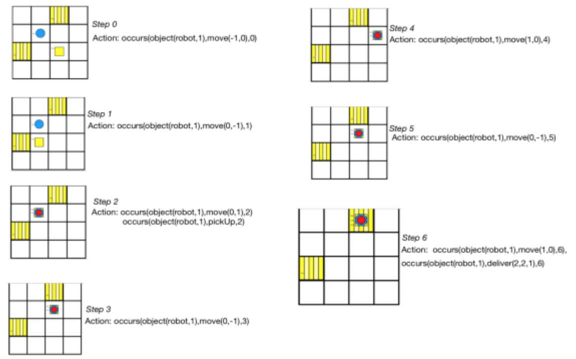
Run the code:

```
$ clingo warehouse.lp instance.asp -c n=4 -c m=7
```

Format the output:

```
occurs(object(robot,1),move(-1,0),0)

occurs(object(robot,1),move(0,-1),1)
occurs(object(robot,1),move(0,1),2)

occurs(object(robot,1),pickUp,2)

 occurs(object(robot,1),move(0,-1),3)

occurs(object(robot,1),move(1,0),4)

occurs(object(robot,1),move(0,-1),5)

occurs(object(robot,1),move(1,0),6), occurs(object(robot,1),deliver(2,2,1),6)
```

Snap:



Conclusion (Self-assessment)

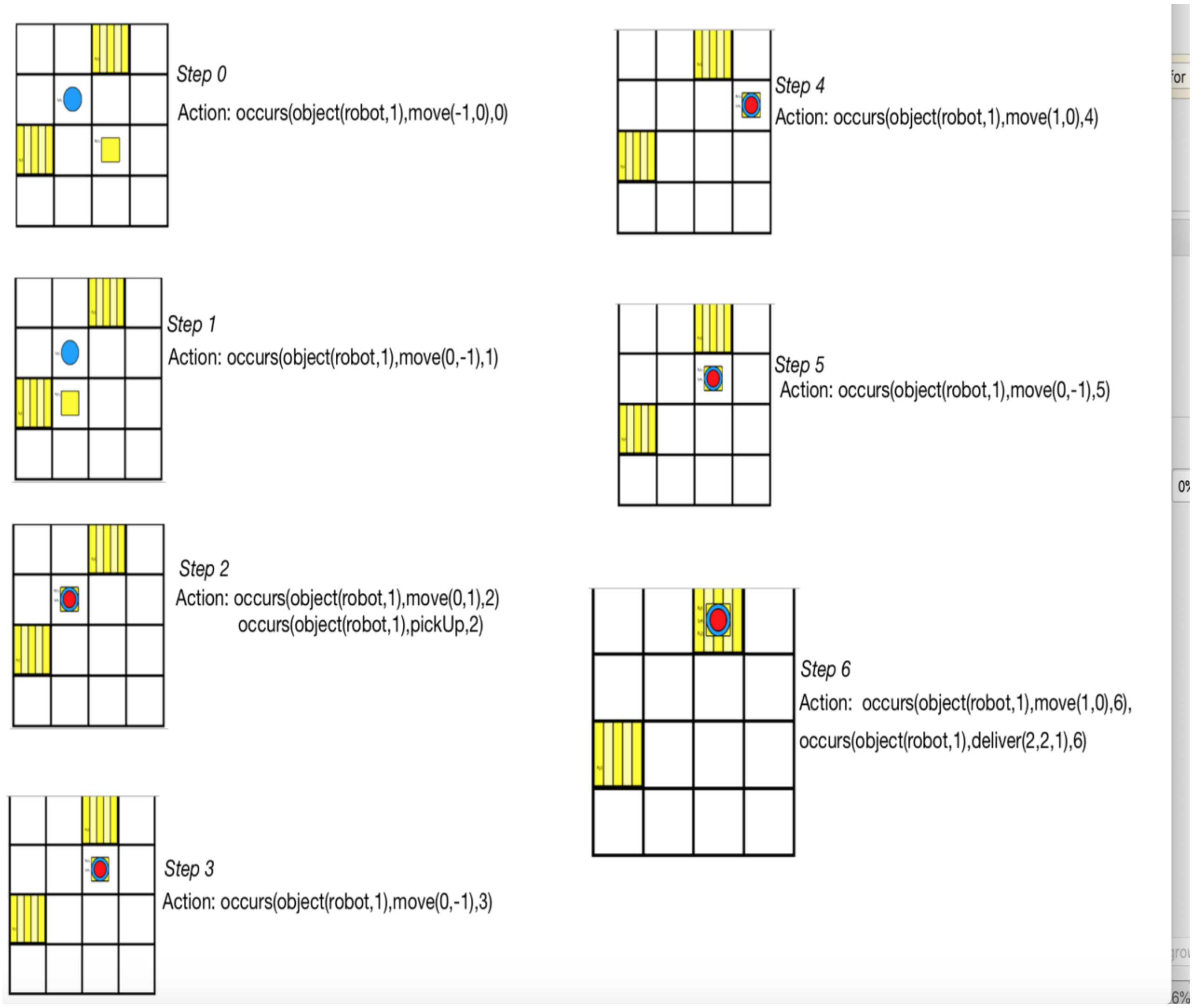
The main purpose of designing such problem statement and find an optimal solution is to solve key bottlenecks of supply chain management, on which the whole world is dependent on. Create an insight of the projects, constraints break down constraints, create satisfiability rules, perform integration tests on the each component

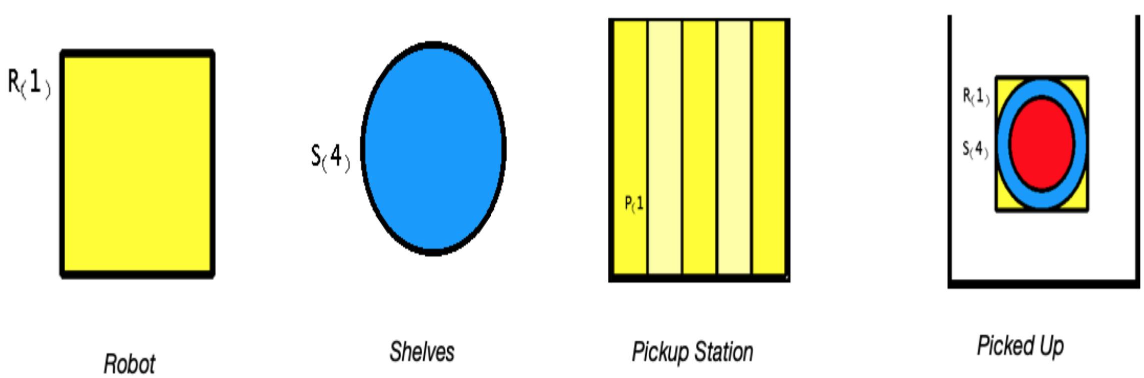
The code is modular with different sections such as: Actions, fluent, law of inertia etc.

Constraints:

```
:- not 1{robot(R,X,Y,T)}1,  robot(R,X,Y,T),  T=0..m.
:- not 1{shelf(S,X,Y,T)}1, shelf(S), T=1..m.
:- not 1{product(I,S,U,T)}1, product(I), T=0..m.
:- not 1{order(O,I,U,T)}1, order(O,P), T=0..m.
:- not 1{carries(R,T,B):boolean(B)}1, robot(R,X,Y,T),  T=0..m.
```

Figures


Step 0
Action: occurs(object(robot,1),move(-1,0),0)
Step 1
Action: occurs(object(robot,1),move(0,-1),1)
Step 2
Action: occurs(object(robot,1),move(0,1),2)
occurs(object(robot,1),pickUp,2)
Step 3
Action: occurs(object(robot,1),move(0,-1),3)
Step 4
Action: occurs(object(robot,1),move(1,0),4)
Step 5
Action: occurs(object(robot,1),move(0,-1),5)
Step 6
Action: occurs(object(robot,1),move(1,0),6),
occurs(object(robot,1),deliver(2,2,1),6)


R(1)
S(4)
P(1
R(1)
S(4)
Robot
Shelves
Pickup Station
Picked Up

Opportunities for future work

ASP and KRR are powerful tools and concepts which goes hand in hand to solve complex real time problem. The idea is to solve problem with logic functional programming.

References

Custodio, L., Machado, R. Flexible automated warehouse: a literature review and an innovative framework. Int J Adv Manuf Technol 106, 533–558 (2020). https://doi.org/10.1007/s00170-019-04588-z

M. Schwarz *et al*., "NimbRo picking: Versatile part handling for warehouse automation," *2017 IEEE International Conference on Robotics and Automation (ICRA)*, Singapore, 2017, pp. 3032-3039, doi: 10.1109/ICRA.2017.7989348.

Hamberg, R. & Verriet, J. eds., 2012. Automation in Warehouse Development. Available at: http://dx.doi.org/10.1007/978-0-85729-968-0.

Guo, W.L. & Huang, Z.J., 2014. Analysis and Research of the Automated Warehouse Management and Control System. Applied Mechanics and Materials, 511-512, pp.1095–1098. Available at: http://dx.doi.org/10.4028/www.scientific.net/amm.511-512.1095.

Maciol, A., 2016. Knowledge-based methods for cost estimation of metal casts. The International Journal of Advanced Manufacturing Technology, 91(1-4), pp.641–656. Available at: http://dx.doi.org/10.1007/s00170-016-9704-z.

[1]F. Negrello, H. S. Stuart, and M. G. Catalano, “Hands in the Real World,” Frontiers in Robotics and AI, vol. 6, Jan. 2020.
[1]T. Beyer, P. Gohner, R. Yousefifar, and K.-H. Wehking, “Agent-based dimensioning to support the planning of Intra-Logistics systems,” 2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA), Sep. 2016.

[1]S. Andova, M. G. J. van den Brand, L. J. P. Engelen, and T. Verhoeff, “MDE Basics with a DSL Focus,” Lecture Notes in Computer Science, pp. 21–57, 2012.

Aksoy et al., 2011. Aksoy, E. E., Abramov, A., Dörr, J., Ning, K., Dellen, B., and Wörgötter, F. (2011). Learning the semantics of object–action relations by observation. The International Journal of Robotics Research, 30(10):1229–1249. Bastianelli et al., 2013.

Bastianelli, E., Bloisi, D., Capobianco, R., Gemignani, G., Iocchi, L., and Nardi, D. (2013). Knowledge representation for robots through humanrobot interaction. CoRR, abs/1307.7351. Chao and L Thomaz, 2012.

Chao, C. and L Thomaz, A. (2012). Timed petri nets for multimodal interaction modeling. ICMI 2012 Workshop on Speech and Gesture Production in Virtually and Physically Embodied Conversational Agents.

Corea, 2018. Corea, F. (2018). Ai knowledge map: How to classify ai technologies. https://www.forbes.com/sites/cognitiveworld/2018/08/22/ai-knowledge-maphow-to-classify-ai-technologies/#78f5a32e7773. Accessed: 2018-15-11. Costelha and Lima, 2007.

Costelha, H. and Lima, P. (2007). Modelling, analysis and execution of robotic tasks using petri nets. pages 1449–1454.

Costelha and Lima, 2012. Costelha, H. and Lima, P. (2012). Robot task plan representation by petri nets: Modelling, identification, analysis and execution. Autonomous Robots, 33.

Floreano and Mattiussi, 2008. Floreano, D. and Mattiussi, C. (2008). Bio-Inspired Artificial Intelligence: Theories, Methods, and Technologies. The MIT Press. Galindo et al., 2008.

Galindo, C., Fernández-Madrigal, J.-A., González, J., and Saffiotti, A. (2008). Robot task planning using semantic maps. Robotics and Autonomous Systems, 56(11):955 – 966. Semantic Knowledge in Robotics.

IGI Global, 2017. IGI Global (2017). Definition: Collaborative knowledge. https://www.igi-global.com/dictionary/collaborative-knowledgeconstruction/36158. Accessed: 2019-02-09. Jain et al., 2015.

Jain, A., Saini, M., and Kumar, M. (2015). Introduction to artificial intelligence. International Journal for Research in Applied Science & Engineering Technology, 3(5):241–247.

Jain et al., 2011. Jain, D., Gleissenthall, K., and Beetz, M. (2011). Bayesian logic networks and the search for samples with backward simulation and abstract constraint learning. pages 144–156.

Jain and Inamura, 2013. Jain, R. and Inamura, T. (2013). Bayesian learning of tool affordances based on generalization of functional feature to estimate effects of unseen tools. 18(1-2):95–103. Exported from https://app.dimensions.ai on 2018/12/19. Kelnar, 2016.

Kelnar, D. (2016). The fourth industrial revolution: a primer on artificial intelligence (ai). https://medium.com/mmc-writes/the-fourth-industrial-revolutiona-primer-on-artificial-intelligence-ai-ff5e7fffcae1. Accessed: 2018-12-19.

Kjellström et al., 2011. Kjellström, H., Romero, J., and Kragic, D. (2011). Visual object-action recognition: Inferring object affordances from human demonstration. Computer Vision and Image Understanding, 115:81–90