An Insertion-Based Linked List Variable and Regular Constraint for Classical Planning with Constraint Programming

Damien Van Meerbeeck

Machine Reasoning Lab, Linköping University, Sweden

CASP:ER Workshop, 29 June 2026

LINKÖPING UNIVERSITY

Machine Reasoning Lab

WASP WALLENBERG AI, AUTONOMOUS SYSTEMS AND SOFTWARE PROGRAM

Motivation

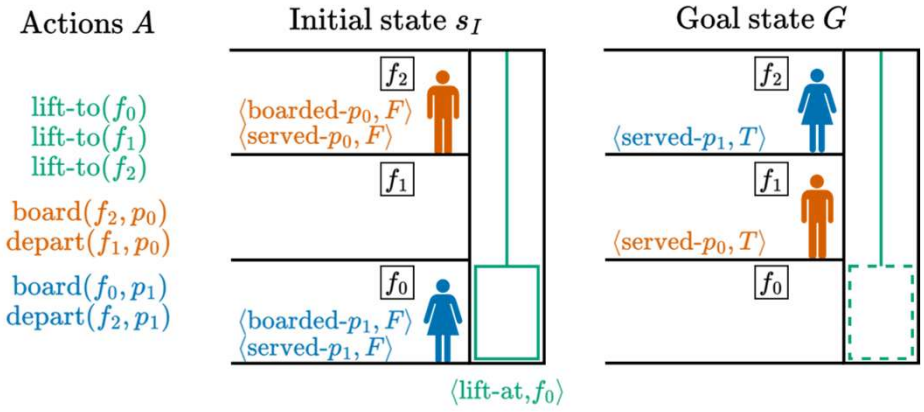
- Often CP or SAT based planner use an array of variables of fixed length for the plan.

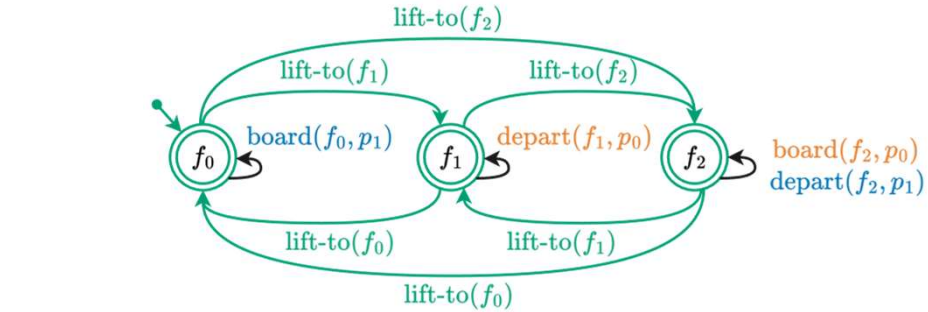
$\boldsymbol{Plan}$	$\{0, \dots, 4\}$	$\{0, \dots, 4\}$	$\{0, \dots, 4\}$	$\{0, \dots, 4\}$	$\{0, \dots, 4\}$	$\{0, \dots, 4\}$

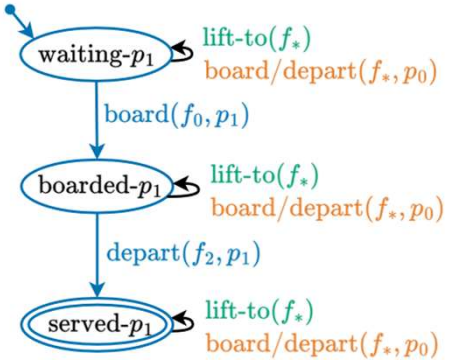
- However, this representation hinders the potential of these planners to generate plans in a non-sequential way.
  - Where to place required action 2?
  - Where to place action 4 that needs to follow action 2?
- Branching on action precedence rather than on action positioning, could lead to stronger branching heuristics.

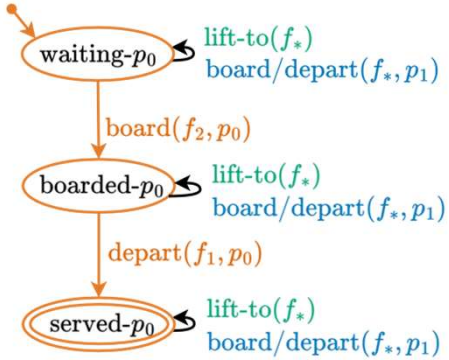
Automata-based CP planning

- Derive factored automata form planning tasks and model the dynamics of the task with Regular constraints.



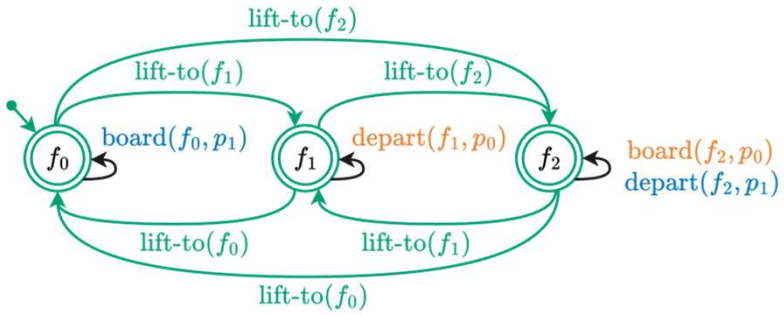


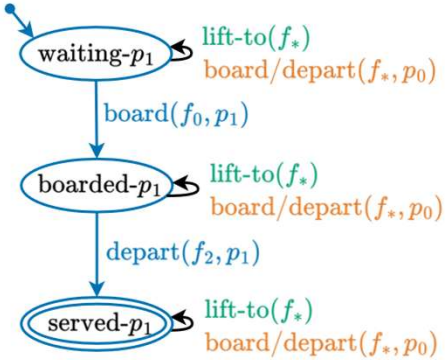


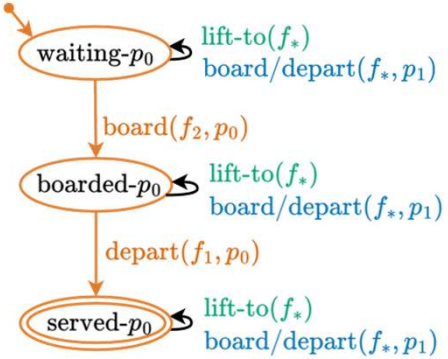


Idea – Planning with insertion-based variable

- Use the automata representation to construct the plan with insertions.







$bos \rightarrow eos$

$bos \rightarrow \text{board}(f_0, p_1) \rightarrow \text{depart}(f_2, p_1) \rightarrow eos$

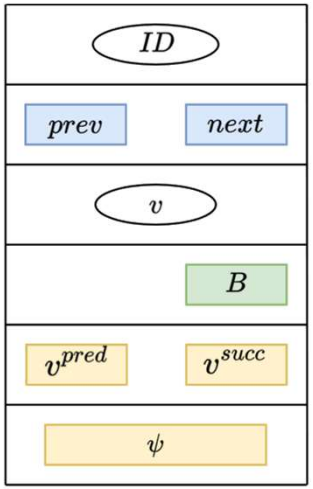
$bos \rightarrow \text{board}(f_0, p_1) \rightarrow \text{lift-to}(f_2) \rightarrow \text{depart}(f_2, p_1) \rightarrow eos$

$bos \rightarrow \text{board}(f_0, p_1) \rightarrow \text{lift-to}(f_2) \rightarrow \text{depart}(f_2, p_1) \rightarrow \text{board}(f_2, p_0) \rightarrow eos$

$bos \rightarrow \text{board}(f_0, p_1) \rightarrow \text{lift-to}(f_2) \rightarrow \text{depart}(f_2, p_1) \rightarrow \text{board}(f_2, p_0) \rightarrow \text{depart}(f_1, p_0) \rightarrow eos$

$bos \rightarrow \text{board}(f_0, p_1) \rightarrow \text{lift-to}(f_2) \rightarrow \text{depart}(f_2, p_1) \rightarrow \text{board}(f_2, p_0) \rightarrow \text{lift-to}(f_1) \rightarrow \text{depart}(f_1, p_0) \rightarrow eos$

Linked list variable – Node



Unique identifier of the node.

Identifier of the previous and next node in the list.

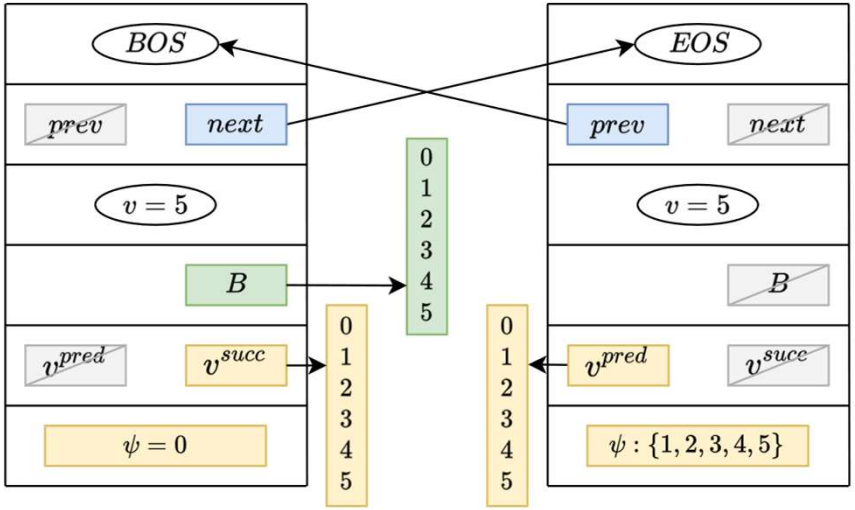
Value of the node.

Values that can be inserted between this node and the next.

Values of the predecessor and successor.

Position of the node in the list.

Linked list variable – Initialization



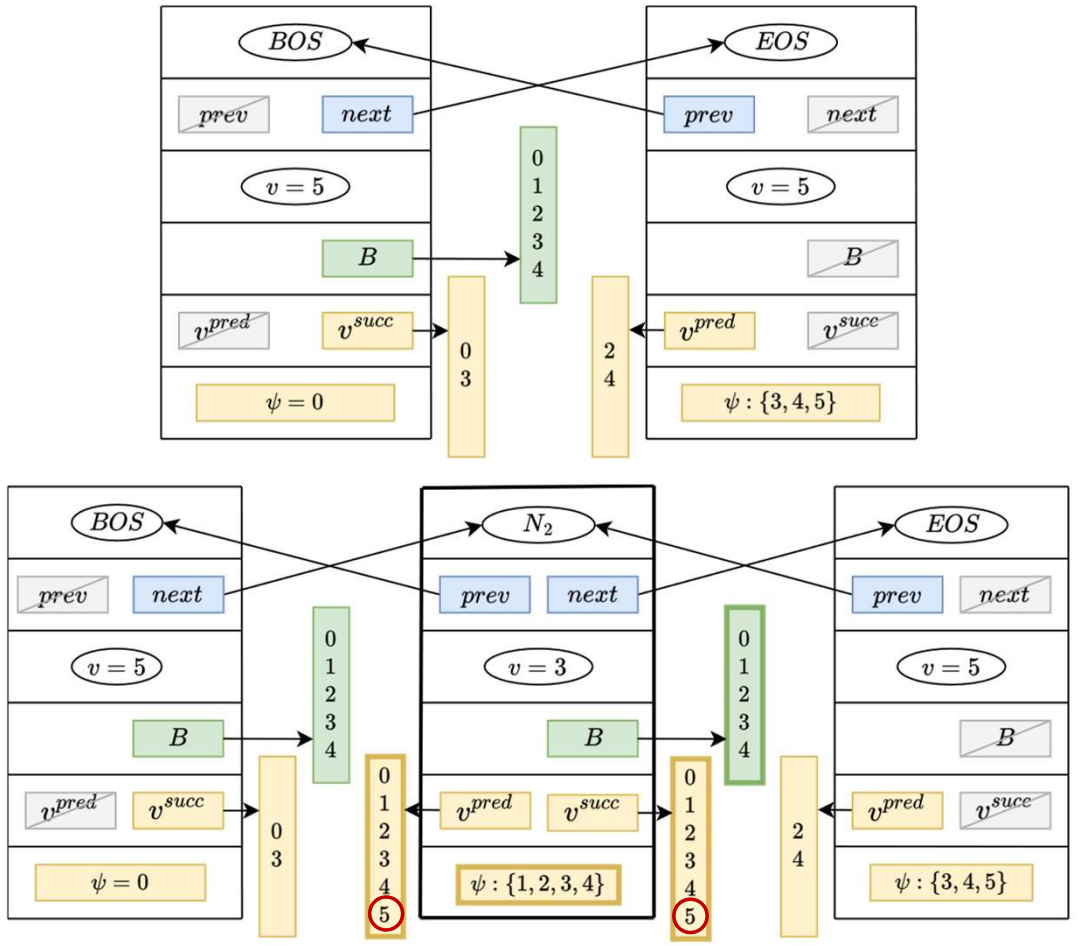
Linked list variable – Variable consistency

$N_i \to N_j$ notes two consecutive nodes of the list
$N_i \mapsto N_j$ notes a direct successor-predecessor relation
$N_i \rightsquigarrow N_j$ notes successive nodes that are not in a successor-predecessor relation

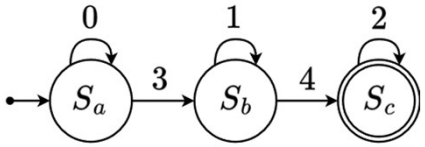
At all times:

- If $N_i \to N_j$ then $\psi_i < \psi_j$
- If $N_i \to N_j$ and $\psi_j - \psi_i = 1$, then $N_i \mapsto N_j$
- $N_i \mapsto N_j$ if and only if $B_i = \emptyset$
- If $N_i \mapsto N_j$, then $v_i^{succ} = v_j$, $v_j^{pred} = v_i$ and $\psi_j - \psi_i = 1$
- If $N_i \to N_j$, and if $v_j \notin v_i^{succ}$ or $v_i \notin v_j^{pred}$, then $N_i \rightsquigarrow N_j$
- $N_i \rightsquigarrow N_j$ if and only if $\psi_j - \psi_i > 1$

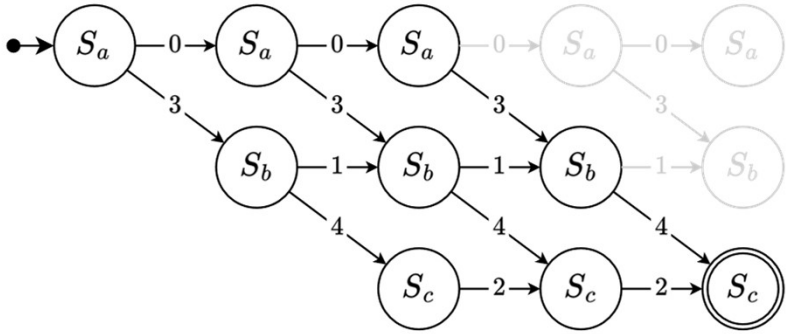
Linked list variable – Variable insertion



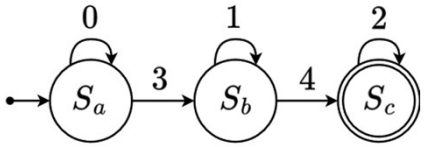
Regular constraint for array of variables



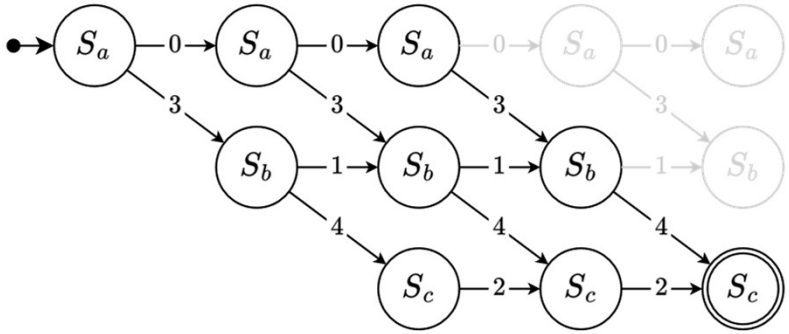
Plan	{0,1,2,3,4}	{0,1,3,4}	{1,2,3,4}	{0,1,2,3,4}



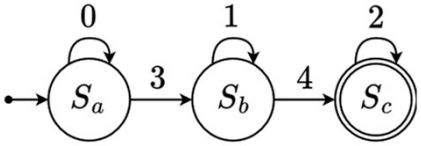
Regular constraint for array of variables

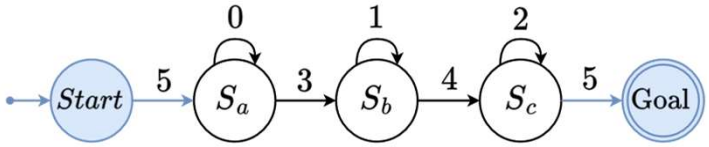


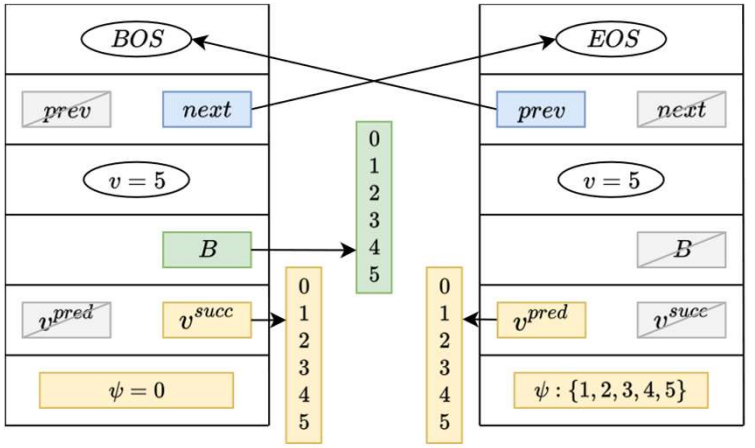
Plan	{0,3}	{0,1,2,3,4}	{1,2,3,4}	{2,4}

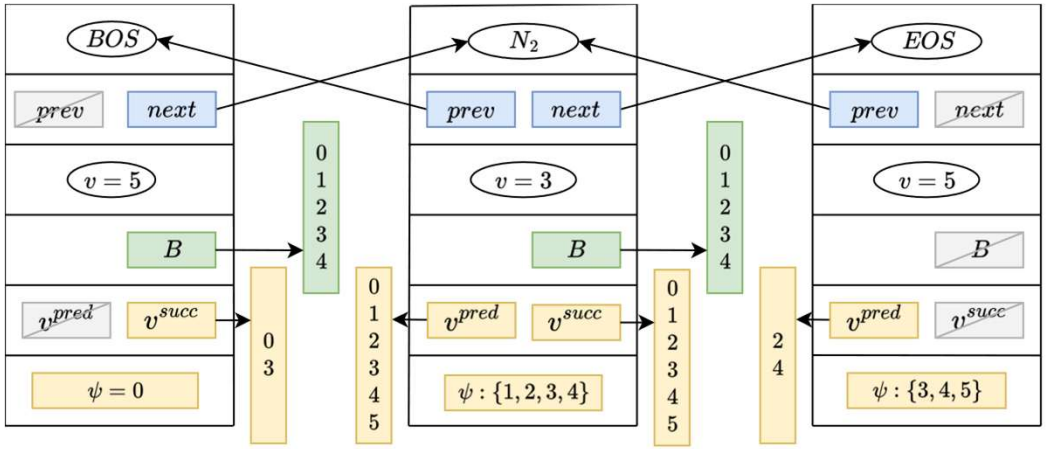


Regular constraint for the linked list variable

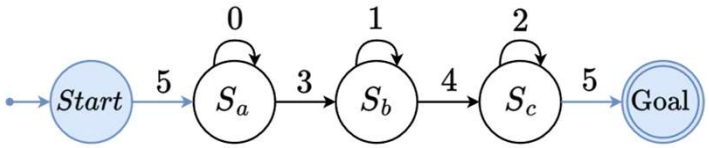




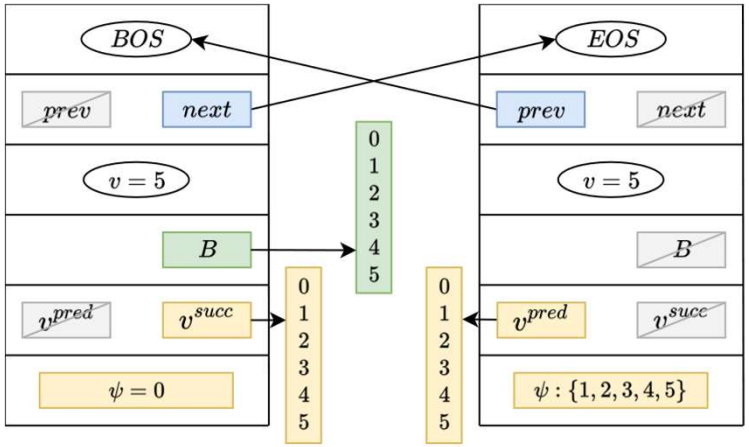




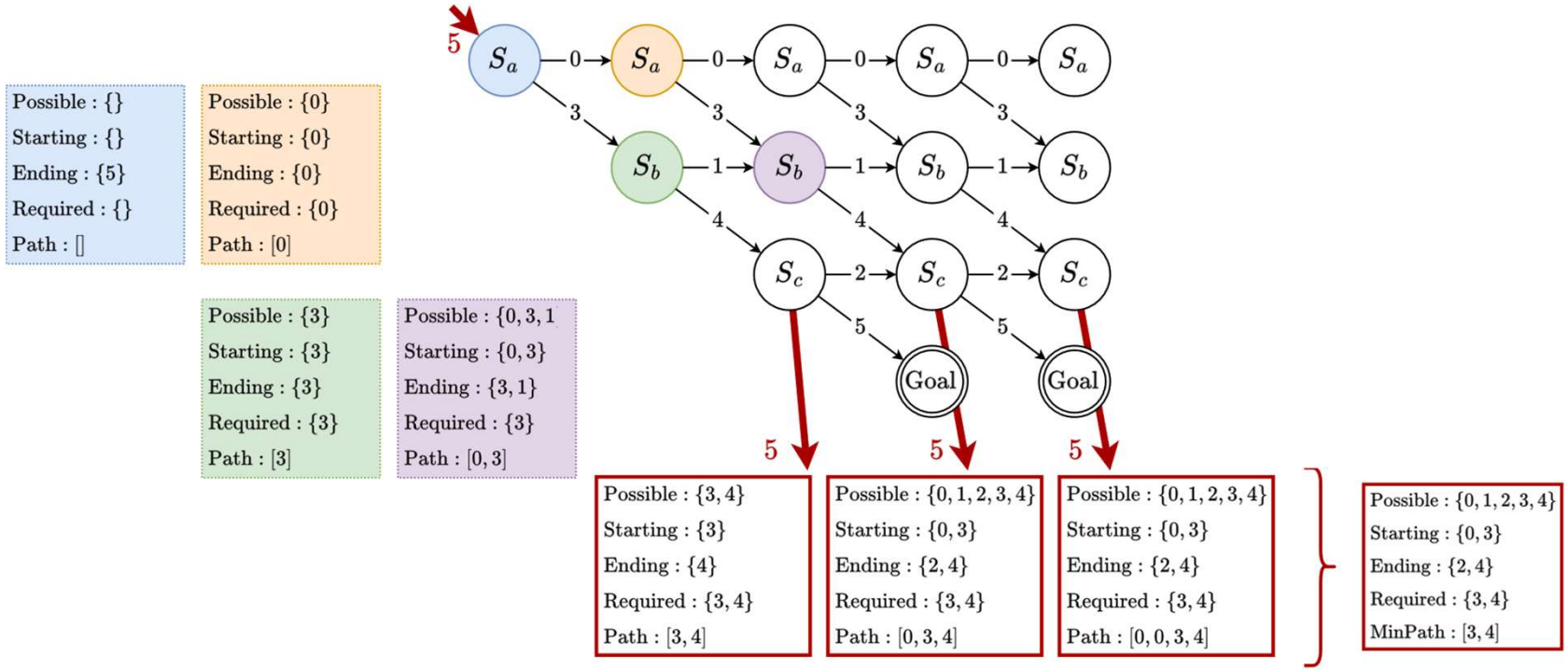
Regular constraint for the linked list variable



Possible : {}	Possible values to insert (B)
Starting : {}	Possible starting values (v^{pred})
Ending : {}	Possible ending values (v^{succ})
Required : {}	Required values for insertion
Path : []	Possible path to reach state (ψ)

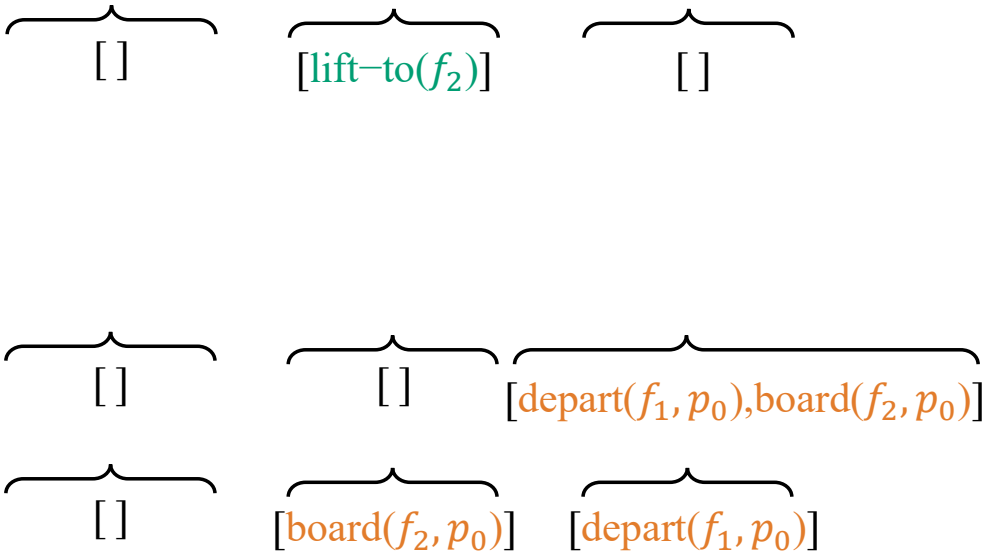


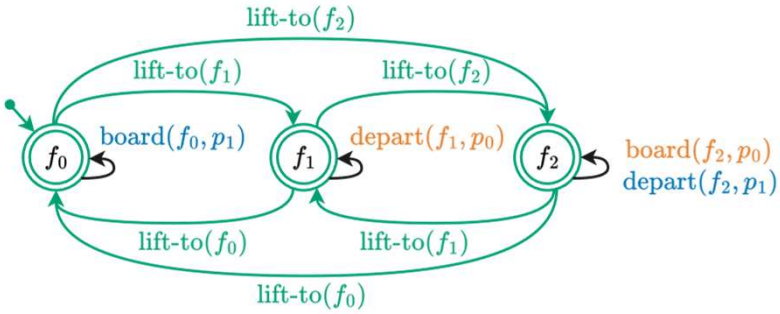
Regular constraint – Propagation

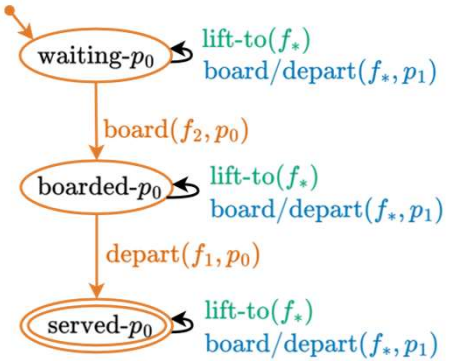


Branching

$bos \rightarrow \text{board}(f_0, p_1) \rightarrow \text{depart}(f_2, p_1) \rightarrow eos$



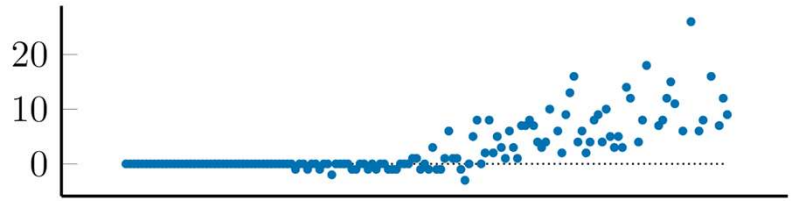




Experimental results

Model (CP solver)	solved	optimal
Array (MiniCPBP)	60	33
Array (CP-SAT)	141	69
Linked list (MaxiCP)	150	63

Comparison of the performance of the model using the linked list variable to the two models using a sequence of integer variables for the two different solvers, MiniCPBP and CP-SAT. Over the 150 tasks of the Miconic domain, the first column reports the number of tasks solved, while the second column reports the number of tasks for which the optimal solution was found.



Difference of the best plan length found by the linked list model and the sequence model with CP-SAT fort he tasks of the Miconic domain ordered by increasing difficulty.

Conclusion

- A novel insertion-based variable representing an ordered sequence of elements with repetitions, the linked list variable.
- A Regular constraint for this variable for which the propagation of the constraint can, in addition to pruning the domain of the variable, provide information to guide the search towards solutions.
- Potential in using Insertion based-variable to solve classical planning.