PATTY: The Symbolic Pattern Planner

Matteo Cardellini, Enrico Giunchiglia

DIBRIS, University of Genova, Genova, Italy
{matteo.cardellini, enrico.giunchiglia}@unige.it

Abstract

In this abstract, we describe the approaches encoded by our planner PATTY, namely the *brave* (PATTY_B) and *greedy* (PATTY_G) approaches for the Agile track, and the *cost-effective brave* ($\text{PATTY}^\text{C}_\text{B}$) and the *cost-effective greedy* ($\text{PATTY}^\text{C}_\text{G}$) for the Satisficing track, which competed in the Numeric Track of the 2026 International Planning Competition.

Introduction

Our planner PATTY, to solve a planning problem, employs the *Planning as Satisfiability* approach (Kautz and Selman 1992), by (i) fixing a *bound* or *number of steps* n, initially set to 0, (ii) computing a logic formula encoding the sequences of actions of length n which are executable in a given starting state, (iii) checks whether one of such sequences is a valid plan by imposing the initial state I and the goal G as conditions on the starting and the final state, and (iv) upon failure, iterates from the second step while increasing n. Symbolic Pattern Planning (SPP) (Cardellini, Giunchiglia, and Maratea 2024, 2026) is a recent logic encoding, which, given an arbitrary finite sequence of actions $\prec$ called *pattern*, models the sequences of actions which are executable from a state and are also a subsequence of $\prec$. Planning as Satisfiability with SPP is very effective in solving numeric and temporal (Cardellini and Giunchiglia 2025b, 2026b) planning problems.

The configurations of PATTY used in the IPC perform symbolic search by iteratively refining patterns. At each iteration, given a reachable state P and a starting state S, we build a formula $\Pi^{\prec}_{S,P}$ enforcing that: (i) the search starts from S; (ii) $\prec = \prec_g; \prec_h$, where $\prec_g$ reaches P from S and $\prec_h$ is computed in P to progress beyond it; and (iii) there exists a state P' closer than P to a goal state, according to a goal-value function GF_P. Initially, $S = P = I$, $\prec_g$ is empty, and $\prec_h$ is computed from I. If the formula yields a goal state P', the corresponding plan is returned. Otherwise, P is updated to P', while S, $\prec_g$, $\prec_h$, and GF_P are recomputed according to the selected strategy. The choices S and the two patterns define the *brave* and *greedy* approaches.

For the Satisficing Track, plan cost is also relevant. Since SPP may introduce redundant actions, $\text{PATTY}^\text{C}_\text{B}$ and $\text{PATTY}^\text{C}_\text{G}$ first compute a plan as above and then remove redundant actions using the procedure of Med and Chrpa (2022), originally introduced by Fink and Yang (1992), later rediscovered by Nakhost and Müller (2010), and adapted to numeric planning by Cardellini, Giunchiglia, and Maratea (2026).

Pointers to Related Literature

Providing a detailed description of all details of the planner would be impossible in the limited space. We thus enumerate related literature pointing to the interested reader:

1. Balyo (2013) for classical planning and Bofill, Espasa, and Villaret (2017) for numeric planning introduce the $R^2\exists$-encoding, i.e., the concept of having a predetermined order between the actions in the encoding.
2. Scala et al. (2016b) introduce the concept of *rolling* of actions, i.e., a technique to model multiple consecutive repetitions of the same action in a single bound.
3. Scala et al. (2016a) introduce the concept of Asymptotic Relaxed Planning Graph (ARPG), used, in the next paper, to construct a pattern in polynomial time.
4. Cardellini, Giunchiglia, and Maratea (2024) is the first paper presenting the SPP approach and its Satisfiability Modulo Theories (SMT) encoding, where the pattern is static and computed with the ARPG.
5. Cardellini, Giunchiglia, and Maratea (2026) is the journal version of the paper above. In the extended version, we also consider various pattern selection procedures and mechanisms for improving the quality of the returned plan together with the approach of Med and Chrpa (2022) which are implemented in $\text{PATTY}^\text{C}_\text{B}$ and $\text{PATTY}^\text{C}_\text{G}$.
6. In Cardellini and Giunchiglia (2025a) it was shown how to symbolically search for a plan by iteratively extending (adding actions to) and compressing (removing actions from) an initially computed pattern. Specifically, by concatenating (i) a pattern $\prec_g$ that allows reaching a state s satisfying a subset of the goals, and (ii) another pattern $\prec_h$ computed from s, which is extended until reaching a state satisfying new additional goals.
7. Cardellini and Giunchiglia (2026a)[1] is the journal version of the paper above, presenting the brave and greedy approaches.

[1] At the moment of the IPC, the paper as yet to be accepted

Brave PATTY

1. The search starts from the initial state I and $\prec_h$ is a complete pattern computed in I. The initial goal value of I is c.

2. We employ the formula $\Pi^{\prec_h}_{I,I}$ to search for a state that can be reached from I using $\prec_h$ and with a goal value less than c.

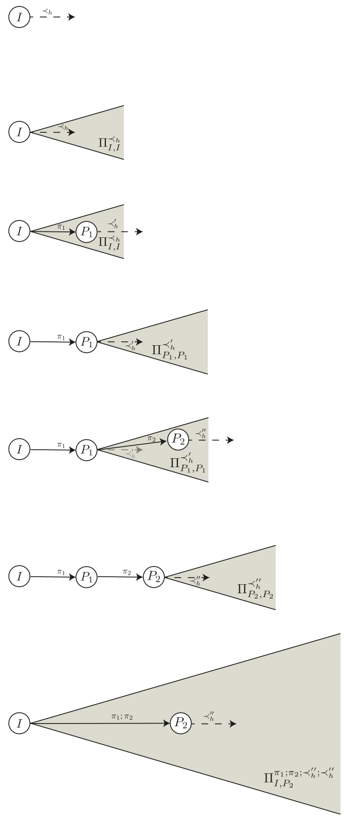
3. Suppose to find, among these states, the state P_1 with goal value function c_1 s.t., $0 < c_1 < c$. The plan to reach P_1 from I is π_1. The complete pattern computed in P_1 is $\prec'_h$.

4. The search restarts from P_1 employing the pattern $\prec'_h$, searching for another state with goal value function less than c_1.

5. Suppose such state P_2 exists, with goal value function c_2 s.t., $0 < c_2 < c_1$. The plan to reach P_2 from P_1 is π_2 and $\prec''_h$ is the complete pattern computed in P_2.

6. The search restarts from P_2, employing the pattern $\prec''_h$, searching for a state with goal value less than c_2.

7. Suppose such a state cannot be reached. The search starts again from I, employing the pattern obtained by concatenating the plan to reach P_2 from I, i.e., $\pi_1; \pi_2$ and the pattern $\prec''_h$ concatenated $n = 2$ times, to reach more states.



Step 7 is repeated, incrementing n, until a new state P_3 with goal value $c_3 < c_2$ is found. If $c_3 = 0$, the plan is returned; otherwise, we keep iterating from Step 6, restarting from P_3 and employing the complete pattern $\prec'''_h$ computed in P_3.

(a) Example of the brave PATTY$_{\mathrm{B}}$ approach.

Greedy PATTY

1. The search starts from the initial state I and $\prec_h$ is an incomplete pattern computed in I, by Alg. 3 with $p = 1$. The goal value of I is c.

2. We employ the formula $\Pi^{\prec_h}_{I,I}$ to search for a state that can be reached from I using $\prec_h$ and with a goal value less than c.

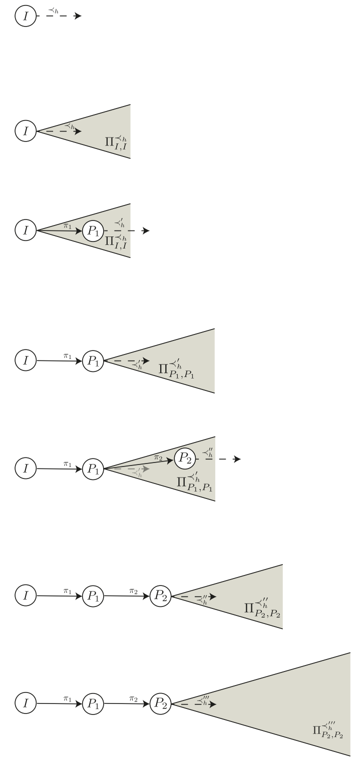
3. Suppose to find, among these states, the state P_1 with goal value function c_1 s.t., $0 < c_1 < c$. The plan to reach P_1 from I is π_1. The incomplete pattern computed with $p = 1$ in P_1 is $\prec'_h$.

4. The search restarts from P_1 employing the pattern $\prec'_h$, searching for another state with goal value function less than c_1.

5. Suppose such state P_2 exists, with goal value function c_2 s.t., $0 < c_2 < c_1$. The plan to reach P_2 from P_1 is π_2 and $\prec''_h$ is the incomplete pattern computed in P_2 with $p = 1$.

6. The search restarts from P_2, employing the pattern $\prec''_h$, searching for a state with goal value less than c_2.

7. Suppose such state cannot be reached. The search starts regardless again from P_2, employing the incomplete pattern $\prec'''_h$ computed from P_2 by Alg. 3 with $p = 2$.



Step 7 is repeated, incrementing p, until a new state P_3 with goal value $c_3 < c_2$ is found. If $c_3 = 0$, the plan is returned; otherwise, we keep iterating from Step 6, restarting from P_3 and employing the pattern $\prec''''_h$ computed from P_3.

(b) Example of the greedy PATTY$_{\mathrm{G}}$ approach.

Brave and Greedy Approaches

Figures 1a and 1b illustrate the two strategies. Both relax a conjunctive goal into a disjunction of *subgoals*. At each iteration, the planner searches for a state P' satisfying at least one additional subgoal. The prefix pattern $\prec_g$ connects the current starting state S to the current intermediate state P, while the suffix pattern $\prec_h$ guides the search from P toward further progress. The suffix may be *complete*, if it contains all task actions, or *incomplete* otherwise.

In the brave strategy, when a better state P' is found, the search restarts from it: S is set to P', $\prec_g$ is reset to the empty sequence ϵ, and a complete $\prec_h$ is computed from P'. If no such state is found, the strategy preserves completeness by restarting from I, setting $\prec_g$ to a plan reaching the last intermediate state, and repeatedly concatenating the last complete $\prec_h$ until progress is obtained. Since a complete pattern contains all task actions, any plan of length n can be represented by concatenating that pattern n times.

The greedy strategy differs in two respects. First, $\prec_h$ is incomplete and contains only actions that the ARPG identifies as promising for reaching a subgoal. Second, when no better state is found, the search continues from the last intermediate state instead of restarting from I. This sacrifices completeness, but can be more effective in domains where the pattern focuses the search on useful actions.

Cost-Effective Approaches

The planners PATTY$^{\mathrm{C}}_{\mathrm{B}}$ and PATTY$^{\mathrm{C}}_{\mathrm{G}}$ follow the same search algorithm of PATTY$_{\mathrm{B}}$ and PATTY$_{\mathrm{G}}$ as shown in Fig. 1a and 1b, respectively. The SPP encoding is known to return redundant plans (Cardellini, Giunchiglia, and Maratea 2026), i.e., plans where actions do not contribute either to the goal or to other actions' preconditions. The PATTY$^{\mathrm{C}}_{\mathrm{B}}$ and PATTY$^{\mathrm{C}}_{\mathrm{G}}$ approaches, from the first computed plan, remove redundant actions employing the Action Elimination Algorithm, quadratic in the length of the plan, presented in Algorithm 1 of Med and Chrpa (2022) – originally introduced by Fink and Yang (1992) and later rediscovered and formulated by Nakhost and Müller (2010) – cast for numeric planning. In a nutshell, the algorithm iterates through the plan trying, for each action a, to remove a together with all following actions which preconditions become inapplicable by removing a. If the goal is still achieved without these actions, they are redundant and can be removed.

References

Balyo, T. 2013. Relaxing the Relaxed Exist-Step Parallel Planning Semantics. In *2013 IEEE 25th International Conference on Tools with Artificial Intelligence*, 865–871. Herndon, VA, USA: IEEE. ISBN 978-1-4799-2972-6 978-1-4799-2971-9.

Bofill, M.; Espasa, J.; and Villaret, M. 2017. Relaxed Exists-Step Plans in Planning as SMT. In Sierra, C., ed., *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, 563–570. ijcai.org.

Cardellini, M.; and Giunchiglia, E. 2025a. Pushing the Envelope in Numeric Pattern Planning. In Ortiz, M.; Wassermann, R.; and Schaub, T., eds., *Proceedings of the 22nd International Conference on Principles of Knowledge Representation and Reasoning, KR 2025, Melbourne, Australia. November 11-17, 2025*.

Cardellini, M.; and Giunchiglia, E. 2025b. Temporal Numeric Planning with Patterns. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(25): 26481–26489.

Cardellini, M.; and Giunchiglia, E. 2026a. Exploiting Search in Symbolic Numeric Planning with Patterns.

Cardellini, M.; and Giunchiglia, E. 2026b. Symbolic Pattern Temporal Numeric Planning with Intermediate Conditions and Effects. *arXiv*, (arXiv:2602.09798).

Cardellini, M.; Giunchiglia, E.; and Maratea, M. 2024. Symbolic Numeric Planning with Patterns. In Wooldridge, M. J.; Dy, J. G.; and Natarajan, S., eds., *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, 20070–20077. AAAI Press.

Cardellini, M.; Giunchiglia, E.; and Maratea, M. 2026. Symbolic pattern planning. *Artif. Intell.*, 352: 104482.

Fink, E.; and Yang, Q. 1992. Formalizing Plan Justifications. In Glasgow, J. I.; and Hadley, R. F., eds., *Proceedings of the Ninth Conference of the Canadian Society for Computational Studies of Intelligence*, 9–14. Canadian Society for Computational Studies of Intelligence.

Kautz, H. A.; and Selman, B. 1992. Planning as Satisfiability. In Neumann, B., ed., *10th European Conference on Artificial Intelligence, ECAI 92, Vienna, Austria, August 3-7, 1992. Proceedings*, 359–363. John Wiley and Sons.

Med, J.; and Chrpa, L. 2022. On Speeding Up Methods for Identifying Redundant Actions in Plans. In Kumar, A.; Thiébaux, S.; Varakantham, P.; and Yeoh, W., eds., *Proceedings of the Thirty-Second International Conference on Automated Planning and Scheduling, ICAPS 2022, Singapore (virtual), June 13-24, 2022*, 252–260. AAAI Press.

Nakhost, H.; and Müller, M. 2010. Action elimination and plan neighborhood graph search: Two algorithms for plan improvement. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 20, 121–128.

Scala, E.; Haslum, P.; Thiebaux, S.; and Ramirez, M. 2016a. Interval-Based Relaxation for General Numeric Planning. In *Proceedings of the Twenty-second European Conference on Artificial Intelligence*, 655–663.

Scala, E.; Ramirez, M.; Haslum, P.; and Thiebaux, S. 2016b. Numeric Planning with Disjunctive Global Constraints via SMT. *Proceedings of the International Conference on Automated Planning and Scheduling*, 26: 276–284.