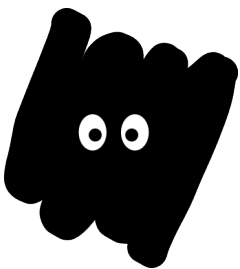
Verifinsta: Verifying If an Instance Belongs to a Domain

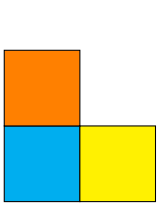
Claudia Grundke Gabriele Röger Malte Helmert

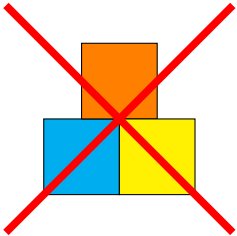
University of Basel

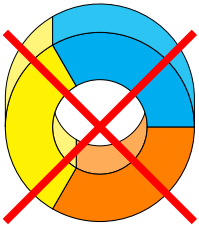


GenPlan Workshop
June 28, 2026

Belongs to Blocksworld Domain?







When Does an Instance Belong to a Domain?

formalised domain: PDDL domain (predicates, constants, actions, axioms) + legality axioms and domain-wide goal

an instance belongs to a formalised domain if

- its initial state is legal and
- its goal matches the domain goal

Grundke, Röger, Helmert. Formal Representations of Classical Planning Domains. ICAPS 2024.

Legality Axioms

Legality axioms define which initial states are legal for a domain.

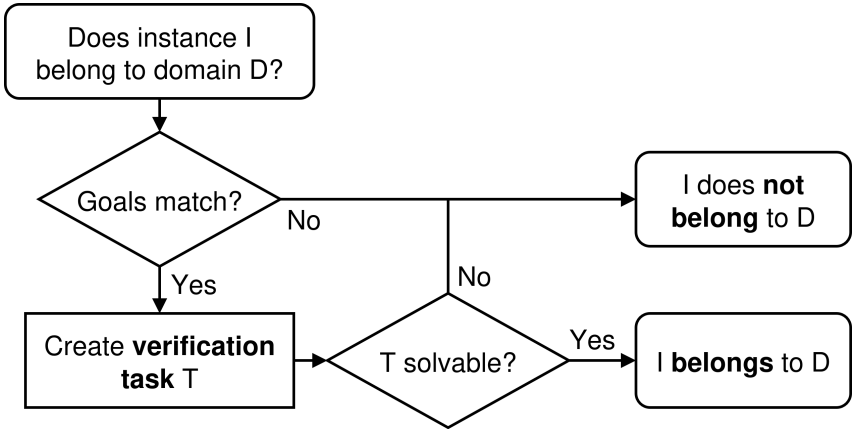
$$\begin{aligned}
\mathit{above}(x, y) &\leftarrow \mathit{on}(x, y) \vee \exists z(\mathit{on}(x, z) \wedge \mathit{above}(z, y)) \\
\mathit{illegal}() &\leftarrow \exists x, y, z(\mathit{on}(x, y) \wedge \mathit{on}(x, z) \wedge y \neq z) \\
\mathit{illegal}() &\leftarrow \exists x\ \mathit{above}(x, x) \\
\mathit{illegal}() &\leftarrow \ldots \\
\mathit{legal}() &\leftarrow \neg \mathit{illegal}()
\end{aligned}$$

PDDL Validity

VERIFINSTA does **not** (yet) check if the input is valid PDDL.

tools like VAL (Fox, Long, Howey, Cresswell. `github.com/KCL-Planning/VAL`)
or the pddl Python package (Favorito, Fuggitti, Muise. `github.com/AI-Planning/pddl`)
can be used for this

VERIFINSTA


Does instance I belong to domain D?
Goals match?
No
I does **not belong** to D
Yes
No
Create **verification task** T
T solvable?
Yes
I **belongs** to D

Verification Task

same as the input task, except

- extend initial state with linear order,
- set goal to *legal*(), and
- remove all actions

The verification task is solvable iff the initial state of the input task is legal.

$\Rightarrow$ Any planner (that supports axioms) can verify legality.

Verification Task

same as the input task, except

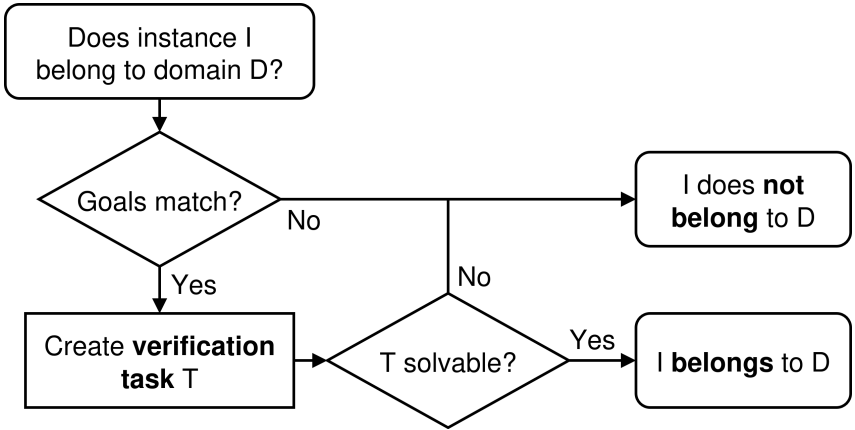
$\texttt{blockA} < \texttt{blockB} < \texttt{blockC} < \ldots$

- extend initial state with linear order,
- set goal to *legal*(), and
- remove all actions

The verification task is solvable iff the initial state of the input task is legal.

$\Rightarrow$ Any planner (that supports axioms) can verify legality.

VERIFINSTA

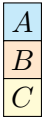

Does instance I belong to domain D?
Goals match?
No
I does **not belong** to D
Yes
No
Create **verification task** T
T solvable?
Yes
I **belongs** to D

STRIPS Goals

STRIPS goals use instance-specific objects and thus cannot be shared between instances. But we can compile them into initial states:

for STRIPS goal $on(A, B) \wedge on(B, C)$

add to initial state: $on^{g}(A, B)$, $on^{g}(B, C)$



goal of all Blocksworld tasks then is
$\forall x, x'(on^{g}(x, x') \rightarrow on(x, x'))$

Evaluating VERIFINSTA

We verified IPC instances against their (formalised) domains:

IPC 2023 learning track instances:

- (partially) hand-crafted base cases
- automatically generated training and test cases

formalised domains[1] replicate the properties applied by the IPC instance generators
Preliminary experiments revealed small mistakes, e.g., swapped parameters.

Fast Downward for solving the verification tasks

[1] Grundke, Helmert, Röger. Domain-Independent Instance Generation for Classical Planning. KR 2025.

Verification Results

VERIFINSTA verified 1573 / 1688 instances within 5 min and 4 GiB

- 1560 legal
- 13 illegal

All illegal instances are hand-crafted base cases that are inconsistent with the IPC instance generators.

Performance Results

VERIFINSTA could not verify 115 / 1688 instances within 5 min and 4 GiB
$\rightsquigarrow$ 56 time-outs + 59 memory-outs while Fast Downward solved the verification tasks

30 min time limit: verified 8 more instances

using VAL for the verification tasks: verified 249 fewer instances

Summary

With VERIFINSTA and any planner that supports axioms
you can verify if an instance belongs to a (formalised) domain.

link to VERIFINSTA

link to the paper