

## Sample Prompt

The prompt is composed of three parts:

1. **Instruction:** A description of the task.
2. **Context Examples (x3):** Example problems with inputs and correct outputs, specifies the problem format. Each of our context examples contains three parts: (1) The list of allowed predicates and their natural language descriptions in the domain (we refer to this as  $\langle \mathcal{F}, N(\mathcal{F}) \rangle$  in the paper). (2) The natural language description of the action, varied according to description class. (3) The PDDL of the action.
3. **Query:** The input for the action we want to generate the PDDL for. Includes both the allowed predicates and natural language input.

The following example prompt is part of the random description class in which a random predicate within the PDDL reference action is described. The final prompt is the concatenation of all three parts.

---

Listing 1: The Instruction. This specific instruction is used in all prompts and was selected after manually testing other similar instruction prompts with no significant changes in results.

---

Given a description of an action in some domain, convert it to Planning Domain Definition Language (PDDL) action. You may only use the allowed predicates provided for each action.

---

---

Listing 2: Context Example 1. The fly-airplane action from the logistics domain.

---

Allowed Predicates:

```
(in-city ?loc - place ?city - city) : a
  place loc in in a city.
(at ?obj - physobj ?loc - place) : a
  physical object obj is at a place loc.
(in ?pkg - package ?veh - vehicle) : a
  package pkg is in a vehicle veh.
```

Input:

The action, "FLY-AIRPLANE" will fly an airplane from one airport to another . After the action, the airplane will be in the new location.

PDDL Action:

```
(:action FLY-AIRPLANE
  :parameters (?airplane - airplane ?loc-
    from - airport ?loc-to - airport)
  :precondition (at ?airplane ?loc-from)
  :effect (and (not (at ?airplane ?loc-
    from)) (at ?airplane ?loc-to))
)
```

---

We note that: (1) All the context examples in this sample are from different domains, but they do not have to be. The actions used for context examples are uniformly sampled from all domains except the domain being used for

---

Listing 3: Context Example 2. The pack-first action from the heavy pack domain.

---

Allowed Predicates:

```
(heavier ?item1 - item ?item2 - item) :
  item1 is heavier than item2.
(packed ?i - item) : item i is packed into
  the box.
(unpacked ?i - item) : item i is unpacked
  from the box.
(nothing-above ?i - item) : nothing is above
  item i in the box.
(box-empty) : the box is empty.
```

Input:

The action, "pack-first" will pack an item into the box . After the action, the item will be packed.

PDDL Action:

```
(:action pack-first
  :parameters (?item - item)
  :precondition (box-empty)
  :effect (and (not (box-empty)) (packed ?
    item) (nothing-above ?item) (not (
    unpacked ?item)))
)
```

---

---

Listing 4: Context Example 3. The Load action from the heavy pack domain.

---

Allowed Predicates:

```
(at ?x - locatable ?y - place) : the
  locatable x is at some place.
(on ?x - crate ?y - surface) : crate x is on
  a surface y.
(in ?x - crate ?y - truck) : crate x is in
  truck y.
(lifting ?x - hoist ?y - crate) : hoist x is
  lifting crate y.
(available ?x - hoist) : hoist x is
  available.
(clear ?x - surface) : surface x is clear.
```

Input:

The action, "Load" will use a hoist to load a crate into a truck at a place if the truck is at the place.

PDDL Action:

```
(:action Load
  :parameters (?x - hoist ?y - crate ?z -
    truck ?p - place)
  :precondition (and (at ?x ?p) (at ?z ?p)
    (lifting ?x ?y))
  :effect (and (not (lifting ?x ?y)) (in ?
    y ?z) (available ?x))
)
```

---

the query. (2) The average prompt length for all description classes was approximately 400 words, well under the context windows of all models experimented with.

---

**Listing 5: The Query**

---

```
Allowed Predicates:
(handempty) : the hand is empty.
(holding ?x - block): block x is held.
(clear ?x - block): block x is clear.
(on ?x - block ?y - block) : block x is on
    block y.
(ontable ?x - block): block x is on the
    table.
```

**Input:**

The action, "put-down" will have the hand put down a block if the block is being held.

PDDL Action:

---

## Sample Responses

Responses are the token sequence generated by the large language model in response to our prompt. We apply simple parenthesis matching based pruning to the model's response and then try and parse it as PDDL. The pruner attempts to match the outermost parenthesis in the response and takes this sub-string of the response to parse as PDDL. Errors from this process result in a syntax error class, specifically either the NoPDDL or PError sub-classes, depending if no PDDL is found or the pruner is unable to match parenthesis respectively. The following seven responses are from the sample prompt provided above:

---

**Listing 6: Pruned StarCoder Response**

---

```
(:action put-down
  :parameters (?x - block)
  :precondition (and (holding ?x))
  :effect (and (handempty) (clear ?x) (not
    (holding ?x)))
)
```

---

---

**Listing 7: Pruned LLaMA 7b Response**

---

```
(:action put-down
  :parameters (?x - block)
  :precondition (and (holding ?x) (
    handempty))
  :effect (and (clear ?x) (handempty))
)
```

---

---

**Listing 8: Pruned LLaMA 7b Chat Response**

---

```
(:action put-down
  :parameters (?block - block)
  :precondition (handempty)
  :effect (and (not (holding ?block)) (
    clear ?block) (ontable ?block))
)
```

---

---

**Listing 9: Pruned LLaMA 13b Chat Response**

---

```
(:action put-down
  :parameters (?x - block)
  :precondition (and (holding ?x))
  :effect (and (not (holding ?x)))
)
```

---

---

**Listing 10: Pruned LLaMA 13b Chat Response**

---

```
(:action put-down
  :parameters (?x - block)
  :precondition (holding ?x)
  :effect (and (not (holding ?x)) (clear ?
    x))
)
```

---

---

**Listing 11: Pruned LLaMA 70b Response**

---

```
(:action put-down
  :parameters (?x - block)
  :precondition (holding ?x)
  :effect (and (not (holding ?x)) (clear ?
    x))
)
```

---

---

**Listing 12: Pruned LLaMA 70b Chat Response**

---

```
(:action put-down
  :parameters (?x - block)
  :precondition (holding ?x)
  :effect (and (not (holding ?x)) (on ?x ?
    table))
)
```

---

## Hyperparameters

We provide additional information on our hyperparameters to assist with reproducibility. The following Hyperparameters were used for all models, including StarCoder:

- Decoding strategy: greedy. We select the most probable token, we found from a manual search that this produced better domains with less syntax and semantic issues.
- Maximum Token Output: 300. We limit the number of new tokens created to 300. This is enough that all reasonable combinations of predicates in all domains can be captured.
- Stop Sequences: "Input:" and "Allowed Predicates:", we stop generating on these sentinel tokens as they signal the PDDL response has likely been completed.