HDDL Parser:

A Realtime Hierarchical Planning Language Validation Toolkit

Mohammad Yousefi, Pascal Bercher

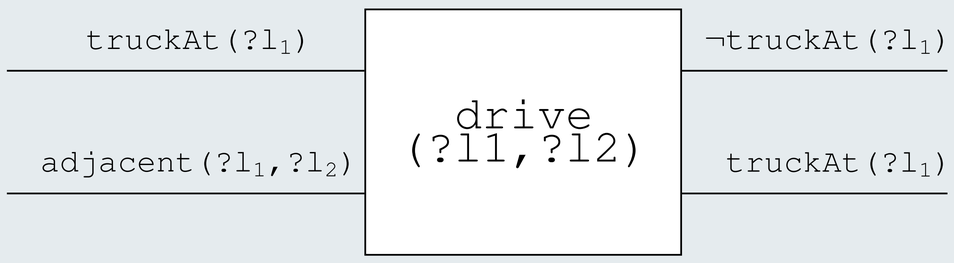
School of Computing, The Australian National University

mohammad.yousefi@anu.edu.au, pascal.bercher@anu.edu.au

Australian National University

What is HDDL?

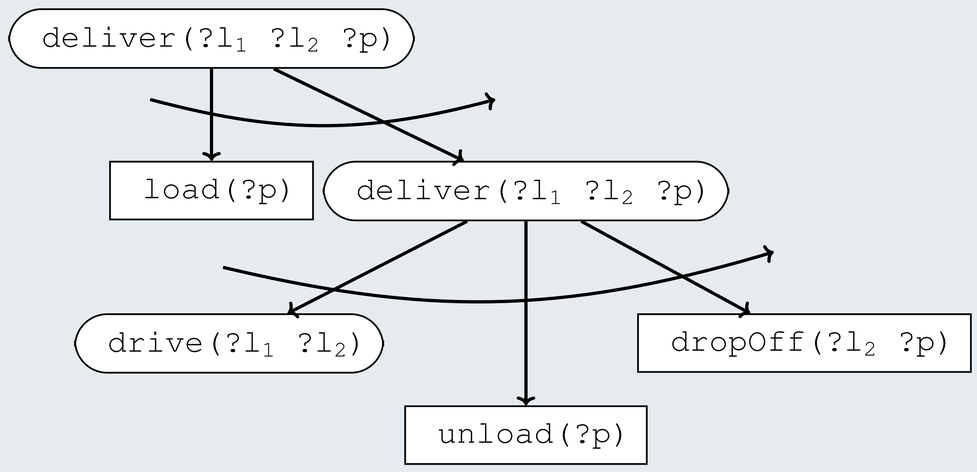
Consider this drive action:



Which we represent in PDDL as:

```
(:action drive
 :parameters (?l1 ?l2 - location)
 :precondition (and (truckAt ?l1)...)
 :effect (and (not (truckAt ?l1))...)
)
```

HDDL adds some grammar magic!



which we represent in HDDL as:

```
(:method m1-deliver
 :parameters (?l1 ?l2 ?p)
 :task ( deliver ?l1 ?l2 ?p)
 :ordered-subtasks (and
    (load ?p)
    (deliver ?l1 ?l2 ?p)
  )
)
```

What is HDDL Parser?

It is a technology that teleports the state of modelling hierarchical planning problems from the good old days of 1990 to 2025!

BEFORE!!!

```
(:predicates
  (on ?x ?y - coord ?s - status)
  (next ?x ?y - coord)
  (max ?x - coord)
  (min ?x - coord)
)

(:method do_play
  :parameters ()
  :tak (play)
  :ordered-subtasks (and
  (turn)
  (play)
  )
)
```

AFTER ^___^

```
(:predicates
  (on ?x ?y - coord ?s - status)
  (next ?x ?y - coord)
  (max ?x - coord)
  (min ?x - coord)
)

(:method do_play
  :parameters ()
  :tak (play)
  :ordered-subtasks (and
  (turn)
  (play)
  )
)
```

35 years of HTN research has culminated in this breakthrough: colored keywords in HDDL!

Detected Errors

HDDL Parser catches a whole bunch of errors, while you are developing a domain! These errors include:

- **Basic Syntax Issues**–deviations from the HDDL grammar
- **Inconsistent Parameter Usage**–type mismatches and incorrect parameter counts in task/predicate usage
- **Undefined Entities**–undefined predicates, types, objects, and tasks
- **Duplicate Definitions**–duplicate definition of tasks, methods, predicates, and constants
- **Cyclic Type Declarations**–cycles in the type hierarchy
- **Undeclared Parameters**–undeclared parameters in tasks or methods.
- **Task Network Issues**–cyclic ordering declaration.
- **Contradictory Formulas**–contradictions in formulae (i.e., preconditions and effects).
- **Unrefinable Tasks**–compound tasks that do not have a primitive refinement.

How Does it Work?