

Lecture 10

CS 6260 Automated Planning and Learning

Temporal Planning: Durative Actions

Department of Computer Science and Engineering

Indian Institute of Technology Madras



Logistics

Announcements

- Office Hours (in SSB 304)
 - Every Tu/We/Th: For one hour after the class starting March 10.
 - Exception This Week:
 - Office Hours for 2 hours on Tuesday (12:00 pm – 02:00 pm)
 - No class and no office hours on We/Th.
- No classes from Feb 23 – Mar 06.
 - Will post video material to watch.
 - Q/A session and Quick Recap on March 10 (and March 11, if needed).

Numeric Planning

:numeric-fluents

```
(:functions
  (time-to-walk ?l1 ?l2 - location)
  (time-to-drive ?l1 ?l2 - location)
  (driven)
  (walked)
)

(:action DRIVE-TRUCK
  :parameters (...)
  :precondition (and ...)
  :effect
    (and (not (at ?truck ?loc-from)) (at ?truck ?loc-to)
          (increase (driven) (time-to-drive ?loc-from ?loc-to))))
)
```

:numeric-fluents

```
(:init
 (at driver1 s2)
 (at driver2 s2)
 (at truck1 s0)
 (empty truck1)
 (at truck2 s0)
 (empty truck2)
 (at package1 s0)
 (at package2 s0)
 (path s1 p1-0)
 (path p1-0 s1)
 (path s0 p1-0)
 (path p1-0 s0)
 (= (time-to-walk s1 p1-0) 43)
 (= (time-to-walk p1-0 s1) 43)
 (= (time-to-walk s0 p1-0) 80)
 (= (time-to-walk p1-0 s0) 80))
```

PDDL 2.1: Durative Actions

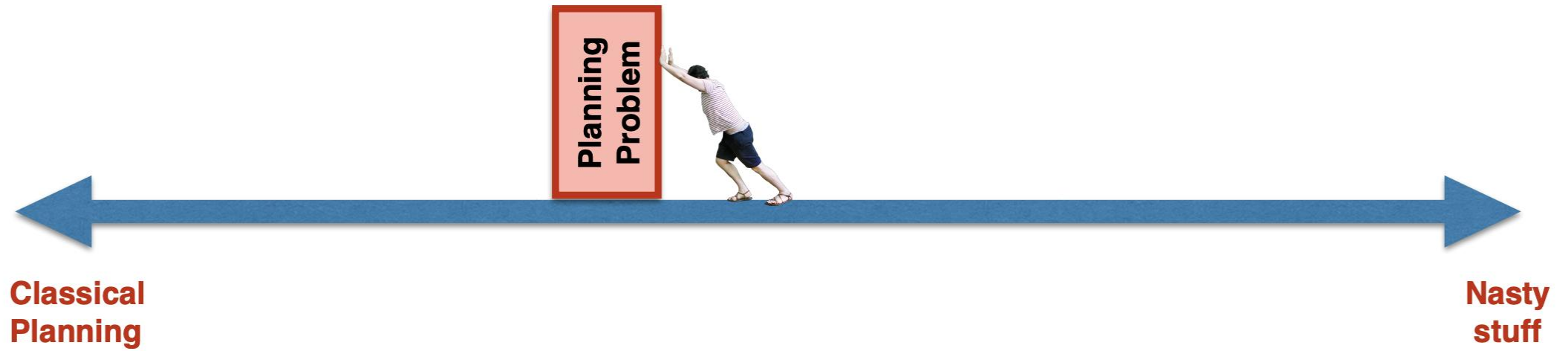
Refer to this document for PDDL 2.1

<https://www.jair.org/index.php/jair/article/view/10352/24759>

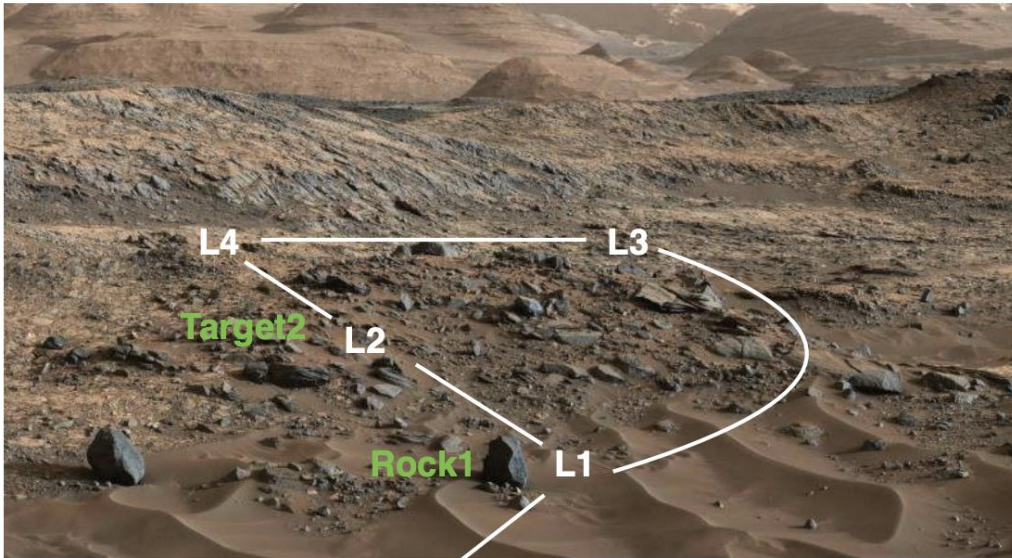
Additional Info: <https://planning.wiki/ref/pddl>

<https://www.cs.cmu.edu/afs/cs/project/jair/pub/volume20/smith03a-html/PDDL-commentary.html>

Temporal Spectrum



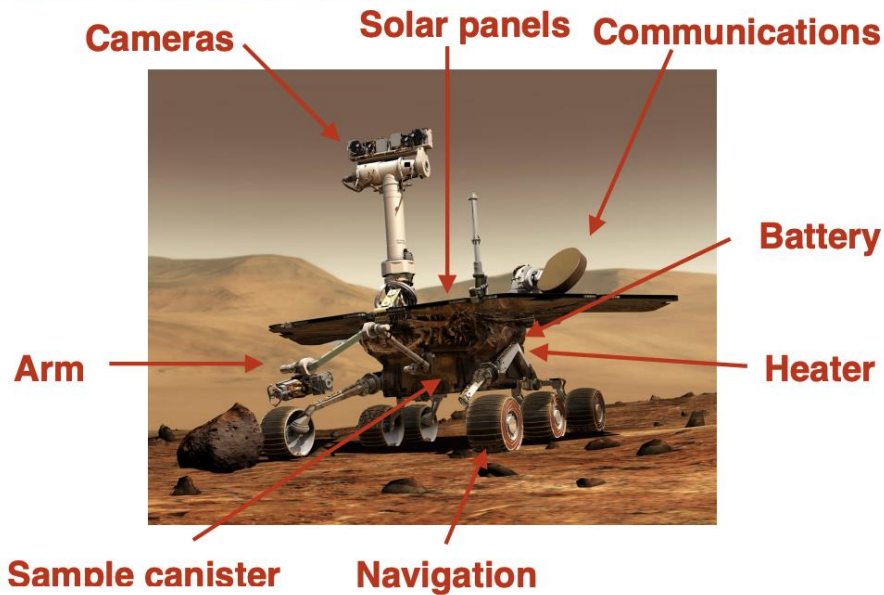
Rover Example



Path(L0,L1)
Visible(Rock1,L1)
Path(L1,L2)
Path(L2,L1)
Reachable(Target2,L2)
Path(L1,L3)
...



L0



At(L0)
Stowed
Empty
Charged

Classical Planning

Initial Conditions

At(L0)
Stowed
Empty
Charged

Path(L0,L1)
Visible(Rock1,L1)
Path(L1,L2)
Path(L2,L1)
Reachable(Target2,L2)
Path(L1,L3)
...

Actions

Pre
Navigate(?loc1, ?loc2)
Eff

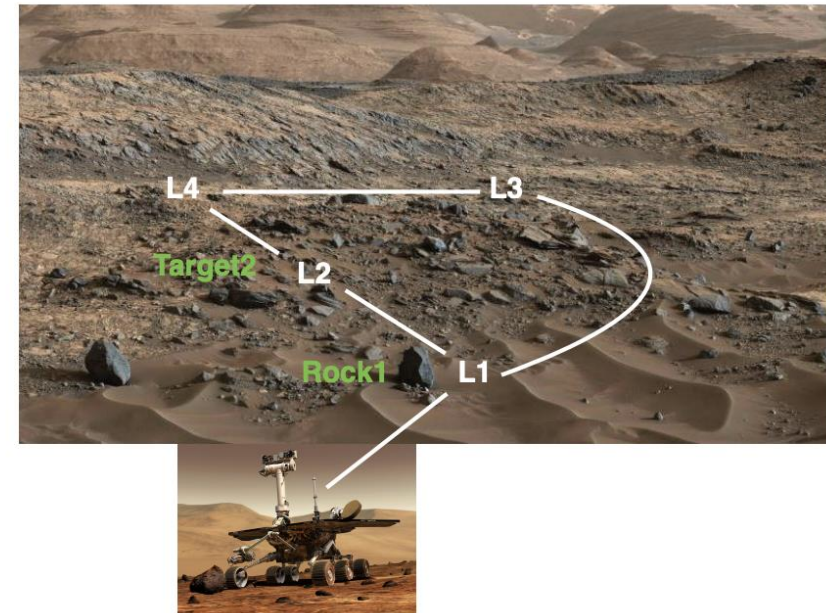
Pre
Snap(?target)
Eff

Pre
PlaceArm(?target)
Eff

Pre
CollectSample()
Eff

Goal(s)

Image(Rock1)
Sample(Target2)
At(L3)



Classical Planning

Initial Conditions

At(L0)
Stowed
Empty
Charged

Path(L0,L1)
Visible(Rock1,L1)
Path(L1,L2)
Path(L2,L1)
Reachable(Target2,L2)
Path(L1,L3)
...

No time constraints

Actions

Pre
| **Navigate(?loc1, ?loc2)**
Eff

Pre
| **Snap(?target)**
Eff

Pre
| **PlaceArm(?target)**
Eff

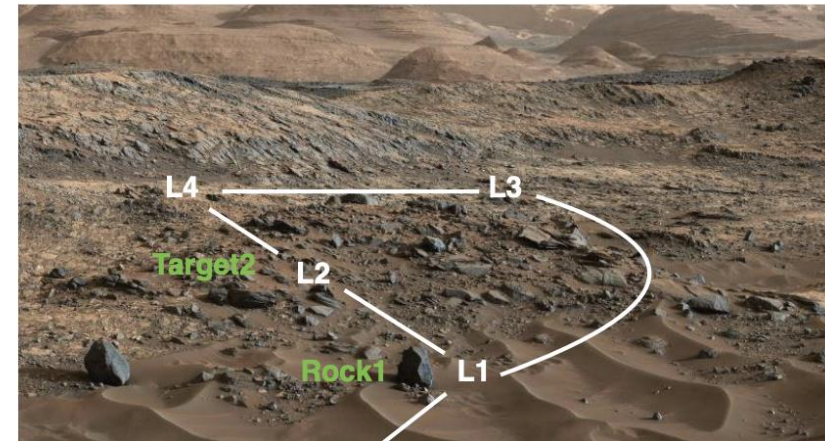
Pre
| **CollectSample()**
Eff

Instantaneous
Instantaneous change
Sequential

Goal(s)

Image(Rock1)
Sample(Target2)
At(L3)

Achievement



Plans:

IC

| | | | | | | |

Goals



Reality

Time constraints

- [0800,1100] Illumination(Rock1)
- [1530,1600] Comm-window
- [0900,1630] Temperature > -20

Actions take time

Continuous change

- location
- energy consumption
- solar recharging

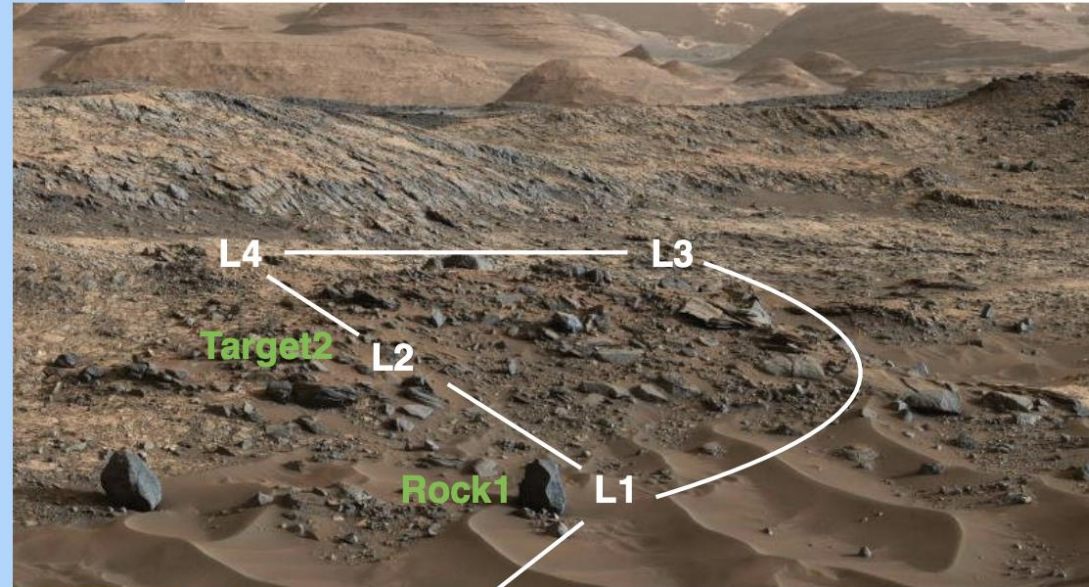
Concurrency

- heating
- sample processing
- communication

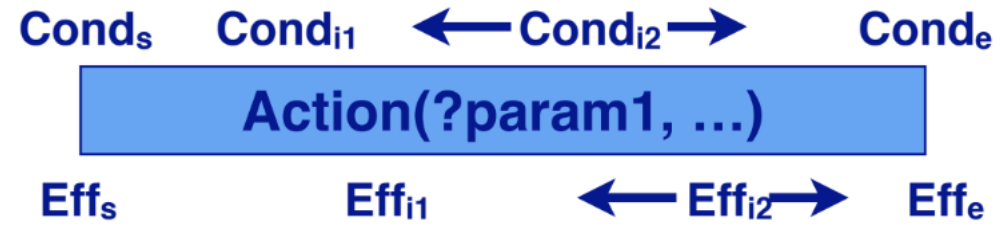
Goal constraints

- Deadlines: [...,1500] At(L3)
- Windows: [0900,1030] Image(Rock1)
- Ordering: Image(Rock1) < Sample(Target2)

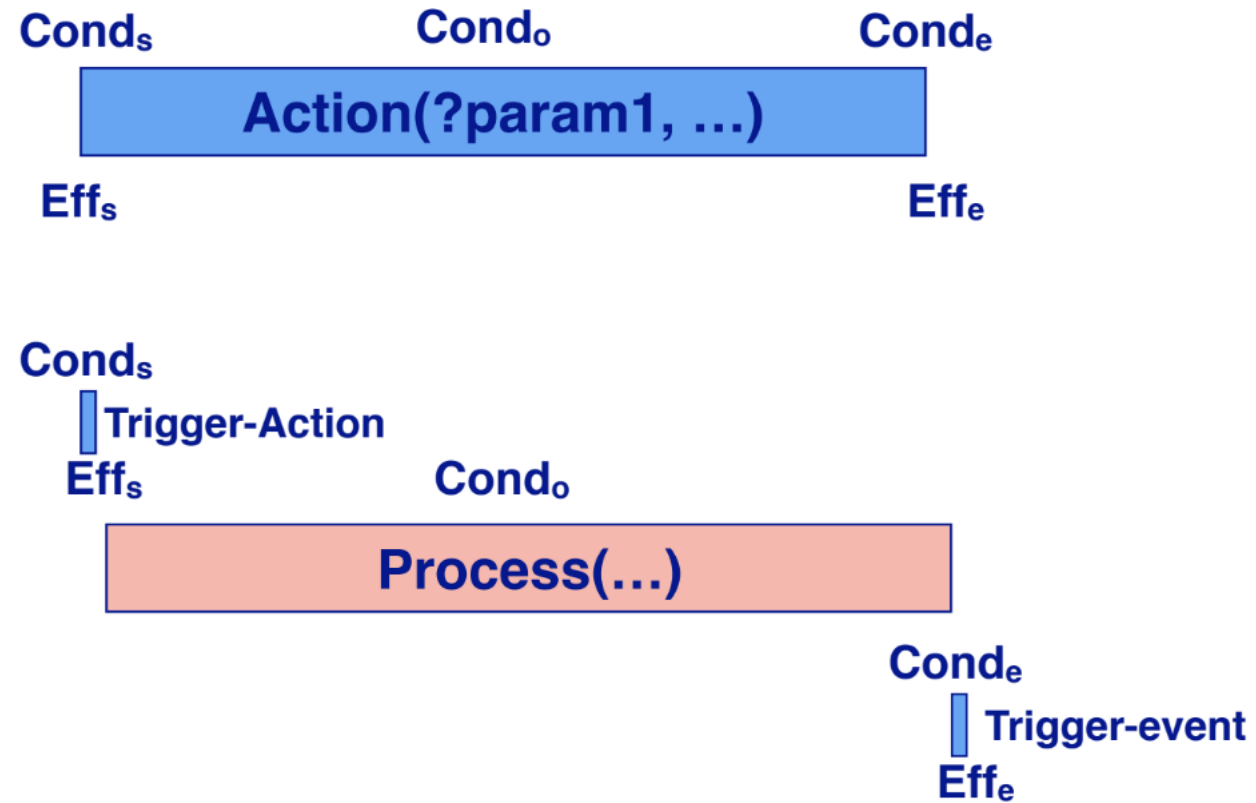
Uncertainty!



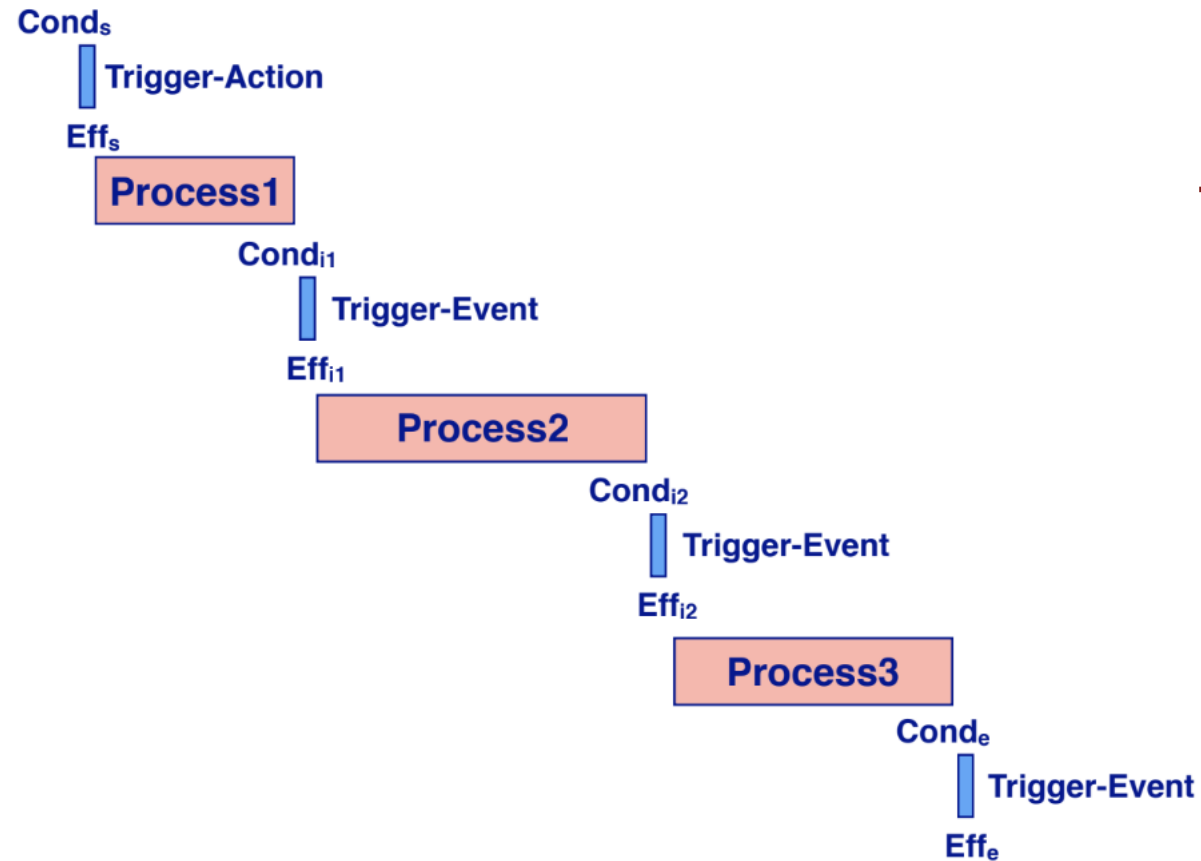
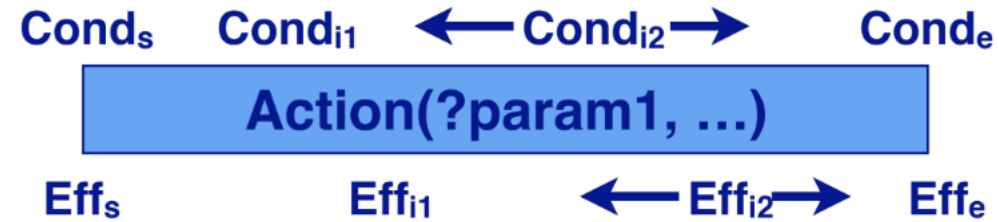
Durative Actions with Discrete Effects



Durative Actions with Discrete Effects

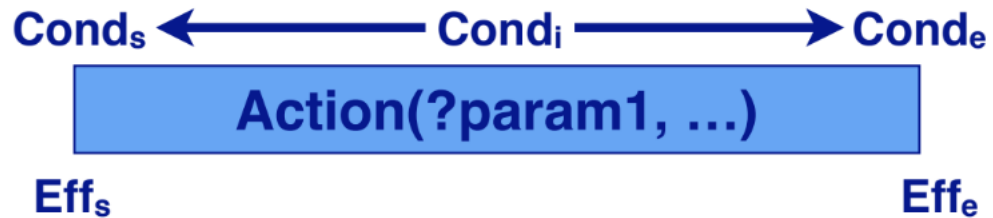
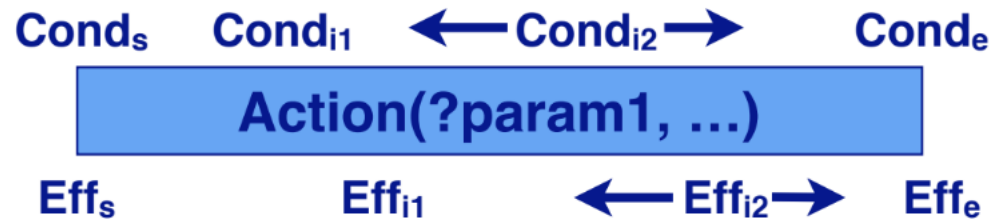


Durative Actions with Discrete Effects



Too detailed!!

Durative Actions with Discrete Effects

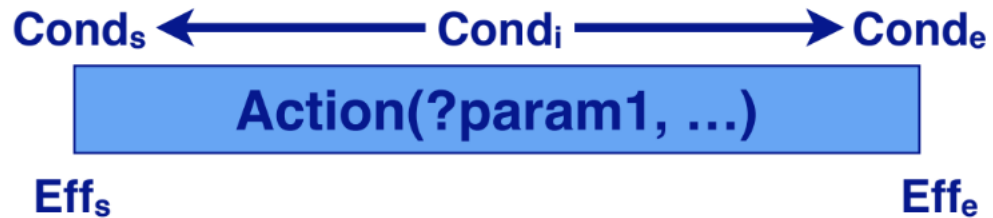
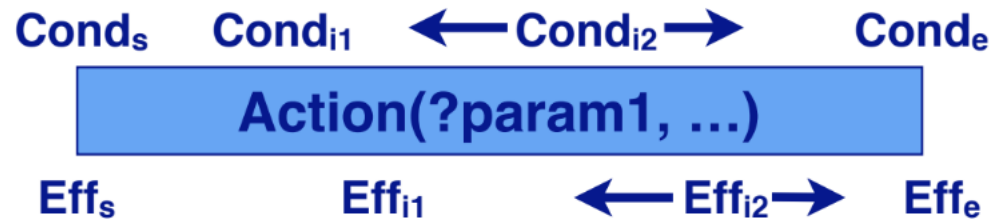


```
(:durative-action move
  :parameters (<arguments>)
  :duration (= ?duration 5)
  :condition (logical_expression)
  :effect (logical_expression)
)
```

:durative-actions

```
(:durative-action move
  :parameters (?r - rover ?fromwp - waypoint ?towp - waypoint)
  :duration (= ?duration 5)
  :condition (and
    (over all (can-move ?from-waypoint ?to-waypoint))
    (at start (at ?rover ?from-waypoint))
    (at start (> (battery-amount ?rover) 8)))
  :effect (and
    (at end (at ?rover ?to-waypoint))
    (at end (been-at ?rover ?to-waypoint))
    (at start (not (at ?rover ?from-waypoint)))
    (at start (decrease (battery-amount ?rover) 8))
    (at end (increase (distance-travelled) 5)))
)
```

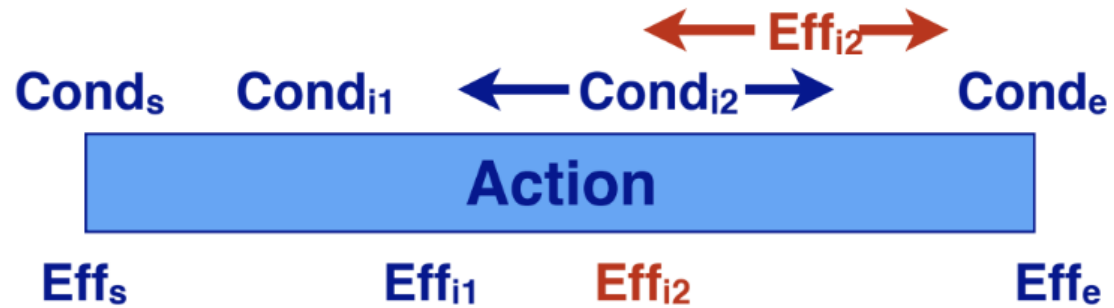
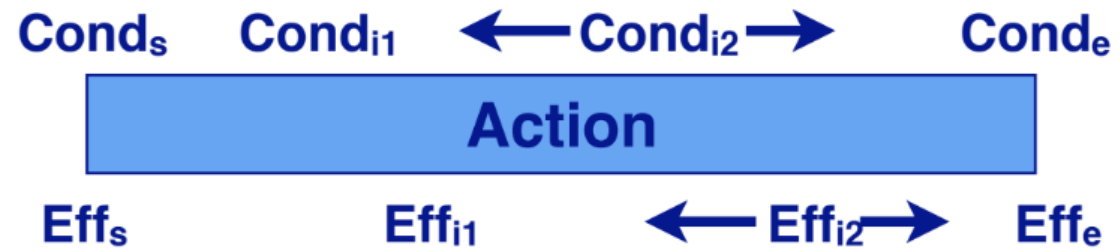
Durative Actions with Discrete Effects



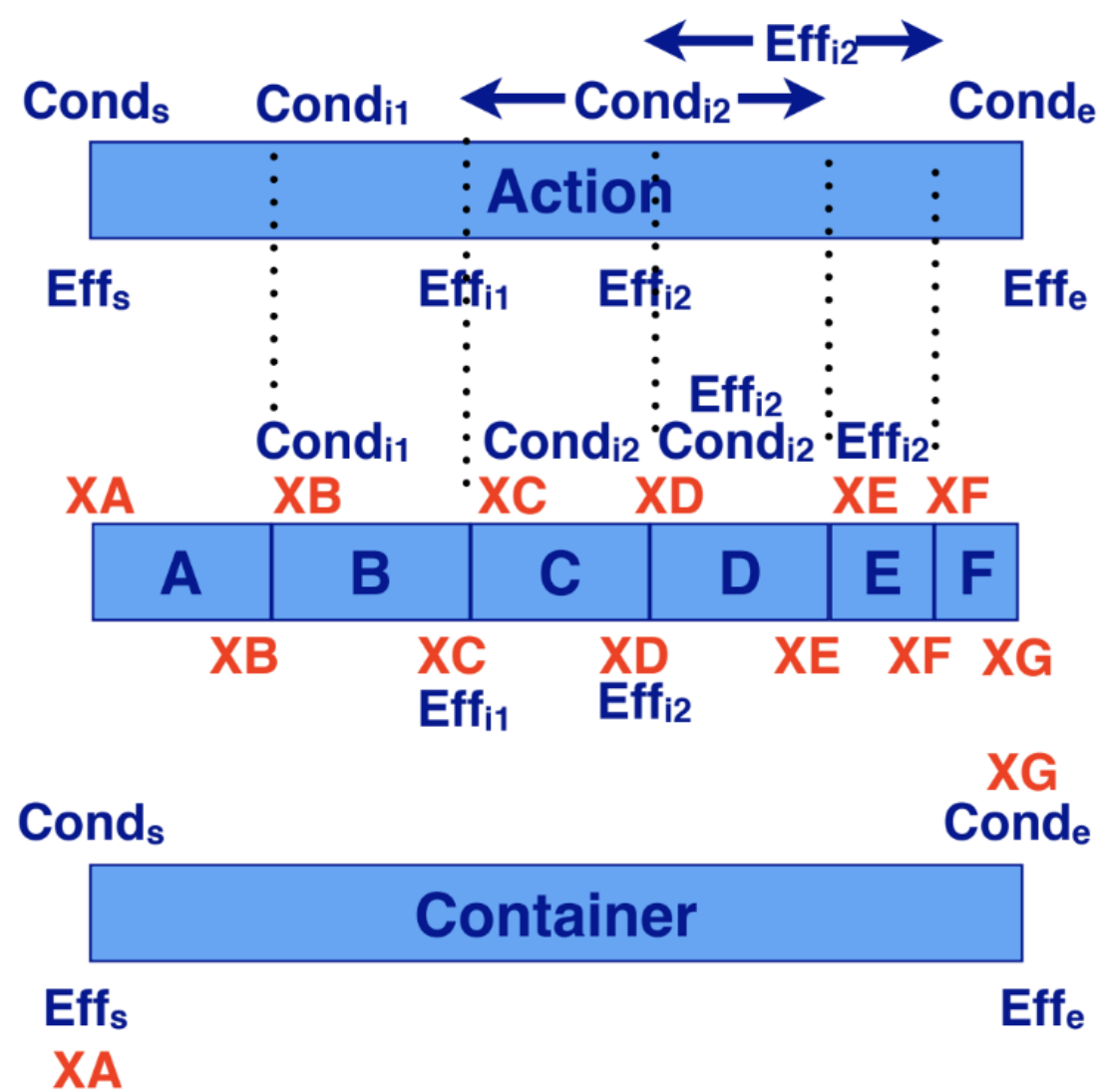
```
(:durative-action move
  :parameters (<arguments>)
  :duration (= ?duration 5)
  :condition (logical_expression)
  :effect (logical_expression)
)
```

Intermediate Conditions and Effects?

Step 1: Eliminate Interval Effects



Step 2: Decomposing Intermediate Conditions and Effects



Simple Durative Planning

Initial Conditions

At(L0)
Stowed
Empty
Charged

Path(L0,L1)
Visible(Rock1,L1)
Path(L1,L2)
Path(L2,L1)
Reachable(Target2,L2)
Path(L1,L3)
...

Actions

Cond

Navigate(?loc1, ?loc2)

Eff

Cond

Snap(?target)

Eff

Cond

PlaceArm(?target)

Eff

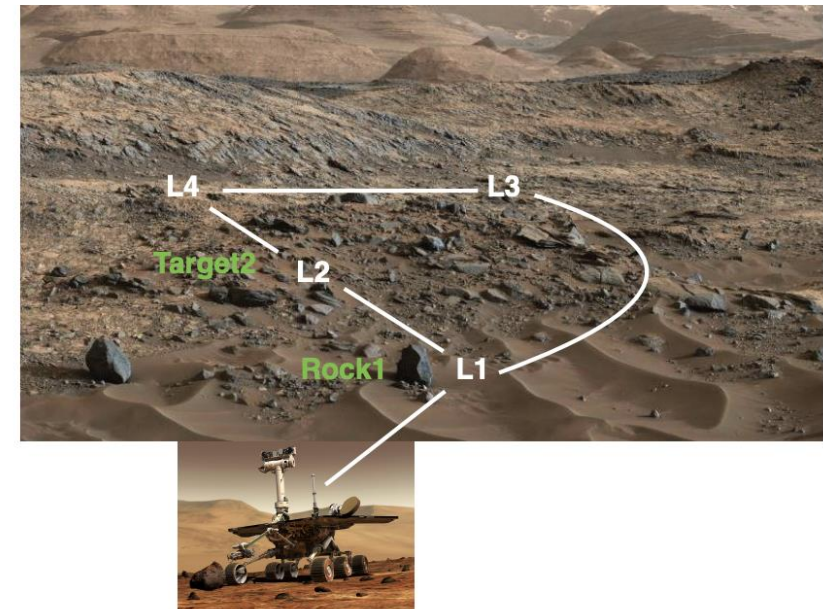
Cond

CollectSample(?target)

Eff

Goal(s)

Image(Rock1)
Sample(Target2)
At(L3)



Simple Durative Actions

Charged
At(?loc1)

Stowed

Navigate(?loc1, ?loc2)

Dur=DriveTime(?loc1, ?loc2)

¬At(?loc1)

At(?loc2)

At(?loc)
Stowed
Viz(?target, ?loc)

Snap(?target)

Dur=1

Image(?target)

Stowed

At(?loc)
Reach(?target, ?loc)

PlaceArm(?target)

Dur=5

¬Stowed

Placed(?target)

Stow()

Dur=1

Stowed

¬Placed(?target)

Empty

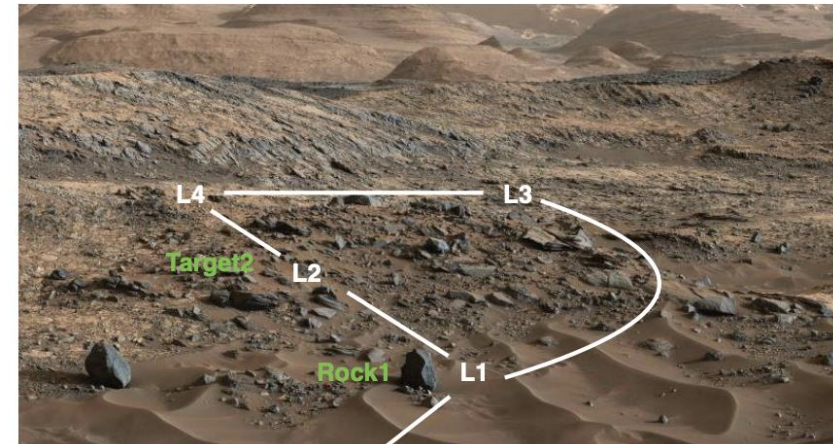
Placed(?target)

CollectSample(?target)

Dur=2

¬Empty

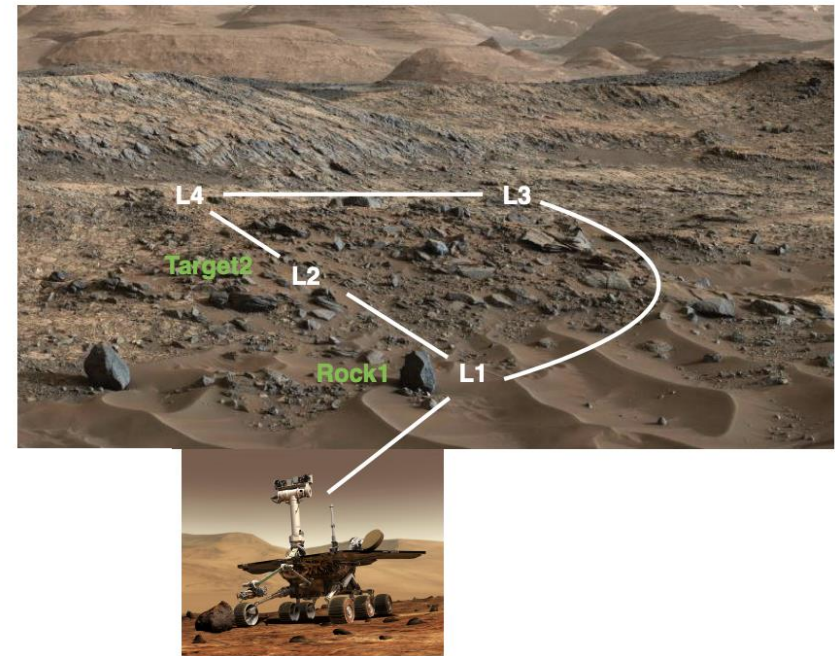
Sample(?target)



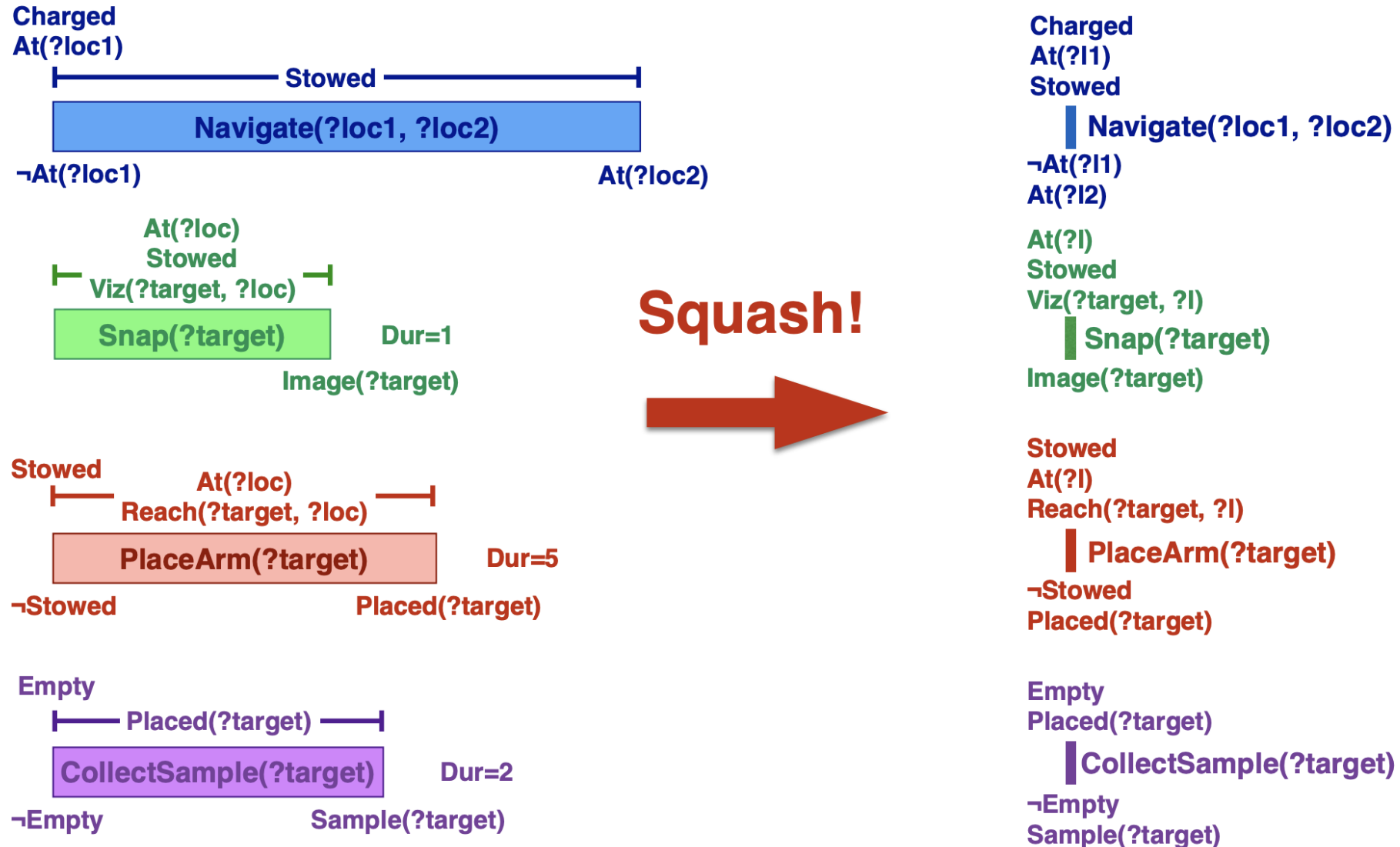
Rover Plan

Goal(s): Image(Rock1), Sample(Target2), At(L3)

At(L0)



Reduction to Classical Planning



Reduction to Classical Planning

Run Classical Planner



Nav(L0,L1)
Snap(Rock1)
Nav(L1,L2)
PlaceArm(T2)
Collect(T2)
Stow()
Nav(L2,L1)
Nav(L1,L3)

Expand

At(L0)



Image(Rock1)

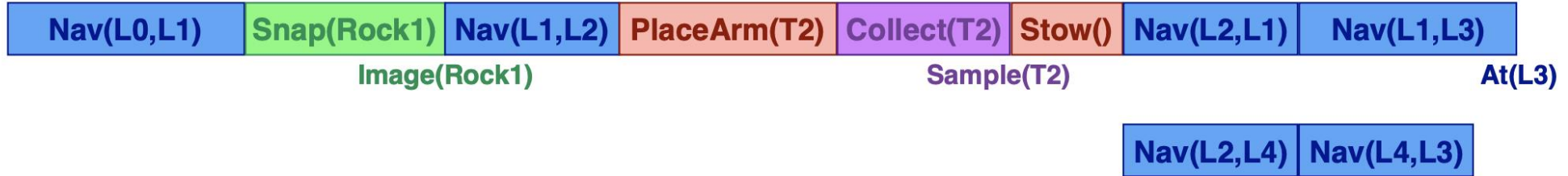
Sample(T2)

At(L3)

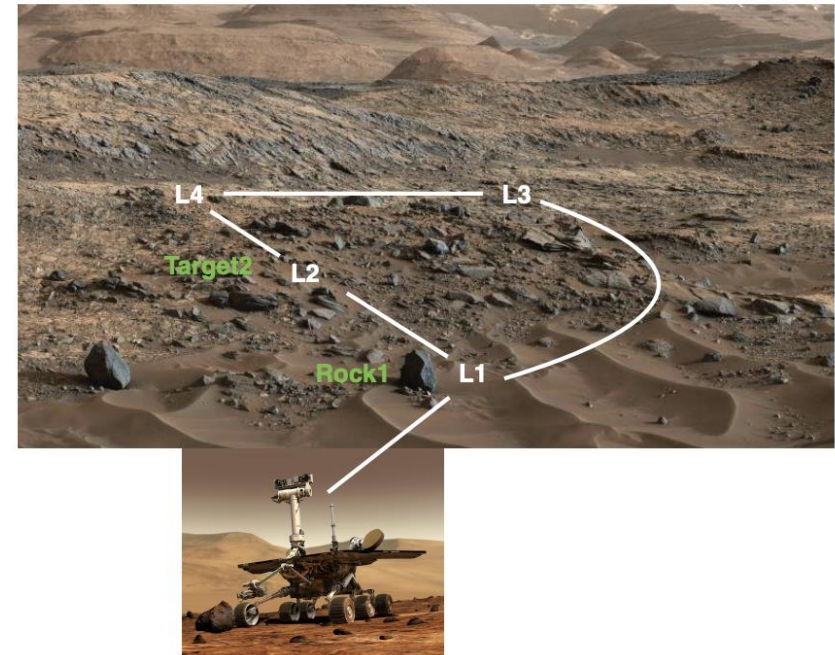
Solution: Minimizing Makespan

Goal(s): Image(Rock1), Sample(Target2), At(Loc3)

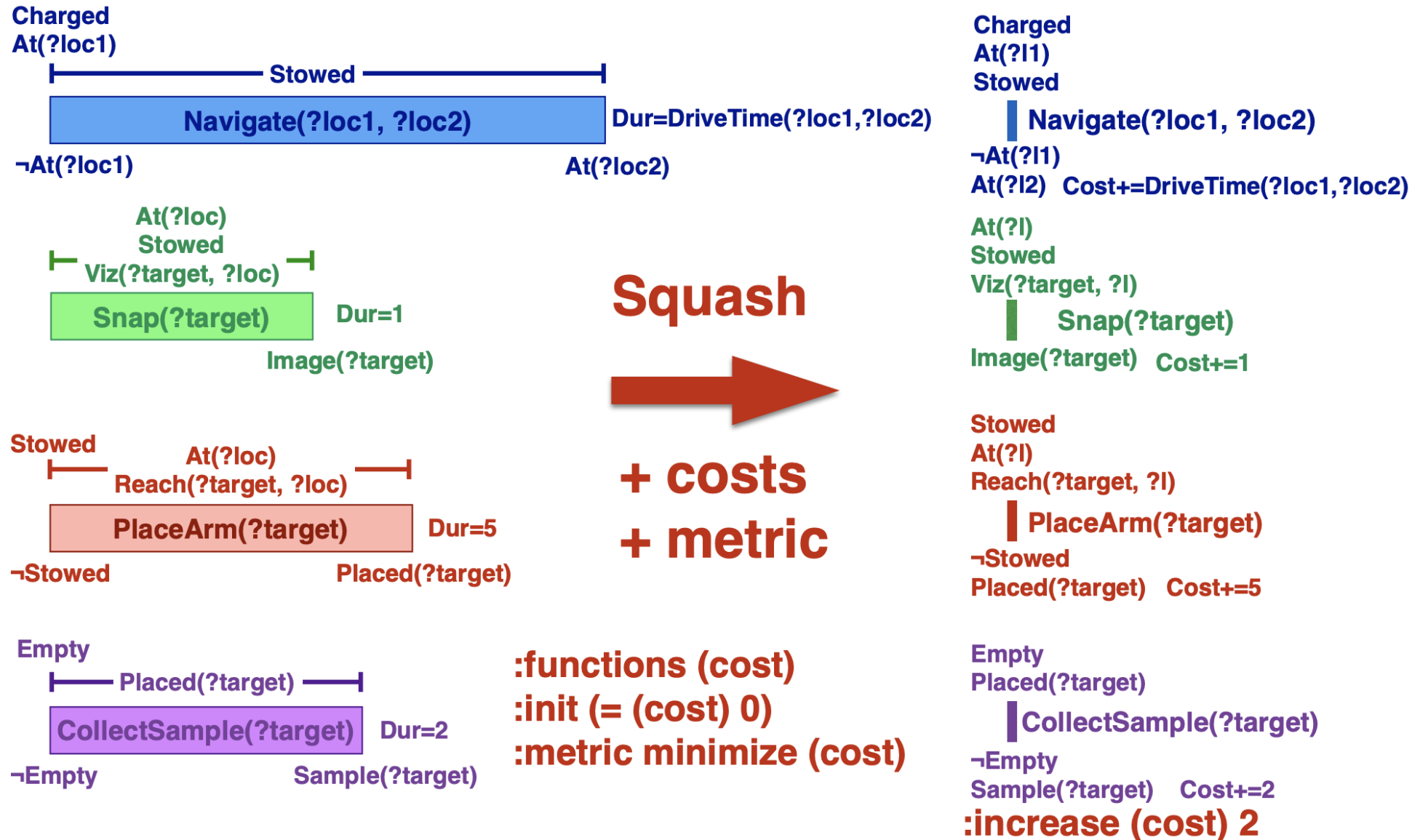
At(L0)



find $\pi : \forall \pi', dur(\pi) \leq dur(\pi')$



Reduction to Optimal Numeric Planning



Reduction to Optimal Numeric Planning

Run Numeric Planner
(minimizing cost)



Nav(L0,L1) Snap(Rock1) Nav(L1,L2) PlaceArm(T2) Collect(T2) Stow() Nav(L2,L4) Nav(L4,L3)

instead of

Nav(L2,L1) Nav(L1,L3)

Expand

At(L0)



Image(Rock1)

Sample(T2)

At(L3)

Numeric planning
– minimize use of a resource (time)

Reality

Time constraints

- [0800,1100] Illumination(Rock1)
- [1530,1600] Comm-window
- [0900,1630] Temperature > -20



Actions take time

ONP

Continuous change

- location
- energy consumption
- solar recharging

Concurrency

- heating
- sample processing
- communication

Goal constraints

- Deadlines: [...,1500] At(L3)
- Windows: [0900,1030] Image(Rock1)
- Ordering: Before(Image(Rock1), Sample(Target2))

Uncertainty!

