PDDL exercise

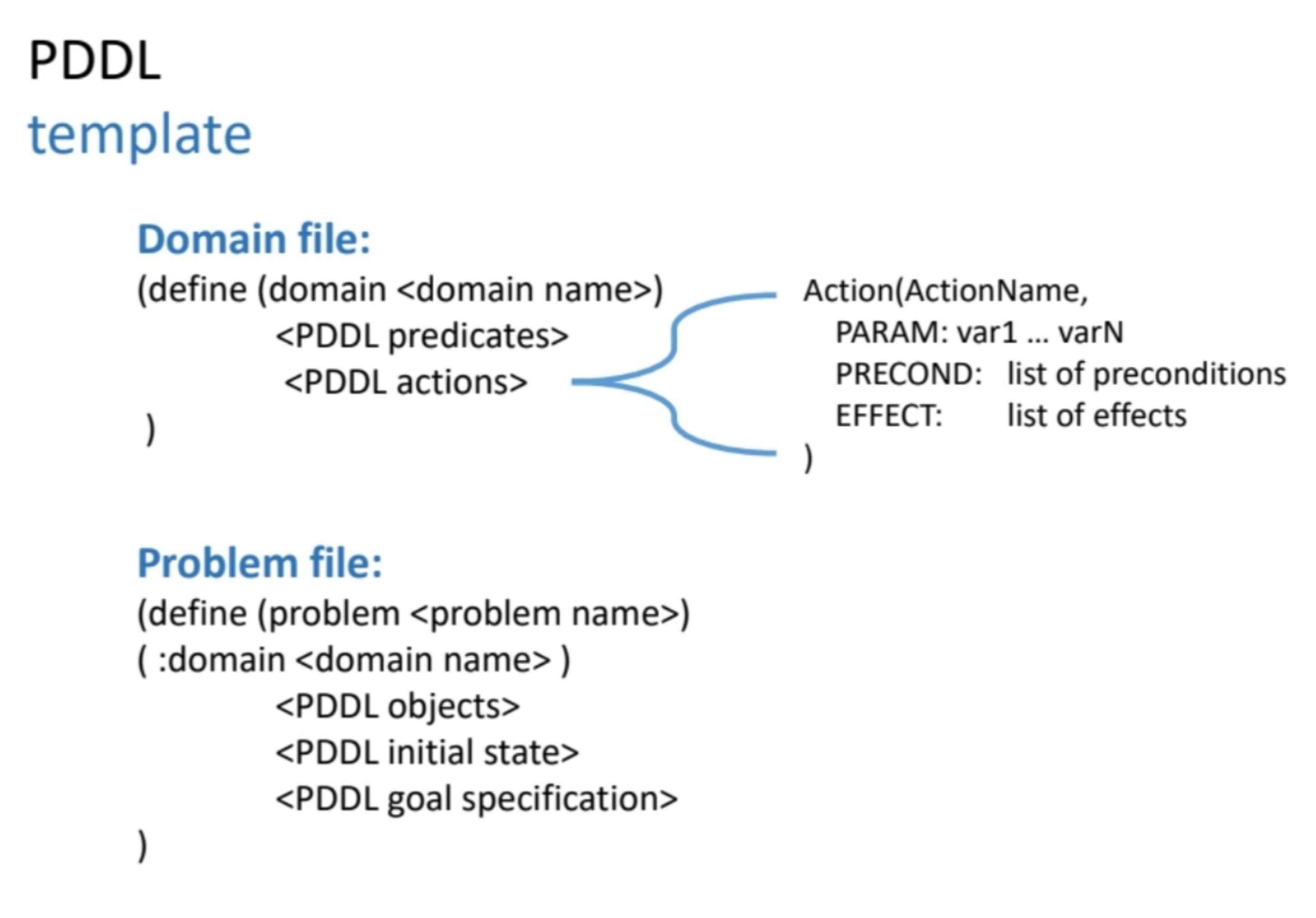
Definire il dominio:

```
(define (domain nome_dominio) //define
    (:requrements :equality) //requrements
    (: predicates  (...)(...)(...) ) //predicates
    (: action  name_action
          :parameters ( ?..  ?..  ?.. )
          :effect (and (...)(...)(...)(not(...))))
          :pre condition ( and (on...) (...) (...) ))//action

    (: action ... ) //action 2
) //domain
```

Definire le instanze

```
define ( problem  name )
    (:domain  name_domain)
    (:objects  a  b  c .. )
    (: init  (action a) (action b)... ) ← STATO INIZIALE
    (: goal  (and  (action a) (action b)... )) ← STATO FINALE
```



PDDL

template

Domain file:

```
(define (domain <domain name>)
            <PDDL predicates>
             <PDDL actions>
 )
```

```
Action(ActionName,
   PARAM: var1 … varN
   PRECOND:   list of preconditions
   EFFECT:       list of effects
)
```

Problem file:

```
(define (problem <problem name>)
( :domain <domain name> )
            <PDDL objects>
            <PDDL initial state>
            <PDDL goal specification>
)
```

Exercises.

1) The monkey and bananas problem

INITIAL STATE: At(x), Level(low), BoxAt(c),
BananasAt(B)

GOAL STATE: Have(bananas)

ACTIONS:

① move to box location

GoBox(x)
PRECONDITION: At(x), Level(low)
EFFECT: notAt(x), At(c)

② climb up on the box

ClimBox(x),
PRECONDITION: At(x), BoxAt(x), Level(low)
EFFECT: Level(high), not Level(low)

③ Move monkey and box to bananas location

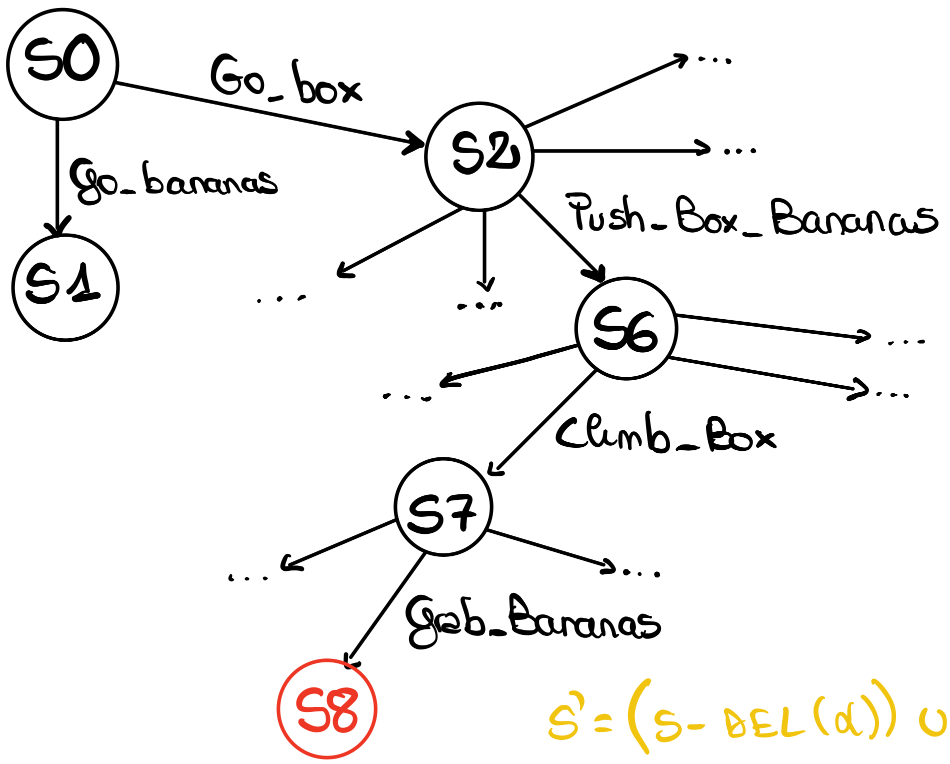
PushBoxBananas(x)
PRECONDITION: At(x), BoxAt(x),
Level(low)
EFFECT: BoxAt(B), not BoxAt(x),
At(B), not At(x)

④ Take the banans

GrabBananas(x)
PRECONDITION: At(x), BananasAt(x),
Level(high)

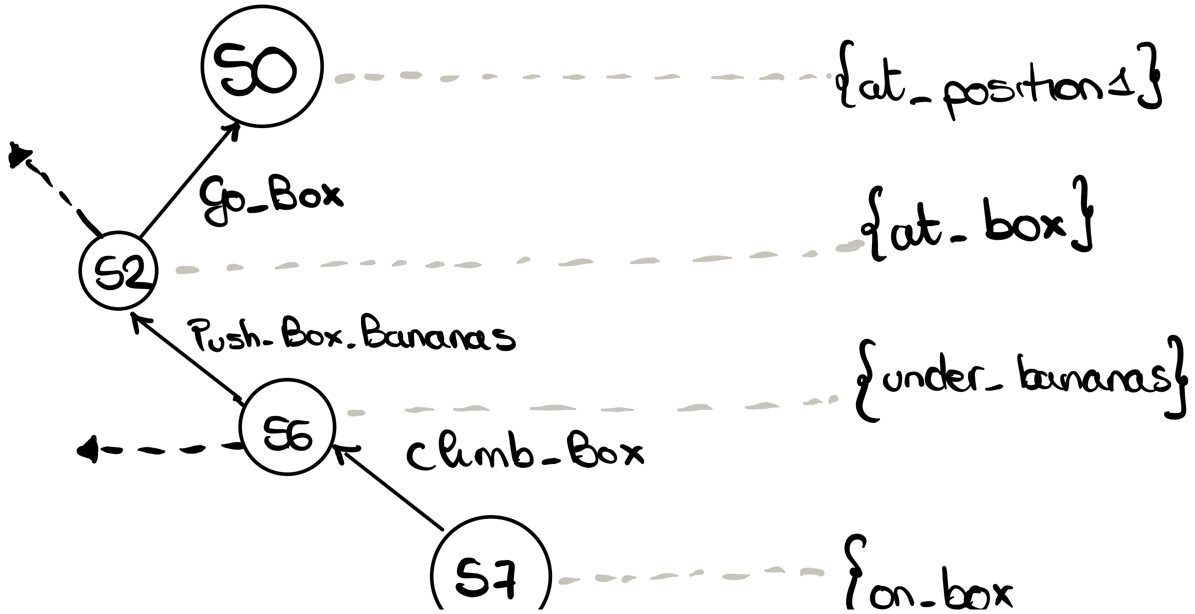
EFFECT: Have(bananas)

Forward search:



$S' = (S - DEL(\alpha)) \cup ADD(\alpha)$

Backward search:



$$g' = (g - ADD(a)) \cup \{PRECOND(a)\}$$

PDDL CODE:

```
define (domain  monkey-bananas )
    (:requirements  :strips   )
    (:predicates  on, box , holding, on-ceiling

    (:action go
      :parameters ( ?who  ?from  ?to)
      :precondition ( At( ?who  ?from))
      :effect ( At( ?who  ?to) , not (At ( ?who  ?from)))
   )

  (:action climb
    :parameters ( ?who  ?what  ?under-what ]
   :precondition ( AND (At( ?who  ?under_what)
                        At( ?what  ?under_what))
   :effect ( On( ?who  ?what))
  )
```

```
(:action push
  :parameters (?who ?what ?from ?to)
  :precondition ( At( ?who ?from) At(?what ?from))
  :effect (AND(At(?who ?to) At(?what ?to)
          not(?what ?from) Not(?who ?from))
 )

(: action grab
  :parameters(?who ?what ?under-what)
  :precondition( AND(At(?who ?under what)
                 On(?what ?under-what)))
  :effect ( Have ( bananas ))
)
) //domain

define ( problem monkey-problem)
  (:domain monkey-bananas)
  (:objects monkey box bananas C1
           C2 C3 )
  (: init At(monkey C1) At (bananas C2)
         At( box C3) not Have(bananas))
  (: goal Have(bananas))
```