Introduction to AI
-Tutorial Planning-

Francesca Toni

STRIPS: rocket domain

1. Formulate the rocket domain below in STRIPS. There are three possible actions:
   - to load a piece of cargo into a rocket
   - to unload a piece of cargo from a rocket
   - to move a rocket from a location to another

   where
   - for a rocket to be moved, it must have fuel, and moving uses up fuel
   - for a piece of cargo to be loaded/unloaded into/from a rocket, cargo and rocket must be at the same location
2. Modify your formulation of the *rocket domain* to accommodate fuel as a resource that can be consumed and produced

STRIPS : rocket domain – example solution to 1.

Move(r,x,y) preconditions: *Rocket(r), At(r,x), Has-fuel(r),* $x \neq y$

effects: *At(r,y), ¬At(r,x), ¬Has-fuel(r)*

Unload(c,r,x) preconditions: *Cargo(c), Rocket(r), In(c,r), At(r,x)*

effects: *¬In(c,r), At(c,x)*

Load(c,r,x) preconditions: *Cargo(c), Rocket(r), At(c,x), At(r,x)*

effects: *¬At(c,x), In(c,r)*

STRIPS : rocket domain – example solution to 2.

- One needs to accommodate **resource constraints** (see Section 11.1.1 in Russel&Norvig for a formal account).
- Intuitively this could be done, assuming every move requires 10l of fuel and each refill adds exactly 10l and the rocket has only up to 20l capacity, by adding a new action Refill and modifying Move as follows

Move(r,x,y) preconditions: *Rocket(r), At(r,x), Fuel(r)>10, x≠y*

effects: *At(r,y), ¬At(r,x), Fuel(r) := Fuel(r) - 10*

Refill (r,x) preconditions: *Rocket(r), At(r,x), Fuel(r)<10*

effects: *Fuel(r) := Fuel(r) + 10*

Graph-plan

Apply GRAPHPLAN to the shopping domain with

actions:

Go(x) *preconditions: At(y), x≠y* *effects: At(x), ¬ At(y)*

Buy(x) *preconditions: At(s),Sells(s,x)* *effects: Have(x)*

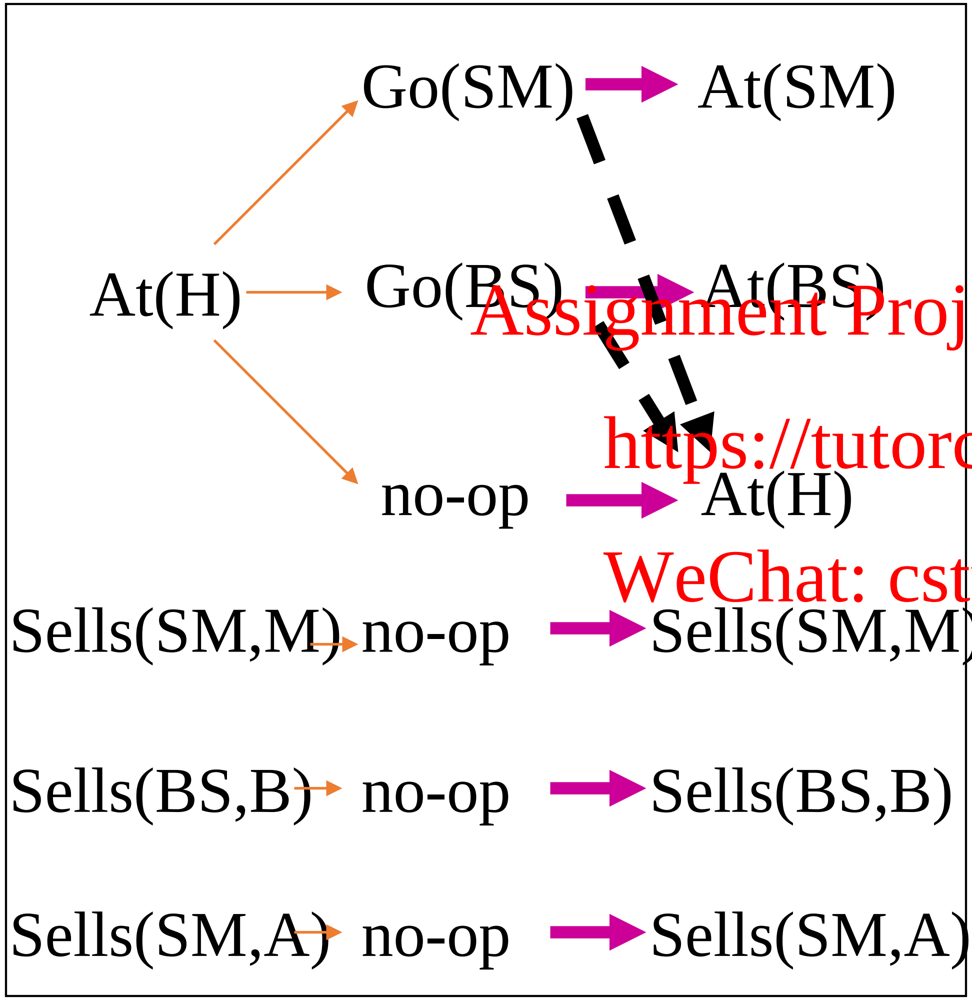
start state: *At(H),Sells(BS,B),Sells(SM,A),Sells(SM,M)*

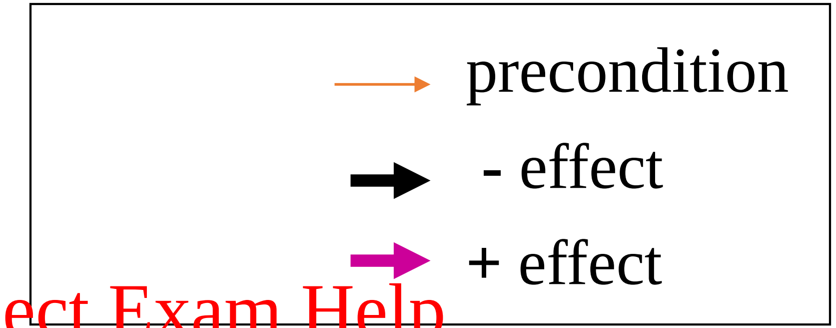
goal: *At(H), Have(M),Have(A),Have(B)*

Show explicitly the graph (including mutex) for S0, A0, S1

Return the computed plan explicitly, and indicate the level at which the plan can be extracted.

Graph-plan solution: S_0, A_0, S_1

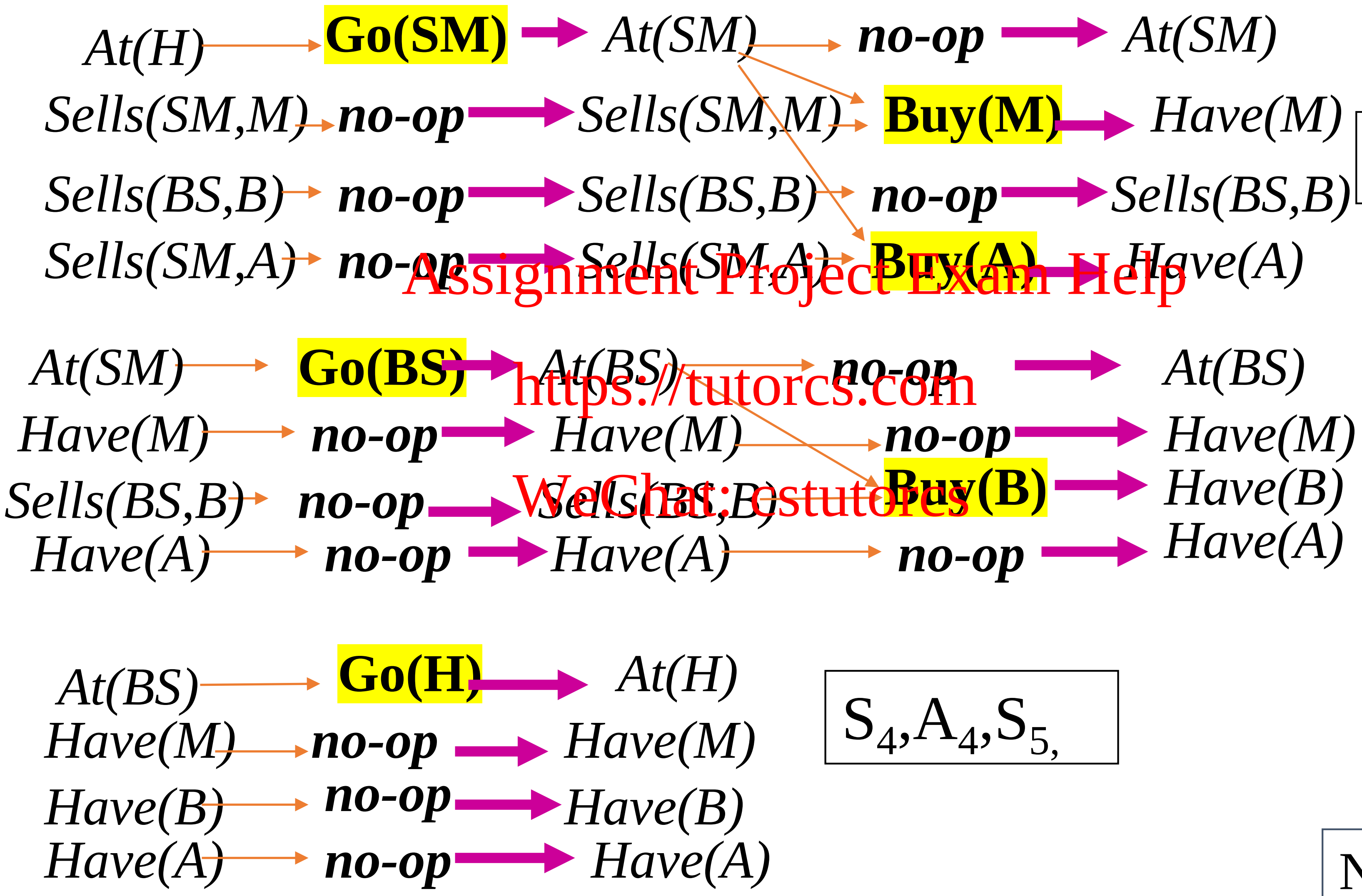




Mutex:

- actions, A_0
  - *Go(SM):* Go(BS) and no-op
  - *Go(BS):* Go(SM) and no-op
  - *no-op:* Go(SM) and Go(BS)
- fluents, S_1
  - *At(SM):* At(BS) and At(H)
  - *At(BS):* At(SM) and At(H)
  - *At(H):* At(BS) and At(SM)

Graph-plan solution: (sub-graph) with plan



S_0, A_0, S_1, A_1, S_2

S_2, A_2, S_3, A_3, S_4

$S_4, A_4, S_5,$

Note parallel actions!