A. Subgoal Generator Prompt

System Prompt

Main agent: agent0
Helper agent: agent1
Generate one clearly stated small independent subgoal for helper agent to help main agent complete the given task. The subgoal should not rely on any main agent actions, and should be executable by the helper agent independently without waiting for any main agent actions. The subgoal should be clearly state with unambiguous terminology. Do not use actions like assist or help. Generate actions that the helper agent can do independently, based on the given steps for completing the task. The main agent should be able to continue working on the remaining task while the helper agent is completing its small subgoal. Do not overtake the full sequence of actions. Remember, the helper agent is only assisting the main agent and acts agnostic to the main agent.

User Prompt and LLM Response

Part 1: Fixed

Example domain scenario:
You have 1 shaker with 3 levels, 3 shot glasses, 3 dispensers for 3 ingredients. The shaker and shot glasses are clean, empty, and on the table. Your left and right hands are empty. The first ingredient of cocktail1 is ingredient3. The second ingredient of cocktail1 is ingredient1. Your goal is to make 1 cocktail. shot1 contains cocktail1.

agent0 takes the following steps to complete the above task:
grasp right shot2
fill-shot shot2 ingredient1 right left dispenser1
pour-shot-to-clean-shaker shot2 ingredient1 shaker1 right l0 l1
clean-shot shot2 ingredient1 right left
fill-shot shot2 ingredient3 right left dispenser3
grasp left shaker1
pour-shot-to-used-shaker shot2 ingredient3 shaker1 right l1 l2
leave right shot2
shake cocktail1 ingredient3 ingredient1 shaker1 left right
pour-shaker-to-shot cocktail1 shot1 left shaker1 l2 l1

Now we have a new problem defined in this domain for which we don't have access to the signle agent plan:
You have 1 shaker with 3 levels, 3 shot glasses, 3 dispensers for 3 ingredients. The shaker and shot glasses are clean, empty, and on the table. Your left and right hands are empty. The first ingredient of cocktail1 is ingredient3. The second ingredient of cocktail1 is ingredient1. The first ingredient of cocktail2 is ingredient1. The second ingredient of cocktail2 is ingredient2. Your goal is to make 2 cocktails. shot1 contains cocktail1. shot2 contains cocktail2.

A possible agent1 subgoal looking at how the domain works based on the plan example provided for another task in this domain could be -
agent1 subgoals: It can help in filling ingredient1 in a shot glass, then pour it in shaker1, while agent0 prepares other cocktail ingredients using other objects. In this way, agent1 would not need to wait for agent0 and it can complete its goal independently. agent1 should also release all objects that the main agent might need for its own actions. Therefore, agent1's clearly stated (with object names) complete and final goal condition is: shaker1 contains ingredient1 and all hands are empty.

Part 2: Domain specific (`BLOCKSWORLD`)

Current domain scenario:
You have 5 blocks. b2 is on top of b5. b5 is on top of b1. b1 is on top of b4. b3 is on top of b2. b4 is on the table. b3 is clear. Your arm is empty. Your goal is to move the blocks. b4 should be on top of b3.

agent0 takes the following steps to complete the above task:
unstack b3 b2
putdown b3
unstack b2 b5
putdown b2
unstack b5 b1
putdown b5
unstack b1 b4
putdown b1
pickup b4
stack b4 b3

Now we have a new problem defined in this domain for which we don't have access to the signle agent plan:
You have 3 blocks. b2 is on top of b3. b3 is on top of b1. b1 is on the table. b2 is clear. Your arm is empty. Your goal is to move the blocks. b3 should be on top of b2. b1 should be on top of b3.

A possible agent1 subgoal looking at how the domain works based on the plan example provided for another task in this domain could be -
agent1 subgoals:

LLM response - English Subgoal (`BLOCKSWORLD`)

It can help in unstacking block b2 from b3 and putting it down on the table, while agent0 works on rearranging the other blocks. In this way, agent1 would not need to wait for agent0 and it can complete its goal independently. agent1 should also release all objects that the main agent might need for its own actions. Therefore, agent1's clearly stated (with object names) complete and final goal condition is: b2 is on the table and the arm is empty.

Part 2: Domain specific (TERMES)

Current domain scenario:
The robot is on a grid with 3 rows and 3 columns.
pos-0-0 pos-0-1 pos-0-2
pos-1-0 pos-1-1 pos-1-2
pos-2-0 pos-2-1 pos-2-2
The robot is at pos-2-0.
The depot for new blocks is at pos-2-0. The maximum height of blocks is 2. Your goal is to build blocks so that the height at pos-1-1 is 2. You cannot have an unplaced block at the end.

agent0 takes the following steps to complete the above task:
create-block pos-2-0
place-block pos-2-0 pos-1-0 n0 n1
create-block pos-2-0
move pos-2-0 pos-2-1 n0
place-block pos-2-1 pos-1-1 n0 n1
move pos-2-1 pos-2-0 n0
create-block pos-2-0
move-up pos-2-0 n0 pos-1-0 n1
place-block pos-1-0 pos-1-1 n1 n2
move-down pos-1-0 n1 pos-2-0 n0
remove-block pos-2-0 pos-1-0 n1 n0
destroy-block pos-2-0

Now we have a new problem defined in this domain for which we don't have access to the signle agent plan:
The robot is on a grid with 4 rows and 3 columns.
pos-0-0 pos-0-1 pos-0-2
pos-1-0 pos-1-1 pos-1-2
pos-2-0 pos-2-1 pos-2-2
pos-3-0 pos-3-1 pos-3-2
The robot is at pos-2-0.
The depot for new blocks is at pos-2-0. The maximum height of blocks is 3. Your goal is to build blocks so that the height at pos-1-2 is 3. You cannot have an unplaced block at the end.

A possible agent1 subgoal looking at how the domain works based on the plan example provided for another task in this domain could be -
agent1 subgoals:

LLM response - English Subgoal (TERMES)

It can help in creating a block at the depot pos-2-0, then move it to pos-1-2, while agent0 prepares other blocks using other positions. In this way, agent1 would not need to wait for agent0 and it can complete its goal independently. agent1 should also ensure that there are no unplaced blocks at the end. Therefore, agent1's clearly stated (with object names) complete and final goal condition is: pos-1-2 has a block of height n1 and all unplaced blocks are destroyed.

Part 2: Domain specific (TYREWORLD)

Current domain scenario:
You have a jack, a pump, a wrench, a boot, 2 hubs, 2 nuts, 2 flat tyres, and 2 intact tyres. The jack, pump, wrench, and intact tyres are in the boot. The boot is unlocked but is closed. The intact tyres are not inflated. The flat tyres are on the hubs. The hubs are on the ground. The nuts are tight on the hubs. The hubs are fastened. Your goal is to replace flat tyres with intact tyres on the hubs. Intact tyres should be inflated. The nuts should be tight on the hubs. The flat tyres, wrench, jack, and pump should be in the boot. The boot should be closed.

agent0 takes the following steps to complete the above task:
loosen nuts2 the-hub2
jack-up the-hub2
undo nuts2 the-hub2
remove-wheel w2 the-hub2
loosen nuts1 the-hub1
jack-up the-hub1
undo nuts1 the-hub1
remove-wheel w1 the-hub1
inflate r2
inflate r1
open boot
fetch r1 boot
put-on-wheel r1 the-hub1
do-up nuts1 the-hub1
jack-down the-hub1
tighten nuts1 the-hub1
fetch r2 boot
put-on-wheel r2 the-hub2
do-up nuts2 the-hub2
jack-down the-hub2
tighten nuts2 the-hub2
put-away w1 boot
put-away w2 boot
close boot

Now we have a new problem defined in this domain for which we don't have access to the signle agent plan:
You have a jack, a pump, a wrench, a boot, 1 hubs, 1 nuts, 1 flat tyres, and 1 intact tyres. The jack, pump, wrench, and intact tyres are in the boot. The boot is unlocked but is closed. The intact tyres are not inflated. The flat tyres are on the hubs. The hubs are on the ground. The nuts are tight on the hubs. The hubs are fastened. Your goal is to replace flat tyres with intact tyres on the hubs. Intact tyres should be inflated. The nuts should be tight on the hubs. The flat tyres, wrench, jack, and pump should be in the boot. The boot should be closed.

A possible agent1 subgoal looking at how the domain works based on the plan example provided for another task in this domain could be -
agent1 subgoals:

LLM response - English Subgoal (TYREWORLD)

It can help in inflating the intact tyre while agent0 works on removing the flat tyre from the hub. In this way, agent1 would not need to wait for agent0 and it can complete its goal independently. agent1 should also release all objects that the main agent might need for its own actions. Therefore, agent1's clearly stated (with object names) complete and final goal condition is: the intact tyre is inflated and all hands are empty.

Part 2: Domain specific (GRIPPERS)

Current domain scenario:
You control 1 robot, each robot has a left gripper and a right gripper. There are 4 rooms and 6 balls. robot1 is in room3. ball1 is in room3. ball2 is in room1. ball3 is in room3. ball4 is in room2. ball5 is in room4. ball6 is in room4. The robots' grippers are free. Your goal is to transport the balls to their destinations. ball1 should be in room4. ball2 should be in room1. ball3 should be in room1. ball4 should be in room2. ball5 should be in room1. ball6 should be in room1.

agent0 takes the following steps to complete the above task:
pick robot1 ball1 room3 lgripper2
move robot1 room3 room1
move robot1 room1 room4
pick robot1 ball5 room4 rgripper2
drop robot1 ball1 room4 lgripper2
pick robot1 ball6 room4 lgripper2
move robot1 room4 room1
drop robot1 ball6 room1 lgripper2
drop robot1 ball5 room1 rgripper2
move robot1 room1 room3
pick robot1 ball3 room3 lgripper2
move robot1 room3 room1
drop robot1 ball3 room1 lgripper2

Now we have a new problem defined in this domain for which we don't have access to the signle agent plan:
You control 1 robots, each robot has a left gripper and a right gripper. There are 2 rooms and 2 balls. robot1 is in room1. ball2 is in room1. ball1 is in room1. The robots' grippers are free. Your goal is to transport the balls to their destinations. ball1 should be in room2. ball2 should be in room2.

A possible agent1 subgoal looking at how the domain works based on the plan example provided for another task in this domain could be -
agent1 subgoals:

LLM response - English Subgoal (GRIPPERS)

It can help in picking up ball1 in room1 and transporting it to room2. In this way, agent1 would not need to wait for agent0 and it can complete its goal independently. agent1 should also release all objects that the main agent might need for its own actions. Therefore, agent1's clearly stated (with object names) complete and final goal condition is: ball1 is in room2 and all grippers are free.

B. Subgoal Translator Prompt

System Prompt

I want you to solve planning problems. Provide me with the PDDL goal that describes the new planning goal directly without further explanations. Make sure to provide only non-conflicting, necessary, and final goal conditions mentioned in the given goal.

User Prompt and LLM Response

Domain and Problem specific (`BLOCKSWORLD`)

The PDDL problem and its initial conditions are given as:
(define (problem BW-rand-3)
(:domain blocksworld-4ops)
(:objects b1 b2 b3)
(:init
(arm-empty)
(on-table b1)
(on b2 b3)
(on b3 b1)
(clear b2)
)

An example planning goal for this problem:
Your goal is to move the blocks. b3 should be on top of b2. b1 should be on top of b3.

The PDDL goal for the example planning goal:
(:goal
(and
(on b3 b2)
(on b1 b3))
)

New planning goal for the same problem:
Your goal is: b2 is on the table and the arm is empty.

The PDDL goal for the new planning goal:

LLM response - PDDL Subgoal (`BLOCKSWORLD`)

(:goal
(and
(on-table b2)
(arm-empty))
)

Domain and Problem specific (`TERMES`)

The PDDL problem and its initial conditions are given as:
(define (problem termes-00038-0036-4x3x3-random_towers_4x3_3_1_3)
(:domain termes)
; termes-00038-0036-4x3x3-random_towers_4x3_3_1_3
; Initial state:
; 0 0 R0D 0
; 0 0 0 0
; 0 0 0 0
; Goal state:
; 0 0 0 0
; 0 0 0 0
; 0 3 0 0
; Maximal height: 3
(:objects
n0 - numb
n1 - numb
n2 - numb
n3 - numb
pos-0-0 - position
pos-0-1 - position
....
pos-3-0 - position
pos-3-1 - position
pos-3-2 - position
)
(:init
(height pos-0-0 n0)
.....
(height pos-3-0 n0)
(height pos-3-1 n0)
(height pos-3-2 n0)
(at pos-2-0)
(SUCC n1 n0)
(SUCC n2 n1)
(SUCC n3 n2)
(NEIGHBOR pos-0-0 pos-1-0)
(NEIGHBOR pos-0-0 pos-0-1)
(NEIGHBOR pos-0-1 pos-1-1)
(NEIGHBOR pos-0-1 pos-0-0)
(NEIGHBOR pos-0-1 pos-0-2)
(NEIGHBOR pos-0-2 pos-1-2)
.....
(NEIGHBOR pos-3-2 pos-2-2)
(NEIGHBOR pos-3-2 pos-3-1)
(IS-DEPOT pos-2-0)
)

An example planning goal for this problem:
Your goal is to build blocks so that the height at pos-1-2 is 3. You cannot have an unplaced block at the end.

The PDDL goal for the example planning goal:
(:goal
(and
(height pos-0-0 n0)
(height pos-0-1 n0)
(height pos-0-2 n0)
(height pos-1-0 n0)
(height pos-1-1 n0)
(height pos-1-2 n3)
(height pos-2-0 n0)
(height pos-2-1 n0)
(height pos-2-2 n0)
(height pos-3-0 n0)
(height pos-3-1 n0)
(height pos-3-2 n0)
(not (has-block))
))

New planning goal for the same problem:
Your goal is: pos-1-2 has a block of height n1 and the robot is at pos-2-0.

The PDDL goal for the new planning goal:

LLM response - PDDL Subgoal (`TERMES`)

(:goal
(and
(height pos-0-0 n0)
(height pos-0-1 n0)
(height pos-0-2 n0)
(height pos-1-0 n0)
(height pos-1-1 n0)
(height pos-1-2 n1)
(height pos-2-0 n0)
(height pos-2-1 n0)
(height pos-2-2 n0)
(height pos-3-0 n0)
(height pos-3-1 n0)
(height pos-3-2 n0)
(not (has-block))
))

Domain and Problem specific (`TYREWORLD`)

The PDDL problem and its initial conditions are given as:
(define (problem tyreworld-1)
(:domain tyreworld)
(:objects
wrench jack pump - tool
the-hub1 - hub
nuts1 - nut
boot - container
r1 w1 - wheel
)
(:init
(in jack boot)
(in pump boot)
(in wrench boot)
(unlocked boot)
(closed boot)
(intact r1)
(in r1 boot)
(not-inflated r1)
(on w1 the-hub1)
(on-ground the-hub1)
(tight nuts1 the-hub1)
(fastened the-hub1)
)

An example planning goal for this problem:
Your goal is to replace flat tyres with intact tyres on the hubs. Intact tyres should be inflated. The nuts should be tight on the hubs. The flat tyres, wrench, jack, and pump should be in the boot. The boot should be closed.

The PDDL goal for the example planning goal:
(:goal
(and
(on r1 the-hub1)
(inflated r1)
(tight nuts1 the-hub1)
(in w1 boot)
(in wrench boot)
(in jack boot)
(in pump boot)
(closed boot)
))

New planning goal for the same problem:
Your goal is: the intact tyre is inflated and the pump is in the boot.

The PDDL goal for the new planning goal:

LLM response - PDDL Subgoal (TYREWORLD)

(:goal
(and
(inflated r1)
)

Domain and Problem specific (GRIPPERS)

The PDDL problem and its initial conditions are given as:
(define (problem gripper-2-2-2)
(:domain gripper-strips)
(:objects robot1 - robot
rgripper1 lgripper1 - gripper
room1 room2 - room
ball1 ball2 - object)
(:init
(at-robby robot1 room1)
(free robot1 rgripper1)
(free robot1 lgripper1)
(at ball1 room1)
(at ball2 room1)
)

An example planning goal for this problem:
Your goal is to transport the balls to their destinations. ball1 should be in room2. ball2 should be in room2.

The PDDL goal for the example planning goal:
(:goal
(and
(at ball1 room2)
(at ball2 room2)
))

New planning goal for the same problem:
Your goal is: ball1 is in room2 and all grippers are free.

The PDDL goal for the new planning goal:

LLM response - PDDL Subgoal (GRIPPERS)

(:goal
(and
(at ball1 room2)
(free robot1 rgripper1)
(free robot1 lgripper1)
)