Mark Grozen-Smith
A:
answers

original state -

    Initial state: r1 q2 a1 b2 ur uq
    Goal state: a2 b1

Solution:
r1
Lar1
q2
Lbq2
Mr12
ar
Mq21
bq
Uar2
Ubq1

Solvable modified state (removed initial aq) -

    Initial state: r1 q2 a1 b2 ur
    Goal state: a2 b1

Solution:
r1
Lar1
b2
Mr12
ar
b2
Uar2
r2
b2
a2
r2
Lbr2
a2
Mr21
br
a2
Ubr1

Unsolvable state:

    Initial state: r1 q2 a1 b2 ur uq

Goal state: r1 r2 q1 q2 ur uq a1 a2 ar aq b1 b2 br bq
Solution requires all states simultaneously. This is not possible.

Part B:
To be an admissible heuristic, it must always underestimate (or exactly estimate) the cost. We know this heuristic always underestimates the cost because it currently just counts how many layers away we are from expanding all goal propositions. This however is not doing enough work since we also must consider the mutex relationships. Thus, we underestimate the cost.

The plans returned by A* were of lower quality (especially the higher the complexity of the problem, e.g. more mutexes), however A* searches very quickly by not worrying itself with these mutexes.
Case 1:
Case 2:
Case 3:

find_housing

Tasks: meet_owner, sign_lease, show_cash, finalize_rent, take_loan, acquire_house

Non-Primitive Tasks:rent_house, buy_house, spend_loan

Actions:

rent_house(x,y) - user x rents house y
Preconditions - exist(x), exist(y)
Effect - gets_housing(x), renting_house(x,y)

buy_house(x,y) - user x buy house y
Preconditions - exist(x), exist(y), has_money(x)
Effect - gets_housing(x), buying_house(x,y)

meet_owner(x,y, z) - user x meets owner y of house z
Preconditions - exist(x), exist(y), exist(z), owns(y,z)
Effect - ready_to_buy(x,z)

sign_lease(x,y, z) - user x meets owner y of house z to sign lease
Preconditions - exist(x), exist(y), exist(z), owns(y,z), ready_to_buy(x,z)
Effect - owns(x,z), not_owns(y,z), finalize_get_housing(x)

show_cash(x,y, z) - user x shows cash to owner y of house z
Preconditions - exist(x), exist(y), exist(z), owns(y,z), has_money(x)

Effect - ready_to_buy(x,z) buying_with_cash(x,z)

finalize_rent(x,y, z) - user x meets owner y of house z to finalize_rent
Preconditions - exist(x), exist(y), exist(z), owns(y,z), ready_to_buy(x,z), buying_with_cash(x,z)
Effect - owns(x,z), not_owns(y,z), finalize_get_housing(x)

take_loan(x,y) - user x takes loan y
Preconditions - exist(x)
Effect - owes_money(x,y)

spend_loan(x,y) - user x spends loan on house y
Preconditions - exist(x), exist(y)
Effect - owns(x,y), finalize_get_housing(x)

attraction_trip

Tasks: buy_tickets, google_city, take_flight, miss_flight

Non-Primitive Tasks: plan_with_friends, wing_trip, plan_alone

Actions:

plan_with_friends(x,y) - plans trip z for user x and friends y
Preconditions: exist(x), exist(y), [has_responsible_friends(x) v has_lame_friends(x)]
Effect: trip_planned(x), trip_planned(y)

plan_alone(x) - plans trip for user
Preconditions: exist(x), [is_ambitious(x) v is_lame(x)]
Effect: trip_planned(x), trip_planned(y)

google_city(x,y) - googles city y for user x
Preconditions: exist(x), exist(y), has_responsible_friends(x)
Effect: can_execute_trip(x)

buy_tickets(x,y) - user x buys tickets for user x and friends y
Preconditions: exist(x), exist(y), can_execute_trip(x)
Effect: trip_finalized(x), trip_finalized(y)

buy_tickets(x) - wing trip for user x
Preconditions: exist(x), exist(y), has_lame_friends(x)

Effect: trip_finalized(x), missed_flight(x)

take_flight(x) - wing trip for user x
Preconditions: exist(x), is_ambitious(x)
Effect: trip_finalized(x), island_reached(x)

enjoy_island(x) - enjoy island as user x
Preconditions: exist(x), island_reached(x)
Effect: enjoy_vacation(x)

take_loan(x) - take loan as user x
Preconditions: exist(x), is_lame(x)
Effect: has_money(x)

buy_car(x, y) - buy car y as user x
Preconditions: exist(x), exist(y), has_money(x)
Effect: has_car(x, y), not_has_money(x)

drive_car(x, y) - drive car y as user x
Preconditions: exist(x), exist(y), has_car(x,y)
Effect: trip_finalized(x), enjoy_vacation(x)