Action Language Description

July 2019

Sorts

$area = \{room, corridor\}$.
$exit = \{door\}$.
$agent = \{robot, human\}$.
$fixed\_element = \{floor, floor_cor, door\}$.
$object = \{box1, box2, box3, box4, box5, cup, chair\}$.
$thing = object \cup agent$.
$obj\_w\_zloc = thing \cup fixed\_element$.
$vertsz = 0..18.$)
bool = {true, false }
substance = {paper, plastic, wood, glass}.

skill_level = {poor, average, good }.
weight = {light, medium, heavy }.
limb = {arm, leg }.

$Inertial fluents$:
fluent (inertial, on (thing, obj_w_zloc)).
fluent (inertial, z_loc(obj_w_zloc, vertsz)).
fluent (inertial, location(obj_w_zloc, area)).
fluent (inertial, in_hand(agent, object)).

$Defined fluents$:
fluent(defined, in_range(obj_w_zloc, obj_w_zloc, vertsz).
fluent(defined, can_support(obj_w_zloc, thing).

$Actions$:
$action(\ go\_to(\ agent,\ obj\_w\_zloc))$.
$action(\ put\_down(\ agent,\ object,\ obj\_w\_zloc))$.
$action(\ go\_through(\ agent,\ exit,\ location))$.
$action(\ pick\_up(\ agent,\ object))$. $action(\ move\_to(\ agent,\ object,\ obj\_w\_zloc))$.

$\#const\ n = 1$
$step(0..n)$.

$height(obj\_w\_zloc,\ vertsz)$
$has\_exit(\ area,\ exit)$
$has\_exit(\ area,\ fixed\_element)$
$has\_surf(obj\_w\_zloc, bool)$
$material(obj\_w\_zloc, substance)$
$has\_weight(thing, weight)$
$limb\_strength(agent, limb, skill\_level)$
$joint\_mobility(agent, limb, skill\_level)$
$affordance\_permits(action, step, id)$
$affordance\_forbids(action, step, id)$

Causal Laws

1. $go\_to(\ A,\ S)$ **causes** $on(A,\ S)$
2. $put\_down(\ A,\ O,\ S)$ **causes** $on(O,\ S)$
3. $put\_down(\ A,\ O,\ S)$ **causes** $\neg(in\_hand(A, O)$
4. $go\_to(\ A,\ S)$ **causes** $z\_loc(\ A,\ Z + H)$ **if** $height(\ A, H), z\_loc(S,\ Z)$
5. $go\_through(A,\ E,\ L)$ **causes** $location(A,\ L)$
6. $go\_through(A,\ E,\ L)$ **causes** $on(A,\ floor)$
7. $put\_down(A,\ O,\ S)$ **causes** $z\_loc(\ O,\ Z + H)$ **if** $height(\ O,\ H), z\_loc(S, Z)$
8. $pick\_up(A,\ O)$ **causes** $in\_hand(A,\ O)$
9. $pick\_up(A,\ O)$ **causes** $\neg on(O,\ S)$ **if** $on(O,\ S)$
10. $pick\_up(A,\ O)$ **causes** $\neg z\_loc(O,\ Z)$ **if** $z\_loc(O,\ Z)$
11. $go\_to(\ A,\ S2)$ **causes** $\neg on(\ A,\ S) if on(\ A,\ S)$ **if** $on(\ A,\ S)$
12. $go\_to(\ A,\ S)$ **causes** $\neg z\_loc(\ A,\ Z)$ **if** $z\_loc(\ A,\ Z)$
13. $go\_through(\ A,\ E,\ Loc)$ **causes** $\neg on(\ A,\ S)$ **if** $on(\ A,\ S)$

State Constraints

1. $\neg on(O, S)$ **if** $on(O_2, S),\ O \neq O_2$
2. $z\_loc(O, Z + H)$ **if** $on(O, S),\ height(O, H),\ z\_loc(S, Z)$
3. $\neg on(O, S)$ **if** $on(O, S_2),\ thing(O),\ S \neq S_2$
4. $\neg height(O, H_2)$ **if** $height(O, H_1),\ H \neq H_2\ \neg has\_weight(O, W_2)$ **if** $has\_weight(O, W_1),\ W_1 \neq W_2$
5. $\neg location(O, L)$ **if** $location(O, L_2),\ L \neq L_2$

6. $in\_range(O_1, O_2, X)$ **if** $z\_loc(O_1, Z_1), z\_loc(O_2, Z_2),$
$Height(O_1, H_1), height(O_2, H_2),$
$(Z_1 - H_1) \geq (Z_2 - H_2),$
$X = (Z_1 - H_1) - (Z_2 - H_2)$

7. $can\_support(S, O)$ **if** $has\_weight(O, light), material(S, glass)$
8. $can\_support(S, O)$ **if** $nothas\_weight(O, heavy), material(S, plastic)$
9. $can\_support(S, O)$ **if** $has\_weight(O, light), material(S, paper).$
10. $can\_support(S, O)$ **if** $material(S, wood)$
11. $\neg can\_support(S, O)$ **if** $holds(on(S, S2), I), not\ holds(can\_support(S2, O), I).$
12. $\neg on(O, O2)$ **if** $has\_surf(O2, false).$

Executability Conditions

1. **Impossible** $pick\_up(A, O)$ **if** $in\_hand(A, O_2)$
2. **Impossible** $put\_down(A, O, S)$ **if** $not\ in\_hand(A, O)$
3. **Impossible** $go\_to(A, S)$ **if** $on(A, S)$
4. **Impossible** $pick\_up(A, O)$ **if** $on(O_2, O)$
5. **Impossible** $go\_to(A, S))$ **if** $on(O_2, S),\ object(S)$
6. **Impossible** $put\_down(A, O, S)$ **if** $on(O_2, S),\ object(S)$
7. **Impossible** $go\_through(A, E, L_1)$ **if** $not\ location(A, L_0),$
$not\ has\_exit(L_0, E),$
$not\ has\_exit(L_1, E).$
8. **Impossible** $go\_to(A, S)$ **if** $in\_hand(A_2, S).$
9. **Impossible** $go\_to(A, S)$ **if** $z\_loc(S, Z),$
$z\_loc(A, Z2),$
$height(A, H),$
$Z2 - H = BASE,$
$Z < BASE - 1.$
10. **Impossible** $go\_to(A, S)$ **if** $z\_loc(S, Z),$
$z\_loc(A, Z2),$
$height(A, H),$
$Z2 - H = BASE,$
$Z > BASE + 1.$

11. **Impossible** $go\_through(A, E, Loc1)$ **if** $location(A, Loc2)$,
$Loc1 = Loc2$.

12. **Impossible** $go\_to(A, S)$ **if** $location(A, Loc1)$,
$location(S, Loc2)$,
$Loc1 \neq Loc2$.

13. **Impossible** $go\_to(A, S)$ **if** $on(A, S2),\ S = S2$.

14. **Impossible** $put\_down(A, O, S)$ **if** $location(A, Loc1)$,
$location(S, Loc2)$,
$Loc1 \neq Loc2$.

15. **Impossible** $pick\_up(A, O)$ **if** $location(A, Loc1)$,
$location(O, Loc2)$,
$Loc1 \neq Loc2$.

16. **Impossible** $go\_to(A, S)$ **if** $agent(S)$.

17. **Impossible** $pick\_up(A, O)$ **if** $height(A, H_A),\ height(O, H_O),\ H_O \geq H_A + 2$.

18. **Impossible** $go\_through(A, D, Loc)$ **if** $in\_range(D, A, X)$,
$height(A, H)$,
$X \geq H + 2$.

19. **Impossible** $go\_through(A, D, Loc)$ **if** $in\_range(A, D, X)$,
$height(D, H)$,
$X \geq H$.

Executability conditions determined by affordance relations

1. **Impossible** $action(X)$ **if** $affordance\_forbids(X, ID)$.

2. **Impossible** $go\_to(R, S)$ **if** $not\ affordance\_permits(go\_to(R, S), 30)$.

3. **Impossible** $pick\_up(R, O)$ **if** $in\_range(O, R, X),\ height(R, H), X \geq H$,
$not\ affordance\_permits(pick\_up(R, O), 17)$.

4. **Impossible** $pick\_up(R, O)$ **if** $in\_range(O, R, X),\ height(R, H), X \geq H$,
$affordance\_permits(pick\_up(R, O), 17)$,
$not\ affordance\_permits(pick\_up(R, O), 13)$,
$not\ affordance\_permits(pick\_up(R, O), 14)$.

5. **Impossible** $pick\_up(A, O)$ **if** $z\_loc(A, Z), height(A, H), z\_loc(O, Z_O), Z - H \geq Z_O,$
$not\ affordance\_permits(pick\_up(A, O), 18).$

6. **Impossible** $pick\_up(A, O)$ **if** $z\_loc(A, Z), height(A, H), z\_loc(O, Z_O), Z - H \geq Z_O,$
$affordance\_permits(pick\_up(A, O), 18),$
$not\ affordance\_permits(pick\_up(A, O), 15),$
$not\ affordance\_permits(pick\_up(A, O), 16).$

7. **Impossible** $pick\_up(R, O)$ **if** $height(A, H),\ height(O, H_O),\ H_O \geq H,$
$not\ affordance\_permits(pick\_up(R, O), 19).$

8. **Impossible** $put\_down(A, O, S)$ **if** $z\_loc(A, Z), z\_loc(S, Z_S), Z_S \geq Z,$
$not\ affordance\_permits(put\_down(A, O, S), 20).$

9. **Impossible** $put\_down(A, O, S)$ **if** $z\_loc(A, Z), z\_loc(S, Z_S), Z_S \geq Z,$
$affordance\_permits(put\_down(A, O, S), 20),$
$not\ affordance\_permits(pick\_up(A, O), 13),$
$not\ affordance\_permits(pick\_up(A, O), 14).$

10. **Impossible** $go\_through(A, D, R)$ **if** $not\ affordance\_permits(go\_through(A, D, L), 26).$

11. **Impossible** $put\_down(A, O, S)$ **if** $z\_loc(A, Z), z\_loc(S, Z_S), height(A, H), Z - H > Z_S,$
$notaffordance\_permits(put\_down(A, O, S), 21),$
$not\ affordance\_permits(put\_down(A, O, S), 22),$
$not\ affordance\_permits(put\_down(A, O, S), 23).$

12. **Impossible** $put\_down(A, O, S)$ **if** $z\_loc(A, Z), z\_loc(S, Z_S), height(A, H), Z_S \geq Z,$
$affordance\_permits(put\_down(A, O, S), 21),$
$not\ affordance\_permits(pick\_up(A, O), 15),$
$not\ affordance\_permits(pick\_up(A, O), 16).$

13. **Impossible** $put\_down(A, O, S)$ **if** $not\ affordance\_permits(put\_down(A, O, S), 24).$

14. **Impossible** $put\_down(A, O, S)$ **if** $not\ affordance\_permits(put\_down(A, O, S), 24),$
$not\ affordance\_permits(put\_down(A, O, S), 25).$

15. **Impossible** $go\_to(A, S)$ **if** $z\_loc(S, Z), z\_loc(A, Z_A), height(A, H), Z \neq Z_A - H,$
$notaffordance\_permits(go\_to(A, S), 28),$
$not\ affordance\_permits(go\_to(A, S), 29).$

16. **Impossible** $go\_through(A, D, Loc)$ **if** $in\_range(D, A, X), X \neq 0,$
$notaffordance\_permits(go\_through(A, D, Loc), 31),$
$not\ affordance\_permits(go\_through(A, D, Loc), 32).$

17. **Impossible** $go\_through(A, D, Loc)$ **if** $not\ affordance\_permits(go\_through(A, D, Loc), 33).$

Affordance Relations

1. $affordance\_permits(pick\_up(A, O), 10)$ **if** $limb\_strength(A, arm, good)$.
2. $affordance\_permits(pick\_up(R, O), 11)$ **if** $height(R, H)$,
   $in\_range(O, R, X)$,
   $X < H$,
   $X >= 0$.
3. $affordance\_permits(put\_down(A, O, S), 12)$ **if** $can\_support(S, O)$.
4. $affordance\_permits(pick\_up(A, O), 13)$ **if** $joint\_mobility(A, arm, good)$,
   $limb\_strength(A, arm, average)$,
   $not\ has\_weight(O, heavy)$.
5. $affordance\_permits(pick\_up(A, O), 14)$ **if** $joint\_mobility(A, arm, good)$,
   $limb\_strength(A, arm, good)$.
6. $affordance\_permits(pick\_up(A, O), 15)$ **if** $joint\_mobility(A, leg, good)$,
   $limb\_strength(A, leg, average)$,
   $joint\_mobility(A, arm, good)$,
   $limb\_strength(A, arm, average)$,
   $not\ has\_weight(O, heavy)$.
7. $affordance\_permits(pick\_up(A, O), 16)$ **if** $joint\_mobility(A, leg, good)$,
   $limb\_strength(A, leg, good)$,
   $joint\_mobility(A, arm, good)$,
   $limb\_strength(A, arm, good)$.
8. $affordance\_permits(pick\_up(A, O), 17)$ **if** $in\_range(O, A, X), height(A, H), X < H + 2$.
9. $affordance\_permits(pick\_up(A, O), 18)$ **if** $z\_loc(A, Z), height(A, H), z\_loc(O, Z_O)$,
   $Z - H \geq Z_O, Z - H - Z_O < 2$.
10. $affordance\_permits\ (pick\_up(A, O), 19)$ **if** $height(A, H), height(O, H_O)$,
    $H_O \geq H,\ H_O \leq H + 1$,
    $has\_weight(O, light)$.
11. $affordance\_permits(put\_down(A, O, S), 20)$ **if** $z\_loc(A, Z), z\_loc(Z, Z_S), Z_S \geq Z, Z_S - Z \leq 2$.
12. $affordance\_permits(put\_down(A, O, S), 21)$ **if** $z\_loc(A, Z), z\_loc(Z, Z_S)$,
    $height(A, H),\ Z - H > Z_S,\ Z - Z_S \leq 2$.

13. $affordance\_permits(put\_down(A, O, S), 22)$ **if** $z\_loc(A, Z), z\_loc(Z, Z_S), height(A, H),$
$Z_S < Z - HA, has\_weight(O, light).$

14. $affordance\_permits(put\_down(A, O, S), 23)$ **if** $z\_loc(A, Z), z\_loc(Z, Z_S), height(A, H),$
$Z_S < Z - HA, has\_weight(O, medium),$
$notmaterial(O, glass), notmaterial(S.glass).$

15. $affordance\_permits(put\_down(A, O, S), 24)$ **if** $can\_support(S, O).$

16. $affordance\_permits(put\_down(A, O, S), 25)$ **if** $height(A, H), height(O, H_O),$
$z\_loc(A, Z_A), z\_loc(Z, Z_S),$
$Z_A - H \leq Z_S, Z_S \leq Z_A.$

17. $affordance\_permits(go\_to(A, S), 28)$ **if** $z\_loc(S, Z),$
$z\_loc(A, Z2),$
$height(A, H),$
$Z2 - H = BASE,$
$Z \leq BASE + 1, joint\_mobility(A, leg, good).$

18. $affordance\_permits(go\_to(A, S), 29)$ **if** $z\_loc(S, Z),$
$z\_loc(A, Z2),$
$height(A, H),$
$Z2 - H = BASE,$
$Z \geq BASE - 1, joint\_mobility(A, leg, good).$

19. $affordance\_permits(go\_to(A, S), 30)$ **if** $can\_support(S, A),\ agent(A),$
$obj\_w\_zloc(S).$

20. $affordance\_permits(go\_through(A, Opening, L), 31)$ **if** $in\_range(Opening, S, X),$
$has\_surf(S, true), height(S, H),$
$X = H + 1, 0$
$on(A, S),$
$joint\_mobility(A, leg, good).$

21. $affordance\_permits(go\_through(R, D, L), 32)$ **if** $in\_range(S, Opening, X), X > 0,$
$z\_loc(Opening, Z), z\_loc(S, Z_S),$
$on(A, S), height(A, H), Z - Z_S \geq H.$

22. $affordance\_permits(go\_through(A, E, L), 33)$ **if** $height(A, H),$
$height(E, H\_exit),$
$H \leq H\_exit.$

23. $affordance\_permits(go\_through(R, D, L), I, 34)$ **if** $on(R, S)$,
$height(R, HR)$,
$height(D, HD)$,
$height(S, HS)$,
$in\_range(D, S, X)$,
$HS + HR > X$,
$HS < X + HD$.

24. $affordance\_forbids(pick\_up(A, O), 35)$ **if** $nothas\_weight(O, light)$,
$limb\_strength(A, arm, poor)$.