High-Level Robot Control Through Logic

Murray Shanahan and Mark Witkowski

Appendices

Appendix A: Navigation Code

```
/* Navigation Compound Actions */

happens(go_to_room(R,R),T,T).

happens(go_to_room(R1,R3),T1,T4) :-
     towards(R2,R3,R1), connects(D,R1,R2),
     holds_at(door_open(D),T1),
     happens(go_through(D),T1,T2),
     happens(go_to_room(R2,R3),T3,T4),
     before(T2,T3),
     not(clipped(T2,in(R2),T3)).

happens(go_to_room(R1,R3),T1,T4) :-
     connects(D,R1,R2),
     holds_at(door_open(D),T1),
     happens(go_through(D),T1,T2),
     happens(go_to_room(R2,R3),T3,T4),
     before(T2,T3),
     not(clipped(T2,in(R2),T3)).

initiates(go_to_room(R1,R2),in(R2),T) :-
     holds_at(in(R1),T).

happens(go_through(D),T1,T2) :-
     holds_at(loc(corner(C1),ahead),T1),
     door(D,C1,C2),
     happens(turn(left),T1),
     happens(turn(left),T2),
     before(T1,T2),
     not(clipped(T1,door_open(D),T2)).

happens(go_through(D1),T1,T3) :-
     holds_at(loc(corner(C1,ahead)),T1),
     door(D2,C1,C2), diff(D1,D2),
     holds_at(door_open(D2),T1),
     happens(go_straight,T1),
     happens(go_through(D1),T2,T3),
     before(T1,T2).

happens(go_through(D),T1,T3) :-
     holds_at(loc(corner(C),behind),T1),
     happens(follow_wall,T1),
     happens(go_through(D),T2,T3),
     before(T1,T2),
     not(clipped(T1,door_open(D),T2)).

happens(go_through(D),T1,T3) :-
     holds_at(loc(corner(C),ahead),T1),
     inner(C),
     happens(turn(right),T1),
     happens(go_through(D),T2,T3),
     before(T1,T2),
     not(clipped(T1,door_open(D),T2)).

/* Navigation Heuristics */

towards(R1,R1,R2).

towards(R1,R2,R3) :- connects(D,R1,R2).

towards(R1,R2,R3) :-
     connects(D1,R1,R4), connects(D2,R4,R2).

/* External Actions */

terminates(close_door(D),door_open(D),T).

initiates(open_door(D),door_open(D),T).
```

Appendix B: Map Building Code

```
/* Map Building Compound Actions */

happens(explore,T1,T6) :-
     holds_at(loc(corner(C1),ahead),T1),
     inner(C1),
     unexplored_corner(C1,T1),
     happens(turn(right),T1,T2),
     happens(follow_wall,T3,T4), before(T2,T3),
     happens(explore,T5,T6), before(T4,T5).

happens(explore,T1,T4) :-
     holds_at(loc(corner(C1),ahead),T1),
     not(inner(C1)),
     unexplored_corner(C1,T1),
     happens(go_straight,T1,T2),
     happens(explore,T3,T4), before(T2,T3).

happens(explore,T1,T4) :-
     holds_at(loc(corner(C1),behind),T1),
     unexplored_corner(C1,T1),
     happens(follow_wall,T1,T2),
     happens(explore,T3,T4), before(T2,T3).

happens(explore,T1,T4) :-
     holds_at(loc(corner(C1),S),T1),
     not(unexplored_corner(C1,T1)),
     unexplored_door(D,T1),
     happens(go_through(D),T1,T2),
     happens(explore,T3,T4), before(T2,T3).

initiates(explore,knows_map,T).
```

```
holds_at(knows_map,T) :-
     not(unexplored_door(D,T)),
     not(unexplored_corner(C,T)).

unexplored_corner(C1,T) :-
     pos(C1,P), not(next_corner(R,C1,C2)).

unexplored_door(D,T) :-
     door(D,C1,C2), not(connects(D,R1,R2)).

/* Integrity constraints */

inconsistent([pos(C1,P1), pos(C2,P2),
     same_pos(P1,P2),
     room_of(C1,R), room_of(C2,R),
     diff(C1,C2)]).

inconsistent([next_corner(R,C1,C2),
     next_corner(R,C1,C3), not(eq(C2,C3))]).

inconsistent([next_corner(R1,C1,C2),
     next_corner(R2,C1,C2), not(eq(R1,R2))]).

/* Constraints */

common_antecedent(pos(C,[X1,X2,Y1,Y2]),
          pos(C,[X3,X4,Y3,Y4]),
          pos(C,[X5,X6,Y5,Y6) :-
     max(X1,X3,X5), min(X2,X4,X6),
     max(Y1,Y3,Y5), min(Y2,Y4,Y6).
```

Appendix C: Shared Code

```
/* Primitive Actions */

initiates(follow_wall,
          loc(corner(C2),ahead),T) :-
     holds_at(loc(corner(C1),behind),T),
     next_visible_corner(C1,C2,left,T).

terminates(follow_wall,loc(corner(C),behind),T).

next_visible_corner(C1,C2,left,T) :-
     holds_at(in(R),T),
     next_corner(R,C1,C2),
     not(invisible_corner(C2,T)).

next_visible_corner(C1,C3,left,T) :-
     holds_at(in(R),T),
     next_corner(R,C1,C2),
     invisible_corner(C2,T),
     next_visible_corner(C2,C3,left,T).

invisible_corner(C1,T) :-
    door(D,C1,C2),holds_at(neg(door_open(D)),T).

invisible_corner(C1,T) :-
    door(D,C2,C1),holds_at(neg(door_open(D)),T).

initiates(go_straight,
          loc(corner(C2),behind),T) :-
     holds_at(loc(corner(C1),ahead),T),
     door(D,C1,C2).

terminates(go_straight,
          loc(corner(C1),ahead),T) :-
     holds_at(loc(corner(C1),ahead),T),
     door(D,C1,C2).

initiates(turn(left),loc(door(D),in),T) :-
     holds_at(loc(corner(C1),ahead),T),
     door(D,C1,C2),
     holds_at(door_open(D),T).

terminates(turn(left),
          loc(corner(C1),ahead),T) :-
     holds_at(loc(corner(C1),ahead),T),
     door(D,C1,C2),
     holds_at(door_open(D),T).

initiates(turn(left),
          loc(corner(C2),behind),T) :-
     holds_at(loc(door(D),in),T),
     holds_at(in(R1),T),
     connects(D,R1,R2), door(D,C1,C2),
     next_corner(R2,C1,C2).

terminates(turn(left),loc(door(D),in),T) :-
     holds_at(loc(door(D),in),T).

initiates(turn(left),in(R2),T) :-
     holds_at(loc(door(D),in),T),
     holds_at(in(R1),T),  connects(D,R1,R2).

terminates(turn(left),in(R1),T) :-
     holds_at(loc(door(D),in),T),
     holds_at(in(R1),T).

initiates(turn(right),
          loc(corner(C),behind),T) :-
     holds_at(loc(corner(C),ahead),T),
     inner(C).

terminates(turn(right),
          loc(corner(C),ahead),T) :-
     holds_at(loc(corner(C),ahead),T), inner(C).

initiates(turn(right),facing(W1),T) :-
     holds_at(facing(W2),T), plus_90(W2,W1).

terminates(turn(right),facing(W),T) :-
     holds_at(facing(W),T).

initiates(turn(left),facing(W1),T) :-
     holds_at(facing(W2),T), minus_90(W2,W1).

terminates(turn(left),facing(W),T) :-
     holds_at(facing(W),T).

initiates(follow_wall,co_ords(P),T) :-
     holds_at(loc(corner(C1),behind),T),
     next_visible_corner(C1,C2,left,T),
     pos(C2,P).

terminates(follow_wall,co_ords(P),T) :-
     holds_at(co_ords(P),T).

initiates(go_straight,co_ords(P),T) :-
     holds_at(loc(corner(C1),ahead),T),
     door(D,C1,C2), pos(C2,P).

terminates(go_straight,co_ords(P),T) :-
     holds_at(co_ords(P),T).

initiates(turn(left),co_ords(P),T) :-
     holds_at(loc(door(D),in),T),
     holds_at(in(R1),T),
     connects(D,R1,R2),
     door(D,C1,C2), next_corner(R2,C1,C2),
     pos(C2,P).
```

```
terminates(turn(left),co_ords(P),T) :-
     holds_at(loc(door(D),in),T),
     holds_at(co_ords(P),T).

/* Sensor events */

happens(left_and_front(X),T,T) :-
     happens(follow_wall,T,T),
     holds_at(co_ords(P1),T),
     holds_at(facing(W),T),
     holds_at(loc(corner(C1),behind),T),
     next_visible_corner(C1,C2,left,T),
     inner(C2),
     displace(P1,X,W,P2), pos(C2,P2).

happens(left(X),T,T) :-
     happens(turn(right),T,T),
     holds_at(loc(corner(C),ahead),T), inner(C).

happens(left(X),T,T) :-
     happens(turn(left),T,T),
     holds_at(loc(door(D),in),T),
     holds_at(in(R1),T),
     connects(D,R1,R2), connects(D,R2,R1),
     holds_at(co_ords(P1),T),
     holds_at(facing(W1),T),
     next_corner(R2,C3,C2), door(D,C3,C2),
     wall_thickness(Y1), displace(P1,Y1,W1,P2),
     pos(C2,P2), door_width(Y2), plus_90(W1,W2),
     displace(P2,Y2,W2,P3), pos(C3,P3).

happens(left(X),T,T) :-
     happens(go_straight,T,T),
     holds_at(co_ords(P1),T),
     holds_at(facing(W),T),
     holds_at(loc(corner(C1),ahead),T),
     holds_at(in(R),T),
     next_corner(R,C1,C2), door(D,C1,C2),
     displace(P1,X,W,P2), pos(C2,P2).

happens(left_gap(X),T,T) :-
     happens(follow_wall,T,T),
     holds_at(co_ords(P1),T),
     holds_at(facing(W),T),
     holds_at(loc(corner(C1),behind),T),
     next_visible_corner(C1,C2,left,T),
     not(inner(C2)),
     displace(P1,X,W,P2), pos(C
```