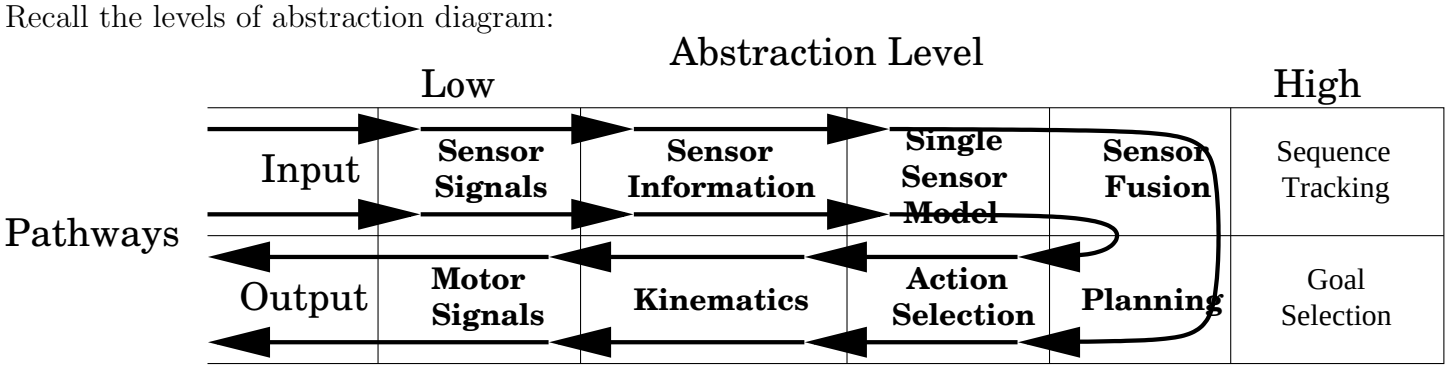
# Interleaved Planning

1. Given that you have a camera and a programming language, how would you program the robix arm to pick up an object in an unknown location?
   (a) What are the steps?
   (b) What if there are multiple different objects?
   (c) What if intermediate steps need to be done?
   (d) You have most likely come up with a rough approximation of traditional AI robotics: the sense, plan, act cycle, or interleaved planning.

2. Sense → Plan → Act
   (a) Gather information from all the sensors.
   (b) Integrate this information together to build a world model.
   (c) Define a search space based on this model.
   (d) Formulate a plan of action to reach a goal
   (e) Decide on the first action to take to execute that plan.
   (f) Calculate the kinematics necessary to take that action
   (g) Take it.
   (h) Repeat.

Recall the levels of abstraction diagram:




Here we see that in a traditional planning system, input from the sensors is taken in, the robot *identifies* what is sees, and then plans a respnse to its input. This response is in the form of a plan: as sequence of steps with (possibly) some contingency actions. The robot then converts the first step of the sequence into motor commands which it then performs.

1. Sensing. Some set of sensors, often including vision are pre-processed in order to recognize features. This often begins with recognizing sub-features and stringing them together.

2. Integration involves matching up pieces from the different sensors together when they correspond to the same object.

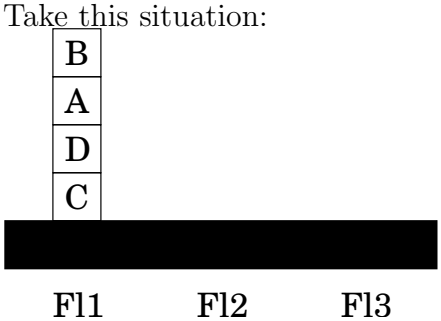
   (a) Known as sensor fusion.

   (b) Builds a more detailed model of the world where sensors overlap, and fills in gaps where they don't.

3. Take these and set them into a world model. The world models are often sets of logical statements about the world.

   (a) Closed World Assumption.

4. Take this situation:




   (a) This can be described with the following 5 strips statements:

      i. On(B,A)
      ii. On(A,D)
      iii. On(D,C)
      iv. On(C,Fl1)
      v. Clear(B)
      vi. Clear(Fl2)
      vii. Clear(Fl3)

   (b) This can be thought of as the state of the world. The goal state could be:

      i. On(A,B)
      ii. On(B,C)
      iii. On(C,D)
      iv. On(D,Fl1)
      v. Clear(A)
      vi. Clear(Fl2)
      vii. Clear(Fl3)

   (c) The robot can also perform actions in the environment:

      i. move(x,y,z)- move block x from y to z.

(d) We can take world states and connect them with links.

   i. A link goes from one state to another if the robot can transform the world from the one state to the other in a single action.

   ii. The result is a (likely infinite) graph.

(e) This defines a search space called a plan space.

   i. The robot will then repeatedly apply operators to states, generating new states until the robot can generate the goal state.

   ii. The ordered set of operators that transforms the current state into the goal state is the plan.

(f) To the graph, we can apply any of the standard search algorithms:

   i. DFS,

   ii. BFS,

   iii. A*, etc.

5. Once the robot has a plan, it selects the first action from the set as the one to execute.

6. The robot then has an action to execute in the form of: move(x,y,z). It must convert this into motor movements: reverse kinematics.

7. Finally, the robot knows what to do, so it does it. Once this is done, the robot must repeat the entire cycle. why?

8. **That** is interleaved planning in a robot.

9. The key here is, how does this planning thing work? That's what we'll talk about now.

10. Preconditions

    (a) How many operators are there?

    (b) How big is the search-space?

    (c) Can all the operators be applied in every situation?

    (d) How do we identify *which* operators can be applied in which situation?

    (e) These are called preconditions and normally take the form:

       i. $On(x, y) \wedge Clear(x) \wedge Clear(z)$.

11. Add and Delete Lists

    (a) When we apply an operator to a state, what state do we get to?

    (b) How can we codify this change?

    (c) These are called the Add list and Delete list:

       i. A: On(x,z), Clear(y), Clear(Fl).

       ii. D: Clear(z), On(x,y).

12. So lets run through a serious mobile robot example, the room example.

13. Predicates:

    (a) INROOM(x,r) - x is an object, r is room.
    (b) NEXTTO(x,t) - x is an object, t is an object.
    (c) STATUS(d,s) - d is a door, s is a status, OPEN or CLOSED.
    (d) CONNECTS(d,rx,ry) - d is a door, rx and ry are rooms.
    (e) HOLDING(r,x) - r is a robot, x is a liftable object
    (f) LIFTABLE(x) - x is an object

14. Imagine an initial state:

    (a) INROOM(IT,R1) - IT is the name of our robot.
    (b) INROOM(Book1,R2)
    (c) INROOM(TABLE1,R1)
    (d) INROOM(PLEBE1,R1)
    (e) CONNECTS(D1, R1,R2)
    (f) CONNECTS(D1,R2,R1) - really, there is a reason for this.
    (g) STATUS(D1,CLOSED).
    (h) NEXTTO(IT,TABLE1)

15. And a goal state:

    (a) HOLDING(IT,BOOK1).
    (b) STATUS(D1,CLOSED).

16. Operators:

| Operator | Preconditions | Add-list | Delete-list |
|---|---|---|---|
| GOTO(IT,x,y) | INROOM(IT,r)∧INROOM(y,r)∧NEXTTO(IT,x) | NEXTTO(IT,y) | NEXTTO(IT,x) |
| OPEN(IT,d) | NEXTTO(IT,d)∧STATUS(d,CLOSED) | STATUS(d,OPEN) | STATUS(d,CLOSI |
| GOTHRU(IT,d) | INROOM(IT,rx)∧NEXTTO(IT,d)∧ CONNECTS(d,rx,ry)∧STATUS(d,OPEN) | INROOM(IT,ry) | INROOM(IT,rx) |
| CLOSE(IT,d) | NEXTTO(IT,d)∧STATUS(d,OPEN) | STATUS(d,CLOSED) | STATUS(d,OPEN |
| PICKUP(IT,x) | NEXTTO(IT,x)∧LIFTABLE(x) | HOLDING(IT,x) | LIFTABLE(x) |

17. Some things to think about:

    (a) Closed world assumption says we only plan accross the things we know about.
        i. There can be no surprises.
        ii. But, what if we forget a detail that is needed for some particular task?
        iii. How many facts would there need to be to cover a situation that is reasonable complicated?

iv. How big is the search space and how much time will we spend?

v. What about all the irrelevant details that we still have to reason over?

A. We can break it into pieces, but these have little portability.

(b) Possibly we'll need to operate in the "open world".

(c) But if we're in the open world, can we realistically plan?

(d) Note this also eliminates any reflexes- "a rock is falling towards me, move *anywhere*."

(e) We may want to consider alternatives...