

# Robot Redux

CSC 242

11 May 2001

*Write your **NAME** legibly on the bluebook. All problems are worth the same; best strategy is to do the easiest ones first. Open book, open handouts, open notes.*

## 1. Robot Resolves (Logic)

Robert's 7-year old mistress Joanna was frowning with concentration. "Wobert," she said, "Mummy told me that: **Anyone who is weckwess and who also pways with matches is going to get aww burnt up.** She also said: **Some people are weckwess,** and that: **Some people pway with matches.** THEN she said that pproved **Somebody's going to get burned.** Is that weawwy TWUE, Wobert?"

Robert puts these statements first into FOPC and then into clause form. Then he attempts a resolution proof. 150 microseconds later he pipes "Mummy's fuwwa cwap" through his Peace-in-the-Family (TM) diplomacy, tact, and discourse civility module. Show what went through his mind, being explicit about all the steps.

Ans: FOPC:

$$\forall(x)(W(x) \wedge P(x)) \Rightarrow B(x)$$

$$\exists(x)W(x)$$

$$\exists(x)P(x)$$

therefore

$$\exists(x)B(x)$$

Clauses:

$$1. \quad \neg W(x) \vee \neg P(x) \vee B(x)$$

$$2. \quad W(a)$$

$$3. \quad P(b)$$

and the negated conclusion

$$4. \quad \neg B(x)$$

Resolution Proof: 4. and 1. cancels out the  $B(x)$  in 1. Then if 2. and 1. unify they leave  $\neg P(a)$ , which then does NOT unify with  $P(b)$ . Or you can do them in the other order, similar problem.

It's obvious to me that this is about all you can do here: I'm not sure whether an ATP would be smart enough to give up at this point. Robert is.

Comment: Too many people forgot Skolemization, and too many talked through their reasoning rather than do the resolution proof. As I've said often, I ALWAYS want fopc, clauses, and a resolution proof showing the substitutions (which in this case fail).

## 2. Robot Run (Path Planning)

Robert (don't call him 'Robbie') represents his world as a 2-D array  $E$  containing 0's for free space and 1's for obstacles, something like this:

```

1111111111111111111111111111... 11111111
1000000000000000000000000000... 00000001
1000000000000000000000000000... 00000001
1000000000000000000000000000... 00000001
1000000000000000000000000000... 00000001
1000000000000000000000000000... 00000001
1000000000000000000000000000... 00000001
1000000000000000000000000000... 00000001
1000000000000000000000000000... 00000001
1000000000000000000000000000... 00000001
1000000111111000000000000000... 00000001
1000000111111000000000000000... 00000001
1000000111111000000000000000... 00000001
1000000111111000000000000000... 00000001
1000000000000000000000000000... 00000001
.....

```

He also represents his shape (projected onto the floor)  $S$  as a set of 1's in X-Y space: When he's carrying a stick it could look something like this (the zeroes indicate his center of coordinates).

```

      0
      |
      V

      1
     111
    1111111111
0--> 11111
      11111
      11111

```

a) In one mode of travel, Robert simply translates in  $X$  and  $Y$  without rotating. Thus his shape (as seen from above) does not rotate or change in any way through time. For path planning, Robert's navigation software uses a configuration space approach. It computes the configuration space obstacle map  $M$  resulting from the environment  $E$  and his current "shape"  $S$ . If there is a connected path of 0's from his start position to his goal position in  $M$  (an easy problem from data structures), there is a path in the real world with his current shape as well (a hard problem

in computational geometry). Unfortunately, the latest update to the navigation software Daddy ordered won't run without something called a "Adult verification Sexxxaxxxxess Password"; Robert will ask Mummy about that soon. Meantime, he can access an old suite of routines that do mathematical morphology operations. Improvising quickly, he computes  $M$  using them. What operation does he use and what structuring element?

b) How can Robert extend his path planner in a brute force way to include rotations of his body?

Ans: a) Dilation with the mirror-reflected (around the origin) shape gives  $M$ . b) At whatever level of angle resolution that seems appropriate, rotate the shape and do a) all over again, and make a stack of the resultant 2-D maps. This will be the needed  $(X, Y, \theta)$  version of  $M$ .

Comment: Too many people ignored my stricture to use the morphology routines. I think nobody got part a completely right (one was close but forgot to flip the shape), and only 1 seemed to get b.

### 3. Robot is Rich (Linear Systems)

Robert gets a royalty check for his latest hit song.

a) He goes to BotMax and buys a new main camera with auto-focus, which adjusts the focus until the image is maximally sharp. He doesn't know how it works but he has helped Joanna with her AI homework. He cogitates on the question: "How could autofocus be implemented if computation were infinitely fast and free?". Then he thinks about: "Now suppose computation and time were limited...." What goes through Robert's mind in detail?

b) At BotMax, Robert likes the packaging on an autocorrelation routine, so like anyone else he seriously considers buying it. The back of the box says, in part: "The autocorrelation of a function is simply its correlation with itself: in one dimension,

$$C(s) = \sum_x f(x)f(x+s).$$

Of course the autocorrelation is symmetric and (for a function limited in time or space) attains its maximum at shift zero ( $s = 0$ ). ... Implemented with Fast Fourier Transform technology." These properties of  $C(s)$  aren't obvious to him, and he wants to know what the FFT has to do with autocorrelation, so he goes up to the Technical Information Booth and asks for an explanation. The person in the booth is you.

c) Robert tries his autocorrelation module on an aerial image of the dunes of White Sands, NM. Then he tries it on an aerial image of the city of Albuquerque, NM. Indeed, both the autocorrelations are symmetric with maxima at zero. In fact, one is shaped something like an Egyptian pyramid and the other is much peakier, like the Washington Monument. Which is which and why?

Ans:

a. One idea is to maximize the energy in high-frequency part of the power spectrum, which would show how much energy was in high-frequency phenomena like edges. So you'd make some sum of the highest frequencies and maximize that. Assumption here is that you can get the frequency spectrum (modulus of the FFT) fast enough, plus all the other filtering and strength calculation. Cheap and dirty is to run a few edge detectors at places around the image and maximize their output. That is checking for "edginess", which presumably is an approximate version of the spatial freq. idea. There are other approaches too..

Comment: My problem here was that I should have had another part saying: How do you tell if the picture is in focus or not? That would have helped elicit what I wanted....

b. The autocorrelation is clearly symmetric because shifting one vector by  $s$  is the same as shifting the other one by  $-s$ , but since they are the same you can swap one with the other invisibly.

At shift 0 the correlation is the dot product of a function (in the finite discrete case here a vector) with itself. The intuition is that it is maximally projected on itself, and that its projection on any other “vector” or function is less. For another thing the other vector is going to have fewer non-zero elements. But the best intuition is that the dot prod is maximized when a vector is dotted with itself.

The autocorrelation would be expensive to calculate by shifting, multiplying and adding adding (order  $N^4$ ), so this module clearly takes the FT of the function, squares it, and takes the inverse FT of the result. This uses the convolution theorem, which says multiplication in freq. domain is equivalent to convolution in the spatial domain. There might be some details in swapping a sign or taking the mirror image of the function to get correlation instead of convolution, but that’s a detail. The reason the FT is useful is that the FastFT is  $O(N \log N)$ , so it’s actually quicker to convert to freq. domain and convert back.

c. White sands dunes are smooth, their brightness varies slowly, and they have fewer high freq. components. Their *energy* will be in the low-freqs, and their power spectrum would be more like the Wash. Monument than the pyramid. But the power spectrum is the FT of the autocorrelation. The autocorrelation measures how well a function matches itself; the dunes match themselves better as they are shifted because their brightness varies slower. The city has lots of edges, fast changes, and one part of it doesn’t look much like another. Thus its autocorrelation falls off faster and it’s the dunes that have the slowly falling autocorrelation, the city has the spikey one.

#### 4. Robot Remembered (Bayes Rule)

Robert is supposed to wash the dog and put a bell on the cat. The dog and cat are never inside at the same time, and he remembers that the dog has been inside 30 of the past 100 days and the cat has been inside the other 70. He has one lousy pet-detector module that goes off 90 percent of the time if it’s aimed at a dog and 20 percent of the time if it’s aimed at a cat. He aims his detector at the furry shape in front of him, and it goes off. He unlimbers his Bayesian reasoning engine and computes the likelihood ratio (the probability it’s a dog divided by the probability it’s a cat) given the sensor reading. Show what goes through Robert’s mind in detail.

Ans.

$$P(d) = .3$$

$$P(c) = .7$$

$$P(m | d) = .9$$

$$P(m | c) = .2$$

$$P(d | m) = .3 * .9 / P(m)$$

$$P(c | m) = .2 * .7 / P(m)$$

$$P(d | m)/P(c | m) = (.3 * .9)/(.2 * .7)$$

Comment: Lots got this 10/10. Good work.

### 5. Robot at Rest (Plans and Utilities)

Robert has been running errands. Now it's 11:15 and he's got a load of groceries, but his legs have just stopped working. Worse, he's on the new Metro Howitzer bullet-train tracks and the 11:16 is running on time.

His self-diagnosis module comes up with this plan.

- Run a test T to determine whether the problem is motor driver failure (M) or a power management failure (P). This binary test costs 10 seconds.
- If the test says "M", short a pair of jumpers with a paper clip (action *jumpers*)... cost 30 seconds. Then run a reconfiguration routine (action *reconfig*) ... cost 15 seconds. Always do both actions before attempting to run; you can't tell if the first one failed until the second one is done.
- If the test says "P", reload an old version of the power management software (action *reload*) ... cost 30 seconds.

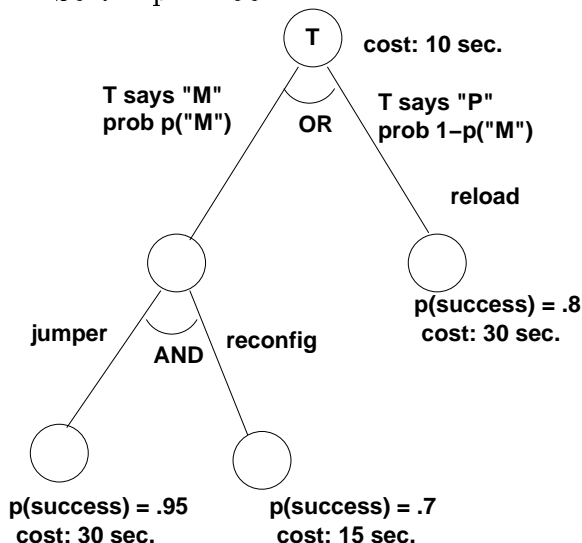
Life's uncertain. The probability that T says "M" or "P" (that is, "not M") given that M or P really is the case is given by this Conditional Probability Matrix:

$$CPM = \begin{array}{c|cc} & M & P \\ \hline \text{"M"} & .7 & .4 \\ \hline \text{"P"} & .3 & .6 \end{array}$$

Given the failure, the prior probability that it is an M is .6 and the prior that it is a P is .4.

The probability of *jumpers* succeeding is .95. The probability of *reconfig* succeeding is .7. The probability of *reload* succeeding is .80.

So the plan looks like this:



a) Robert doesn't place any value on his life (he's backed up on a zip drive), but he does want to know the expected cost of his plan (in seconds). What goes on in his mind as he computes it? Be explicit, show all steps, etc.

b) If T mis-diagnoses the problem or if any attempted action fails, Robert is rubbish. Like anyone else he wonders what his chances of survival are. What are they? Be explicit, show all steps, etc.

(Hint:  $p("M")$ , for instance, is a *marginal* probability that Robert computes from the joint probabilities of (P,M, "P", "M"), which in turn comes from the CPM and the priors.)

Ans. I hate these numbers, they lead to sloppy work and are hard to grade. so let's say

The probability of *jumpers* succeeding is  $p_1$

The probability of *reconfig* succeeding is  $p_2$

The probability of *reload* succeeding is  $p_3$

The cost of T is  $c_0$

The cost of jumpers is  $c_1$

The cost of reconfig is  $c_2$

The cost of reload is  $c_3$

The CPM gives the prob. of the row GIVEN the column.

$$CPM = \begin{array}{c|cc} & M & P \\ \hline "M" & a & b \\ "P" & c & d \end{array}$$

Joint probabilities =  $p(\text{row AND column})$ :

$$\begin{array}{c|cc} & M & P \\ \hline "M" & p("M", M) & p("M", P) \\ "P" & p("P", M) & p("P", P) \end{array} = \begin{array}{c|cc} & M & P \\ \hline "M" & ap(M) & bp(P) \\ "P" & ap(M) & dp(P) \end{array}$$

We'll need the probs of going down the two branches.

$$p("M") = p("M" | M) + p("M" | P) = (ap(M) + bp(P)).$$

$$p("P") = 1 - p("M").$$

a) Then expected cost is

$$c_0 + p("M")(c_1 + c_2) + p("P")(c_3) = 10 + (.7 * .6 + .4 * .4)(30 + 15) + (.3 * .6 + .6 * .4)(30) = 10 + 26.10 + 12.6 = 48.7 \text{ sec.}$$

b) Probability of survival is:

$$p("M", M)(p_1 p_2) + p("P", P)(p_3) = (.7 * .6)(.95 * .7) + (.6 * .4)(.8) = .2793 + .192 = .4713$$

Comment: At least one person got this 10/10, and several seemed to know what they were doing mostly. The marginals threw most people, tho.

**6. Robot sees Red (Learning)** Robert has three color sensors, for Red, Green and Blue, each of which signals one (1) or zero (0) depending on how much R, G, or B it senses. There are dangerous places in Robert's world, but luckily they are always associated with particular colors. The three danger signals are

$(R, G, B) = (0, 1, 1)$

$(R, G, B) = (1, 0, 0)$

$(R, G, B) = (1, 1, 0)$ .

a) Joanna wants to teach Robbie to sound his siren for help if he sees these danger signals. Robbie came free with a perceptron: can he learn to classify danger signals with it? Why or why not?

b) Joanna saves her allowance and buys Robbie neural net recognizer. How should she set up the problem (what does the net look like, what are the inputs and outputs...)? Can this approach work any better? Why or why not?

c) Joanna's daddy tells her he's selling Robert for scrap if he keeps hearing that siren. Joanna thus needs to teach Robert how to avoid dangerous places, not just how to detect them. She knows he has a grid-world representation for his navigation calculations. What sort of learning module should she ask Santa for, and what's the idea behind it?

Ans. a) No, these points aren't linearly separable. One way to see that is to draw the 2x2x2 cube of (RGB) triples.

b) The net would have three input units for the RGB signals and one output for danger or not. Inside it would have some number of hidden units, which J. gets to design. My guess would be a single hidden layer of 4 units would be a good place to start. Theoretically the NN can learn this function because it can fit nonlinear functions. This is a glorified XOR problem for it.

c) Reinforcement learning is what's needed here, since it is a way to assign credit (or in this case, blame) to maneuvers and motions that put R. into danger. The idea is to learn to avoid dangerous places by remembering how he got to them and propagating the badness (a negative reward, if you like) back along the path so the utility of approaching them goes down in Roberts policy, which is how he maps states to actions. You can copy from the book (not to sound cynical) all the buzzwords like Q learning, policy iteration, dynamic programming, etc, as ways to implement reinforcement learning.

Comment: mostly OK but only 1 person identified reinforcement learning as the appropriate tool. Dunno why this blind spot.

## **7. Robot Refers (NLU)**

Joanna says: "I chased the doggie with the stick."

a) Robert creates two parse trees using his simplest English-language phrase structure grammar. Give an idea of what they look like; that is, show how the two different ways to parse the sentence result in different trees.

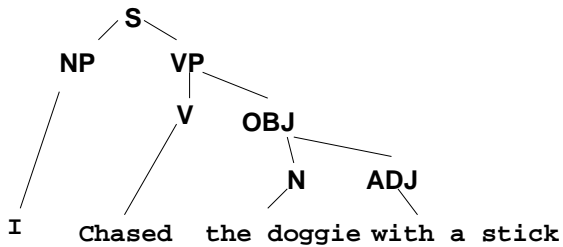
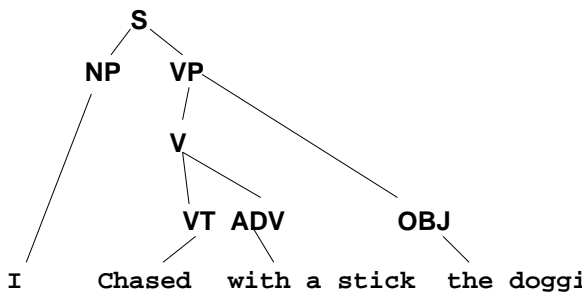
b) Does the sentence have referential ambiguity? semantical ambiguity? lexical ambiguity? syntactic ambiguity? Say why or why not in each case.

c) What approaches might Robert use to disambiguate the sentence: 1) just by itself, 2) in the context of an ongoing dialog?

Ans.

*Deja vu* all over again. I only know one NLU question!

a) This sort of thing is all I wanted for the "parse tree":



b) Syntactic yes (that's the parse tree(s), thus semantic since the meanings are different owing to the different role of the stick; Referential yes: we don't know what dog or what stick we're talking about here. lexical no, we know what all the words mean and they seem unambiguous in this case.

c) By itself, rather a tough one. High-level models and probabilities about J's likelihood of using a stick while chasing a dog would help. In context, one would search for bindings for the dog and the stick which would restrict it to something playing a being-carried role for either J. or Fido. Simpler ideas would be to use recency of reference to figure out the bindings: If "I picked up the stick." immediately preceded our S. we would bind to that. For more on ambiguous reference, ask your TA Joel, who is a world authority (it's true!).

## 8. Robot Regularizes (Vision)

a) Joanna can ask Robert to bring her clothing items based on their fabric: corduroy, angora, herringbone tweed, cable-knit sweater, seersucker, etc. How can Robbie visually distinguish (and classify) these different textures? Specifically and technically, what texture features are you proposing and how are they used?

b) Joanna has been to ceramics class. Robbie says: "What a beautiful polka-dot flowerpot!". Joanna says: "Wobert, how did you know my white cewamic with bwack polkadots is shaped wike a fwowerpot?". Robbie says: "I used the fact that the dots are round to get the surface normal at each dot and then proceeded with a regularization assumption to derive the surface normals everywhere." "Oh", says Joanna, thinking "Daddy can expwain what Wobert means and exacwy what he did!" What does Daddy say?

[picture of pot]

Ans.



a) Co-occurrence matrices, edgeiness, edge-direction histograms, edge-contrast variances, image variance, texel detection, maybe frequency-space measures (Ken Law's texture energy measures, autocorrelations, etc.). Lots of stuff here, but I want technical definitions, not just words! Given all these features, then you'd just heave them into a classifier of some variety (describe your favorite), like parametric, nearest centroid, nearest neighbor, neural net, k-nearest nbr. etc. OR a decision tree if you like....

b) The minor axis of the ellipses points in the direction of the normal (direction of tilt) and the ratio of major/minor gives its slant (how much tilt there is). These are the polar coord versions of (p,q) space, which as we've seen is triv. to integrate into a depth map. So some simple (s,t)  $\rightarrow$  (p,q) conversion allows R. to integrate. But first he assumes smoothness (a regularizing function in some sense) and averages (interpolates) between these known data points. Assuming he can tell where the pot is smooth also allows him to know the (p,q) all around its smooth contours (here the sides, not the top), which helps. Isn't Daddy wonderful?

Comment: Answers generally not technical enough. This vision stuff is math and geometry, not words.