MCS 380/590 Intro To Artificial Intelligence
Dr. William C. Regli
Drexel University Fall 2000
Homework Assignment Solutions #1
Artificial Intelligence (Russell / Norvig)

1.1 Basically everyone got credit for this problem if you were able to reasonably explain each one of Turing's objections in detail, as well as answering the other listed questions which required mostly opinions. These were intended to just get you to think about certain systems.

1.3 In short, No. It means that AI systems should avoid trying to solve intractable problems. Anyone who clearly mentioned this, and gave a good reason got credit for this problem. Notice that humans don't solve NP-complete problems either. If brains are modellable as Turing machines, as most neuroscientists think – in the worst case, the Turing machine can simply simulate the operation of each neuron – then it would be surprising indeed if they can solve NP-complete problems in general.

1.5 Mostly everyone got credit for this problem. It simply required you to think about the feasibility of an AI system in different scenarios.

1.9 This depends on your definition of "intelligent" and "tell". In one sense computers only do what the programmers command them to do, but in another sense what the programmers consciously tells the computer to do often has very little to do with what the computer actually does. Anyone who has written a program with an ornery bug knows this, as does anyone who has written a successful macine learning program. So in one sense Samuel "told" the computer "learn to play checkers better than I do, and then play that way, " but in another sense he told the computer "follow this learning algorithm" and it learned to play. So we're left in the situation where you may or may not consider learning to play checkers to be a sign of intelligence (or you may think that learning to play in the right way requires intelligence, but not in this way), and you may think the intelligence resides in the programmer or in the computer.

1.10 The point of this exercise is to notice the parallel with the previous one. Whatever you decided about whether computers could be intelligent in 1.9, you are committed to making the same conclusion about animals (including humans), unless your reasons for deciding whether something is intelligent take into account the mechanism (programming via genes versus programming via a human programmer). Note that Searle makes this appeal to mechanism in his Chinese Room argument (see Chapter 26).

Note that you could take this to another level: "Surely neither computers nor animals can be intelligent – they can only do what the laws of physics tell their atoms to do".

2.1 A performance measure is used by an outside observer to evaluate how successful an agent is. A utility function is used by an agent itself to evaluate how desirable states or histories are. In our framework, the utility function may not be the same as the preformance measure; furthermore, an agent may have no explicit utility function at all, whereas there is always a performance measure.

2.3 Mostly everyone got this right if you provided a legitimate example and properly specified all the properties of the environment (ie. accessibile, deterministic, etc).

2.8 A lot of people answered this problem incorrectly. It very simple by definition of just what a reflex agent is. An agent starts at home, so it gets a percept that is identical to the percept it would get when it returns home. Therefore, it must either turn itself off right at the start, or not turn itself off when it returns home. In general, the problem with reflex agents is that they have to do the same thing in situations that look the same, even when the situations are actually quite different.

2.10 Everyone (with the exception of one person) got this problem wrong. The following answer is done by assigning probabilities, but this was not required to get the correct answer.

The first thing to be aware of is that the size of the table depends on the policy of the agent. If the agent wanders forever, the length of the percept sequence will be unbounded, and the table size will be unbounded. Let's assume an agent that takes 1 shut-off action, nm forward steps, 2(n + m) turns, and 0.05 nm sucks on average, but we'll cut that off at 0.10nm maximum sucks. So the longest percept sequence we will consider is 1.10 nm + 2(n + m) + 1. There are eight possible percepts, so we need to raise 8 to this power to get the number of entries in the table. For n = m = 12, say, this is $8\hat{2}07$, which isquite a bit more than will fit in most computers.