Some notes and solutions to Russell and Norvig's Artificial Intelligence: A Modern Approach (AIMA, 3rd edition)

Peter Danenberg

June 29, 2012

Contents

1 DONE 1.1

`CLOSED:` *2011-10-10 Mon 03:03*

Intelligence A spontaneous faculty for associating impressions (more general than ideas); synthesizing abstractions from disparate stimuli; deducing conclusions from abstractions.

Intelligence is an emergent property of simples like e.g. neurons.

Artificial intelligence Mechanism for performing association, abstraction, deduction which appears to be spontaneous; may also be an emergent property of bit-pushing.

Agent Self-contained, autonomous input-processing mechanism.

Rationality The appropriate application of or *ratio*; this includes the mechanical process of deduction, as well as an ill-defined notion of common-sense.

Logical reasoning The mechanical aspect of rationality.

2 DONE 1.2

`CLOSED:` *2011-10-10 Mon 03:03*

The Mathematical Objection (3) still holds up: the halting problem; on the other hands, humans are also susceptible to the halting problem, aren't they? If one falls towards the humanity side of the humanity-rationality AI-axis, this deficit is reducible.

Lady Lovelace's Objection (6) is interesting: it denies *ex nihilo*; are genetic algorithms a counter-example?

The Argument from Informality of Behaviour (8) could be solved by fuzzy dispatch.

A modern refutation might be that there are not enough graduate students to make a satisfactory ontology of world-knowledge; thank the gods, then, for mechanical turks and unsupervised learning!

We came pretty damn close to 30% in the 2008 Loebner prize; why not double it to 60% in 2058? Despite Moore's law, let's say that AI proceeds linearly.

3 DONE 1.3

`CLOSED:` *2011-10-12 Wed 12:58*

Reflex actions are rational in the sense that they are the result of induction on e.g. hot objects and the scientific method (see Turing); though the acquisition may require intelligence (induction, storage), the reflex itself is not intelligent in the sense that it requires no induction: it is immediate.

Reflex actions are not irrational, either, in the sense that someone does a cost-benefit analysis and decides to contravene it; let's call reflex actions *pararational*, therefore: neither rational nor irrational. There's no time to apply a utility function and behave accordingly (or discordingly (sic)).

4 DONE 1.4

Tom Evan's ANALOGY is an ad-hoc geometric solver, and would not therefore program. In people, you might be able to generalize from IQ-tests to success; but not so with domain-specific AI.

5 DONE 1.5

`CLOSED:` *2012-05-28 Mon 21:35*

Aplysia, Eric Kandel

20,000 neurons; memory-updates/second: 10^{-9}; cycle time: $10^{1}5$, high end;

Is memory-updates/second merely memory / cycle time? In which case: $20000/10^{-}9 = 10^{5}(20000)$ neurons, cycle time: 10^{-3}; memory updates per second? Not sure what the relationship between operations/sec and memory updates/sec; the former is an upper bound, though. Could it be that memory updates/sec is also bounded, somehow, by storage units? There is also the relationship between neuros and synapses.

In humans, 7,000 synapses per neuron; hence 10^{14} from 10^{11}. How many synapses per aplysia-neuron?

From this paper:

> On average, we found 24 contacts per pair of neurons.

Let's say, then, that sea slugs have 10^{6} synapses; let's also say that, like humans, this is an upper bound on memory updates per second due to the e.g. refractory period.

That gives 10^6 memory updates per second; which means that a supercomputer houses the potential of 10^8 sea slugs.

6 DONE 1.6

`CLOSED:` *2012-05-28 Mon 21:43*

This post on the limits of introspection posits that:

> Mental processes are the results of opaque preferences, and . . . our own "introspected" goals and preferences are a product of the same machinery that infers goals and preferences in others in order to predict their behavior.

Accordingly, introspection is accurate to the extent that we can infer our own thoughts from the mental model we've extrapolated from watching others.

In other words, the processes which lead to thought are to thought opaque.

7 DONE 1.7

`CLOSED:` *2012-05-31 Thu 02:17*

Bar code scanners should hopefully be a trivial mapping from codes to products; if, on the other hand, you could scan and select similar products someone might be interested in: well, then.

The search engine problem is probably AI-complete; current solutions are some AI-complete-like heuristics.

Voice-activated telephone menus might be artificially intelligent in the sense that they have to recover signal from noise and make sense of it.

Internet routing algorithms are classic agents in the sense that they have environments (connection data), sensors (the ability to peer into network devices) and actuators (the ability to re-route traffic).

8 DONE 1.8

`CLOSED:` *2012-05-31 Thu 02:17*

Isn't it the case that humans do do some kind of implicit calculation? Another example is the ability to catch a ball: there are complex physics at play, and yet the human has evolutionarily honed and ad-hoc facilities to perform the same.

Something like Gaussian blur, in other words, is hard-coded into our neurons; vision system, on the other hand, don't have the advantage of fuzzy connections between analog neurons and have to simulate these biological heuristics with algorithms.

9 DONE 1.9

`CLOSED:` *2012-05-31 Thu 02:19*

Evolution might tend to result in systems that maximize certain utility functions (e.g. propagate genes, to that end: stay alive for a while; &c.); this process is pseudo-rational. Pseudo-rational in the sense that it is not rational for rationality's sake; but accidentally rational as it strives to maximize utility.

Maybe there's no distinction to be drawn there after all: ends justifying means.

10 DONE 1.10

`CLOSED:` *2012-05-31 Thu 02:28*

AI is science in the sense that it benefits from the scientific method (work done, for instance, on the relationship between goals and actions; cooperation; how brains cause minds; &c.) and precise mathematics.

AI is engineering, on the other hand, in the sense that it inheres in the world; it must find solutions in messy situations: solutions which might be approximate but nevertheless useful.

11 DONE 1.11

`CLOSED:` *2012-05-31 Thu 02:35*

"Surely computers . . . can do only what their programmers tell them" might have been the case, if it weren't for the fact that programmers can program machines to do things even they couldn't do (cf. chess programs that outstrip their masters).[1]

This seems like a paradox I don't adequately know how to explain; if it proceeds apace, prepare for the singularity.

12 DONE 1.12

`CLOSED:` *2012-05-31 Thu 02:41*

The relationship between nature and nurture is probably complex; suffice to say: genes might provide an upper bound on the intelligence of an animal that it has to strive to meet. Luck helps; so does discipline.

There is a nature-nuture/code-intelligence analogy only insofar as there is code that adapts to its environment; or a programmer can translate intelligence into code (bounded by the programmer's intelligence, of course).

13 DONE 1.13

`CLOSED:` *2012-05-31 Thu 02:48*

It's true that animals, humans and computers are bound by the laws of physics; nevertheless, there is this bizarre phenomenon of emergent behavior wherein the sum is more than its whole of parts.

Consciousness, after all, is an emergent behavior from the propagation of current through neurons; and the world-wide-web has emerged from a decentralized connection of web pages.

[1]See this article from 2007 on Google's machine translation system: "Using a brute-force statistical approach, the Google machine translation team has developed top performing translation software to and from languages that not even one of the teams members understands, such as Arabic and Chinese."

14 DONE 1.14

`CLOSED:` *2011-10-10 Mon 03:52*

1. The Japanese got this one; just a toy, though.
2. There is at least one driverless car in Cairo; it's not self-controlling, though, but rather remotely driven. Driving in clusterfuck-Cairo (like Athens) is taxing for humans, let alone AI. (Google's making political inroads in Nevada, though.) Sufficiently sensitive sensation of surrounding objects, conditions; physics; navigation; are required.
3. DARPA Grand Challenge
4. This robot fetches a sandwich.
5. Grocery IQ will order groceries; a week's worth, though?
6. Zia Mahmood got clowned once or twice; like poker, though, bridge is probabilistic and psychological.
7. TheoryMine is selling new computer-generated proofs for 15; standard objections apply.
8. The Bulhak-Larios Postmodernism Generator is funny; intentionally so?
9. Hilariously-named SHYSTER: ad-hoc expert system
10. Google Translate
11. Mechanically, but there is a human agent (telemanipulator); see this, though, where "In May 2006 the first AI doctor-conducted unassisted robotic surgery on a 34 year old male to correct heart arythmia."

15 DONE 1.15

`CLOSED:` *2012-05-31 Thu 03:05*

TREC appears to dissolve tracks as they become "solved" (e.g. the spam and terabyte tracks) and take new ones up as they emerge (e.g. the microblog and crowdsourcing tracks).

The Grand Challenge is attempting to solve the problem of driverless transportation (see Google's driverless car); despite recent legislation approving driverless cars (in e.g. California, Nevada, New Jersey), it is still cutting edge.

ICKEPS 2012, for instance, has a track for planning solar array operations on the ISS; seems relevant.

RoboCup is interesting in the sense that it requires advanced perception and cooperation among autonomous agents; I suspect it does not detract much from new ideas, despite the fact that it is still wrestling with some of the oldest (and unsolved) problems in AI (*vide supra*).

The Loebner Prize, on the other hand, seems a little anachronistic; do people care whether their AI counterparts really act human?

16 DONE 2.1

`CLOSED:` *2012-06-11 Mon 00:51*

It follows directly from the definition of a rational agent, which "maximizes its performance measure, given the evidence provided by the percept sequence" (p. 37), that its action "depends . . . on the time step it has reached."

This is because the lifetime of an agent is measured by the total number of percepts it receives [2] [3].

Let t be the time step the agent has reached; if $t \leq T$, the agent's performence measure depends upon the time step it has reached. If $t > T$, on the other hand, the rationality of the agent is undefined; since its performance measure is undefined.

At $t > T$, the agent has become pararational (neither rational nor irrational).

A rational agent's action, therefore, depends upon t only insofar as its performance measure depends upon t.

[2]"Let T be the lifetime of the agent (the total number of percepts it will receive)" (p. 47). Percepts are the granularity of time.

[3]The current state of the environment is the same as the current percept, incidentally, if the environment is fully observable.

Take Opportunity, for instance, which had a performance measure of $T = 90$ sol; as of 2012, it's overstepped T by eight years. If it fails after T to e.g. characterize soil, could you say that it acts rationally? In other words, is Spirit irrational; now that it has failed to meet its original performance measure?

No: by their original performance measure, Opportunity and Spirit are pararational; which is not to say that you couldn't define another performance measure u' depending upon another time T'.

See page 38, by the way, where the authors talk about rationality in terms of expected performance; could it be that an agent transcends T with respect to expected performance?

Example: given a penalty for each move, a reflex agent's expected performance would be just as good as any other's given T = 2; but not when T = 1000 (it would require a state-based agent to realize that the world is clean and stop moving).

17 DONE 2.2

`CLOSED:` *2012-06-11 Mon 17:18*

17.1 DONE a

`CLOSED:` *2012-06-11 Mon 17:18*

Page 38 describes an environment which is partially observable, deterministic and static; as such, the tabular agent in Fig. 2.3 can expect to maximize its utility in no more than four actions (the worst case is A: dirty, B: dirty; which results in either `suck, right, suck, left, ...` or `suck, left, suck, right, ...`).

There is no time to e.g. build a model of dirt, since the dirt doesn't replentish itself.

17.2 DONE b

`CLOSED:` *2012-06-11 Mon 17:18*

The agent does require internal state: it should know, for instance, whether it has cleaned every square; and, if so, should stop.

17.3 DONE c

CLOSED: *2012-06-11 Mon 17:18*

It should learn the geography of its environment to avoid wasting time trying to move off of it; it could maintain, furthermore, a dirt-distribution across the grid and favor those squares that tend to get dirty.

18 DONE 2.3

CLOSED: *2012-06-13 Wed 05:40*

a False. Page 42 mentions that, even in unobservable environments, "the agent's goals may still be achievable, sometimes with certainty;" the reflexive vacuum agent on page 38 is an example.

b True. In an unknown environment, there is no opportunity for the reflex agent to learn the "laws of physics" of the environment (p. 44); or for the programmer to endow the agent with them *a priori*.

c True. It's possible to imagine a task environment in which there are no decisions to be made: merely existing, for instance, satisfies the performance measure.

d False. According to page 46, the agent program takes the current percept; the agent function, on the other hand, takes the entire percept history.

e False. If the agent function is to e.g. determine whether a program will return an answer or run forever (see p. 8); it is not implementable by a program/machine combination. Unless, of course, the author (or agent) has solved the halting problem.

f True. Take the performance measure, for instance, where an agent is supposed to simulate the roll of a fair-sided die.

g True. If an agent is a rational NxN tic-tac-toe player, it will perform just as well in a 2x2 as in a 3x3 environment.

h False. See a: page 138 describes a sensorless vacuum agent that knows the geography of its world; it's possible to search its belief space and even coerce the world into certain states.

i False. Even rational poker-playing agents fall prey to luck.

19 DONE 2.4

`CLOSED:` *2012-06-13 Wed 06:47*

19.1 Soccer

Performance measure Score and defend

Environment Field

Actuators Kicking, thrwing, catching

Sensors Topology, ball, agents

Characteristics Fully observable, multiagent, stochastic, sequential, dynamic, continuous, known

19.2 Titan

Performance measure Like TiME for surface lakes, it would determine the presence of biological compounds.

Environment Titan

Actuators Drill, satellite, landing gear

Sensors Mass spectrometer, camera

Characteristics Partially observable, multiagent? stochastic, sequential, dynamic, continuous, known

19.3 Shopping on the internet

Performance measure Finding used AI books

Environment The internet

Actuators Form completion, HTTP request, cookie storage

Sensors HTML parser

Characteristics Partially observable, multiagent, stochastic, sequential, dynamic, continuous, known

19.4 Playing a tennis match

Performance measure Winning the match

Environment Tennis court

Actuators Tennis racket

Sensors Location, trajectory of ball, opponent; topology

Characteristics Fully observable, multiagent, stochastic, sequential, dynamic, continuous, known

19.5 Practicing tennis against a wall

Performance measure Length of rally

Environment Half-court with wall

Actuators See above.

Sensors See above (sans opponent).

Characteristics Fully observable, single agent, stochastic, sequential, dynamic, continuous, known

19.6 Performing a high jump

Performance measure Height jumped

Environment Measuring stick

Actuators Spring

Sensors Balance

Characteristics Fully observable, single agent, deterministic, episodic, static, continuous, known

19.7 Knitting a sweater

Performance measure Consistency of stitch, conformance to the recipient's body

Environment Yarn, recipient's body

Actuators Needle

Sensors Yarn on needle

Characteristics Fully observable, single agent, deterministic, sequential, static, continuous, known

19.8 Bidding on an item

Performance measure Win, save cash

Environment Auction

Actuators Signify bid

Sensors See the artifact, understand the auctioneer

Characteristics Partially observable [4], stochastic, sequential, dynamic, continuous, known

20 DONE 2.5

`CLOSED:` *2012-06-14 Thu 06:44*

Agent An agent is a black box with inputs and outputs that conspires to perform something

Agent function The agent function maps inputs to outputs.

Agent program The agent program implements the agent function.

[4] The agent's minds are unobservable: we have to operate in belief space.

Rationality Rationality usually means the application of reason; but because the authors have given up on AI as "thinking humanly" (p. 2), it has been cheapened to mean: "act in accordance with this performance measure we've set up."

Autonomy Autonomy is the ability of an agent to select actions beyond the *a priori* programming of its maker.

Reflex agent A reflex agent acts according to the immediate percept; it has amnesia.

Model-based agent A model-based agent acts according to a model of the world it has synthesized from percepts.

Goal-based agent Not merely reacting to the environment (or its model thereof), the goal-based agent has a Vorhaben (so to speak) that can inform sequences of actions.

Utility-based agent Utility-based agents have internalized their own performance measure; and, as such, are able to decide between conflicting goals.

Learning agent Learning agents hone their sense of appropriate actions by modifying the weights associated with environmental features.

21 DONE 2.6

`CLOSED:` *2012-06-14 Thu 12:06*

a There are infinite agent programs that implement a given agent function; take, for instance, an agent that perceives flashes of light and maps them to some output (say, an integer).

The percept sequence could be mapped as an integer encoded as a bit-string of light and dark moments; or a bit-array representing the same thing.

b Yes; an agent function whose performance measure is to determine whether a program stops or not cannot be implemented as a progrm (unless one first solves the Halting Problem).

c Yes; which is to say: a program implements a mapping from percepts to actions; to change the mapping, you have to change the program.

d There would be 2^n possible agent programs on an n-bit machine (not all of them functional).

(According to this, there are a^{2^n} possible programs; 2^n possible states and a choices for each state. I don't think they're factoring the program into the storage, are they?)

e Speeding up the agent program does not change the agent function; they are orthogonal: the former is concrete, the latter abstract.

If they don't behave like Hegelian dialectic, they are at least Platonic forms and instantiations.

22 DONE 2.7

`CLOSED:` *2012-06-20 Wed 03:52*

Clever: the goal-based agent mutates belief-space based on its best guess; acts accordingly.

22.1 Goal-based agent

```
- let
  - state
  - model
  - goals
  - action
    - define (goal-based-agent percept)
      - set! state (update-state state action percept model)
      - let
        # Shouldn't we distinguish between many different
        action-sequences; and, if so, how to do so without a utility
        function: evaluate them against the performance measure?
        - action-sequence (search goals state)
          - return (first action-sequence)
```

22.2 Utility-based agent

```
- let
```

```
- state
- model
- goals
- action
  - define (utility-based-agent percept)
    - set! state (update-state state action percept model)
    - let
      - probabilities (map probability goals)
      - utilities (map utility goals)
        - let
          - expected-utilities (map * probabilities utilities)
          - goal-of-maximum-expected-utility (max goals expected-utilities)
          - action-sequence (search goal-of-maximum-expected-utility state)
            - return (first action-sequence)
```

23 DONE 2.8

`CLOSED:` *2012-06-21 Thu 04:39*

- CLOSING NOTE *2012-06-21 Thu 04:39*
  See aima-chicken.

```
(use debug
     foof-loop
     lolevel
     srfi-1
     srfi-8
     srfi-13
     srfi-69
     vector-lib)

(define (simulate environment)
  (loop ((while (environment)))))

(define (compose-environments . environments)
  (lambda ()
    (every identity (map (lambda (environment)
                           (environment))
                         environments))))
```

```scheme
(define (make-performance-measuring-environment
         measure-performance
         score-update!)
  (lambda () (score-update! (measure-performance))))

(define (make-step-limited-environment steps)
  (let ((current-step 0))
    (lambda ()
      (set! current-step (+ current-step 1))
      (< current-step steps))))

;;; What about pairs of objects and optional display things.
(define make-debug-environment
  (case-lambda
   ((object) (make-debug-environment object pp))
   ((object display)
    (lambda () (display object)))))

(define (vacuum-world-display world)
  (pp
   (vector-append '#(world)
                  (vector-map
                   (lambda (i clean?)
                     (if clean? 'clean 'dirty))
                   world))))

(define clean #t)
(define clean? identity)

(define dirty #f)
(define dirty? (complement clean?))

(define left 0)
(define left? zero?)

(define right 1)
(define right? (complement zero?))

(define make-vacuum-world vector)
```

```
(define vacuum-world-location vector-ref)

(define vacuum-world-location-set! vector-set!)

(define-record vacuum-agent
  location
  score
  program)

(define-record-printer vacuum-agent
  (lambda (vacuum-agent output)
    (format output
            "#(agent ~a ~a)"
            (if (left? (vacuum-agent-location vacuum-agent))
                'left
                'right)
            (vacuum-agent-score vacuum-agent))))

(define (make-vacuum-environment world agent)
  (lambda ()
    (let* ((location (vacuum-agent-location agent))
           (action ((vacuum-agent-program agent)
                    location
                    (vacuum-world-location world location))))
      (case action
        ((left) (vacuum-agent-location-set! agent left))
        ((right) (vacuum-agent-location-set! agent right))
        ((suck) (vacuum-world-location-set! world location clean))
        (else (error (string-join
                      "make-vacuum-environment --"
                      "Unknown action")
                     action))))))

(define (reflex-vacuum-agent-program location clean?)
  (if clean?
      (if (left? location)
          'right
          'left)
      'suck))
```

```
(define make-reflex-vacuum-agent
  (case-lambda
   ((location)
    (make-reflex-vacuum-agent location reflex-vacuum-agent-program))
   ((location program)
    (make-vacuum-agent
     location
     0
     program))))

(define (make-vacuum-performance-measure world)
  (lambda ()
    (vector-count (lambda (i square) (clean? square)) world)))

(define (make-vacuum-score-update! agent)
  (lambda (score)
    (vacuum-agent-score-set! agent (+ (vacuum-agent-score agent)
                                      score))))

(define simulate-vacuum
  (case-lambda
   ((world agent) (simulate-vacuum world agent 1000))
   ((world agent steps)
    (simulate
     (compose-environments
      (make-step-limited-environment steps)
      (make-performance-measuring-environment
       (make-vacuum-performance-measure world)
       (make-vacuum-score-update! agent))
      (make-debug-environment agent)
      (make-debug-environment world vacuum-world-display)
      (make-vacuum-environment world agent)))
    (vacuum-agent-score agent))))

(simulate-vacuum (make-vacuum-world dirty clean)
                 (make-reflex-vacuum-agent
                  left
                  (lambda (location clean?)
                    'right))
```

```
          10)
```

I want environmental combinators, incidentally; such that I can compose an e.g. step-limited environment with an agent with a vacuum one.

We can compose steps; but how do you compose score: do you have to specify a reducer of some kind; e.g. addition? Is it really environment reduction we're talking about here?

I'm beginning to suspect that the performance score is a property of the agent, not the environment; this is consistent with the book's use of "reward" and "penalty." It also makes sense in a multi-agent environment.

On page 37, however, the authors state that:

> This notion of desirability [for a sequence of actions leading to a sequence of states] is captured by a **performance measure** that evaluates any given sequence of environment states.

I suspect that, whereas the environment is an arbiter of the performance score (i.e. applies the performance measure), the score inheres in the agents.

This is corroborated by the following:

> Notice that we said *environment* states, not *agent* states. If we define success in terms of agent's opinion of its own performance, an agent could achieve perfect rationality simply by deluding itself that its performance was perfect.

Since only the environment has access to its true states, it alone can measure performance. Is this problematic in cases where we don't have an omniscient environment that directly communicates performance scores? In such cases, we'd have to rely on the imperfect self-judgement of the agent; and attempt to converge on rationality by internal coherence.

What I'm calling environments, incidentally, are now just functions: step-functions, at that; and can be reduced by `every`.

Agent combinators are a little tough, though; the performance measure has to be aware of the combined features. Can we use some kind of message-passing mechanism?

What stops us, for instance, as modelling the agents as lambdas; too? Part of the problem is the inversion of control: we'd have to pass a message to the agent to store its score, as opposed to manipulating the score directly.

Every agent would be a dispatch-mechanism that would manage its own meta-variables (including score and e.g. location) on the basis of messages. Is it problematic, however, to have agents managing their own score? Could we have an agent $\rightarrow$ score mapping in the environment itself? That way, agents only maintain state according to its percepts.

Score, for instance, is not a percept in the vacuum world; location, however, is. Agents, then, are functions with closures; functions which take as many parameters as their percepts have components. The performance-measuring-environment, therefore, maintains an `agent->score` table. Yes!

Problem is, though, that we'd have to break the nice contract we have: environments are niladic lambdas. To maintain the performance measure table, we'd have to receive the agent and the new score.

How to make the performance measure part of the environment, so that we can relieve the agent from metadata?

By taking the metadata out of the agent, we have to maintain agent $\rightarrow$ metadata mappings in the environment; this is kind of a pain in the ass.

By maintaining agents-as-lambda, we get a certain flexibility; on the other hand, we shunt some complexity onto the environment: as it has to maintain agent-metadata: score, location, &c.

Is this an acceptable tradeoff? The alternative, where I need to guess what agents need (program, score, location) seems onerous; for some reason. In practice, however, it may be simpler. We haven't even solved the agent-hashing-problem, for instance (wherein hashing fails if we mutate a field).

Can we hash closures?

I want to follow this environment-maintains-agent-¿metadata-mapping thing and see how far it goes. (I see now why objects are interesting; closures, of course, do the same thing.)

If `make-*-environment` returned multiple values: the thunk followed by e.g. `agent->score`, `agent->location`; you can ignore the latter values, if you want to.

Or, we can demand that the user furnish them; better yet, we can give the user the option of furnishing and ignoring them.

Also, shouldn't we be able to name agents at some point? This would also have to fall within an external data structure. Maybe the record solution isn't problematic if we create ad-hoc agents for each problem.

If we really need to decouple the program from the agent metadata (do we?), one solution is to have an agent-¿metadata table in the environment; the metadata would be a record containing location, score, name, &c.

This metadata table, on the other hand, would have to be passed to each subenvironment for composition. Seems like a pain.

We found that, since environments consist of a step function, we could reduce them to a lambda; let's see if this continues to be the case. For the time being, however, I think using agent-records is simplifying.

I wouldn't mind agents being lambdas with closures; problem is: can't access the closure without some kind of message passing. (Message passing simulates records.) We could possibly do it with some kind of multiple-return-values hack, in which the subsequent values are ignored (the agent effectively does a state dump every time its program is invoked). The problem with that is that I have to pass a percept in to access its state, or store its state some other way.

To avoid namespacing everything (like e.g. `vacuum-agent`, &c.), I'd like to have separate modules; that way, if we need to, we can import with a prefix.

For learning purposes, we should allow the student to specify no more than the agent program; worry about all the bootstrapping on the back end.

We may have to copy worlds, incidentally, to compare how e.g. reflex- vs. state-agents behave; thank goodness for `vector-copy`. (Copy by default?)

To give feedback to students, should have an e.g. `environment-print` that we can pass around (this sort of function-passing, incidentally, is what Norvig sought to avoid); `environment-print` might happen at every step in e.g. `simulate`. Oh, `make-debugging-environment`.

24 DONE 2.9

`CLOSED:` *2012-06-28 Thu 12:47*

Using the aima-chicken framework:

```
(use aima-vacuum
     test)
```

```
(let ((worlds
       (list (make-world clean clean)
             (make-world clean clean)
             (make-world clean dirty)
             (make-world clean dirty)
             (make-world dirty clean)
             (make-world dirty clean)
             (make-world dirty dirty)
             (make-world dirty dirty)))
      (agents
       (list (make-reflex-agent left)
             (make-reflex-agent right)
             (make-reflex-agent left)
             (make-reflex-agent right)
             (make-reflex-agent left)
             (make-reflex-agent right)
             (make-reflex-agent left)
             (make-reflex-agent right))))
  (let* ((scores (map simulate-vacuum worlds agents))
         (average-score (/ (apply + scores) 8)))
    (test
     "Scores for each configuration"
     scores
     '(2000 2000 1998 1999 1999 1998 1996 1996))
    (test
     "Average overall score"
     1998.25
     average-score)))
```

25 DONE 2.10

CLOSED: *2012-06-29 Fri 17:52*

25.1 DONE a

CLOSED: *2012-06-29 Fri 13:29*

With a partially observable environment, a simple reflex agent will not be rational (in the sense that its expected performance is not as good as any other's); in other words, it should be scoring about twice as much as this:

```
(use aima-vacuum
     test)

(test
 "Penalizing vacuum with reflex agent"
 998
 (simulate-penalizing-vacuum (make-world dirty dirty)
                             (make-reflex-agent left)))
```

The reflex agent would require state to determine that e.g. the world was clean and that it didn't need to move anymore.

25.2 DONE b

`CLOSED:` *2012-06-29 Fri 17:52*

```
(use aima
     aima-vacuum
     test
     vector-lib)

(debug? #f)

(define-record unknown)

(define unknown (make-unknown))

(define (all-clean? world)
  ;; Vector bleeds a little world.
  (vector-every (lambda (location) (clean? location)) world))

(test
 "Stateful agent in penalizing environment"
 1995
 (simulate-penalizing-vacuum
  (make-world dirty dirty)
```

```
  (make-reflex-agent
   left
   ;; We could also make an initial pessimistic hypothesis of all-dirty.
   (let ((world (make-world unknown unknown)))
     (lambda (location clean?)
       (if clean?
           (begin
             ;; Extra work here every time; otherwise, we'd have an
             ;; extra `all-clean?' check after we set the state.
             ;; `vector-set!', I'd wager, is cheaper than
             ;; `all-clean?'.
             (vector-set! world location clean)
             (if (all-clean? world)
                 ;; Symbols appropriate here, or should we have predefined
                 ;; go-left, go-right, clean, do-nothing? We're message
                 ;; passing, after all; I suppose a lambda wouldn't make any
                 ;; sense?
                 ;;
                 ;; Can't be lambdas unless we redefine e.g. `go-right'
                 ;; to penalize in the case of
                 ;; `make-penalizing-environment'; better to keep as
                 ;; symbols and dispatch, right? There should be some
                 ;; sort of data-directed model we could use, though,
                 ;; instead of the case-based dispatch.
                 'noop
                 (if (right? location)
                     'left
                     'right)))
           'suck))))))

(test
 "Stateful agent in penalizing environment (from the egg)"
 1995
 (simulate-penalizing-vacuum
  (make-world dirty dirty)
  (make-stateful-reflex-agent left)))
```

25.3 DONE c

`CLOSED:` *2012-06-29 Fri 17:52*

- CLOSING NOTE *2012-06-29 Fri 17:52*
  Should we actually implement it?

If the simple and stateful reflex agents are omniscient w.r.t. the environment, they are equivalent; the stateful agent will simply update its state according to its omniscient percept and the simple one will simply act accordingly.

26 TODO 2.11

A simple reflex agent wouldn't be able to explore an environment of unknown extent without exhausting all possible paths of the corresponding $n \times n$ space; given sufficient time, such an exhaustive agent would asymptotically approach rationality toward $t = \infty$.

27 Meetups

27.1 Mon Jun 11 2012

- Had to redefine the rational from "exercizing reason" to "maximizing utility function" because they gave up an AI as thinking machines in the 60s.
- Mitochondria were once autonomous agents; cells as composite agents
- Thin vs. thick agents and skynet
- In games like poker, the mind of the adversarial agents are part of the environment; requires a theory of mind to discern things like: "is he bluffing?"

27.2 Mon Jun 18 2012

- David has the international version, which would have you write an essay on evolution and autonomy; see e.g. Turing on child AI:

We have thus divided our problem into two parts. The and child-programme the education process. These two remain very closely connected. We cannot expect to find a good child-machine at the first attempt. One must experiment with teaching one such machine and see how well it learns. One can then try another and see if it is better or worse. There is an obvious connection between this process and evolution, by the identifications

Structure of the child machine Hereditary material

Changes Mutations

Natural selection Judgment of the experimenter

One may hope, however, that this process will be more expeditious than evolution. The survival of the fittest is a slow method for measuring advantages. The experimenter, by the exercise of intelligence, should be able to speed it up. Equally important is the fact that he is not restricted to random mutations. If he can trace a cause for some weakness he can probably think of the kind of mutation which will improve it.

27.2.1 CANCELED Test a simple agent in each (Python, Java, Clojure) implementation.

`CLOSED:` *2012-06-20 Wed 03:19*

- CLOSING NOTE *2012-06-20 Wed 03:19*
  Looks like we're going to standardize on Clojure.

27.2.2 DONE Get some standard cables to connect to the projector.

`CLOSED:` *2012-06-20 Wed 03:19*

27.2.3 DONE See if we can use `xrandr` to get twin-view with an external HDMI.

`CLOSED:` *2012-06-15 Fri 06:04*

```
xrandr --output eDP1 --off
xrandr --output HDMI1 --mode 1280x720
```

27.3 Mon Jun 25 2012

27.3.1 Discussion

- 2.1

  Change performance measure; utility-based agent aware of its own performance measure: can react accordingly? Not a reflex agent, though, that's pre-programmed.

  Example: given a penalty for each move, a reflex agent's expected performance would be just as good as any other's given T = 2; but not when T = 1000 (it would require a state-based agent to realize that the world is clean and stop moving).

- 2.2

  c Memory, motor, bump sensor (or penalty); learn geography, probability of becoming dirty. Clustering algorithm: centers of mass for dirt.

- 2.3

  a Best action given available information.

  c One-square, not dirty.

  d Page 51 (program) vs. page 36 (function).

  f See **c** above.

  g Stochastic vs. deterministic vacuum world: reflex agents are still rational. Performance measure still the same.

  i Even an omniscient poker player is subject to luck.

27.3.2 TODO Get a minimal Clojure example up here.

27.3.3 TODO Set up `csrg.org` with a mailing list.

This is an alternative to e.g. Google groups and whatever mechanism Meetup has.

(Or is it `csrg.com`? It is indeed `csrg.org`.)

28 Notes

28.1 1

- Two dimensions: thought vs. action, humanity vs. rationality.
- Physical simulation of a person is unnecessary for intelligence.
  - Mind-body dualism of Descartes?
- Cognitive science brings together computer models from AI and experimental techniques from psychology.
- Real cognitive science, however, is necessarily based on experimental investigation of actual humans.
- The standard of rationality is mathematically well defined and completely general.
- We will adopt the working hypothesis that perfect rationality is a good starting point for analysis.
- Limited rationality: acting appropriately when there is not enough time
- Materialism, which holds that the brain's operation according to the laws of physicas constitutes the mind.
- Logical positivism
- Carnap, The Logical Structures of the World, was probably the first theory of mind as a computational process.
- Intelligence requires action as well as reasoning.

- Actions are justified by a logical connection between goals and knowledge of the action's outcome.
- Regression planning system
- The leap to a formal science required a level of mathematical formalization: logic, computation, probability.
- The world is an extremely large problem instance.
- Models based on satisficing—making decisions that are "good enough"—gave a better description of actual human behavior.
- Searle: brains cause minds.
- Behaviorism
- "A cognitive theory should be like a computer program."
- Intelligence and an artifact
- Parallelism—a curious convergence with the properties of the brain.
- The state of a neuron was conceived of as "factually equivalent to a proposition which proposed its adequate stimulus." McCulloch and Pitts (1943)
  - Neural events and the relations among them can be treated by means of propositional logic.
  - For any logical expression satisfying certain conditions, one can find a net behaving in the fashion it describes.
  - For every net behaving under one assumption, there exists another net which behaves under the other and gives the same results.
- Perhaps "computational rationality" would have been more precise and less threatening, but "AI" stuck.
- AI from the start embraced the idea of duplicating human faculties such as creativity.
- John McCarthy referred to this period as the "Look, Ma, no hands!" era.
- "A physical symbol system has the necessary and sufficient means for general intelligent action."

- 1958 . . . McCarthy define Lisp, which was to become the dominant AI programming language for the next 30 years.
- It is useful to have a formal, explicit representation of the world and its workings and to be able to maniplutae that representation with deductive processes.
- McCarthy, Programs with Common Sense
  - In this program the procedures will be described as much as possible in the language itself and, in particular, the heuristics are all so described.
  - If one wants a machine to be able to discover an abstraction, it seems most likely that the machine must be able to represent this abstraction in some relatively simple way.
  - The improving mechanism should be improvable.
  - Must have or evolve concepts of partial success.
    - Something about ˜1995 that made for a cute blog.
  - For example, to mest people, the number 3812 is not an object: they have nothing to say about it except what can be deduced from its structure. On the other hand, to most Americans the number 1776 is an object because they have filed somewhere the fact that it represents the year when the American Revolution started.
  - One might conjecture that division in man between conscious and unconscious thought occurs at the boundary between stimulus-response heuristics which do not have to be reasoned about but only obeyed, and the others which have to serve as premises in deductions.
- Machine evolution (genetic algorithms): Friedberg, 1958, 1959.
  - Friedberg. 1958. A learning machine Part 1. IBM Journal of Research and Development, 2, 2–13.
    - From and intent, to be sure, are related quite discontinuously in the compact, economical programs that programmers wrte.

  - Friedberg, Dunham, North. 1959. A learning machine, Part 2. IBM Journal of Research and Development, 3, 282–287.

- Failure to come to grips with the “combinatorial explosion”
- The new back-propagation learning algorithms for multilayer netwrks that were to cause an enormous resurgence in neural-net research in the late 1980s were actually discovered first in 1969.
- Bruce Buchanan: a philosopher turned computer scientist
- DENDRAL was the first successful knowledge-intensive system (expert system).
- AI Winter
- Parallel Distributed Processing (Rumelhart, McClelland. 1986)
- Connectionist models: competitors to symbols models and logicist approach
- Ones that act rationally according to the laws of decision theory and do not try to imitate the thought steps of human experts
- Control theory deals with designing devices that act optimally on the basis of feedback from the environment.

28.2 2

- Rational agents
- Agents behaves as well as possible (utility function?)
- Agent perceives its environment through sensors and acts through actuators.
  - Hands are actuators and sensors.
- Percept :: agent’s perceptual inputs at any given instant
- Agent’s choice depends on percept sequence to date.
- Agent function :: maps percept sequence to action.
- External characterization of agent (agent function): table mapping percept sequences to actions; internally: agent program.

- In a sense, all areas of engineering can be seen as designing artifacts that intaract with the world.
  - Trivializing agents to view e.g. calculators as such.
- Intelligent agents, on the other hand: non-trivial decision making.
- Rational agents: does the right thing (utility).
- Performance measure
  - (This all sounds reminiscent of Mitchell, by the way.)
- Sequence of actions causes the environment to go through states: environmental states are distinct from agent states.
  - Basing performance merely off of agent-states is a form of coherentism.
- Design performance measures according to what one actually wants in the environment.
- "We leave these question as an exercise for the diligent reader."
  - Classic.
- Rationality: performance measure, agent's prior (i.e. *a priori*) knowledge, agent's actions, agent's percept sequence.[5]
  - "Percept," it turns out, is the converse of "concept": "A Percept or Intuition is a single representation . . . a Concept is a collective (general or universal) representation of a whole class of things." (F. C. Bowen Treat. Logic)
- For each percept sequence, a rational agent should select an action that is expected to maximize its performance measure, given its percept sequence and a priori knowledge.
- Omniscience vs. rationality
- Rationality maximizes *expected* performance; perfection, *actual* performance.
- Our definition of rationality does not require omniscience.
  - It's possible sometimes, by the way, to detect transitions in authorship.

[5] Bizarre to me that a programmer is responsible for the *a priori*; playing god, anyone?

- Information gathering: actions in order to modify future percepts.
- *a priori* rather than percepts: lacks autonomy.
- Ration agent: autonomous; boostrap with *a priori*, though.
- Just as evolution provides animals with built-in reflexes to survive long enough to learn for themselves
- Task environments
- PEAS :: Performance, Environment, Actuators, Sensors
  - Mitchell has: task, performance measure, training experience, target function, target function representation.
- Fully observable vs. partially observable environment.
- Task environment effectively fully observable if the sensors detect all aspects that are *relevant* to the choice of action, performance measure.
- Single agent vs. multiagent
- Entity *may* vs. *must* be viewed as an agent.
- Competitive vs. cooperative multiagent environment
- Communication
- In some competitive environments, randomized behavior is rational because it avoids predictability.
- Deterministic vs. stochastic environment
- "Uncertain" environment: not fully observable or not deterministic
- Stochastic: uncertainty about outcomes quantified in terms of probabilities; nondeterministic: actions characterized by possible outcomes, no probabilities attached.
- Episodal vs. sequential: atomic episodes: receives percept and performs single action; sequential: current decision affect all future decisions.
- Static vs. dynamic: environment change while agent is deliberating.
- Discrete vs. continuous: state of the environment, time, percepts, actions.
- Known vs. unknown: "laws of physics" of the environment

- Hardest: partially observable, multiagent, stochastic, sequential, dynamic, continuous, unknown.
- Code repository includes environment simulator that places one or more agents in a simulated environment, observes their behavior over time, evaluates them according to a given performance measure.
  - Shit: this is something we could implement in Scheme (java, python, lisp, data); lot of work, though? Glory?
    - A lot of the utilities are in SRFI-1; e.g. `transpose` is `zip`.
    - Infinity is there.
    - Might have to write `rms-error`; `ms-error`.
    - `sample-with-replacement`; `sample-without-replacement`
      - Combinatorics SRFI, anyone?
    - `fuzz`
    - `print-grid`, &c.
    - `the-biggest`, `the-biggest-random-tie`, `the-biggest-that`, &c.
    - Binary tree stuff
    - Queue
    - Heap
    - They did CLOS-like stuff
  - Damn, they put some work in; could do it incrementally? Will be ad-hoc, I guarantee it.
  - Maybe we can program it, given the agents API.
    - `run-environment` looks like something central.
  - What would happen if we merely translated the code to Chicken? Could do so, perhaps, without fully understanding it; write an idiomatically Scheme-port later.
  - In that case, find some alternative to CLOS; or use tinyCLOS?
  - Also, beginning to think that we misnamed our repo: we're calling it `aima`, but we'd like to write an `aima` egg; with `aima-agents`,

    `aima-search`, `aima-logic`, `aima-planning`, `aima-uncertainty`, `aima-learning`, `aima-language` modules.

      * Call it `aima-egg`? `aima-chicken`?

  - Translation seems like the way to go: relatively mechanical.
  - Incidentally, "We need a new language for logical expressions, since we don't have all the nice characters (like upside-down A) that we would like to use." We can use in Scheme, can't we? Sure. Tough to type? Maybe. Also, think font-lock.
  - May not be up-to-date for 3e; let's see; also, rife with `defmethod` and other OOisms. Can ignore it, possibly, and its type-checking; `defstructure` is similar, I think, to SRFI-9.
  - Damn, they implemented unification.
  - Not to mention: the learning stuff (e.g. decision trees).
  - Man, should we implement this stuff ad-hoc; or otherwise depend on the existing implementations?
  - Path of least resistance: do it in Allegro? Ouch.

- The job of AI is to design an agent program that implements the agent function, the mapping from percepts to actions.
- Agent = architecture + program
- The agent program takes the current percept as input; the agent function, which takes the entire percept history.
- The agent function that the program embodies
- Write programs that produce rational behavior from a smallish program rather than a vast table.
- Reflex agents; model-based reflex agents; goal-based agents; utility-based agents.
- ```
  - table-driven-agent percept
    - persistent
      - percepts ()
      - table
    - append percept to percepts
    - lookup percepts table
  ```
  ```

- ```
  - reflex-vacuum-agent location status
    - if dirty? status
      - suck
      - else if location = A
        - right
      - else
        - left
  ```
- Simple reflex agent: select actions on the basis of the current precept
  - Learned responses, innate reflexes
  - ```
    - simple-reflex-agent percept
      - persistent
        - rules
      - state = interpret-input(percept)
      - rule = rule-match(state rules)
      - rule.action
    ```
  - Works only if the correct decision can be made on the current percept: i.e. if the environment is fully observable.
  - Escape from infinite loops is possible if the agent can randomize its actions.
  - In single-agent environments, randomization is usually not rational.
  - In most cases we can do better with more sophisticated deterministic agents.
- Model-based reflex agents
  - Partial observability: keep track of the part of the world it can't see now.
  - Internal state
  - Knowledge of how world evolves independently from agent
  - Knowledge of actions affect the world
  - Model of the world
  - ```
    - model-based-reflex-agent percept
      - persistent
        - state
    ```
    ```

```
    - model
    - rules
    - action
  - state = update-state(state action percept model)
  - rule = rule-match(state)
  - (action rule)
```

  - State of the world can contain goals.

- Goal-based agents
  - In addition to current state, goal information that describes situations that are desirable
  - Search, planning
  - Flexible, reason about world vis vis goals

- Utility-based agents
  - Whereas goals are happy/unhappy, more general performance measure: utility
  - Utility functional: internalization of performance measure
  - This is not the only way to be rational: rational agent for vacuum has no idea what its utility function is (see exercise 1.3).
  - When there are conflicting goals, utility function opts for the appropriate tradeoff.
  - Partial observability, stochasticity: decision making under uncertainty.
  - Expected utility of the action outcomes: derive, given the probabilities and utilities of each outcome.
  - Any rational agent must behave as if it possesses a utility function.
  - An agent that possesses an explicit utility function can make rational decisions with a general-purpose algorithm that does not depend on the specific utility function being maximized.
  - Global rationality: local constraint on rational-agent designs.
  - Choosing utility-maximizing course of action

- Learning agents
  - Now we're getting into some Mitchell-action: critic, learning element, performance element, problem generator, &c.
    - Need a book on big data?
  - Operate in initially unknown environments and become more competent.
  - Learning element
    - Making improvements
  - Performance element
    - Selecting external actions
  - Critic
    - How performance element should be modified vis vis fixed performance standard.
    - Performance standard must be fixed (i.e. checkmate).
    - Performance standard outside agent; conforms thereto.
  - Problem generator
    - Suggesting actions that will lead to new and informative experiences.
    - Suboptimal actions short run, better actions long run.
  - Utility-based agents learning utility information
  - Performance standard distinguishes percept as reward or penalty.
  - Process of modification of each component
- Atomic, factored, structured environments
  - Atomic
    - Each state of the world indivisible
  - Factored
    - Variables, attributes
  - Structured

    - Objects and relationships can be described explicitly
  - Increasing expressiveness
  - Intelligent systems: operate at all points along the expressiveness-axis simultaneously.
- Agents perceives and acts; agent function maps percept seq -¿ action.
- Performance measure evaluates behavior.
- Maximize expected performance measure.
- Task environment: performance measure, external environment, actuators, sensors.
- Nicomachean Ethics
- McCarthy, Programs with Common Sense
- Newell and Simon, Human Problem Solving
- Horvitz suggests the use of rationality conceived as the maximization of expected utility as the basis for AI. Pearl, 1988.[6]
  - Horvitz, E., 1988. Reasoning Under Varying and Uncertain Resource Constraints, Proc. of the 7th National Conference on AI, Minneapolis, MN, Morgan Kauffman, pp:111-116.
  - Horvitz, E. J., Breese, J.S., Henrion, M. 1988. Decision The- ory in Expert Systems and Artificial Intelligence, Journal of Approximate Reasoning, 2, pp247-302.

28.3 Lectures

28.3.1 1

- AI: mapping from sensors to actuators
  - Voice, child-like engagement
- Fully vs. partially observable
- Deterministic vs. stochastic

[6] Which of these following papers do you think he's talking about? Probably the latter: it carries an *et al.*

- Discrete vs. continuous
- Benign vs. adversarial
- Uncertainty management

28.3.2 2

- Initial state
- `actions`$(state) \to a_1, a_2, a_3, \ldots$
- `result`$(state, action) \to state'$
- `goal-test`$(state) \to T|F$
- `path-cost`$(state \xrightarrow{action} state \xrightarrow{action} state) \to n$
- `step-cost`$(state, action, state') \to n$
- Navigate the state space by applying actions
- Separate state into three parts: ends of paths (frontier); explored and unexplored regions.
- Step-cost
- Tree-search
  - Family-resemlance; difference: which path to look at first.
- Depth-first search: shortest-first search

28.4 Turing, Computing Machinery and Intelligence

- Can machines think?
- It is A's object in the game to try and cause C to make the wrong identification.
  - Didn't realize there was an adversarial element to the Turing test.
- What will happen when a machine takes the part of A in this game?
- . . . drawing a fairly sharp line between the physical and the intellectual capacities of man.

  - A reasonable dualism

- May not machines carry out something which ought to be described as thinking but which is very different from what a man does?
  - The humanity/rationality plane of AI?

- Imitation game
  - Simulacrum sufficeth

- It is probably possible to rear a complete individual from a single cell of the skin (say) of a man . . . but we would not be inclined to regard it as a case of "constructing a thinking machine".

- Digital computer:
  1. Store
  2. Executive unit
  3. Control

- It is not normally possible to determine from observing a machine whether it has a random element, for a similar effect can be produced by such devices as making the choices depend on the digits of the decimal for π.

- Discrete state machines: strictly speaking there are no such machines. Everything really moves continuously.

- This is reminiscent of Laplace's view that from the complete state of the universe at one moment of time, as described by the positions and velocities of all particles, it should be possible to predict all future states.

- This special property of digital computers, that they can mimic any discrete state machine, is described by saying that they are universal machines.

- "Are there imaginable digital computers which would do well in the imitation game?" $\rightarrow$ "Are there discrete state machines which would do well?"

- I believe that in about fifty years' time it will be possible to programme computers, with a storage capacity of about 10^9, to make them play the imitation game so well that an average interrogator will not have more than 70% chance of making the right identification after five minutes of questioning.

  - Russell/Norvig, 12: storage units: $10^{1}5$
  - Loebner prize:

    > Elbot of Artificial Solutions won the 2008 Loebner Prize bronze award, for most human-like artificial conversational entity, through fooling three of the twelve judges who interrogated it (in the human-parallel comparisons) into believing it was human. This is coming very close to the 30% traditionally required to consider that a program has actually passed the Turing test.

    * From a judge:

      > He predicted that by the end of the century, computers would have a 30 per cent chance of being mistaken for a human being in five minutes of text-based conversation.

      I thought this was mistaken (should be 70), but it is indeed correct.

  - In other words, a damn-good guess.

- Conjectures are of great importance since they suggest useful lines of research.
- We might expect that He would only exercise this power in conjunction with a mutation which provided the elephant with an appropriately improved brain to minister to the needs of this soul.
- We like to believe that Man is in some subtle way superior to the rest of creation.
- "The consequences of machines thinking would be too dreadful." I do not think that this argument sufficiently substantial to require refutation. Consolation would would be more appropriate: perhaps this should be sought the transmigration of souls.
- There are limitations to the powers of discrete-state machines. The best known of these results is known as Gdel's theorem, and shows that in any sufficiently powerful logical system statements can be for-

mulated which can neither be proved nor disproved within the system, unless possibly the system itself is inconsistent.

- "Will this machine every answer 'Yes' to any question?" It can be shown that the answer is either wrong or not forthcoming.
- The only way to know that a man thinks is to be that particular man. It is in fact the solipsist point of view.
- I do not wish to give the impression that I think there is no mystery about consciousness. There is, for instance, something of a paradox connected with any attempt to localise it.
- When a burnt child fears the fire and shows that he fears it by avoiding it, I should say that he was applying scientific induction.
- It would deliberately introduce mistakes in a manner calculated to confuse the interrogator.
- By observing the results of its own behaviour it can modify its own programmes so as to achieve some purpose more effectively.
- This is the assumption that as soon as a fact is presented to a mind all consequences of that fact spring into the mind simultaneously with it.
- The undistributed middle is glaring.
- I would defy anyone to learn from these replies sufficint about the programme to be able to predict any replies to untried values.
- A smallish proportion are super-critical. An idea presented to such a mind may give rise to a whole "theory" consisting of secondary, tertiary and more remote ideas.
  - Sponteneity
- These last two paragraphs should be described as "recitations tending to produce belief."
- The only satisfactory support that can be given will be that provided by waiting for the end of the century and then doing the experiment described.
- Estimates for the storage capacity of the brain vary from $10^{1}0$ to $10^{1}5$ binary digits.

- Russell/Norvig (12): 10^13 synapses
- At my present rate of working I produce about a thousand digits of programme a day, so that about sixty workers, working steadily through the fifty years might accomplish the job, if nothing went into the waste-paper basket.
  - Mythical man-month?
- The child-programme and the education process
- One might have a complete system of logical inference "built in". The store would be largely occupied with definitions and propositions. Certain propositions may be described as "imperatives". As soon as an imperative is classed as "well-established" the appropriate action takes place.
  - Compare McCarthy, Programs with Common Sense, regarding imperatives.
- These choices make the difference between a brilliant and a footling reasoner.
- We can only see a short distance ahead, but we can see plenty there that needs to be done.

29 Tasks

29.1 TODO Find a reasonable pseudocode package in LaTeX.

See this survey. Algorithm2e isn't bad; doesn't seem to have a function, though. Pseudocode seems to be relatively natural; even if the output is a little ugly.

The alternative, I suppose, is straight up lists.

Algorithm2e has to be wrapped in dollars, which sucks; also: bizarre camel-case macros. Looks good, otherwise. Has no functions, apparently, either.

29.2 TODO Should we tangle to a bunch of text files?

Looking for an alternative to the big-ass pdf.

29.3 DONE Reimplement the Lisp environment in Scheme.

`CLOSED:` *2012-06-01 Fri 03:09*

Should we try to map CLOS to coops? Or maybe TinyCLOS would suffice. This takes balls. See aima-chicken.

29.4 DONE Personal notes as footnotes.

`CLOSED:` *2012-06-01 Fri 03:09*

29.5 CANCELED Should we try to release an e.g. Wumpus World egg?

`CLOSED:` *2012-06-01 Fri 03:08*

- CLOSING NOTE *2012-06-01 Fri 03:08*
  This is superseded by the Chicken port of the Lisp implementation (aima-chicken).

I wonder if it would be worthwhile to study the canonical Lisp examples.