My Short Course on Machine Self-Reference

John Case

Department of Computer and Information Sciences
University of Delaware
Newark, DE 19716 USA
Email: case@cis.udel.edu

Course Outline:

- Introductory talk on Machine Self-Reference And The Theater Of Consciousness.
- Relevant Mathematical Preliminaries in Theory of Computation.
- Large number of **illustrative** results proved by Machine Self-Reference.
  - Most from General computability Theory.
  - Some from Computability Theoretic Learning Theory.
- As time permits: Survey of results attempting to insightfully mathematically characterize or otherwise understand Machine Self-Reference.

Course Math References

[Cas91] J. Case. Effectivizing inseparability. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 37:97–111, 1991. Typos in journal version corrected in version at http://www.cis.udel.edu/∼case/papers/mkdelta.pdf.

[Cas94] J. Case. Infinitary self-reference in learning theory. *Journal of Experimental and Theoretical Artificial Intelligence*, 6:3–16, 1994.

[Cas74] J. Case. Periodicity in generations of automata. *Mathematical Systems Theory*, 8:15–32, 1974.

[CM07a] J. Case and S. Moelius. Characterizing programming systems allowing program self-reference. *Computation and Logic in the Real World - 3rd Conference of Computability in Europe*, volume 4497 of *LNCS*, pages 115–124. Springer, 2007. Journal version accepted for the associated special issue of *Theory of Computing Systems*, 2008.

[CM07b] J. Case and S. Moelius. Properties complementary to program self-reference. *Proceedings of the 32nd International Symposium on Mathematical Foundations of Computer Science 2007*, volume 4708 of *LNCS*, pages 253–263. Springer, 2007.

[JORS99] S. Jain, D. Osherson, J. Royer, and A. Sharma. *Systems that Learn: An Introduction to Learning Theory*. MIT Press, Cambridge, Mass., 2nd edition, 1999.

[Odi99] P. Odifreddi. *Classical Recursion Theory*, volume II. Elsivier, 1999.

[Ric81] G. Riccardi. The independence of control structures in abstract programming systems. *Journal of Computer and System Sciences*, 22:107–143, 1981.

[Rog58] H. Rogers. Gödel numberings of partial recursive functions. *Journal of Symbolic Logic*, 23:331–341, 1958.

[Rog67] H. Rogers. *Theory of Recursive Functions and Effective Computability*. McGraw Hill, New York, 1967. Reprinted, MIT Press, 1987.

[Roy87] J. Royer. *A Connotational Theory of Program Structure*. Lecture Notes in Computer Science 273. Springer-Verlag, 1987.

[RC94] J. Royer and J. Case. *Subrecursive Programming Systems: Complexity and Succinctness*. Progress in Theoretical Computer Science. Birkhäuser Boston, 1994.

Machine Self-Reference And The Theater Of Consciousness

John Case

Department of Computer and Information Sciences
University of Delaware
Newark, DE 19716 USA
Email: case@cis.udel.edu
Talk: www.cis.udel.edu/∼case/siena.pdf

Talk Outline:

- Brief history of linguistic self-reference in mathematical logic.
- Meaning, achievement & applications of machine self-reference.
- Self-modeling/self-reflection: segue from machine case to the human refective component of consciousness (other aspects of the complex phenomenon of consciousness, e.g., awareness and qualia, are not treated).
- What use is self-modeling/reference? Lessons from machine cases. Summary and What the Brain Scientist Should Look For!

Background:
Self-Referential
Paradoxes of LANGUAGE

Epimenedes' Liar Paradox
(7th Century BC)

Modern Form:
"This sentence is false."

Mathematical Logic (1930's+):

Paradox $\xrightarrow{\text{Resolved}}$ **Theorems**

Examples:

Gödel (1931) & Tarski (1933)

Liar Paradox $\overset{\text{Resolved}}{\longrightarrow}$ Suitable Mathematical Systems cannot express their own truth.

Gödel (1931)

Liar Paradox $\overset{\text{Transformed}}{\longrightarrow}$

"This sentence is not provable" $\overset{\text{Resolved}}{\longrightarrow}$ Suitable Mathematical Systems with Algorithmically Decidable Sets of Axioms are Incomplete (have unprovable truths).

An Essence of These Arguments:

Sentences which assert something about them**selves**

" . . . blah blah blah . . . about **self**."

This talk is about **self-referential** (syn: self-reflecting) **MACHINES** (Kleene 1936) — not sentences.

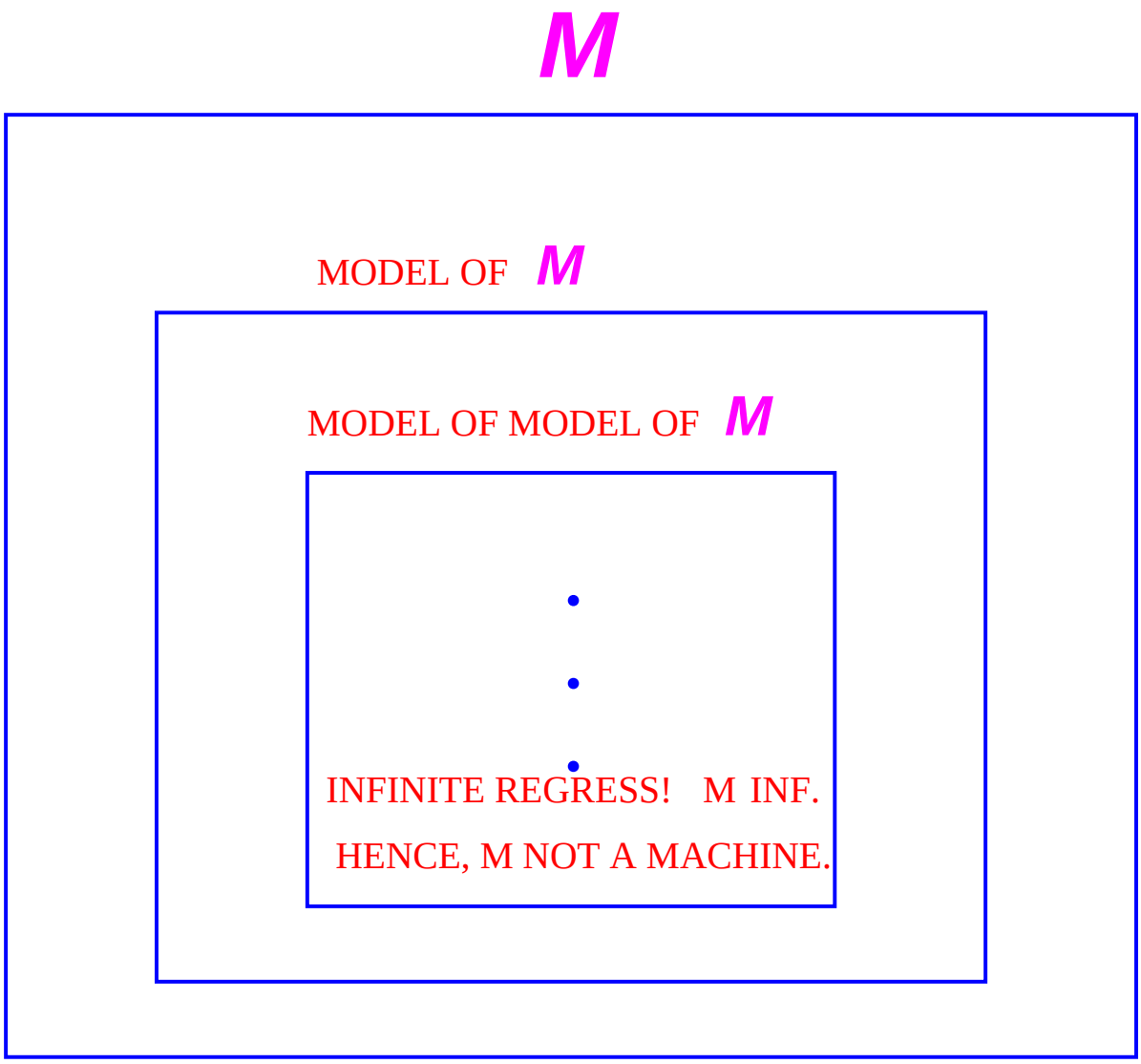
While self-referential sentences assert something about themselves, self-referential machines compute something about themselves.

Problem

Can machines take their **entire** internal mechanism into account as data? Can they **have** "complete self-knowledge" **and use it** in their decisions and computations?

We need to make sure there is **not** some **inherent** paradox in this — Not a problem in the linguistic case.

1. CAN MACHINES CONTAIN A COMPLETE MODEL OF THEMSELVES?



THEREFORE, M CANNOT CONTAIN A MODEL OF ITSELF!

So —

2. Can machines **create** a model of themselves — **external to** themselves?

YES! — by:

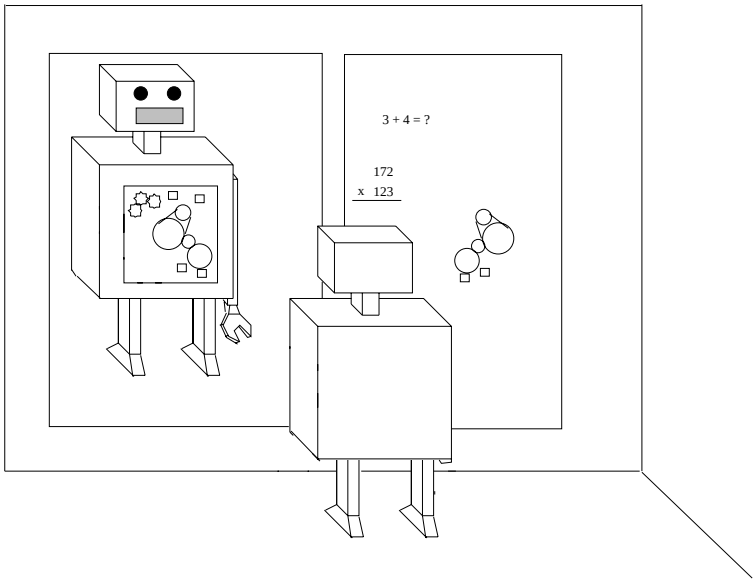
a. Self-Replication or

b. Mirrors.

We're gonna do it with mirrors!

— No smoke, just mirrors.

Later in course we'll explore Self-Replication approach.



The robot has a **transparent** front so its internal mechanism is visible. It faces a **mirror** and a writing board, the latter for "calculations."

It is shown having copied already a portion of its internal mechanism, corrected for mirror reversal, onto the board. It will copy the rest.

Then it can do **anything preassigned and algorithmic** with its board data consisting of: its **complete** (low-level) self-model **and** any other data.

As we will see, above essentially depicts **Kleene's Strong Recursion Theorem (1936)** from Computability Theory (see [Cas94,RC94]).

As an informal application of Kleene's Recursion Theorem, i.e., of machine self-reference/self-reflection, I'll give a very informal, pictorial proof of a **fundamental**, standard theorem about the **limitations** of machines —

More particularly:

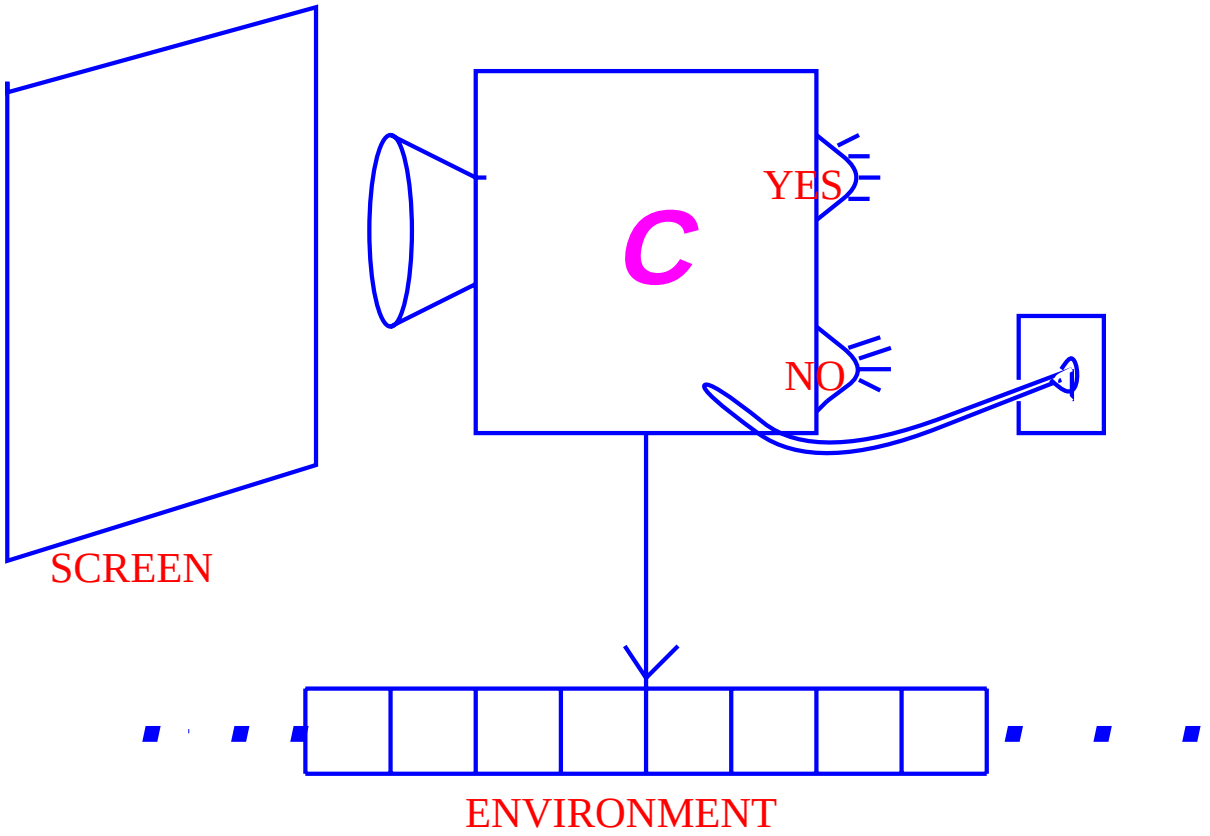
Q. Is there a (possible) machine which, when shown the underlying (static) mechanism of any machine M, predicts correctly whether or not M, once started, will ever (in principle) halt?

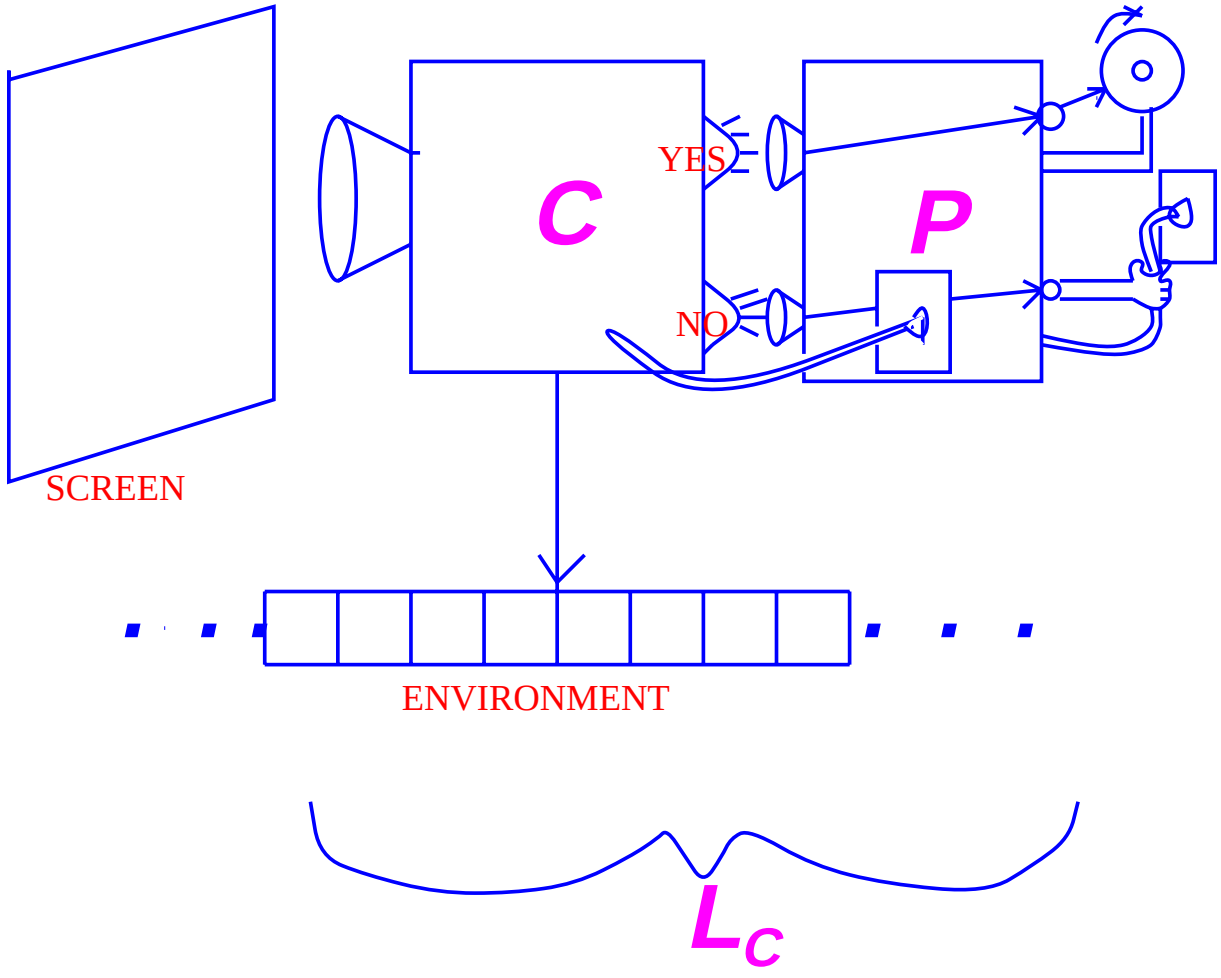
A.

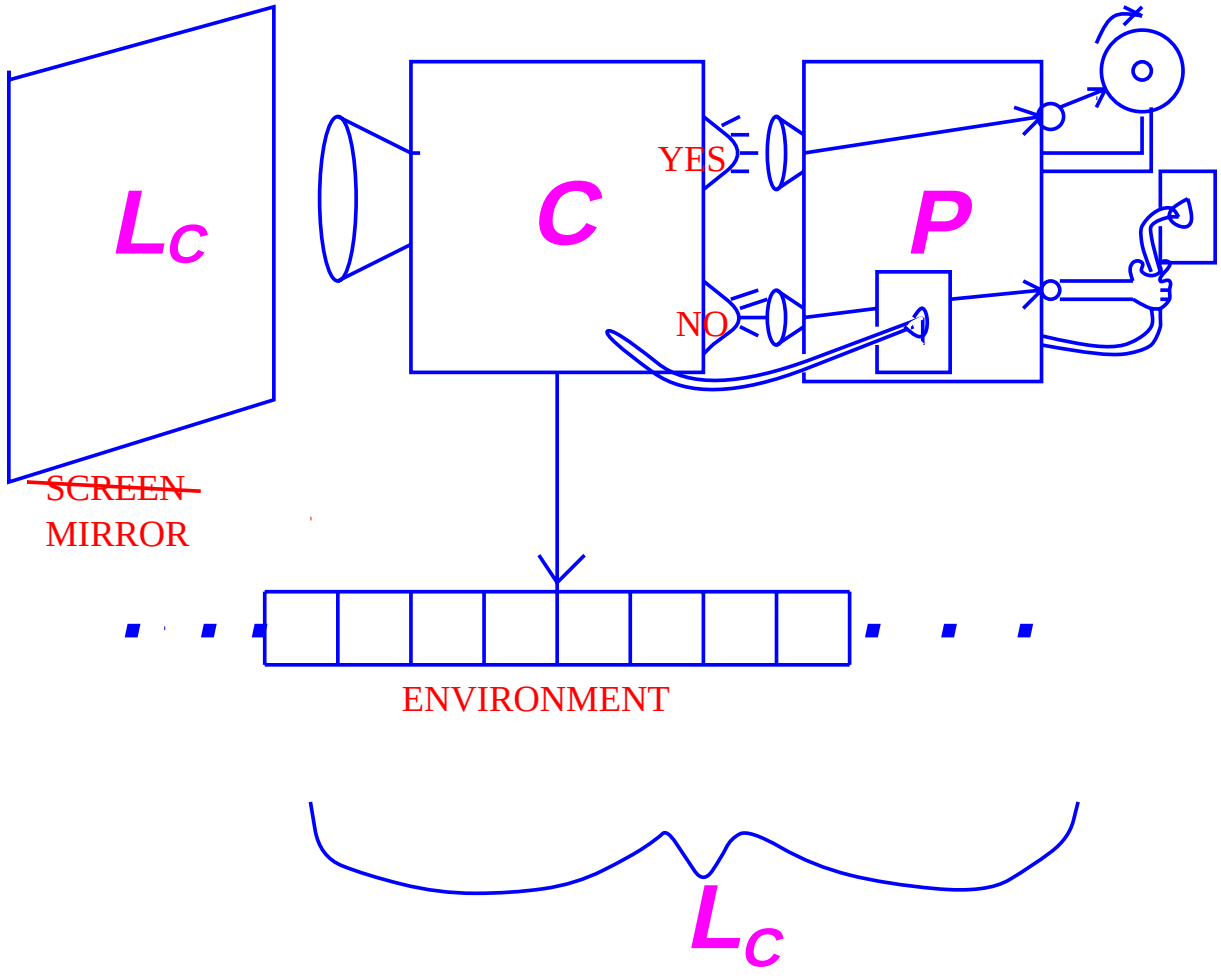
Theorem (≈ Turing 1936)

NO!

Informal Non-Standard Proof:


YES
C
NO
SCREEN
ENVIRONMENT


C
YES
NO
P
SCREEN
ENVIRONMENT
L_C


L_C
C
YES
NO
P
SCREEN
MIRROR
ENVIRONMENT
L_C

Next some formality: Fix a standard formalism for computing all the (partial) computable functions mapping tuples from $\mathbb{N}$ (the set of non-negative integers) into $\mathbb{N}$. Numerically name/code the programs/machines in this formalism onto $\mathbb{N}$. Let $\varphi_p(\cdot, \ldots, \cdot)$ be the (partial) function (of the indicated number of arguments) computed by program number p in the formalism.

Kleene's Theorem

$$(\forall p)(\exists e)(\forall x)[\varphi_e(x) = \varphi_p(e, x)].$$

p plays role of an arbitrary preassigned use to make of self-model. e is a self-knowing program/machine corresponding to p. x is any input to e. Basically, e on x, creates a self-copy (by a mirror or by replicating like a bacterium) and, then, runs p on (the self-copy, x).

In any natural programming system with efficient (linear time) numerical naming/coding of programs, passing from any p to a corresponding e can be done in linear time; furthermore, e itself efficiently runs in time $\mathcal{O}$(the length of p in bits + the run time of p) [RC94].

Following provides a program e which, shown any input x, decides whether x is a (perfect) self-copy (of e).

Proposition

$$(\exists e)(\forall x)[\varphi_e(x) = \begin{cases} 1, & \text{if } x = e; \\ 0, & \text{if } x \neq e \end{cases}].$$

Proof.

e on x creates a self-copy and, then, compares x to the self-copy, outputting 1 if they match, 0 if not. p here is **implicit**; it's the use **just described** that e makes of its self-copy.

□

Some Points:

a. There are not-so-natural programming systems with**out** Kleene's Theorem but which suffice for computing **all** the partial computable functions (mapping tuples from $\mathbb{N}$ into $\mathbb{N}$). Proof later in the course.

b. Self-simulation can be practical, e.g., a **Science** article [BZL06] reports experiments showing that **self-modeling in robots** enables them to compensate for injuries to their locomotive functions.

c. Each of next two slides provides a **succinct**, **game-theoretic** application of machine self-reference which shows a result about program **succinctness**.*

*Our pictorial proof of the Algorithmic Unsolvability of the Machine Halting Problem is also succinct & game-theoretic: In a two move, two player game, think of Candidate machine C as the move of player 1 and the self-referential machine $e = L_C$ as the move of player 2. Player 2's goal is to have the theorem be true; 1's is the opposite. Player 2's strategy involves e's using self-knowledge (and knowledge of C) to do the opposite of what C says $e = L_C$ will do regarding halting.

Let $s(p) \stackrel{\text{def}}{=} \lceil \log_2 p \rceil$, the size of program/machine number p in bits.

Proposition Let H be any (possibly horrendous) computable function (e.g., $H(x) = 100^{100} + 2^{2^{2^{2^x}}}$). Then

$$(\exists e)(\exists D, \text{ a } \textbf{finite} \text{ set} \mid \varphi_e = C_D)[|D| > H(s(e))].$$

Intuitively, e does **not** decide D by table look-up since a table for the huge D would not fit in the H-smaller e.

Proof.

By Kleene's Theorem,

$$(\exists e)[\varphi_e = C_{\{x \mid x \leq H(s(e))\}}].$$

Let $D = \{x \mid x \leq H(s(e))\}$. Clearly, $|D| = H(s(e)) + 1 > H(s(e))$.

□

In a two move, two player game, think of (a program for) H as the move of player 1 and e as the move of player 2. Player 2's goal is to have the proposition be true; 1's is the opposite. Player 2's strategy involves e's using self-knowledge (and knowledge of a program for H) to compute $H(s(e))$ and make sure it says `Yes` to a **finite** number of inputs which number is (one) more than $H(s(e))$.

The theorem on the next slide provides an improvement of the just previous result. It's proof is also game-theoretic. First:

Definition h is a **limiting-computable** function $\stackrel{\text{def}}{\Leftrightarrow}$ for some computable function g, for each x, the sequence $g(x,0), g(x,1), g(x,2), \ldots$ is, **past some point**, $h(x), h(x), h(x) \ldots$.

Proposition There is a (big) limiting computable function h such that, for each computable f, for all but finitely many x, $h(x) > f(x)$.

Proof.

For each x, let $h(x) = 1 + \max\{\varphi_p(x) \mid p \leq x \wedge \varphi_p(x) \text{ is defined}\}$.

For each x, t, let $g(x,t) = 1 + \max\{\varphi_p(x) \mid p \leq x \wedge \varphi_p(x) \text{ defined in } \leq t \text{ steps}\}$.

Clearly, this computable g witnesses that h is limiting-computable. □

Theorem Let H be any (possibly horrendous) **limiting-computable** function. Then

$$(\exists e)(\exists D, \text{ a } \textbf{finite} \text{ set } \mid \varphi_e = C_D)[|D| > H(s(e))].$$

Proof.

Let G be a computable function witnessing H is limiting computable. By Kleene's Theorem there is a self-referential program e such that

$$\varphi_e = C_{\{x\mid \text{ card}(\{w<x \mid \varphi_e(w)=1\}) \leq G(s(e),x)\}}.$$

$\approx$ $\square$

Levels of Self-Modeling?

The complete wiring diagram of a machine provides a **low-level** self-model.

Other, **higher-level** kinds of self-modeling are of interest, e.g., **general descriptions of behavioral propensities**.

A nice **in**human example (provided by a machine) is: **I compute a strictly increasing mathematical function**.

A **human** example is: **I'm grumpy, upon arising, 85% of the time**.

For machines, which we likely are [Jac90,Cas99*], such higher-level self-knowledge may be proved from some powerful, correct mathematical theory **provided the theory has access to the complete low-level self-model**. Hence, the complete, low-level self-model is more basic.

*The expected behaviors in a discrete, quantum mechanical world with computable probability distributions are computable!

Human Thoughts and Feelings

We take the point of view that conscious human thought and feeling **inherently** involve (attenuated) **sensing** in any one of the sensory modalities. E.g.,

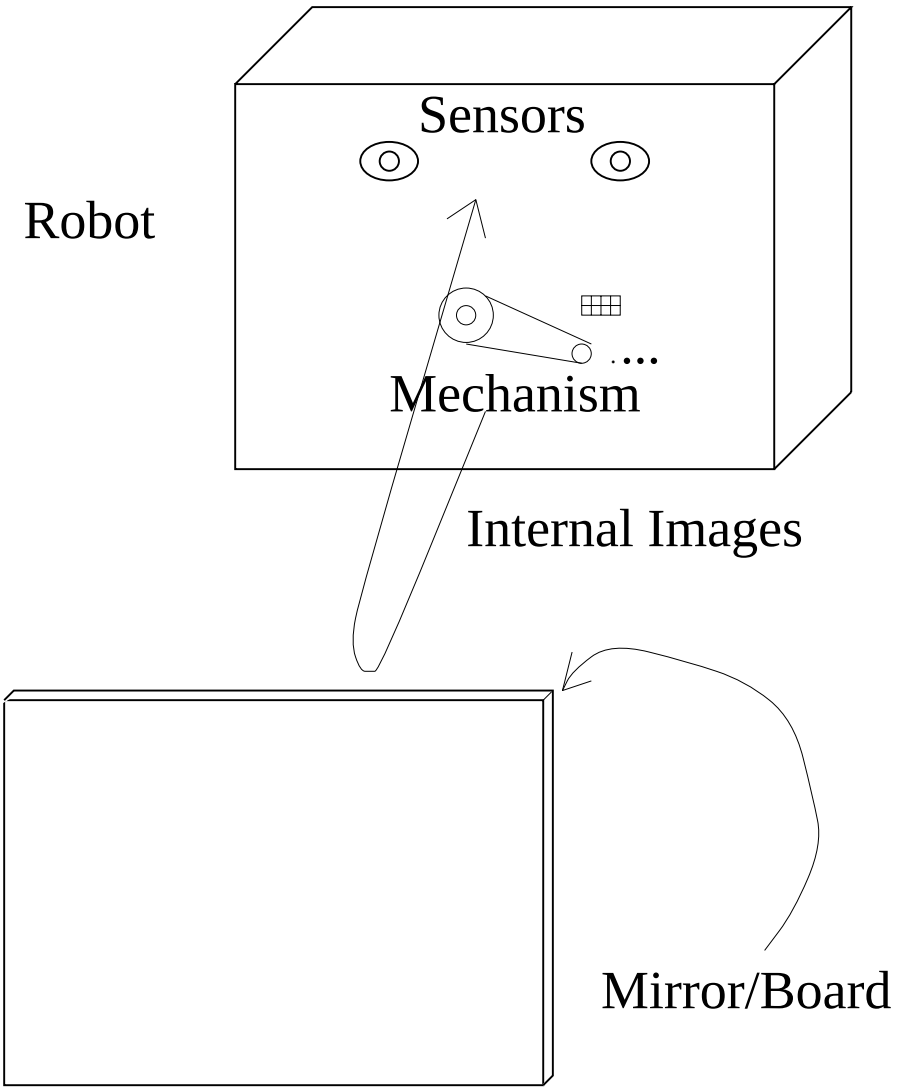
a. Vocal tract "kinesthetic" [Wat70] and/or auditory sensing for inner speech.

b. There is important sharing of brain machinery between **vision** and production and manipulation of **mental images**. Many ingenious experiments show that the same unusual perceptual effects occur with both real images **and imagined ones** [Jam90,FS77,Fin80,She78,Kos83,KPF99].

In the following we will exploit for exposition the **visual** modality since it admits of pictorially, metaphorically representing the other modalities: inner speech, feelings, ...

Generally the only aspects of our inner cognitive mechanism and structure we humans can know by consciousness are by such means as: detecting our own inner speech, our own somatic and visceral concomitants of emotions, our own mental images, ...

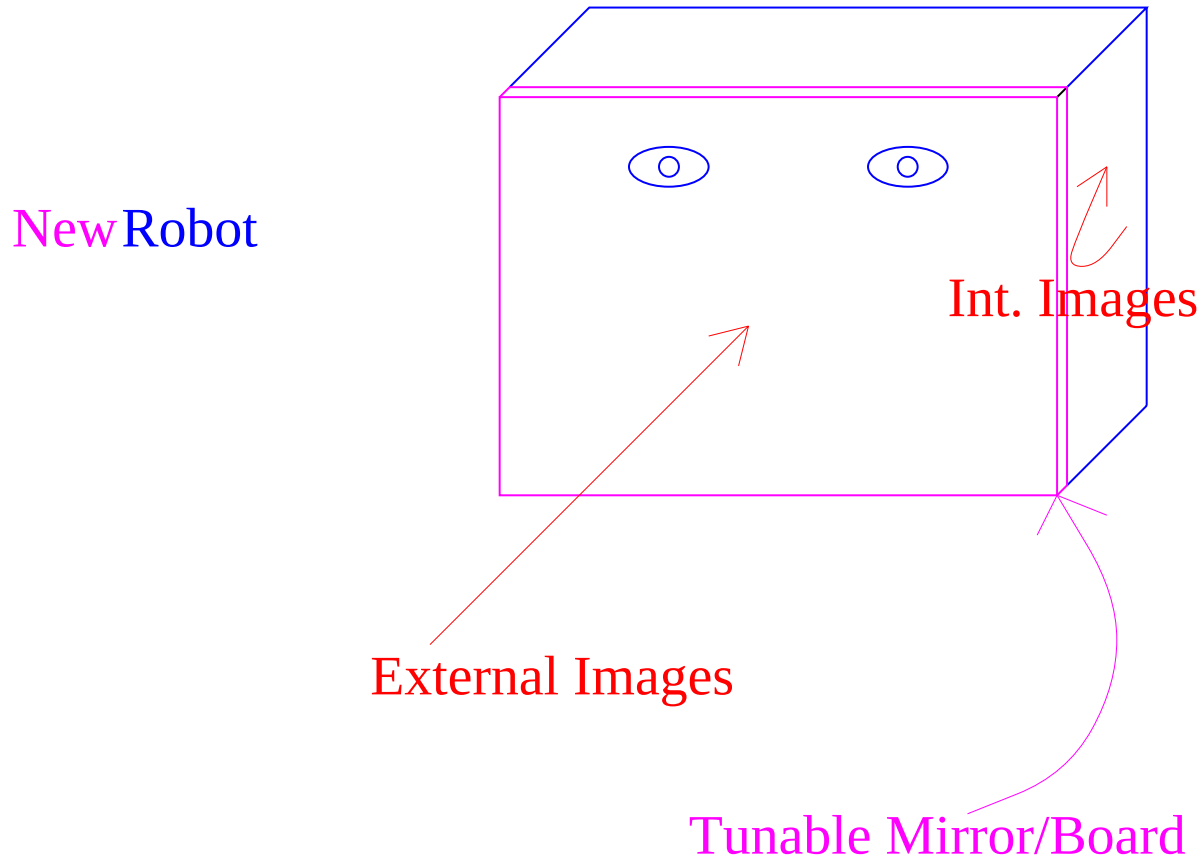
The Robot Revisited



Now, make the mirror/board **tunable**, e.g., as to its degree of "silvering," the degree to which it lets light through vs. reflects it.

The Robot Modified

Attach, then, the tunable mirror/board to the transparent **and sensory** front of the robot to obtain the new robot:



The new robot controls how much it looks at externally generated data and how much it looks at internally generated data, e.g, images of its own mechanism.*

The **attached**, tunable mirror/board is now part of the new robot.

*For humans 'external' means roughly 'external to the brain', e.g., for affect, the concomitant felt somatic and visceral sensations are from the body.

More About The Human Case

The robot's tunable mirror/board is analogous to the human sensory "surface." The latter is also tunable as to how much it attends to internal "images" and how much it attends to external (external to brain, not body).

However, we humans can only "see" the part of our internal cognitive structure originally built from sense data and sent back to our sensory surface to be re-experienced as modified and, typically, attenuated, further sense data. We don't see our own neural net, synaptic chemistry, etc. This is not surprising since we likely evolved from sensing-only organisms.

I recommend that brain scientists locate in the human brain a functional decomposition corresponding to the elements of our modified robot with tunable mirror/sensory surface! A lot is already known, e.g., regarding where in the visual cortex **both** real and imagined pictures are processed [KPF99]!

Lessons Of Machine Case?

From Kleene's Recursion Theorem (eventually) came our modified robot with attached, tunable mirror/board.

In applications of Kleene's Recursion Theorem [Cas94,RC94] (within Computability Theory) we see that, while is it **not** needed to compute all that is computable,

a. It provides very **succinct** proofs **and** program constructs [RC94]: Our example proofs are succinct & tight.

b. As we saw, from a **game-theoretic** viewpoint, in some cases, a (machine) player's **self-knowledge** is an important component of its winning strategy [Cas94].

Quite possibly, then, our own, less complete, human version of self-reflection evolved thanks to a premium on compact (i.e., succinct) brains and the need to win survival games. Emotions and reflection on them useful to survival too.* Of course, self-simulations and simulations of variants of self can be useful.

*Wonder if right-brain whole picture [Kin82] reflection on negative affect and possible left-brain detailed non-whole picture reflection on positive affect evolved also for survival.

Summary

Kleene's Strong Recursion Theorem provides for non-paradoxical self-referential **machines/programs**.

In effect, such a machine/program **externally projects** onto a **mirror** a complete, low level model of itself (i.e., wiring diagram, flowchart, program text, . . .).

We modified this machine self-reference to produce an idealization of the self-modeling component of human consciousness by attaching the mirror to the "sensory surface."

The analog of the mirror above is the human sensory "surface," **tunable** as to its degree of "silvering!"

Brain scientists should further map a Functional Decomposition Corresponding to Our Model.

From applications of Kleene's Theorem in Computability Theory: complete machine self-modeling aids with machine/program **succinctness** and with winning **games**. Perhaps the **uses of human reflective thought** are similar: need to have a compact brain and to win survival games. Emotions and reflection on them useful to survival too. Simulations of self and variants is clearly useful.

Talk References

[BZL06] J. Bongard, V. Zykov, and H. Lipson. Resilient machines through continuous self-modeling. *Science*, 314:1118–1121, 2006.

[Cas94] J. Case. Infinitary self-reference in learning theory. *Journal of Experimental and Theoretical Artificial Intelligence*, 6:3–16, 1994.

[Cas99] J. Case. The power of vacillation in language learning. *SIAM Journal on Computing*, 28:1941–1969, 1999.

[Fin80] R. A. Finke. Levels of equivalence in imagery and perception. *Psychological Review*, 87:113–139, 1980.

[FS77] R. A. Finke and M. J. Schmidt. Orientation-specific color after-effects following imagination. *Journal of Experimental Psychology: Human Perception and Performance*, 3:599–606, 1977.

[Jac90] R. Jackendoff. *Consciousness and the Computational Mind*. Bradford Books, 1990.

[Jam90] W. James. *Principles of Psychology*, volume II. Henry Holt & Company, 1890. Reprinted, Dover, 1950.

[Kin82] M. Kinsbourne. Hemispheric specialization and the growth of human understanding. *American Psychologist*, 35:411–420, 1982.

[Kos83] S. Kosslyn. *Ghosts in the Mind's Machine: Creating and Using Images in the Brain*. Harvard Univ. Press, Cambridge, Massachusetts, 1983.

[KPF99] S. Kosslyn, A. Pascual-Leone, O. Felician, S. Camposano, J. Keenan, W. Thompson, G. Ganis, K. Kukel, and N. Alpert. The role of area 17 in visual imagery: Convergent evidence from PET and rTMS. *Science*, 284:167–170, 1999.

[RC94] J. Royer and J. Case. *Subrecursive Programming Systems: Complexity and Succinctness*. Research monograph in *Progress in Theoretical Computer Science*. Birkhäuser Boston, 1994.

[She78] R. N. Shepard. The mental image. *American Psychologist*, 33:123–137, 1978.

[Wat70] J. Watson. *Behaviorism*. W.W. Norton, 1970.