Problems of self-reference in self-modifying AGI

Benja Fallenstein and Nate Soares

Machine Intelligence Research Institute

August 3, 2014
Motivation

- Can we create a **self-modifying** AGI...
  - ...that goes through a **billion modifications**...
  - ...*without ever going wrong*?

- Need **extremely reliable** way for an AI to reason **about itself**.
  - Much more reliable than a human!

- Is **self-referential** reasoning problematic?
  - See Gödel, the halting problem, etc...
1. The “procrastination paradox”

2. A formal toy model

3. Partial solutions
The “procrastination paradox”

• AGI in a deterministic, known world; discrete timesteps.

• In each timestep, the AI chooses whether to press a button:
  • If pressed in 1st round: Utility = 1/2
  • If pressed in 2nd round (and not before): Utility = 2/3
  • If pressed in 3rd round (and not before): Utility = 3/4
  • ...
  • If never pressed: Utility = 0

• (No optimal strategy, but sure can beat 0!)

• The AGI is programmed to press the button immediately...
  • ...unless it finds a “good argument” that the button will get pressed later.
The AGI reasons:

- Suppose I don’t press the button now.
- Either I press the button in the next step, or I don’t.
  - If I \textit{do}, the button gets pressed, good.
  - If I \textit{don’t}, I must have found a good argument that the button gets pressed later. So the button gets pressed, good!
  - Either way, the button gets pressed.

So the AGI can always find a “good argument” that the button will get pressed later…

- …and therefore never presses the button!

\textit{If we want to have \textbf{reliable} self-referential reasoning, we must understand how to avoid this paradox (and others like it).}
So what went wrong? (And how do we fix it?)

- The paradox doesn’t go through with finite time horizons—
  - —or with temporal discounting:
  - Utility \( = \sum_{t=0}^{\infty} \gamma_t \cdot R_t \), where \( \sum_{t=0}^{\infty} \gamma_t < \infty \) and \( R_t \in [0, 1] \).

- Does using temporal discounting fix all such problems?

- In our toy model:
  - **No**, not by itself.
    - Still get (more technical) paradoxes of self-reference.
  - But: there are ways to fix these problems...
  - ... which work **if** we use finite horizons or discounting.
    - (Suggests this is key to avoiding the problem.)
1. The “procrastination paradox”

2. A formal toy model

3. Partial solutions
For our toy model, use formal logic.

But *not* because we think realistic AGIs work like this.
- The **problem** seems to be much more general.
- Any scheme for highly reliable self-referential reasoning will need to deal with it somehow.

Rather: because we can prove theorems about it—
- and then see what this tells us about the real problem.
Write $P(n)$ for “the button is pressed in the $n^{th}$ timestep”.

Define computable function $f(n)$:

- $f(n)$ searches for proofs
  - in Peano Arithmetic (PA)
  - of length $\leq 10^{100+n}$
  - of “$\exists k > n. P(k)$” — i.e., “button pressed later”.

- If proof found $\implies$ returns 0 ("don’t press button").
- Else $\implies$ returns 1 ("press button").

$\text{PA} \vdash P(n) \iff [f(n) = 1].$

(Self-referential definition by Kleene’s second recursion thm.)
• By looking at $f(n + 1)$, can prove (in $\ll 10^{100+n}$ symbols):
  • “Either the button will be pressed in the next timestep or not”:
    \[ \text{PA } \vdash P(n + 1) \lor \neg P(n + 1) \]
  • “If button not pressed in next step, must have found proof it will be pressed later”:\footnote{Notation: $\square_{\text{PA}} \vdash \varphi^\frown$ means “$\varphi$ is provable in PA”.}
    \[ \text{PA } \vdash \neg P(n + 1) \rightarrow \square_{\text{PA}} \exists k > n + 1. P(k)^\frown \]
  • (???) “If there’s a proof that the button will be pressed, then it will indeed be pressed.”
    \[ \text{PA } \vdash \square_{\text{PA}} \exists k > n + 1. P(k)^\frown \rightarrow \exists k > n + 1. P(k) \]
  • “Hence, either way, the button is pressed.”
    \[ \text{PA } \vdash P(n + 1) \lor \exists k > n + 1. P(k) \]
    \[ \text{PA } \vdash \exists k > n. P(k) \]

• Hence, $f(n) = 0$ (for all $n \in \mathbb{N}$)… button never pressed.
• $\implies$ So $\text{PA } \not\vdash \square_{\text{PA}} \vdash \varphi^\frown \rightarrow \varphi$. 

Benja Fallenstein and Nate Soares
Problems of self-reference in self-modifying AGI
PA avoids the paradox since $PA \not \vdash \Box_{PA} \neg \varphi \rightarrow \varphi$.

Generalize this beyond our logic-based toy example?

Why do we think our AGI will work correctly?

We reason: “It will take only actions if it has very good reason to believe these actions will be safe — therefore, any actions it will take will be almost certainly safe.”

An AGI should be able to use the same argument when reasoning about rewriting itself!

Need *something* like $T \vdash \Box_{T} \neg \varphi \rightarrow \varphi$...

Gödel/Löb: But that’s inconsistent, finite time horizons or not!
1. The “procrastination paradox”

2. A formal toy model

3. Partial solutions
Partial solutions

1. Can have theories $T_0, T_1, T_2, \ldots$ s.t. $T_{n+1} \vdash \Box_{T_n} \neg \varphi \rightarrow \varphi$.
   - AGI using $T_{n+1}$ can rewrite into AGI using $T_n$.
   - Stops working when we reach $T_0$.
   - Works for finite time horizons.

2. Can have theories s.t. $T_n \vdash \Box_{T_{n+1}} \neg \varphi \rightarrow \varphi$ for all $\varphi \in \Pi_1$.
   - AGI using $T_n$ can rewrite into AGI using $T_{n+1}$.
   - Can rewrite forever!
     - (But: AI doesn’t know this! :-()}
   - Works with temporal discounting (see paper).

Do these approaches generalize beyond formal logic?
Conclusions

- Gave example of self-referential reasoning gone wrong.
  - Any **reliable** system for self-referential reasoning will need to deal with this somehow.

- **Analyzed** the problem using a **toy model**, and looked for solutions that generalize.

- In the paper:
  - Detailed proofs.
  - Extension to space-time embedded agents:
    - actions, observations, probabilities, utilities.

- Extremely reliable self-referential reasoning isn’t trivial...
  - but we can make progress towards it!  **Thanks for listening!**