ELECTRONIC FRONTIER FOUNDATION

Cambridge, March 25, 2018

Incompossibilities:

Ubiquitous Engineering Tradeoffs

Seth Schoen
Senior Staff Technologist, EFF

Income Possibilities!

MAKE MONEY FAST

THROUGH

CRYPTOCURRENCY ARBITRAGE!

DON’T MISS THESE

INCREDIBLE INVESTMENT OPPORTUNITIES

Income Possibilities!

~~MAKE MONEY FAST~~

~~THROUGH~~

~~CRYPTOCURRENCY ARBITRAGE!~~

~~DON'T MISS THESE~~

~~INCREDIBLE INVESTMENT OPPORTUNITIES~~

Incompossibilities

- Things that can’t exist at the same time
- Raymond Smullyan attributes the term to Ambrose Bierce (*The Devil’s Dictionary*); it seems to have been introduced earlier by Leibniz in discussions of the concept of “possible worlds”
- Bierce gives it as a super-classy way of threatening someone:

“Sir, we are *incompossible*.”

Incompossibilities

- Familiarly, unfortunate tradeoffs when “you can’t always get what you want”
- MIT joke: “Work, friends, sleep—pick two!”
- Another engineering joke: “Good, fast, cheap—pick two!” *[Yielding $_3C_2$=3 total options.]*
- Hence, situations when we have to sacrifice *something* that we want or value

In software, too?

- **We might like to think that software is perfectible in a much stronger sense than physical objects, because it doesn't suffer from physical limitations**
  - And it's often designed "from scratch"
- **But researchers keep discovering limitative theorems in many disciplines and fields that prove various properties are incompossible**

In software, too?

- Limitative results may show that no mathematical object with a certain combination of properties exist
- This object could be an algorithm, process, or software system!
- In other cases we have strong reason to believe in tradeoffs, even without a theorem

A famous computer science example

- The CAP Theorem for distributed databases: a distributed database system cannot provide
- Consistency,
- Availability, and
- Partition-tolerance
- Eric Brewer (1999, 2000); Seth Gilbert and Nancy Lynch (2002)

A voting/social choice example

- Kenneth Arrow showed in 1951 that there’s no way of aggregating preferences that always ensures several kinds of fairness:
- Deterministic based on preferences, all options achievable
- No single “dictator” making the overall decision
- Independence of irrelevant alternatives (adding a less-preferred option shouldn’t change the outcome)
- If everyone likes A better than B, A should be chosen over B
- Incentive to vote honestly according to one’s preferences

National Resident Matching Program

- A large-scale algorithmic preference aggregation: matches medical students to residencies considering students’ and hospitals’ preferences
- “Stability” criterion (nobody has incentive to make a deal outside the program), based on Gale and Shipley (1962)
- Process to redesign algorithm (effective 1998), considering things like couples who want to live together
- Used to give higher priority to hospitals’ preferences, now gives higher priority to students’ preferences!

National Resident Matching Program

- **Some Arrow-like criteria (e.g. strategy-proof—nobody should have an incentive to lie!)**
- **Some desirable criteria are incompossible :-(**
- **See Roth and Peranson (1999)**
  - Roth won the Nobel Prize for this and related work
- **They say they chose details based on empirical simulations and their judgments about tradeoffs**

Ethical theories

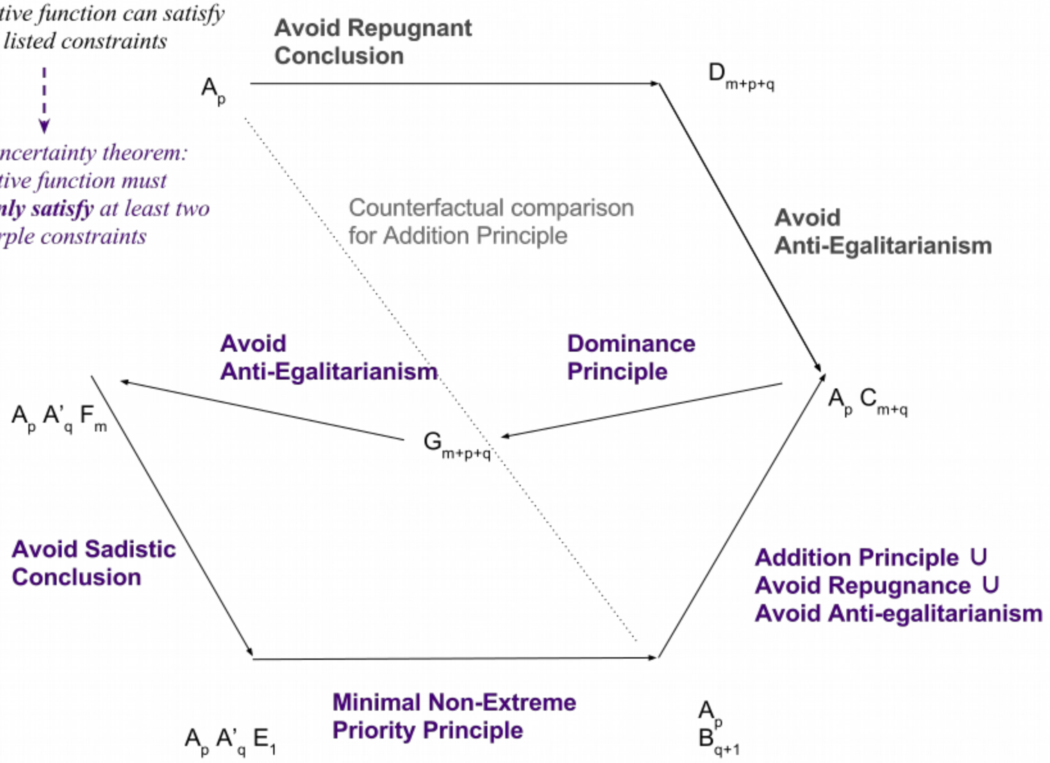
- Gustaf Arrhenius has seven theorems on how strong moral intuitions can sometimes conflict
- Paradoxes in axiology (attempts at saying what makes the world better or worse overall), inspired by Derek Parfit
- Finding *cycles* where different principles imply A is better than B, B is better than C, yet C is better than A!

Ethical uncertainty for AI

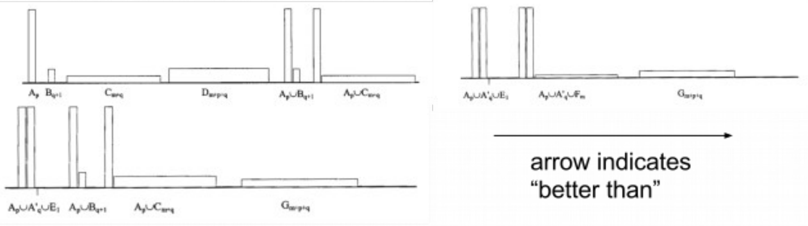
- Increasingly, machines may have to implement ethical rules when making practical decisions in the world
- In a forthcoming paper, Peter Eckersley shows that paradoxes like Arrhenius's imply *ethical uncertainty* in formalizations of ethics in AI objective functions
- At least 2 principles in a cycle must allow "I'm torn" rather than "A > B" or "B > A"

Ethical impossibility theorem:
No objective function can satisfy
all of the listed constraints

Ethical uncertainty theorem:
An objective function must
***uncertainly satisfy** at least two*
of the purple constraints


Avoid Repugnant Conclusion
A_p
D_{m+p+q}
Counterfactual comparison for Addition Principle
Avoid Anti-Egalitarianism
Avoid Anti-Egalitarianism
Dominance Principle
$A_p\ A'_q\ F_m$
G_{m+p+q}
$A_p\ C_{m+q}$
Avoid Sadistic Conclusion
Addition Principle ∪
Avoid Repugnance ∪
Avoid Anti-egalitarianism
Minimal Non-Extreme Priority Principle
$A_p\ A'_q\ E_1$
A_p
B_{q+1}

Key to compared world states



arrow indicates
“better than”

Fairness for AI

- Whether AI decisions are “fair” has been a hot topic
- Researchers have formalized several different intuitions about what this could mean
- A recent theorem: Some of these notions of fairness are incompossible; no AI system is “fair” in all senses
- See Kleinberg, Mullainathan, and Raghavan, “Inherent Trade-Offs in the Determination of Risk Scores” (2017); Google also made an interesting visualization

https://research.google.com/bigpicture/attacking-discrimination-in-ml/

Zooko's Triangle

- Zooko says (a conjecture, not a theorem) that no naming system can be
- Decentralized,
- Human-memorable, and
- Secure (unambiguous)
- We have several examples of naming systems that violate each individual property

Padding for traffic-analysis resistance

```
$ for url in                                                        \
https://www.webmd.com/skin-problems-and-treatments/acne/default.htm \
https://www.webmd.com/mental-health/addiction/default.htm           \
https://www.webmd.com/cancer/default.htm                            \
https://en.wikipedia.org/wiki/Abortion-rights_movements             \
https://en.wikipedia.org/wiki/Anti-abortion_movements;             do
wget -O- "$url" | wc -c; done

111151
110738
109543
128575
99585
```

A harsh tradeoff

- Add padding data to disguise which article someone is viewing

  → The service will consume extra data

  → Users who pay per byte may be upset and/or reduce use of the service

- Don’t add extra padding data

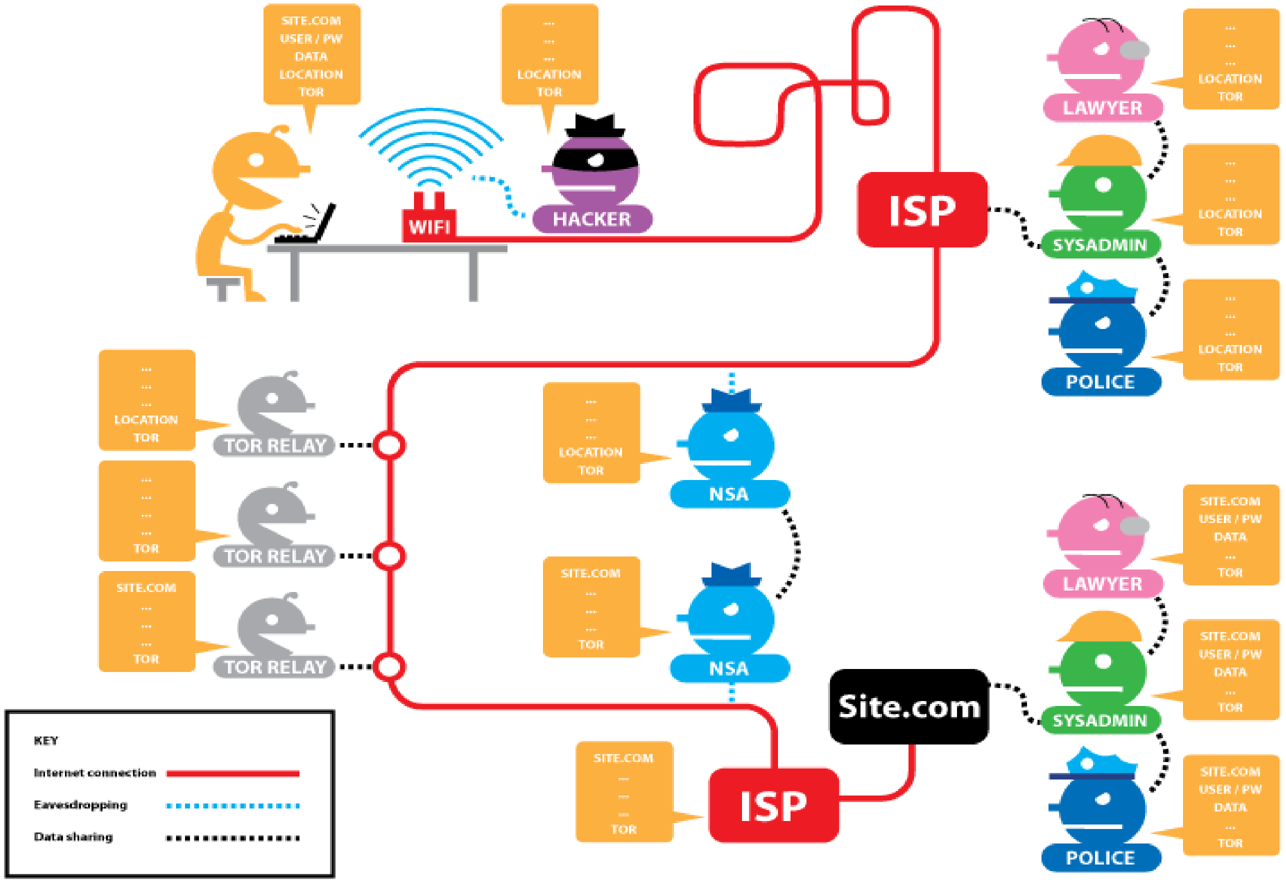
  → It will be pretty clear who’s reading what

Anonymity vs. latency

- **Some old anonymity systems deliberately added delay to communications to create ambiguity about who was responsible for messages**
  - Other related options: padding, synchrony
- **Low-latency systems like Tor don’t add these delays**

  → Someone watching both ends of a communication can infer their connection

Anonymity vs. latency



Pond

- **A “non-instant messaging system” by Adam Langley**
  - No longer maintained, but shows what a modern design for high-latency messaging might look like
- **(Deliberately) slow**
- **(Deliberately) low message size limits and high overhead**
- **Not very partition-tolerant**
- **Probably needs lots of people to use it consistently in order to get useful anonymity**

Web user tracking

- **As you expose more of the web platform to mobile code, you have more individuation that leads to persistent identifiers**
  - See EFF's Panopticlick tool
- **Web developers (and users) resist disabling features because of reduced functionality**

Conjectures on social media tradeoffs

- Social media has been strongly criticized recently, and there are many things people demand from these systems
- A colleague at a social media company has conjectured that not all are compossible
- Even if we all used Mastodon :-)

(in other words, even with decentralization)

Do these results really matter?

- We might hope that limitative theorems are the exception rather than the rule
- Yet they seem to arise over and over in many contexts and sometimes affect very practical engineering decisions
- Problem spaces and values are complex!

Why think about these limitations? (1)

- Clarifying goals and possibilities
- Distributed and federated systems, for example, offer choices about whose responsibility each function is
- Each choice has some adverse consequences for some scenario (including UX, in terms of users' heightened responsibilities in exchange for heightened autonomy)

Why think about these limitations? (2)

- **Thinking and deliberating explicitly rather than choosing by default**
- **E.g. Debian Project deliberated explicitly about unavoidable tradeoffs of electoral methods in designing its own internal system**
  - See Debian Constitution §A.6

Why think about these limitations? (3)

- Not running in circles trying to solve inherently unsolvable problems
- But understanding whether formal impossibility results really apply to the things we care about in practice
- Maybe a theorem’s definition of “security” or “fairness” or “infeasibility” or “always” doesn’t match yours

Why think about these limitations? (4)

- Not assuming that we can get to perfect software, or that software can necessarily be made to solve every problem
- Not blaming software developers and communities for not doing the impossible

Thanks!

Have a great LibrePlanet and, for those from out of town, have a great time in Boston!

(You might want to try the hot chocolate at Burdick's in Harvard Square—just a personal opinion!)