Review

- WHILE program
- UTM theorem, Normal form theorem, s_{mn} theorem
- Fixed point theorem, quines

Oracle WHILE program

- $x_i := \varphi(x_j)$ where $\varphi: \mathbb{N} \to \mathbb{N}$ is fixed
- φ doesn't have to be computable
- Example: $\varphi = \mathbf{1}_H$ "decides" halting problem and complement
- φ computable $\Rightarrow$ no need for oracle

Turing reduction

- $A \preccurlyeq_T B$: an oracle program with oracle $\mathbf{1}_B$ can decide A
- $A \preccurlyeq B \Rightarrow A \preccurlyeq_T B$ but not vice versa ($H \preccurlyeq_T \overline{H}$)
- Preorder (reflexive and transitive)
- $A \equiv_T B$ (Turing equivalent): $A \preccurlyeq_T B$ and $B \preccurlyeq_T A$

Turing reduction – basic results

- A is decidable $\Rightarrow A \preccurlyeq_T B$
- $A \preccurlyeq_T B$ and B is decidable $\Rightarrow A$ is decidable
- A is semidecidable $\Rightarrow A \preccurlyeq_T H$

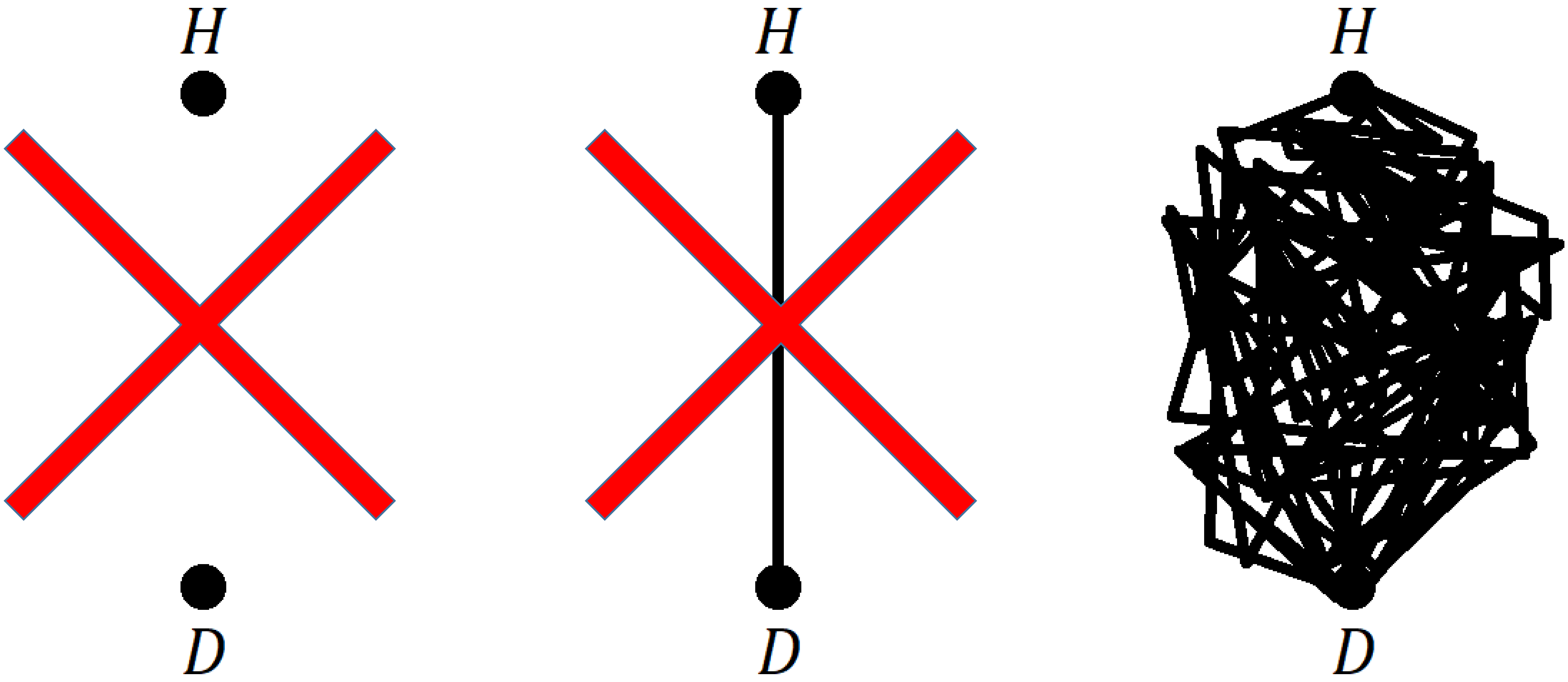
$$D \preccurlyeq_T S \preccurlyeq_T H$$

for any D decidable, S semidecidable

What happens in between?

Emil Post: "Is there $S \not\equiv_T D, H$?"

Between D and H



Friedberg–Muchnik Theorem

There exist two semidecidable sets A, B that are not Turing-reducible to each other

Friedberg–Muchnik Theorem

There exist two semidecidable sets A, B that are not Turing-reducible to each other

Friedberg–Muchnik Theorem

There exist semidecidable sets A, B where:

- For all program TB(i) with oracle $\mathbf{1}_B$, it doesn't decide A
- ...and vice versa

Friedberg–Muchnik Theorem

There exist semidecidable sets A, B where:

- For all program TB(i) with oracle $\mathbf{1}_B$, there exists x where
    - either $x \in A$, but TB(i) doesn't halt or it outputs not 1,
    - or $x \notin A$, but TB(i) doesn't halt or it outputs not 0
- ...and vice versa

Friedberg–Muchnik theorem
Proof idea

- For each TB(i) find an input ("witness") that breaks TB(i)
- Construct A, B iteratively, only add elements
- $A_0 \subseteq A_1 \subseteq A_2 \subseteq \cdots$ and $A = \bigcup_{r=0}^{\infty} A_r$
- TB(i) queries B_r instead

Friedberg–Muchnik theorem
Breaking all programs

- G_i = list of possible witnesses to break TB(i)
- G_i, H_i infinite, increasing, partition $\mathbb{N}$
- x where TB(i) outputs not 1 $\Rightarrow$ put $x \in A$
- Do in rounds so all programs run

Friedberg–Muchnik theorem
Construct in rounds

- $A_0 = B_0 = \emptyset$
- Round r: run TB(i) on $x_i = \min G_i$ with oracle B_{r-1} for r steps, for $i = 1, 2, \ldots, r$
- Case TB(i) outputs 1: do nothing
- Case TB(i) doesn't halt: do nothing
- Case TB(i) outputs not 1: put x_i in A_r
- Then do for TA(i) with oracle A_r

Friedberg–Muchnik theorem
A lot of problems

- $B_{r-1} \neq B$: a program with oracle B_{r-1} is different from oracle B
- Choosing some witness from G_i might invalidate other witnesses
- Put **priority order** on G_i, H_i:
$$G_1 > H_1 > G_2 > H_2 > G_3 > H_3 > \cdots$$
- If two choices conflict, lower-priority must fix it

Friedberg–Muchnik theorem
Lower-priority invalidates higher-priority

- $y \in H_i$ breaks $\text{TA}(i)$, but only because $x \notin A_r$
- $x \in G_j$ breaking $\text{TB}(j)$ is put into A in a later round
- $\text{TA}(i)$ queries maximum m
- Fix: delete $\leq m$ from lower-priority sets

Friedberg–Muchnik theorem
Higher-priority invalidates lower-priority

- $x \in G_j$ breaks TB(j), but only because $y \notin B_r$
- $y \in H_i$ breaking TA(i) is put into B in a later round
- Cannot do same fix (lower-priority must give in)
- Fix: G_j chooses another x

Friedberg–Muchnik theorem
Finite injury argument

- G_j might need to choose another witness because H_i: "injured"
- Induct: G_i, H_i injured at most $4^{i-1} - 1,\ 2 \times 4^{i-1} - 1$ times
- Eventually chooses a witness that is not invalidated

Friedberg–Muchnik theorem
Never got a witness

- $x = \min G_i$ satisfy either $\text{TB}(i)$ never halts, or $\text{TB}(i)$ outputs 1
- Leave x out of A: x is a witness

Friedberg–Muchnik theorem Conclusion

- For each TB(i) we found a witness x where...
  - TB(i) on x halts with output not 1: we've put $x \in A$
  - TB(i) on x doesn't halt, or halts with output 1 (i.e. not 0): we keep $x \notin A$
- Made sure the final witnesses don't break each other
- Each round is finite time, no element comes out
- $\Rightarrow A, B$ semidecidable but neither is Turing-reducible to the other