Finite Injury Priority Method

To each prog. $P$ exists $x[P]$ s.t.: $x \in A \iff P^B$ accepts $x$

To each prog. $Q$ exists $y[Q]$ s.t.: $y \in B \iff Q^A$ accepts $y$

Maintain lists $(P,x)$ and $(Q,y)$ with ‘candidate’ witnesses $(P,x)$ active if simulation $P^B$ on $x$ still running; else inactive

E.g. $L_A = (P_1,x_1), (P_2,x_2), (P_3,x_3)$; $L_B = (Q_1,y_1), (Q_2,y_2)$.

- For each $n:=0,1,...$
  - Add entry $(n,x)$ to list. For active $(P,a)$ increasing in $P$
  - If $P^B$ accepts $a$ within $\leq n$ steps, set $A := A \cup \{a\}$
    and $y := 1 + \max\{y$, largest oracle query by $P^B$ on $a\}$
    and make $(P,a)$ inactive. For all $(Q,b)$ with $Q > P$ do
    - replace $(Q,b)$ with $(Q,y++)$ made active.
  - Add entry $(n,y)$ to list. For all active $(Q,b)$ in list:
Finite Injury Priority Method

Candidates for “\( y \in B \iff Q^A \) accepts \( y \)” change („injury“) but only a finite number of times:

- namely when some \( P < Q \) terminates („priority“) and, once settled, does satisfy the witness condition!

Both \( A, B \) are enumerated, hence semi-decidable.

- For each \( n := 0, 1, \ldots \)
  - Add entry \((n,x)\) to list. For active \((P,a)\) increasing in \( P \)
    - If \( P^B \) accepts \( a \) within \( \leq n \) steps, set \( A := A \cup \{a\} \)
      and \( y := 1 + \max\{ y, \text{largest oracle query by } P^B \text{ on } a \} \)
      and make \((P,a)\) inactive. For all \((Q,b)\) with \( Q > P \) do
        - replace \((Q,b)\) with \((Q,y++)\) made active.
  - Add entry \((n,y)\) to list. For all active \((Q,b)\) in list:
Priority Diagonalization: Trading with the Devil

- You have countably many coins
  - Devil takes one of them
  - and gives you two new ones,
  - Then repeat.

- How many coins do you ultimately own?

NONE!

Courtesy of Joel D. Hamkins
Partially Ordered Sets

Sacks (1964), ..., Soare (1980): detailed study of which posets arise as Turing degrees
Summary and Perspective

No prerequisites: just clear thinking! (and BF)

- Introduction to Diagonalization: Cantor, Barber
- Model of Computation, Computability
- Undecidability, Halting Problem, Rice’s Theorem
- Oracle Computation, Degrees of Uncomputability
- Time Hierarchy Theorem
- Relativization of the "P versus NP" question
- Post’s Question, Solution by Friedberg and Muchnik: Finite Injury Priority Method

Richard E. Ladner (1975) , Uwe Schöning

If $P \neq NP$, there exists problem
- not in $P$  
- in $NP$  
- but not $NP$-complete