Finite Injury Priority Method

HEINZ NIXDORF INSTITUTE
University of Paderborn
Algorithms and Complexity

To each prog. P exists $\underline{x}[P]$ s.t.: $\underline{x} \in A \Leftrightarrow P^B$ accepts $\underline{x}$
To each prog. Q exists $\underline{y}[Q]$ s.t.: $\underline{y} \in B \Leftrightarrow Q^A$ accepts $\underline{y}$

Maintain lists $(P,\underline{x})$ and $(Q,\underline{y})$ with 'candidate' witnesses
$(P,\underline{x})$ **active** if simulation P^B on $\underline{x}$ still running; else *inactive*
E.g. $L_A = (P_1,\underline{x}_1)$, $(P_2,\underline{x}_2)$, $(P_3,\underline{x}_3)$; $L_B = (Q_1,\underline{y}_1)$, $(Q_2,\underline{y}_2)$.

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

Finite Injury Priority Method

HEINZ NIXDORF INSTITUTE
University of Paderborn
Algorithms and Complexity

Candidates for "$\underline{y} \in B \Leftrightarrow Q^A$ accepts $\underline{y}$" change („injury") but only a finite number of times:

- namely when some $P<Q$ terminates („priority")

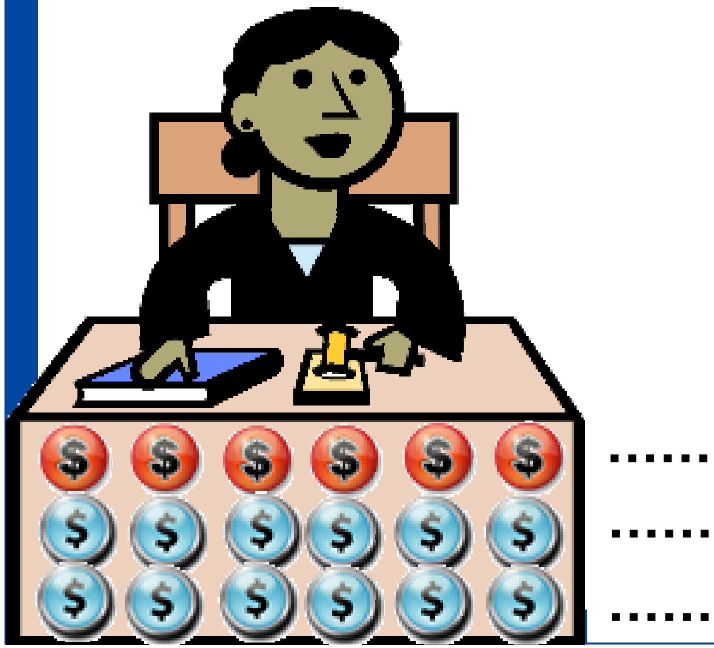
and, once settled, does satisfy the witness condition!

Both A,B are enumerated, hence semi-decidable.

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

Priority Diagonalization: Trading with the Devil

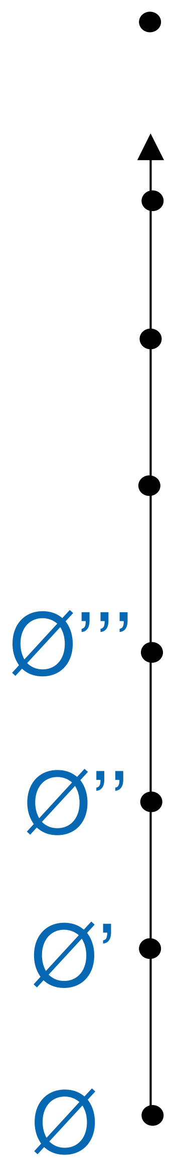
- You have countably many coins
  - Devil takes one of them
  - and gives you two new ones,
  - Then repeat.
- How many coins do you ultimately own ?



NONE!

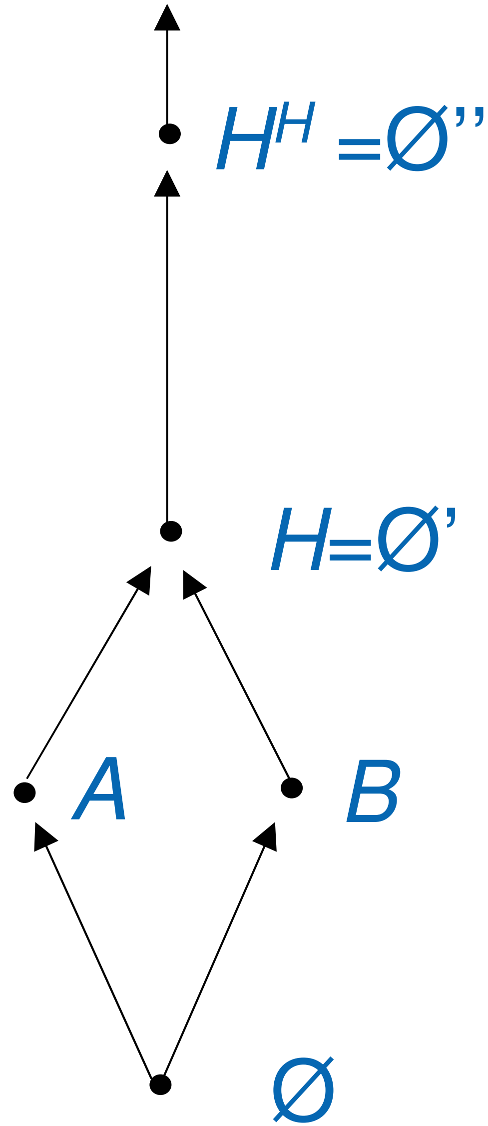
Courtesy of Joel D. Hamkins

Partially Ordered Sets



Sacks (1964), …,
Soare (1980):

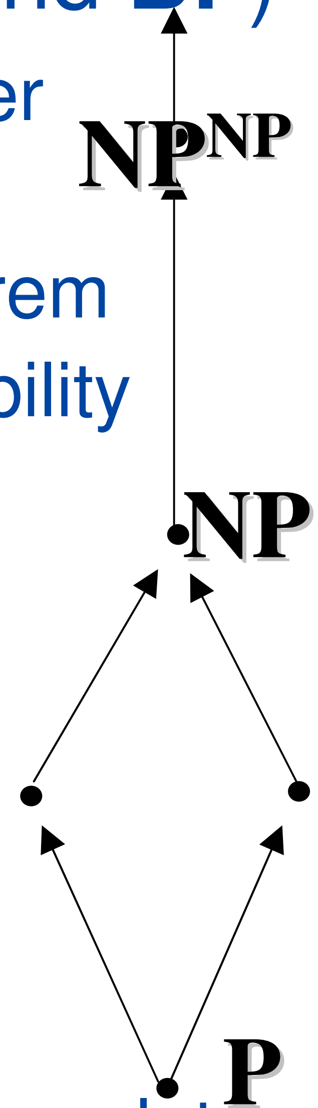
detailed study of
which posets arise
as Turing degrees



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**≠**NP**, there exists problem

- not in **P**
- in **NP**
- but not **NP**-complete