Probabilistic Reduction

Kaspars Balodis

Reduction

- Transform an instance of one problem to an instance of another problem

- What is a problem?
- What is an instance of a problem?

Decision problem

- Arbitrary yes-or-no question on an infinite set of inputs
  - yes-or-no
  - infinite set of inputs

- Is *x* a prime?

Decision problem

- P = "Given two positive integers x and y, does x evenly divides y?"

	1	2	3	4	5	6	7	8	9	10
10	Y	Y	N	N	Y	N	N	N	N	Y
9	Y	N	Y	N	N	N	N	N	Y	N
8	Y	Y	N	Y	N	N	N	Y	N	N
7	Y	N	N	N	N	N	Y	N	N	N
6	Y	Y	Y	N	N	Y	N	N	N	N
5	Y	N	N	N	Y	N	N	N	N	N
4	Y	Y	N	Y	N	N	N	N	N	N
3	Y	N	Y	N	N	N	N	N	N	N
2	Y	Y	N	N	N	N	N	N	N	N
1	Y	N	N	N	N	N	N	N	N	N

Encoding the inputs

- P = {0, 1, 3, 4, 6, 10, 11, 12, 15, 21, 22, ...}

10	45	56	68	81	95	110	126	143	161
9	36	46	57	69	82	96	111	127	144
8	28	37	47	58	70	83	97	112	128
7	21	29	38	48	59	71	84	98	113
6	15	22	30	39	49	60	72	85	99
5	10	16	23	31	40	50	61	73	86
4	6	11	17	24	32	41	51	62	74
3	3	7	12	18	25	33	42	52	63
2	1	4	8	13	19	26	34	43	53
1	0	2	5	9	14	20	27	35	44
	1	2	3	4	5	6	7	8	9

Decision problems

- Sets of nonnegative integers
- $x \in A$?
- Characteristic function

$$c_A(x) = \begin{cases} 1, \ if \ \ x \in A \\ 0, \ if \ \ x \notin A \end{cases} \qquad \chi_A(x) = \begin{cases} 1, \ if \ \ x \in A \\ not \ \ defined, \ if \ \ x \notin A \end{cases}$$

- If $c_A(x)$ is computable then *A* is a recursive set
- If $\chi_A(x)$ is computable then A is a recursively enumerable set

Reductions

- A, B – not computable
- But if we had a machine that computes B, could we compute A?
- Reduction
- $A \leq B$

Many-one reduction

- A is many-one reducible to B if there is an algorithm that given x will produce y such that

  $x \in A \Leftrightarrow y \in B$

$$A \leq_m B$$

Truth-table reduction

- Truth-table condition $\langle\langle x_1, x_2, \ldots, x_k\rangle, \alpha\rangle$
- α – k-ary Boolean function
- Truth-table condition is satisfied by A if

$$\alpha(c_A(x_1), c_A(x_2), \ldots, c_A(x_k)) = 1$$

- *A* is truth-table reducible to *B* if there is an algorithm that given *x* will produce a truth-table condition that is satisfied by *B* iff $x \in A$

$$A \leq_{tt} B$$

Turing reduction

- A is Turing reducible to B if there is an algorithm which has an access to oracle on B that when given x will output “yes” if $x \in A$ and “no” if $x \notin A$

$$A \leq_T B$$

Probabilistic reductions

- A is probabilistically many-one reducible to B with probability p if there is a probabilistic algorithm that given x will with probability p produce y such that $x \in A \Leftrightarrow y \in B$

$$A \leq_m^{\Pr \geq p} B$$

Probabilistic reductions

- A is probabilistically truth-table reducible to B with probability p if there is a probabilistic algorithm that given x will with probability p produce a truth-table condition that is satisfied by B iff $x \in A$

$$A \leq_{tt}^{\Pr \geq p} B$$

Probabilistic reductions

- A is probabilistically Turing reducible to B with probability p if there is an probabilistic algorithm which has an access to oracle on B that when given x with probability p will output "yes" if $x \in A$ and "no" if $x \notin A$

$$A \leq_T^{\Pr \geq p} B$$

Probabilistic reductions

- Note that in all reductions with probability 1-p the algorithm is allowed to produce an incorrect result or even not halt at all

Probabilistic reductions

- Are probabilistic reductions more powerful than deterministic reductions?

Results

- If $A \leq_T^{\Pr > 1/2} B$, then $A \leq_T B$

- For every k, l and p such that $\frac{k}{l} \leq 2 - \frac{1}{p}$ there exists A, B such that

  $A \leq_m^{\Pr \geq k/l} B$ $A \not\leq_m^{\Pr > p} B$

- If $A \leq_m^{\Pr > 1/2} B$ and B is recursively enumerable, then A is recursively enumerable

Results

- There exist sets A and B such that $A \not\leq_{tt} B$

  but $A \leq_{tt}^{\Pr \geq 2/3} B$

- For every A and B if $A \leq_{tt}^{\Pr > 2/3} B$ then $A \leq_{tt} B$

Proof

$$K = \{ x \mid \varphi_x(x) \text{ is defined} \}$$

$$\tilde{K} = \{ x \mid (\exists y)[\varphi_x(x) = y \;\&\; \text{truth–table condition } y \text{ is satisfied by } K] \}$$

$$\tilde{K} \not\leq_{tt} K$$

$$\tilde{K} \leq_{tt}^{\Pr \geq 2/3} K$$

Proof

$$\tilde{K} \not\leq_{tt} K$$

$$K = \{x \mid \varphi_x(x) \text{ is defined}\}$$
$$\tilde{K} = \{x \mid (\exists y)[\varphi_x(x) = y \,\&\, \text{truth–table condition } y \text{ is satisfied by } K]\}$$

- Let's assume the contrary that $\tilde{K} \leq_{tt} K$
- Then also $\overline{\tilde{K}} \leq_{tt} K$
- Assume $\overline{\tilde{K}} \leq_{tt} K$ via Turing machine x_0
- $x_0 \in \overline{\tilde{K}}$?
- $x_0 \in \overline{\tilde{K}} \Leftrightarrow \left(\varphi_{x_0}(x_0) \text{ is satisfied by } K\right)$
- $\left(\varphi_{x_0}(x_0) \text{ is satisfied by } K\right) \Leftrightarrow x_0 \in \tilde{K}$
- Contradiction

Proof

- $\tilde{K} \leq_{tt}^{\Pr \geq 2/3} K$
- Algorithm:
  - With probability ⅓ answer “no”
  - With probability ⅓ answer that $x \in \tilde{K}$ *iff* $x \in K$
  - With probability ⅓ simulate the Turing machine x on input x. When it stops and gives y, answer that $x \in \tilde{K}$ iff the truth-table condition y is satisfied by K

Proof

- $\tilde{K} \leq_{tt}^{\Pr\geq 2/3} K$
- Algorithm:
  - With probability ⅓ answer "no"
  - With probability ⅓ answer that $x \in \tilde{K}$ *iff* $x \in K$
  - With probability ⅓ simulate the Turing machine x on input x. When it stops and gives y, answer that $x \in \tilde{K}$ iff the truth-table condition y is satisfied by K

Turing machine x never stops $x \notin \tilde{K}$

Proof

- $\tilde{K} \leq_{tt}^{\Pr\geq 2/3} K$
- Algorithm:
  - With probability ⅓ answer "no"
  - With probability ⅓ answer that $x \in \tilde{K}$ *iff* $x \in K$
  - With probability ⅓ simulate the Turing machine x on input x. When it stops and gives y, answer that $x \in \tilde{K}$ iff the truth-table condition y is satisfied by K

Turing machine x produces y and $x \in \tilde{K}$
truth-table condition y is satisfied by K

Proof

- $\tilde{K} \leq_{tt}^{pr} K$ with probability ⅔
- Algorithm:
  - With probability ⅓ answer "no"
  - With probability ⅓ answer that $x \in \tilde{K}$ *iff* $x \in K$
  - With probability ⅓ simulate the Turing machine x on input x. When it stops and gives y, answer that $x \in \tilde{K}$ iff the truth-table condition y is satisfied by K

Turing machine x produces y and $x \notin \tilde{K}$ truth-table condition y is not satisfied by K

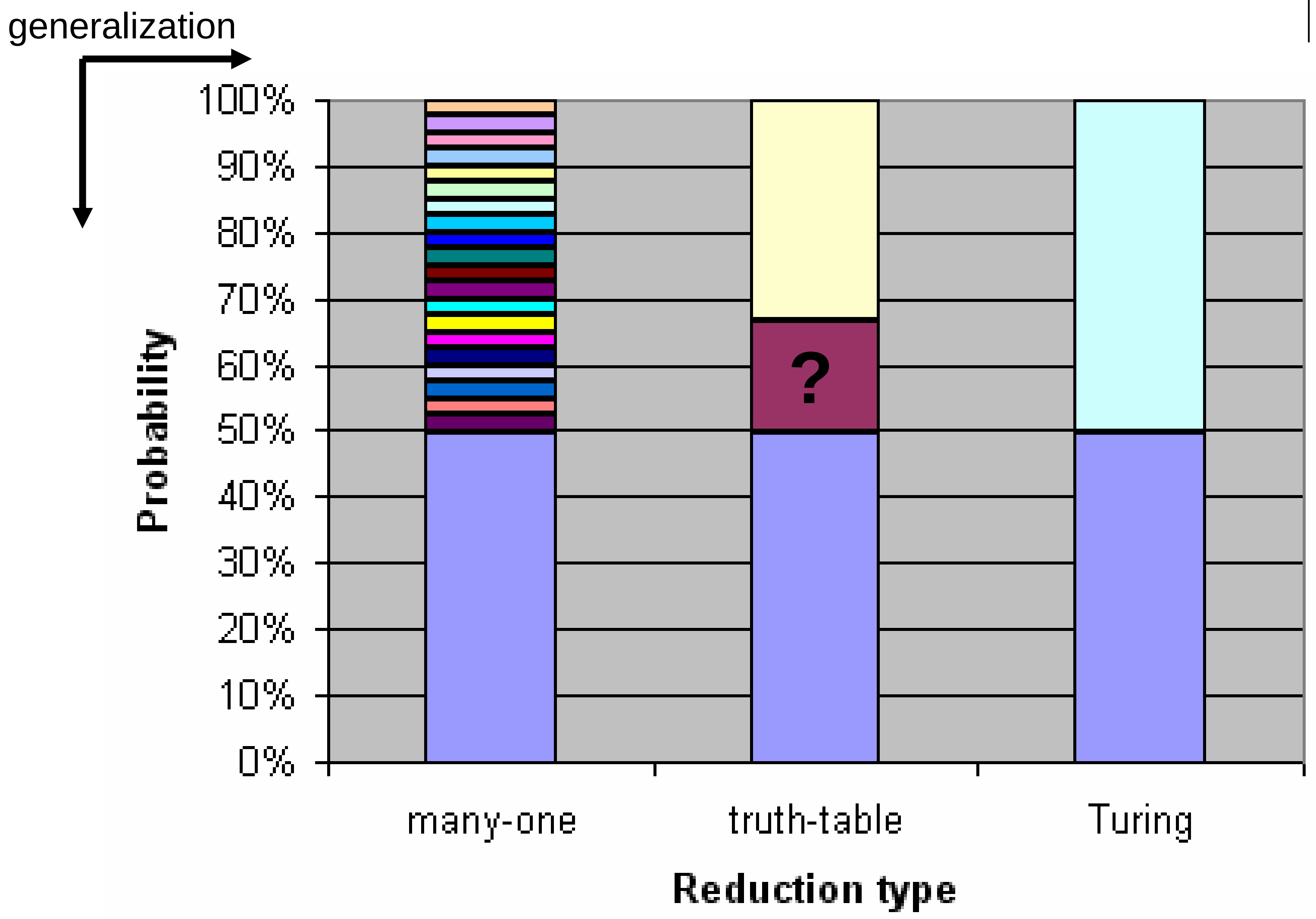
Proof

- In all cases the algorithm gives the correct answer with probability $\frac{2}{3}$

Proof $A \leq_{tt}^{\Pr>2/3} B \Rightarrow A \leq_{tt} B$

- Simulate the reducing Turing machine
- When truth-tables are produced in computation paths with probability >⅔, combine them all into a new truth-table
- If the probability to say "yes" or "no" is greater than ⅓ then it is the correct answer

Conclusion


generalization
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
Probability
?
many-one
truth-table
Turing
Reduction type

Thank you!

Questions?