CS 6810 – Theory of Computing

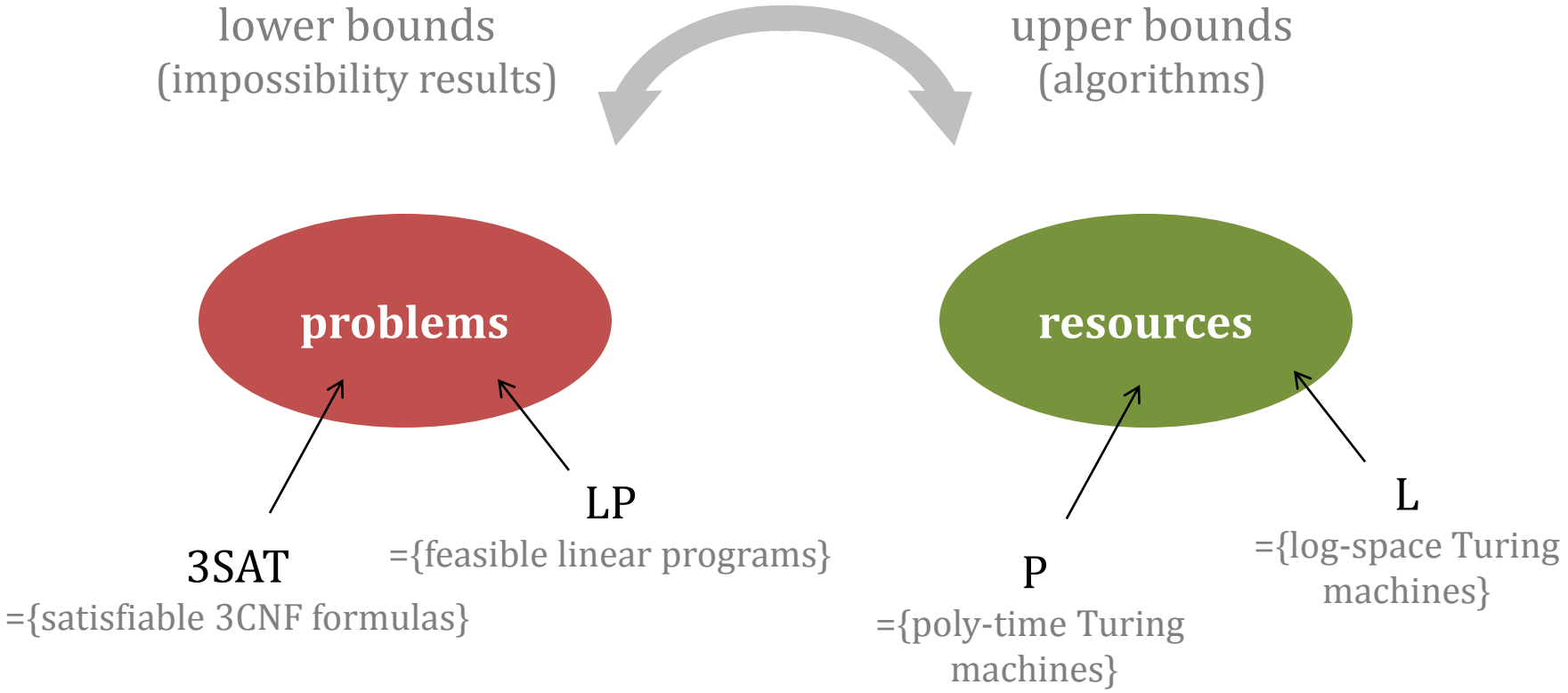
Lecture 1: Introduction & Review

David Steurer

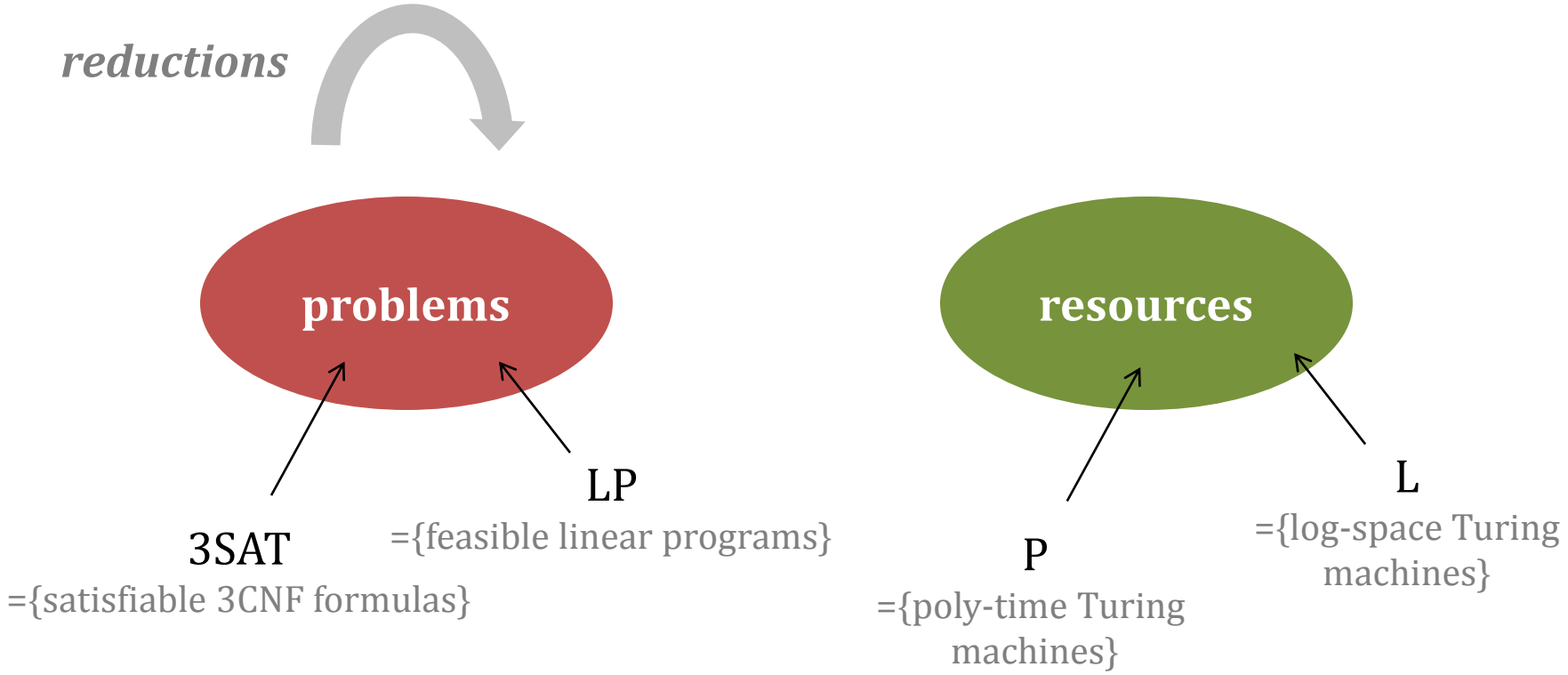
August 23, 2012

What is complexity theory?

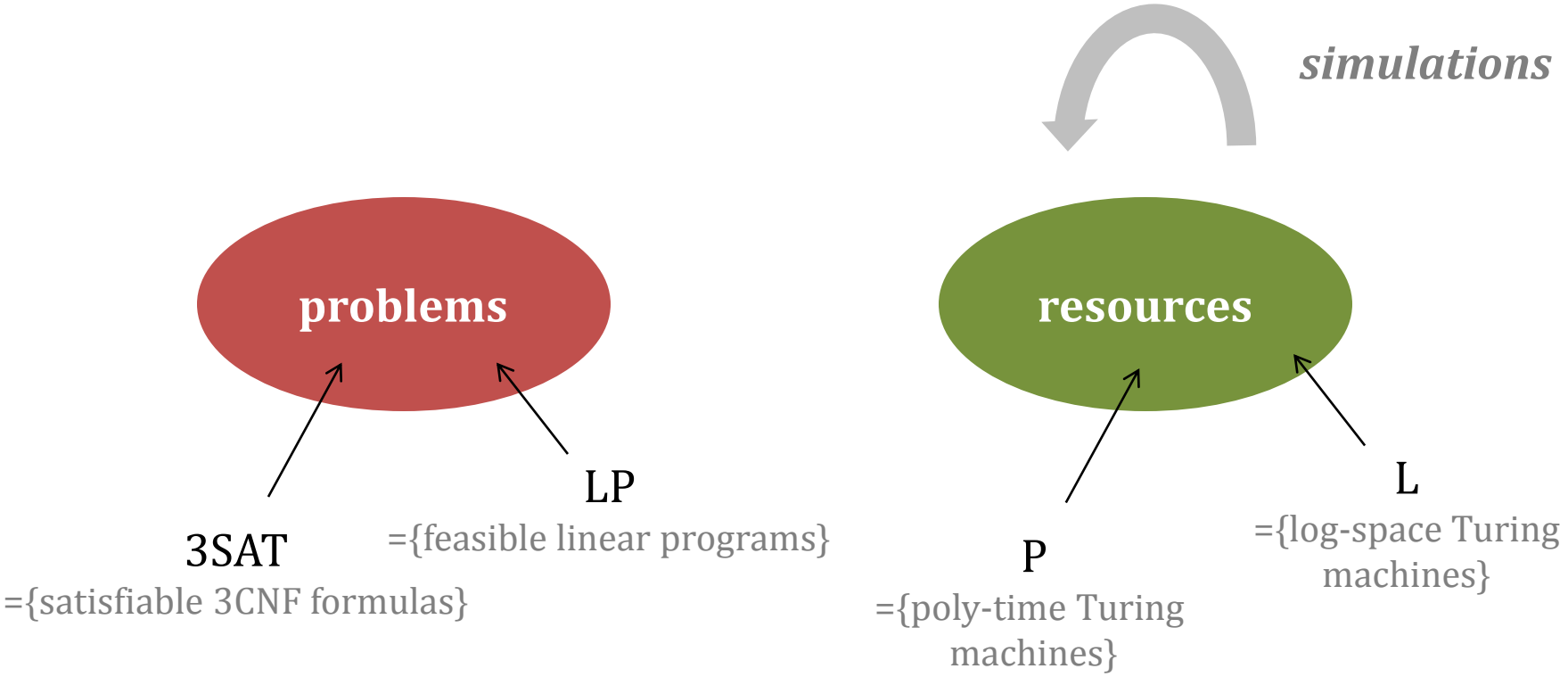
How do resource limitations impact our ability to solve problems?



Which problems are easier than others?



What resources are more powerful than others?



decision problems *(simplest, default)*

search problems *(most general)*

promise problems *(very useful case between decision and search)*

distributional problems *(average-case complexity, later in course)*

decision problems *(simplest, default)*

$f: \{0,1\}^* \rightarrow \{0,1\}$ or $L \subseteq \{0,1\}^*$

(YES/NO answer for each input) (set of YES inputs)

Example

3SAT = {satisfiable 3CNF formulas ϕ}

search problems *(most general)*

for every input x, there is a set of acceptable outputs y

relation $R = \{(x, y) |$ output y is acceptable on input $x\}$

Examples

$3SAT^s = \{(\phi, z) \mid$ assignment z satisfies 3CNF formula $\phi\}$

$Max3SAT^s = \{(\phi, z) \mid$ assignment z satisfies as many clauses as possible in 3CNF formula $\phi\}$

promise problems *(very useful case between decision and search)*

for every input x, there is a set of acceptable outputs $y \in \{0,1\}$

partition into YES inputs, NO inputs, and DON'T-CARE inputs

Example

Max3SAT(α)

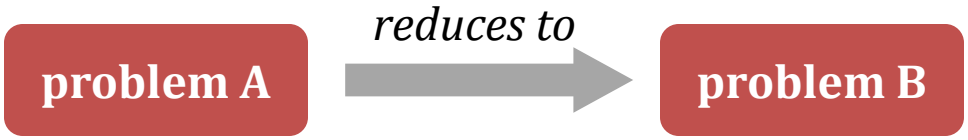
YES: satisfiable 3CNF formula

NO: at most α fraction of clauses satisfiable

resources

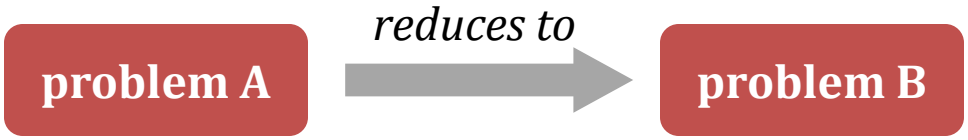
model	measure
Turing machine	time and space
	non-determinism, alternation
	randomness
	advice
circuits	size and depth
protocols	communication cost
resolution proofs	length
linear program	number of constraints (facets)

Reductions



Turing/Cook	solve A using an algorithm for B as a subroutine (use B as an oracle)
Karp	map instances of A to instances of B such that answer is preserved *(for decision & promise problems)*
Levin	map A-instances to B-instances such that any answer to B-instance can be pulled back *(for search problems)*

Reductions



Shows:

B is harder than A

meaningful if A is hard *(plausibly)*

Example: NP-hardness reductions

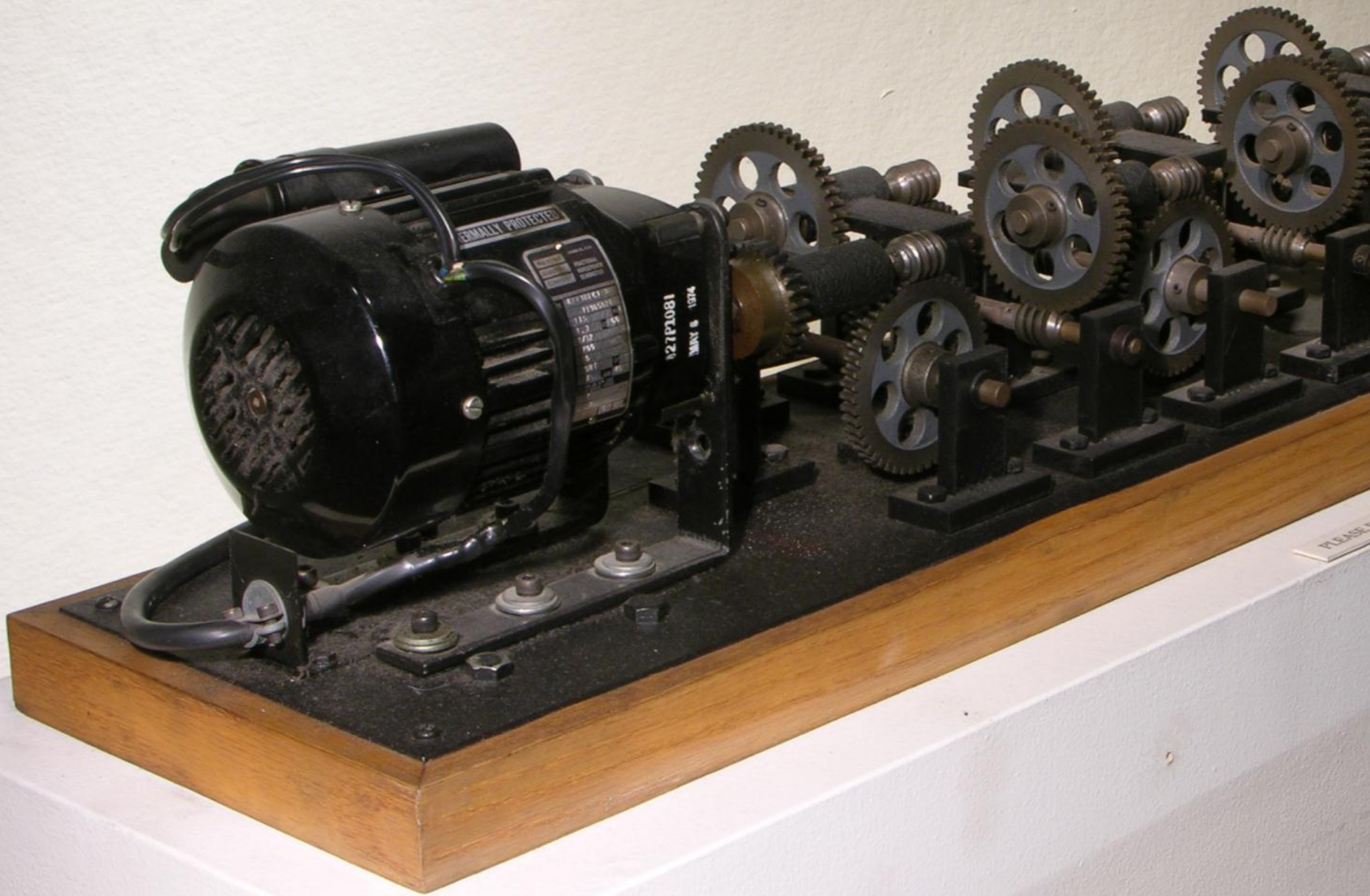
Warning:
3UNSAT is NP-hard w.r.t.
Cook reductions but *not* w.r.t.
Karp reductions (unless NP=coNP)

A is easier than B

meaningful if B is easy *(plausibly)*

Example: LP relaxations
(reduction to LP)

rounding algorithm is pull-back
from LP to original problem
(constructive Levin reduction)


THERMALLY PROTECTED

Machine with Concrete

Arthur Ganson