15-780: Graduate AI

Lecture 4. SAT, Reductions

Geoff Gordon (this lecture)

Tuomas Sandholm

TAs Byron Boots, Sam Ganzfried

Admin

HW1

- *Reminder: HW1 is out*
- *Download from course website*
- *We'll provide hardcopies on request*

Review

Proofs in FOL

- *Proof by contradiction*
- *Unification, variable substitutions*
- *First-order resolution & factoring*
- *Above is a sound, complete proof system*
  - *Herbrand universe, propositionalizing*
  - *Lifting lemma*

Variations on FOL & proofs

- *Wh-questions (who killed JR?), answer literals*
- *Equality (paramodulation or axiom schema)*
- *Second-order logic*

Knowledge engineering

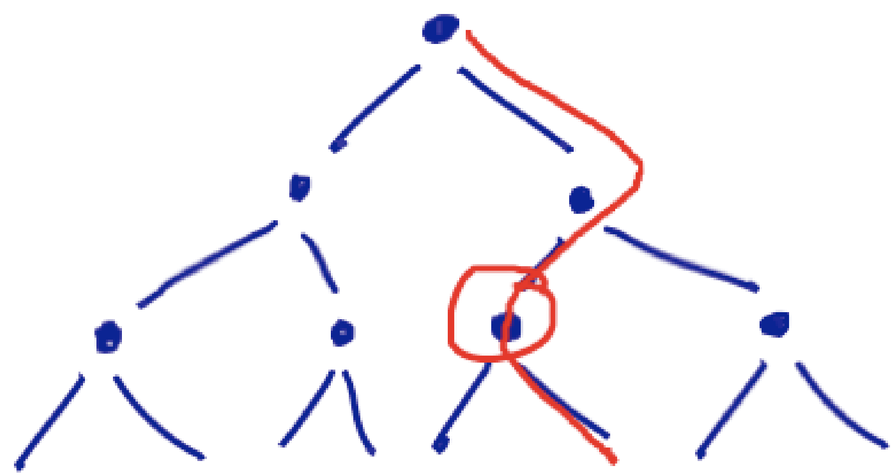
- *Hierarchies (isa, partOf), inheritance*
- *Fluents*
- *Frame problem*
- *Debugging*

Temporal Logics

QCTL*

- *Quantified Computation-Tree Logic **
- *FOL + representation of time + quantifiers over time*

Time in QCTL*



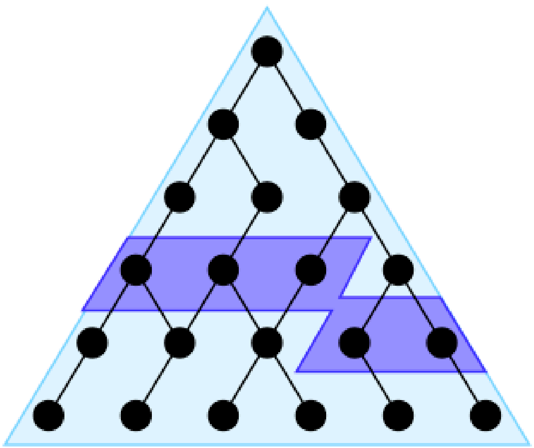
- *Statements interpreted relative to a current history and time point (h, w)*

New quantifiers in QCTL*

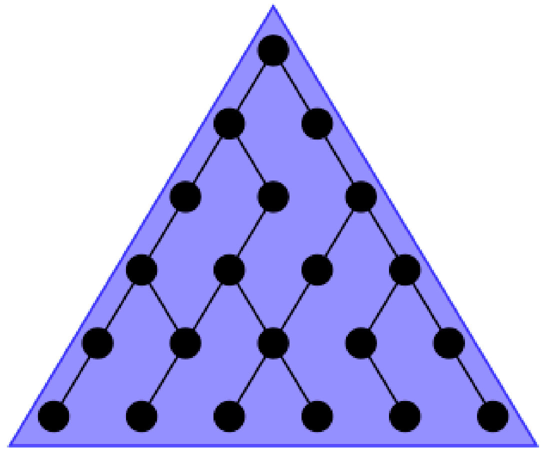
- A S = *"always S"*
  - *S is true for (h', w) for all h' passing through w*
- P U Q = *"P until Q"*
  - *there is some time v in h, with v > w, such that P holds for (h, u) for all u with w < u < v, and Q holds at (h, v)*

Derived quantifiers

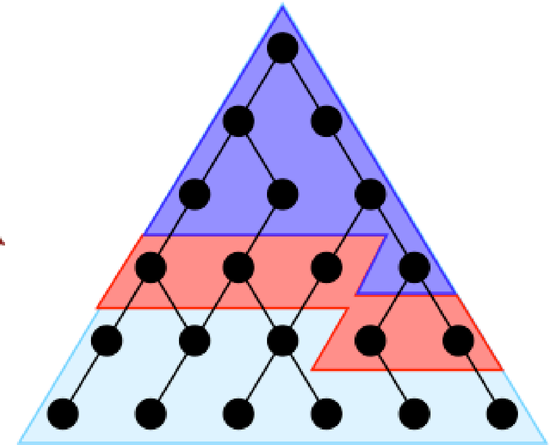
- F P = *"in the future, P"* = T U P
  - *also called "finally, P"*
- G P = *"globally, P"* = P U F
- E P = *"possibly, P"* = $\neg$A $\neg P$
  - *labeled* E *for existential*



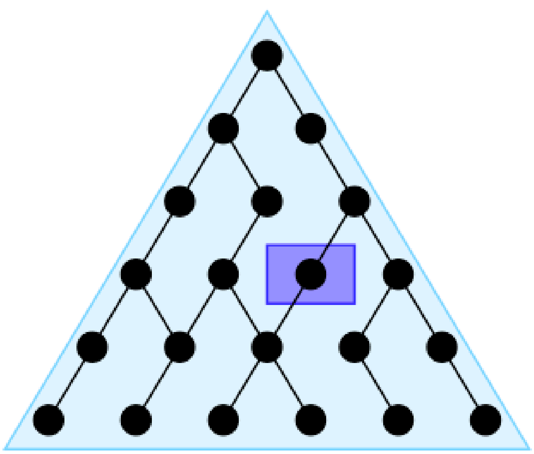
AF p



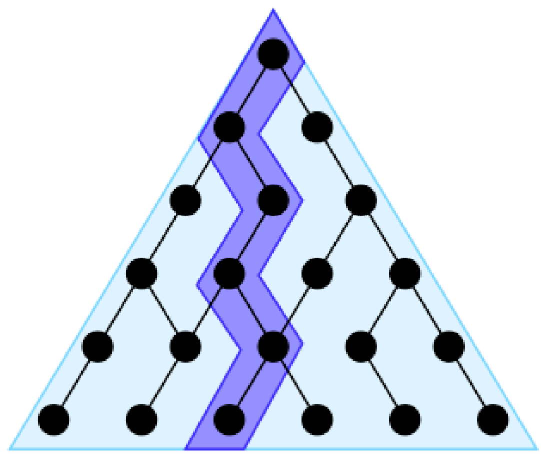
AG p



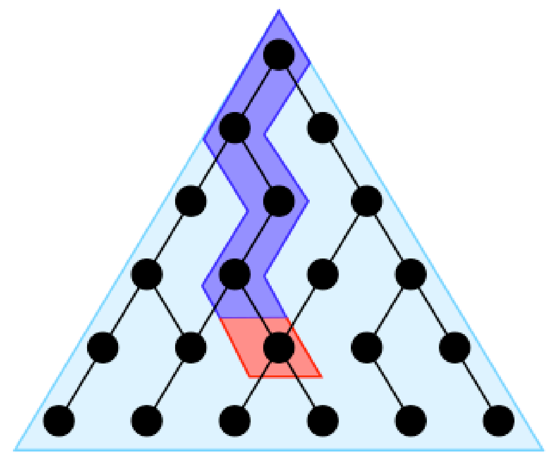
A[p U q]



EF p



EG p



E[p U q]

image credit: Alessandro Artale

Completeness for QCTL*

- *Theorem: the set of valid formulas of QCTL* is not recursively enumerable, even if we restrict to formulas with at most two variables and only unary or 0-ary predicates.*

I Hodkinson, F Wolter, M Zakharyaschev. Decidable and undecidable fragments of first-order branching temporal logics. Proc. 17th Annual IEEE Symposium on Logic in Computer Science (LICS 2002), pp. 393-402.

Intuition

- *The length of a single future is countably infinite, and the set of all futures is uncountably infinite*
- *QCTL* quantifies over uncountable sets*
- *FOL only can quantify over countable sets*
    - *e.g., no FOL translation of* AF*P*

SAT

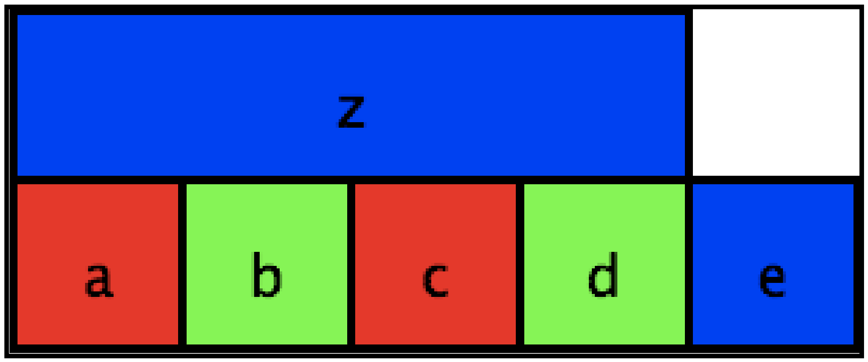
Satisfiability

- *SAT is the problem of determining whether a given propositional logic sentence is satisfiable*
- *A* ***decision problem****: given an instance, answer yes or no*
- *A fundamental problem in CS*

SAT is a general problem

- *Many other useful decision problems* ***reduce*** *to SAT*
- *Informally, if we can solve SAT, we can solve these other problems*
- *So a good SAT solver is a good AI building block*

Example decision problem



- *k-coloring: can we color a map using only k colors in a way that keeps neighboring regions from being the same color?*

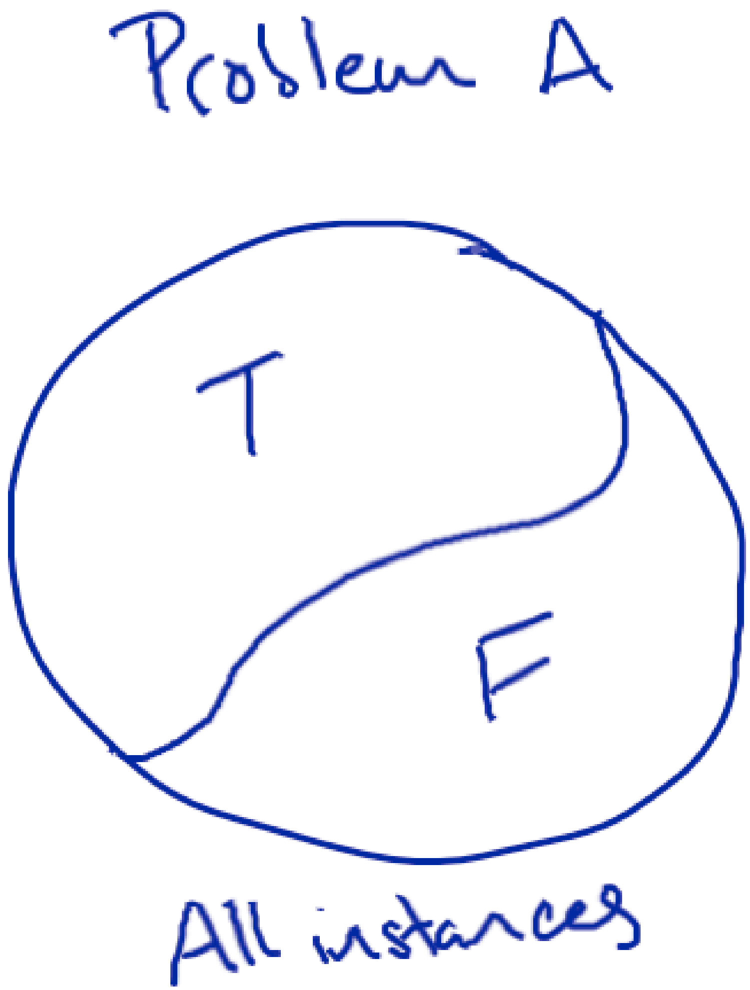
Example decision problem

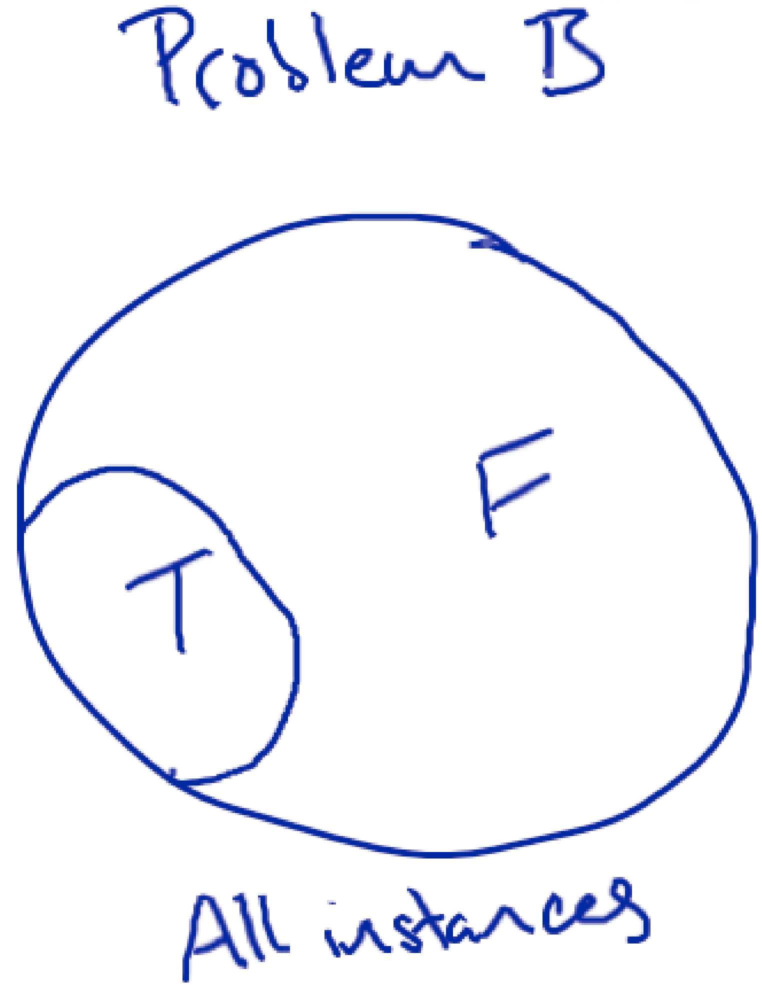
- *Propositional planning: given a list of operators (w/ preconditions, effects), can we apply operators in some order to achieve a desired goal?*
- *Have cake & eat it too example*
- *We’ll see later how to represent as SAT*

Reduction

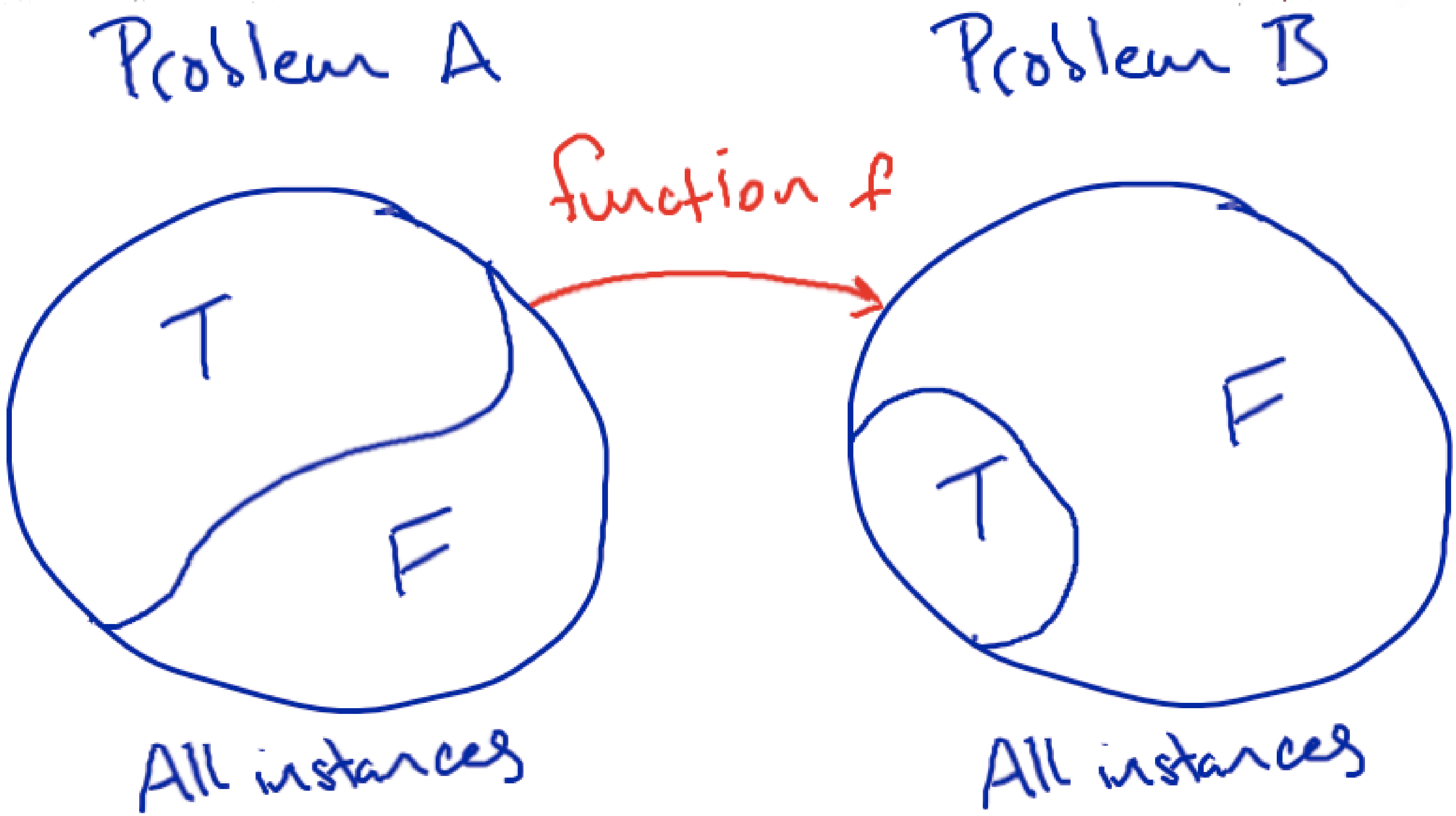
- *Loosely, "A reduces to B" means that if we can solve B then we can solve A*
- *More formally, A, B are decision problems (instances* $\mapsto$ *truth values)*
- *A reduction is a poly-time function f such that, given an instance a of A*
  - *f(a) is an instance of B, and*
  - $A(a) = B(f(a))$

Reduction picture

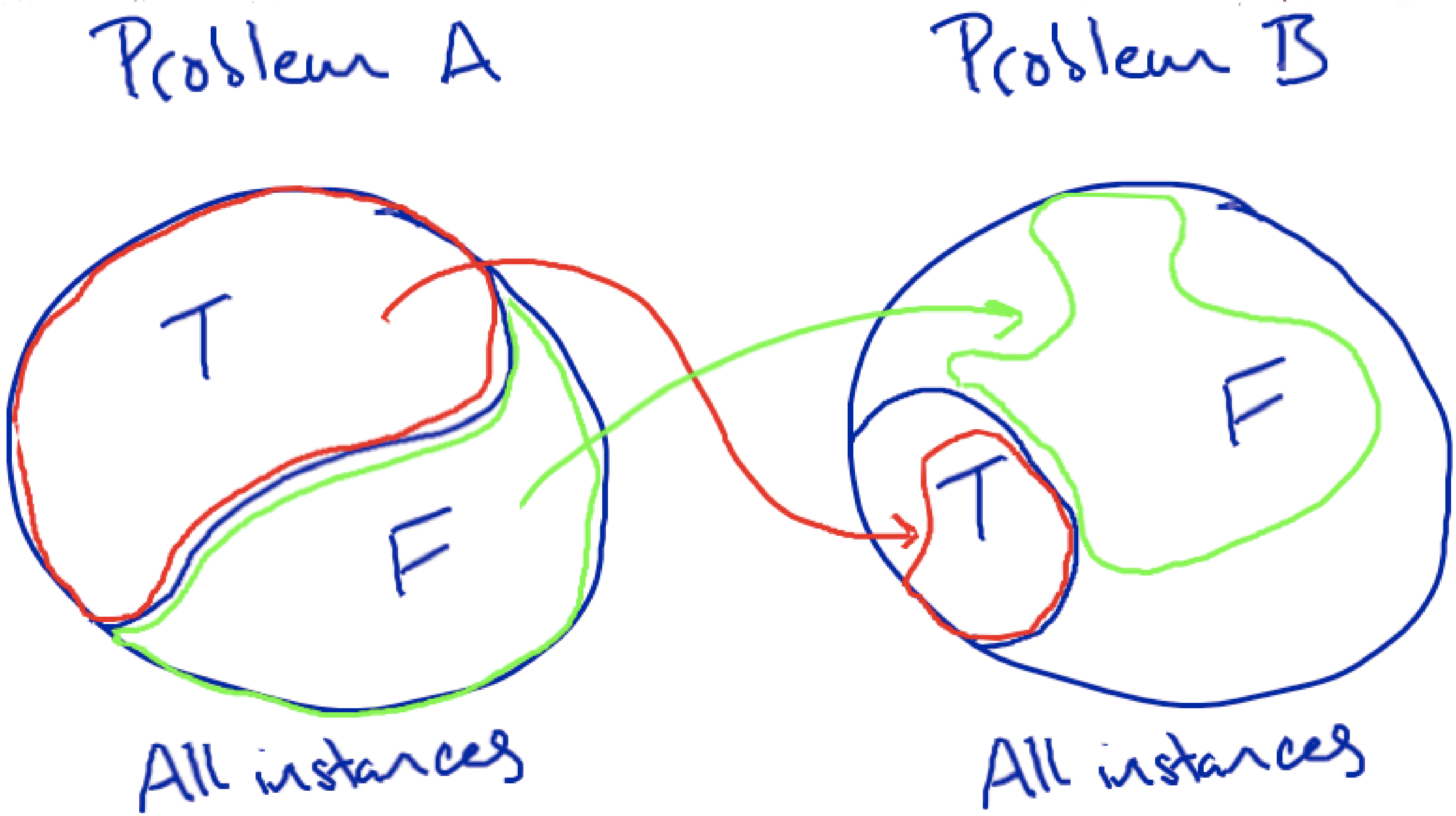




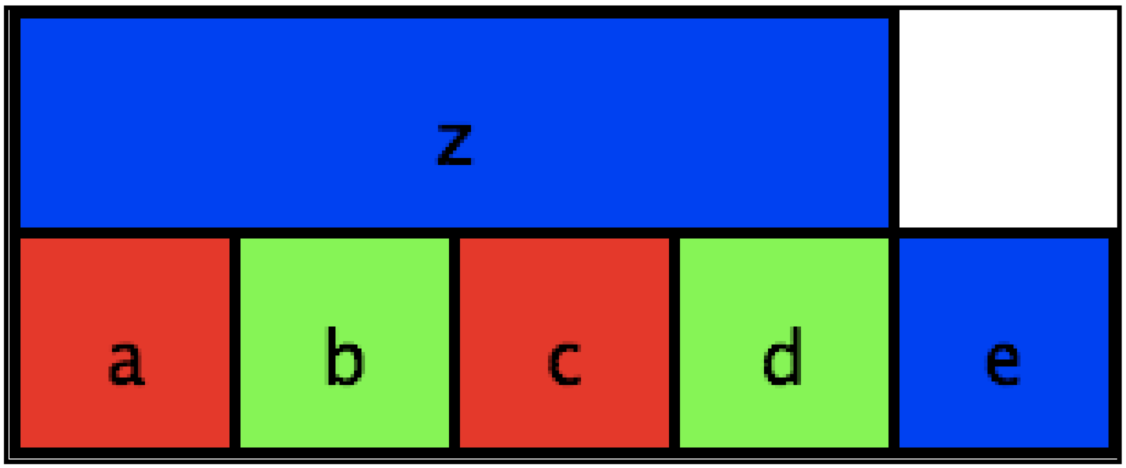
Reduction picture



Reduction picture



Back to example

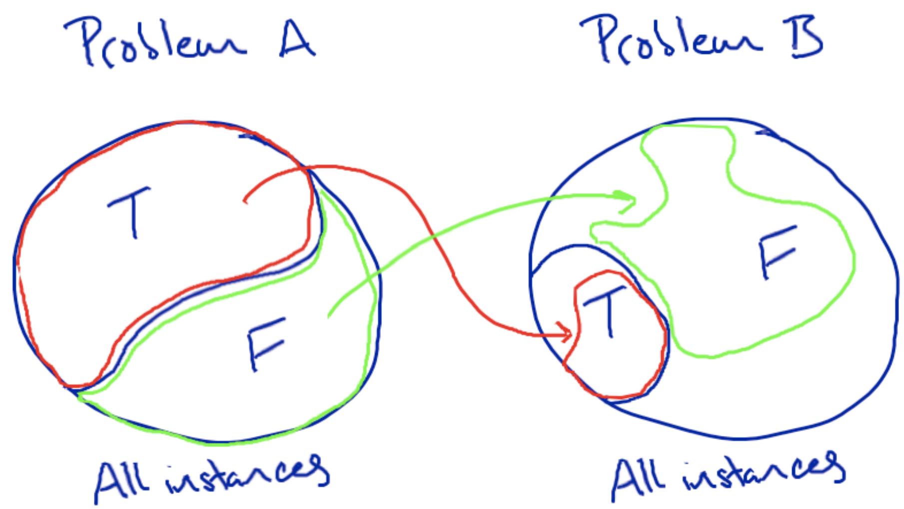


- *Each square must be red, green, or blue*
- *Adjacent squares can't both be red (similarly, green or blue)*

Back to example

- $(a_r \vee a_g \vee a_b) \wedge (b_r \vee b_g \vee b_b) \wedge (c_r \vee c_g \vee c_b) \wedge (d_r \vee d_g \vee d_b) \wedge (e_r \vee e_g \vee e_b) \wedge (z_r \vee z_g \vee z_b)$
- $(\neg a_r \vee \neg b_r) \wedge (\neg a_g \vee \neg b_g) \wedge (\neg a_b \vee \neg b_b)$
- $(\neg a_r \vee \neg z_r) \wedge (\neg a_g \vee \neg z_g) \wedge (\neg a_b \vee \neg z_b)$
- …

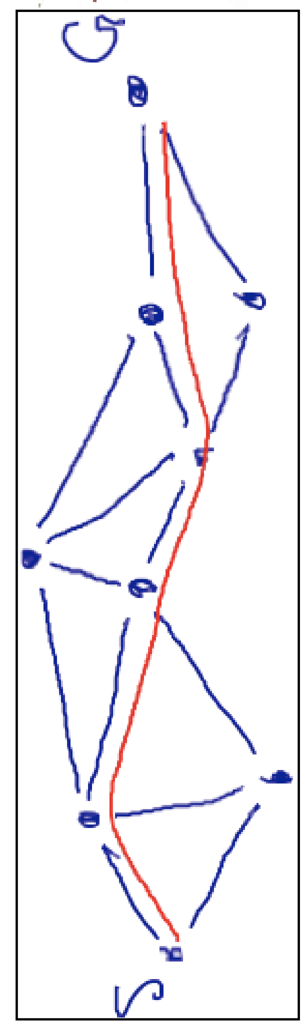
Direction of reduction



- *If A reduces to B:*
  - *if we can solve B, we can solve A*
  - *so B must be at least as hard as A*
- *Trivially, can take an easy problem and reduce it to a hard one*

Not-so-useful reduction

- *Path planning reduces to SAT*
- *Variables: is edge e in path?*
- *Constraints:*
    - *exactly 1 path-edge touches start*
    - *exactly 1 path-edge touches goal*
    - *either 0 or 2 touch each other node*



More useful: reduction to 3SAT

- *We saw that decision problems can be reduced to SAT*
  - *is CNF formula satisfiable?*
- *Can reduce even further, to 3SAT*
  - *is 3CNF formula satisfiable?*
- *Useful if reducing SAT/3SAT to another problem (to show other problem hard)*

SAT reduces to 3SAT

- *Must get rid of long clauses*
- *E.g.,* $(a \lor \neg b \lor c \lor d \lor e \lor \neg f)$
- *Replace with*

$$(a \lor \neg b \lor x) \land (\neg x \lor c \lor y) \land$$
$$(\neg y \lor d \lor z) \land (\neg z \lor e \lor \neg f)$$

NP-completeness

- *S. A. Cook in 1971 proved that many useful decision problems reduce back and forth to SAT*
  - *showed how to simulate poly-size-memory computer w/ (very complicated, but still poly-size) SAT problem*
- *Equivalently, SAT is exactly as hard (in theory at least) as these other problems*

S. A. Cook. The complexity of theorem-proving procedures, Proceedings of ACM STOC'71, pp. 151–158, 1971.

Cost of reduction

- *Complexity theorists often ignore little things like constant factors (or even polynomial factors!)*
- *So, is it a good idea to reduce your decision problem to SAT?*
- *Answer: sometimes…*

Cost of reduction

- *SAT is well studied ⇒ fast solvers*
- *So, if there is an efficient reduction, ability to use fast SAT solvers can be a win*
  - *e.g., 3-coloring*
  - *another example later (SATplan)*
- *Other times, cost of reduction is too high*
  - *usu. because instance gets bigger*
  - *will also see example later (MILP)*

Choosing a reduction

- *May be many reductions from problem A to problem B*
- *May have **wildly** different properties*
  - *e.g., solving transformed instance may take seconds vs. days*