Class 6: Quantifiers and More

Schedule

Problem Set 2 is due **Friday at 6:29pm**.

Programs and Proofs

What does it mean to *test* a computing system? What does it mean for a computing system to *always behave correctly*?

Can a mathematical proof guarantee a real computing system will behave correctly?

Minima

The *minimum* of a set with respect to some comparator operator is the element which is "less than" (according to that comparator) every other element: $m \in S$ is the *minimum* of S if and only if $\forall x \in S - \{m\}. m \prec x$.

$$\forall S \in \text{pow}(\mathbb{N}) - \{\varnothing\}.\, \exists m \in S.\, \forall x \in S - \{m\}.\, m < x$$

Formulas, Propositions, and Inference Rules

$P \implies Q$ is a *formula*.
$\forall P \in \{T, F\}. \forall Q \in \{T, F\}. P \implies Q$ is a *proposition*.
$\frac{P}{Q}$ is an *inference rule*.

A *formula* is *satisfiable* if there is some way to make it true.

$P \implies Q$ is satisfiable:

$\exists P \in \{T, F\}. \exists Q \in \{T, F\}. P \implies Q$

We can show a formula is satisfiable by giving *one* choice for the variable assignments that makes it true. For example, $P = T$, $Q = T$.

A *formula* is *valid* if there is no way to make it false.

$P \implies Q$ is *not* valid:

$\forall P \in \{T, F\}. \forall Q \in \{T, F\}. P \implies Q$

This proposition is false, we can chose $P = T$, $Q = F$.

An *inference rule* is sound if it never leads to a false conclusion.

Negating Quantifiers

What is the negation of $\forall x \in S. P(x)$?

What is the negation of $\exists x \in S. P(x)$?

Satisfiability

Definition. A formula is in *SAT* if it is in CNF form and it is satisfiable.

Definition. A formula is in *3SAT* if it is in 3CNF form and it is satisfiable.

$$(x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \overline{x_2} \vee x_3) \wedge (\overline{x_1} \vee x_2 \vee \overline{x_3})$$

$$(x_{48} \vee x_{4} \vee \overline{x_{9}}) \wedge (\overline{x_{44}} \vee x_{50} \vee \overline{x_{37}}) \wedge (\overline{x_{8}} \vee \overline{x_{1}} \vee x_{28}) \wedge (x_{21} \vee x_{27} \vee \overline{x_{32}}) \wedge (x_{17} \vee x_{29} \vee \overline{x_{30}}) \wedge (x_{30} \vee x_{24} \vee x_{37}) \wedge (\overline{x_{22}} \vee \overline{x_{27}} \vee \overline{x_{44}}) \wedge (x_{8} \vee \overline{x_{25}} \vee \overline{x_{24}}) \wedge (\overline{x_{44}} \vee x_{50} \vee x_{14}) \wedge (x_{45} \vee x_{15} \vee x_{37}) \wedge (\overline{x_{16}} \vee x_{14} \vee \overline{x_{36}}) \wedge (\overline{x_{33}} \vee x_{5} \vee x_{26}) \wedge (x_{18} \vee \overline{x_{7}} \vee \overline{x_{24}}) \wedge (x_{31} \vee x_{38} \vee x_{28}) \wedge (x_{31} \vee \overline{x_{33}} \vee \overline{x_{8}}) \wedge (x_{49} \vee x_{7} \vee \overline{x_{6}}) \wedge (x_{34} \vee \overline{x_{8}} \vee x_{46}) \wedge (x_{4} \vee \overline{x_{5}} \vee \overline{x_{35}}) \wedge (x_{43} \vee x_{27} \vee x_{39}) \wedge (\overline{x_{46}} \vee \overline{x_{40}} \vee \overline{x_{27}}) \wedge (\overline{x_{25}} \vee x_{14} \vee \overline{x_{49}}) \wedge (x_{38} \vee x_{5} \vee x_{15}) \wedge (x_{9} \vee x_{14} \vee \overline{x_{19}}) \wedge (x_{45} \vee \overline{x_{42}} \vee \overline{x_{39}}) \wedge (x_{34} \vee \overline{x_{22}} \vee \overline{x_{28}}) \wedge (\overline{x_{20}} \vee x_{15} \vee \overline{x_{8}}) \wedge (\overline{x_{44}} \vee \overline{x_{10}} \vee \overline{x_{9}}) \wedge (x_{22} \vee \overline{x_{31}} \vee x_{14}) \wedge (\overline{x_{9}} \vee \overline{x_{42}} \vee \overline{x_{15}}) \wedge (\overline{x_{40}} \vee x_{12} \vee \overline{x_{32}}) \wedge (\overline{x_{20}} \vee \overline{x_{6}} \vee \overline{x_{15}}) \wedge (\overline{x_{37}} \vee x_{39} \vee \overline{x_{23}}) \wedge (\overline{x_{3}} \vee \overline{x_{40}} \vee \overline{x_{32}}) \wedge (\overline{x_{4}} \vee \overline{x_{25}} \vee x_{7}) \wedge (\overline{x_{20}} \vee \overline{x_{36}} \vee \overline{x_{37}}) \wedge (\overline{x_{40}} \vee \overline{x_{35}} \vee x_{39}) \wedge (\overline{x_{43}} \vee \overline{x_{40}} \vee \overline{x_{7}}) \wedge (x_{34} \vee x_{44} \vee x_{26}) \wedge (x_{13} \vee x_{27} \vee x_{28}) \wedge (x_{12} \vee \overline{x_{36}} \vee x_{7}) \wedge (\overline{x_{16}} \vee x_{9} \vee \overline{x_{24}}) \wedge (\overline{x_{48}} \vee x_{14} \vee x_{28}) \wedge (x_{16} \vee x_{4} \vee x_{40}) \wedge (\overline{x_{25}} \vee x_{15} \vee x_{37}) \wedge (x_{47} \vee \overline{x_{26}} \vee \overline{x_{23}}) \wedge (x_{4} \vee \overline{x_{13}} \vee x_{36}) \wedge (x_{48} \vee \overline{x_{13}} \vee \overline{x_{37}}) \wedge (x_{4} \vee x_{35} \vee \overline{x_{27}}) \wedge (\overline{x_{22}} \vee x_{47} \vee x_{26}) \wedge (\overline{x_{22}} \vee \overline{x_{46}} \vee x_{27}) \wedge (\overline{x_{20}} \vee x_{49} \vee x_{11}) \wedge (x_{42} \vee \overline{x_{10}} \vee x_{28}) \wedge (\overline{x_{45}} \vee x_{28} \vee \overline{x_{37}}) \wedge (x_{14} \vee \overline{x_{32}} \vee \overline{x_{23}}) \wedge (x_{22} \vee x_{14} \vee x_{23}) \wedge (\overline{x_{17}} \vee \overline{x_{46}} \vee \overline{x_{7}}) \wedge (\overline{x_{31}} \vee x_{46} \vee \overline{x_{50}}) \wedge (x_{34} \vee \overline{x_{41}} \vee x_{43}) \wedge (x_{17} \vee \overline{x_{9}} \vee x_{15}) \wedge (x_{46} \vee x_{14} \vee \overline{x_{12}}) \wedge (\overline{x_{20}} \vee x_{12} \vee x_{14}) \wedge (x_{41} \vee x_{42} \vee \overline{x_{15}}) \wedge (x_{48} \vee x_{46} \vee \overline{x_{36}}) \wedge (\overline{x_{22}} \vee \overline{x_{4}} \vee \overline{x_{49}}) \wedge (x_{22} \vee x_{12} \vee \overline{x_{42}}) \wedge (x_{13} \vee \overline{x_{38}} \vee x_{39}) \wedge (x_{48} \vee \overline{x_{16}} \vee \overline{x_{27}}) \wedge (x_{17} \vee \overline{x_{18}} \vee \overline{x_{26}}) \wedge (x_{48} \vee \overline{x_{40}} \vee \overline{x_{35}}) \wedge (\overline{x_{43}} \vee \overline{x_{40}} \vee \overline{x_{49}}) \wedge (x_{29} \vee x_{11} \vee \overline{x_{32}}) \wedge (x_{33} \vee \overline{x_{17}} \vee x_{39}) \wedge (\overline{x_{25}} \vee \overline{x_{9}} \vee \overline{x_{6}}) \wedge (x_{40} \vee \overline{x_{50}} \vee x_{19}) \wedge (x_{8} \vee x_{10} \vee \overline{x_{27}}) \wedge (x_{5} \vee x_{9} \vee \overline{x_{26}}) \wedge (x_{45} \vee \overline{x_{38}} \vee \overline{x_{27}}) \wedge (\overline{x_{4}} \vee \overline{x_{40}} \vee \overline{x_{42}}) \wedge (x_{21} \vee x_{50} \vee x_{12}) \wedge (\overline{x_{8}} \vee \overline{x_{14}} \vee \overline{x_{42}}) \wedge (\overline{x_{17}} \vee x_{47} \vee \overline{x_{27}}) \wedge (x_{49} \vee \overline{x_{12}} \vee \overline{x_{6}}) \wedge (x_{27} \vee x_{49} \vee \overline{x_{32}}) \wedge (\overline{x_{29}} \vee \overline{x_{12}} \vee \overline{x_{26}}) \wedge (x_{48} \vee \overline{x_{2}} \vee x_{6}) \wedge (x_{16} \vee x_{36} \vee x_{49}) \wedge (x_{33} \vee \overline{x_{12}} \vee \overline{x_{26}}) \wedge (\overline{x_{33}} \vee x_{29} \vee x_{49}) \wedge (\overline{x_{48}} \vee x_{2} \vee x_{19}) \wedge (x_{25} \vee x_{36} \vee x_{49}) \wedge (x_{21} \vee x_{40} \vee \overline{x_{14}}) \wedge (\overline{x_{34}} \vee \overline{x_{44}} \vee \overline{x_{6}}) \wedge (x_{48} \vee \overline{x_{50}} \vee \overline{x_{1}}) \wedge (x_{5} \vee \overline{x_{12}} \vee x_{7}) \wedge (x_{21} \vee \overline{x_{35}} \vee \overline{x_{27}}) \wedge (\overline{x_{22}} \vee \overline{x_{16}} \vee \overline{x_{14}}) \wedge (\overline{x_{13}} \vee \overline{x_{35}} \vee \overline{x_{12}}) \wedge (\overline{x_{4}} \vee \overline{x_{35}} \vee \overline{x_{42}}) \wedge (\overline{x_{50}} \vee \overline{x_{40}} \vee x_{7}) \wedge (x_{25} \vee x_{47} \vee \overline{x_{12}})$$

Converting Truth Tables to CNF

P	Q	$P \implies Q$	$P \oplus Q$
T	**T**	**T**	**F**
T	**F**	**F**	**T**
F	**T**	**T**	**T**
F	**F**	**T**	**F**

The output of the operator is **T** if and only if the inputs do not match *any row* where the output is **F**. We can ensure the inputs do not match a row, but OR-ing the negation of each input: in the disjunction, at least one must be **T** to satisfy the clause.

Problems

Definition. A *problem* is a precise description of set of possible inputs and desired property of an output corresponding to each input.

Define the *ADDITION* problem (adding two integers):

Definition. A *decision problem* is a problem where the output is either *T* or *F*. Equivalently, we can view a decision problem as testing set membership: $x \in S$.

The *SUM* problem:

> **Input.** Three integers, x, y, and z.
> **Output.** **T** iff $z = x + y$.

How could we solve *ADDITION* using *SUM*?

Definition. A *procedure* is a precise description of an information process.

Definition. An *algorithm* for a particular problem is a procedure that *solves* that problem. To *solve* a problem, an algorithm must always (eventually) produce the correct output for any problem input.

Definition. A *program* is adescription of a procedure that can be executed by a computer.

Definition. The *3SAT decision problem* takes as input a logical formula written in CNF, and outputs *T* if the input formula is satisfiable and outputs *F* otherwise.

How many uses of a solver for the *3SAT* decision problem are sufficient to always find a satisfying assignment for a satisfiable formula?

