

COMP 410 Lecture 2

Kyle Dewey

SAT and Semantic Tableau

SAT Background

SAT

- Short for the Boolean satisfiability problem
- Given a Boolean formula with variables, is there an assignment of true/false to the variables which makes the formula true?

SAT

- Short for the Boolean satisfiability problem
- Given a Boolean formula with variables, is there an assignment of true/false to the variables which makes the formula true?

$$(x \vee \neg y) \wedge (\neg x \vee z)$$

SAT

- Short for the Boolean satisfiability problem
- Given a Boolean formula with variables, is there an assignment of true/false to the variables which makes the formula true?

$$(x \vee \neg y) \wedge (\neg x \vee z)$$

Yes: x is true, z is true

SAT

- Short for the Boolean satisfiability problem
- Given a Boolean formula with variables, is there an assignment of true/false to the variables which makes the formula true?

$$(x \vee \neg y) \wedge (\neg x \vee z)$$

Yes: x is true, z is true

$$(x \wedge \neg x)$$

SAT

- Short for the Boolean satisfiability problem
- Given a Boolean formula with variables, is there an assignment of true/false to the variables which makes the formula true?

$$(x \vee \neg y) \wedge (\neg x \vee z)$$

Yes: x is true, z is true

$$(x \wedge \neg x)$$

No

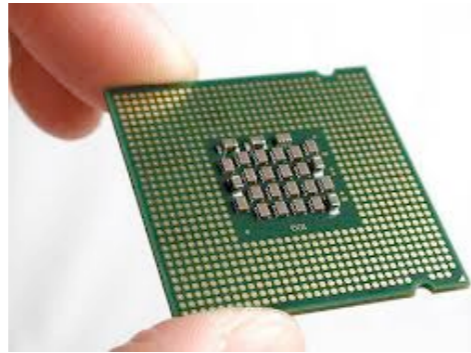
Relevance

Widespread usage in hardware and software verification

- Verification as in proving the system does what we intend
- Much stronger guarantees than testing
- Testing can prove the existence of a bug (a failed test), whereas verification proves the absence of bugs (relative to the theorems proven)

Relevance

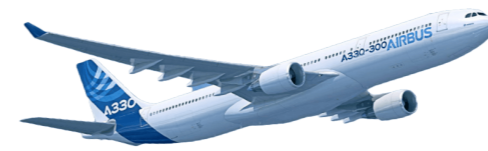
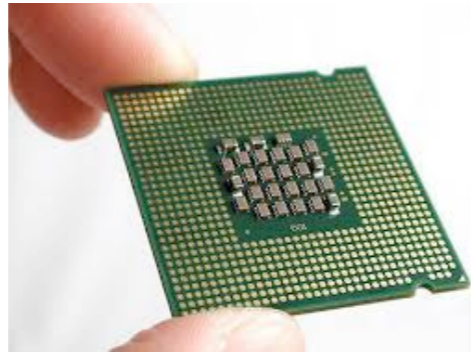
Widespread usage in hardware and software verification



- Circuits can be represented as Boolean formulas
- Can basically phrase proofs as $\text{Circuit} \wedge \text{BadThing}$. If unsatisfiable, then BadThing cannot occur. If satisfiable, then the solution gives the circumstance under which BadThing occurs.
- Many details omitted (entire careers are based on this stuff)

Relevance

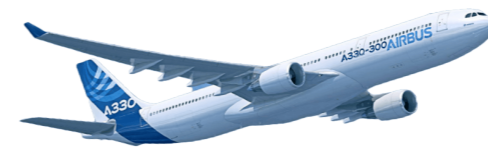
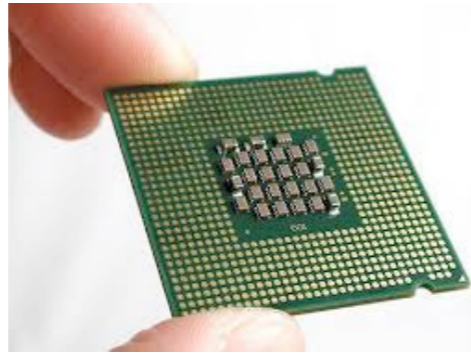
Widespread usage in hardware and software verification



- (Likely) used by Airbus to verify that flight control software does the right thing
- Lots of proprietary details so it's not 100% clear how this verification works, but SAT is still relevant to the problem

Relevance

Widespread usage in hardware and software verification



-Nasa uses software verification for a variety of tasks; SAT is relevant, though other techniques are used, too

Relevance to Logic Programming

- Methods for solving SAT can be used to execute logic programs
- Logic programming can be phrased as SAT with some additional stuff

Semantic Tableau

- One method for solving SAT instances
- Basic idea: iterate over the formula
 - Maintain subformulas that must be true
 - Learn which variables must be true/false
 - Stop at conflicts (unsatisfiable), or when no subformulas remain (have solution)

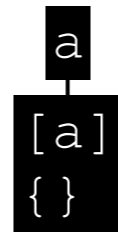
-There are many methods to this

Positive Literals

a

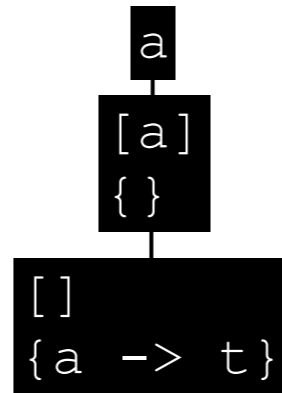
-As in, the input formula is simply "a"

Positive Literals



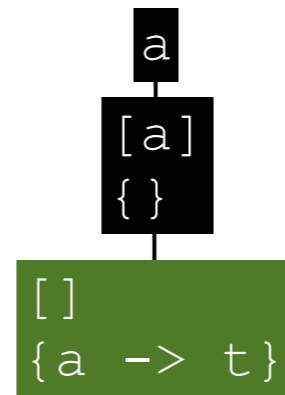
- One subformula must be true: a
- Initially, we don't know what any variables must map to

Positive Literals



-For formula "a" to be true, it must be the case that a is true

Positive Literals



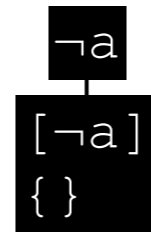
-No subformulas remain, so we are done. The satisfying solution is that a must be true.

Negative Literals

$\neg a$

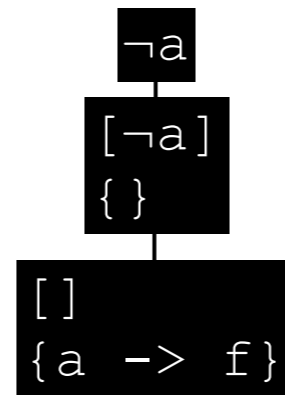
-As in, the input formula is simply " $\neg a$ "

Negative Literals



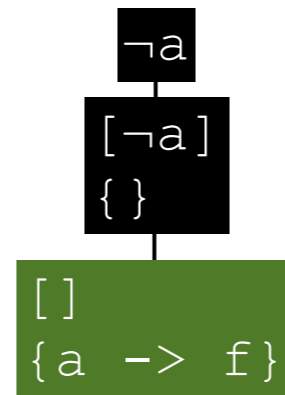
- One subformula must be true: $\neg a$
- Initially, we don't know what any variables must map to

Negative Literals



-For subformula " $\neg a$ " to be true, it must be the case that a is false

Negative Literals

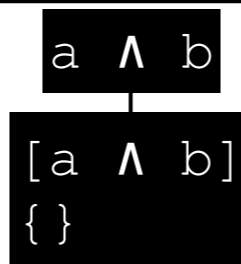


-No subformulas remain, so we are done. The satisfying solution is that “a” must be false.

Logical And

$a \wedge b$

Logical And



- Initially, one subformula must be true: $a \wedge b$
- Initially, we don't know what any variable must map to

Logical And

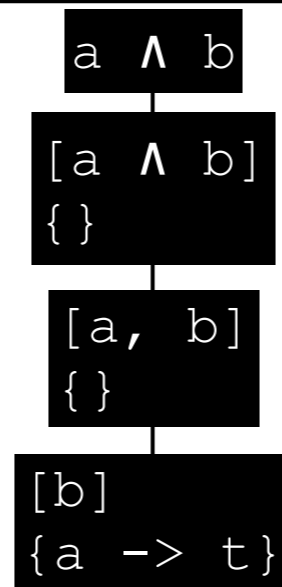
$a \wedge b$

$[a \wedge b]$
{ }

$[a, b]$
{ }

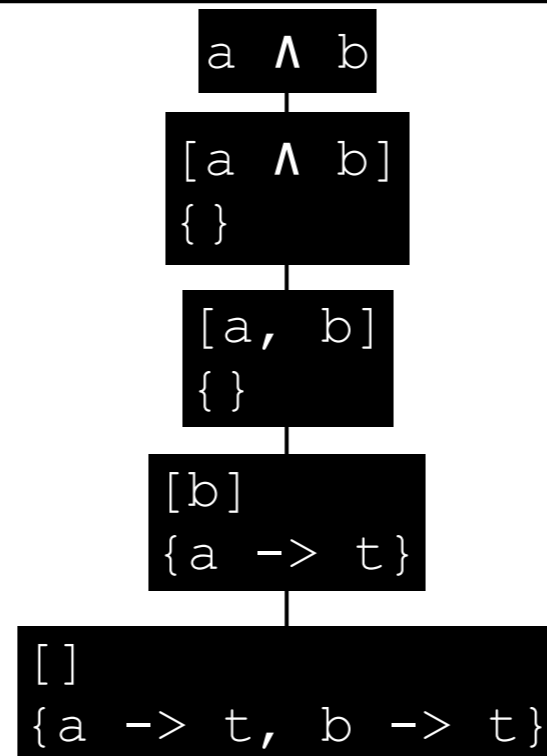
-For $a \wedge b$ to be true, subformulas a and b must both be true

Logical And



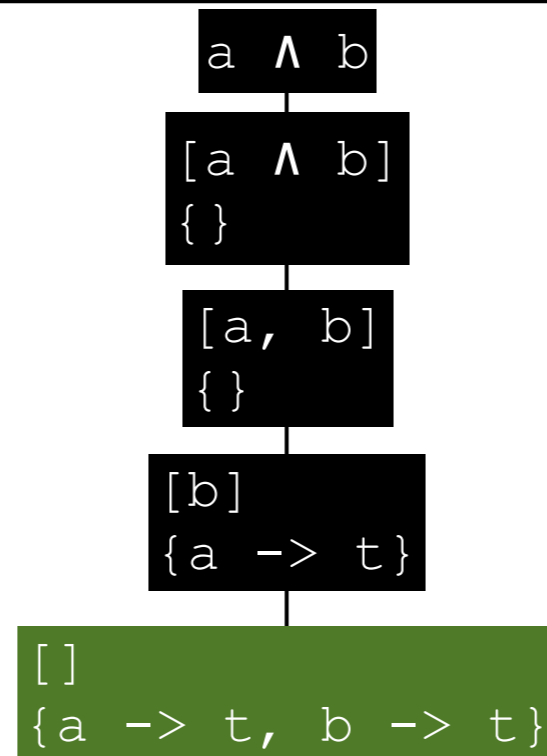
-From the positive literal case, for formula a to be true, variable a must be true

Logical And



-From the positive literal case, for formula b to be true, variable b must be true

Logical And



-No subformulas remain, so we are done with the solution that both a and b must be true

Logical And

$a \wedge \neg a$

-Alternative example, showing a conflict

Logical And

$a \wedge \neg a$

$[a \wedge \neg a]$

$\{\}$

Logical And

$a \wedge \neg a$

$[a \wedge \neg a]$
{}

$[a, \neg a]$
{}

$a \rightarrow f$

Conflict

$[\neg a]$
{ $a \rightarrow t$ }

$[\]$
{ $a \rightarrow t$ }

Logical And

$a \wedge \neg a$

$[a \wedge \neg a]$
{ }

$[\neg a]$
{ $a \rightarrow t$ }



- Now we have a problem: for formula $\neg a$ to be true, it must be the case that variable a is false
- We've already recorded that variable a must be true, which is the opposite of what we expect.
- As such, we have a conflict - this formula is unsatisfiable

Exercise: First Side of SAT Sheet

Logical Or

$$a \vee \neg a$$

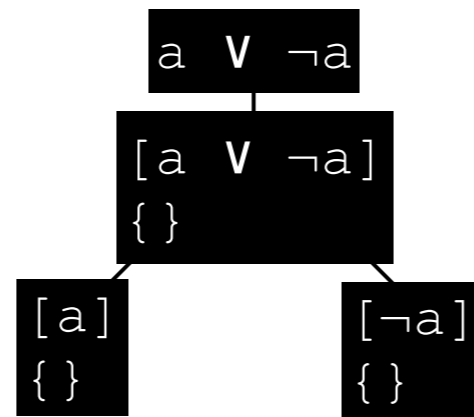
Logical Or

$a \vee \neg a$

$[a \vee \neg a]$

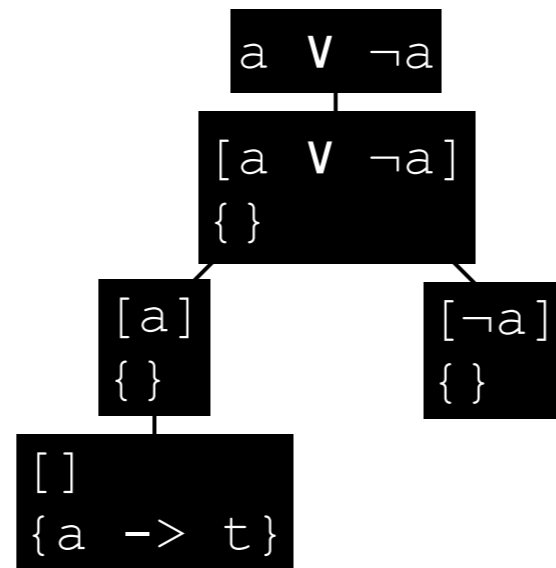
$\{\}$

Logical Or



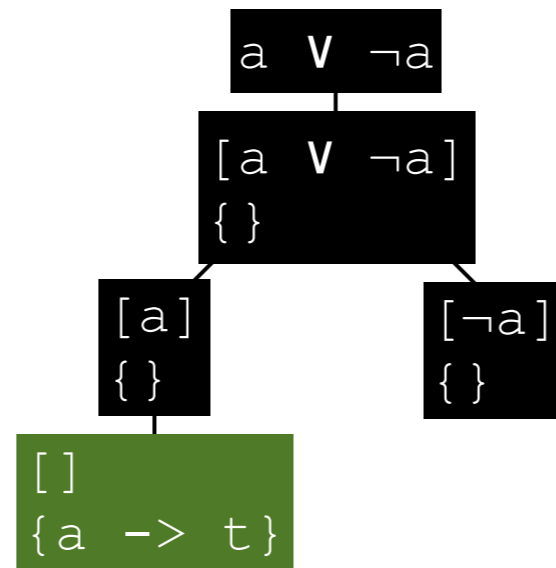
-World splits on or: in one world we pick the left subformula, and in another we pick the right

Logical Or



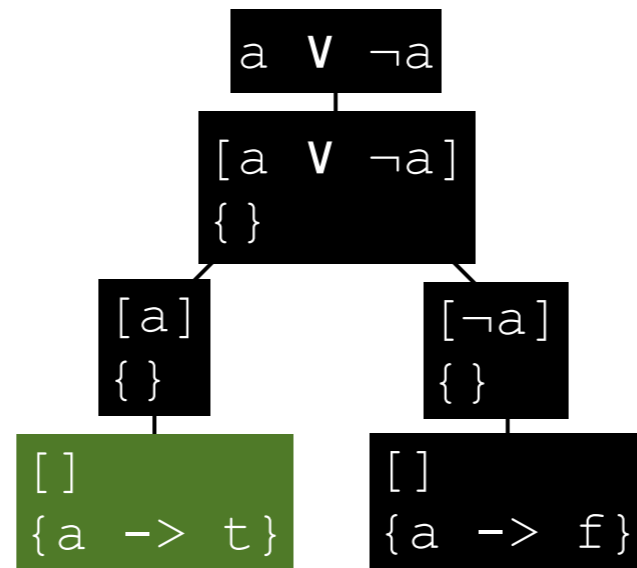
-World splits on or: in one world we pick the left subformula, and in another we pick the right

Logical Or



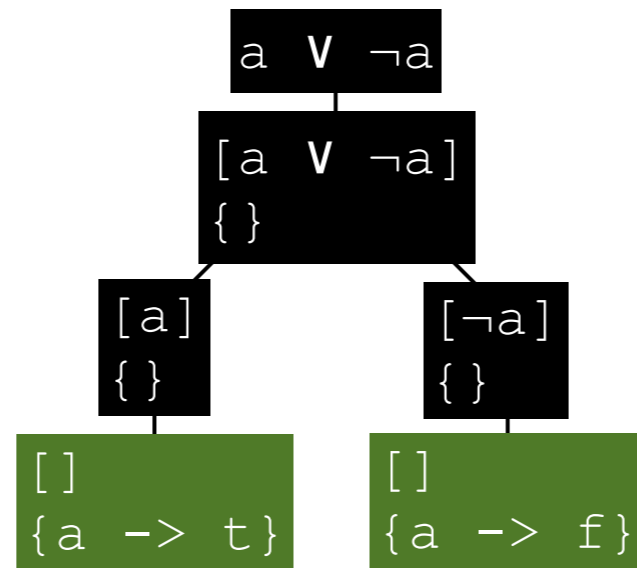
-World splits on or: in one world we pick the left subformula, and in another we pick the right

Logical Or



-World splits on or: in one world we pick the left subformula, and in another we pick the right

Logical Or



-World splits on or: in one world we pick the left subformula, and in another we pick the right

Examples

Example 1:

$(\neg b \vee a) \wedge b$

$(\neg b \vee a) \wedge b$

$(\neg b \vee a) \wedge b$

$[(\neg b \vee a), b]$
 $\{\}$

$(\neg b \vee a) \wedge b$

$[(\neg b \vee a), b]$
 $\{\}$

$[\neg b, b]$
 $\{\}$

$(\neg b \vee a) \wedge b$

$[(\neg b \vee a), b]$
 $\{\}$

$[\neg b, b]$
 $\{\}$

$[b]$
 $\{b \rightarrow f\}$

$(\neg b \vee a) \wedge b$

$[(\neg b \vee a), b]$
 $\{\}$

$[\neg b, b]$
 $\{\}$

$[b]$
 $\{b \rightarrow f\}$



$(\neg b \vee a) \wedge b$

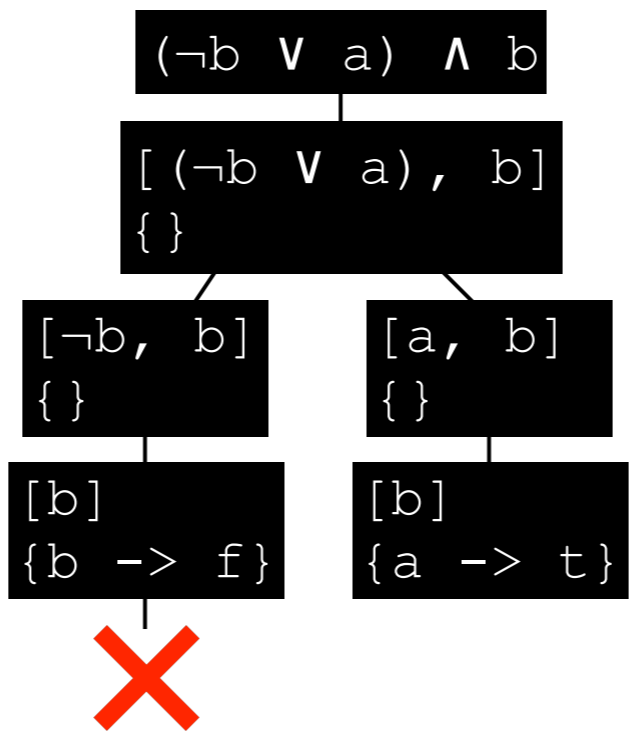
$[(\neg b \vee a), b]$
 $\{\}$

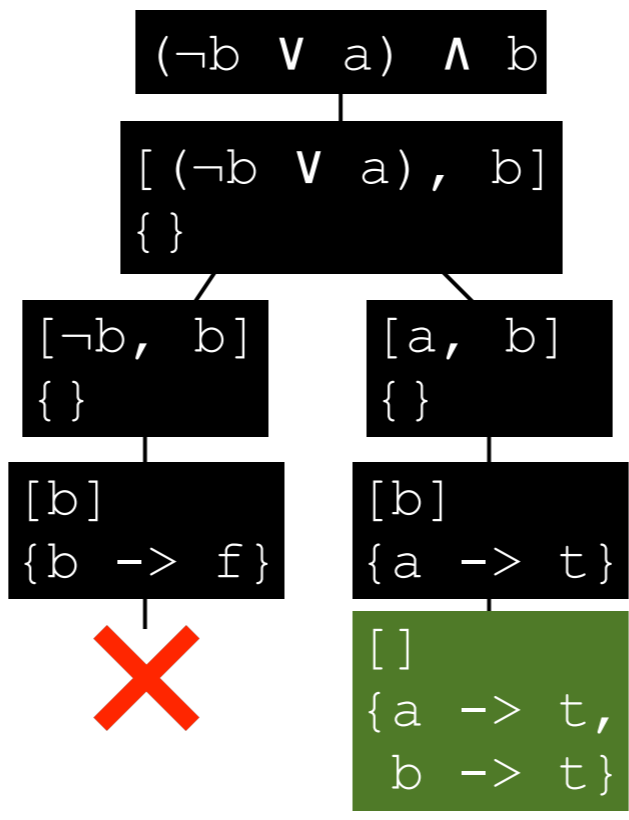
$[\neg b, b]$
 $\{\}$

$[a, b]$
 $\{\}$

$[b]$
 $\{b \rightarrow f\}$







Example 2:

$$(x \vee \neg y) \wedge (\neg x \vee z)$$

$$(x \vee \neg y) \wedge (\neg x \vee z)$$

$(x \vee \neg y) \wedge (\neg x \vee z)$

$[(x \vee \neg y), (\neg x \vee z)]$
 $\{ \}$

$(x \vee \neg y) \wedge (\neg x \vee z)$

$[(x \vee \neg y), (\neg x \vee z)]$
 $\{\}$

$[x, (\neg x \vee z)]$
 $\{\}$

$(x \vee \neg y) \wedge (\neg x \vee z)$

$[(x \vee \neg y), (\neg x \vee z)]$
 $\{\}$

$[x, (\neg x \vee z)]$
 $\{\}$

$[(\neg x \vee z)]$
 $\{x \rightarrow t\}$

$(x \vee \neg y) \wedge (\neg x \vee z)$

$[(x \vee \neg y), (\neg x \vee z)]$
 $\{\}$

$[x, (\neg x \vee z)]$
 $\{\}$

$[(\neg x \vee z)]$
 $\{x \rightarrow t\}$

$[\neg x]$
 $\{x \rightarrow t\}$

$(x \vee \neg y) \wedge (\neg x \vee z)$

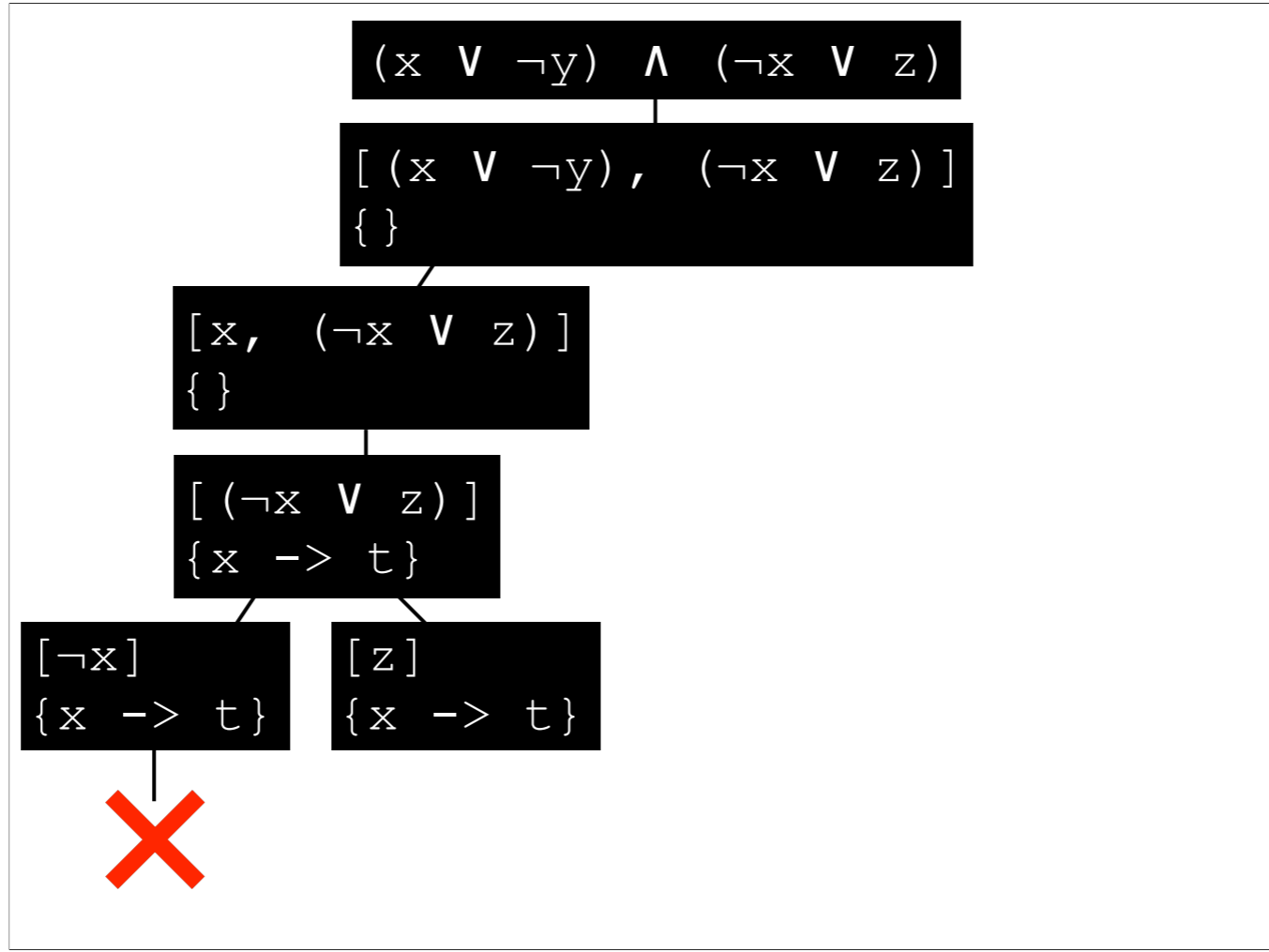
$[(x \vee \neg y), (\neg x \vee z)]$
 $\{\}$

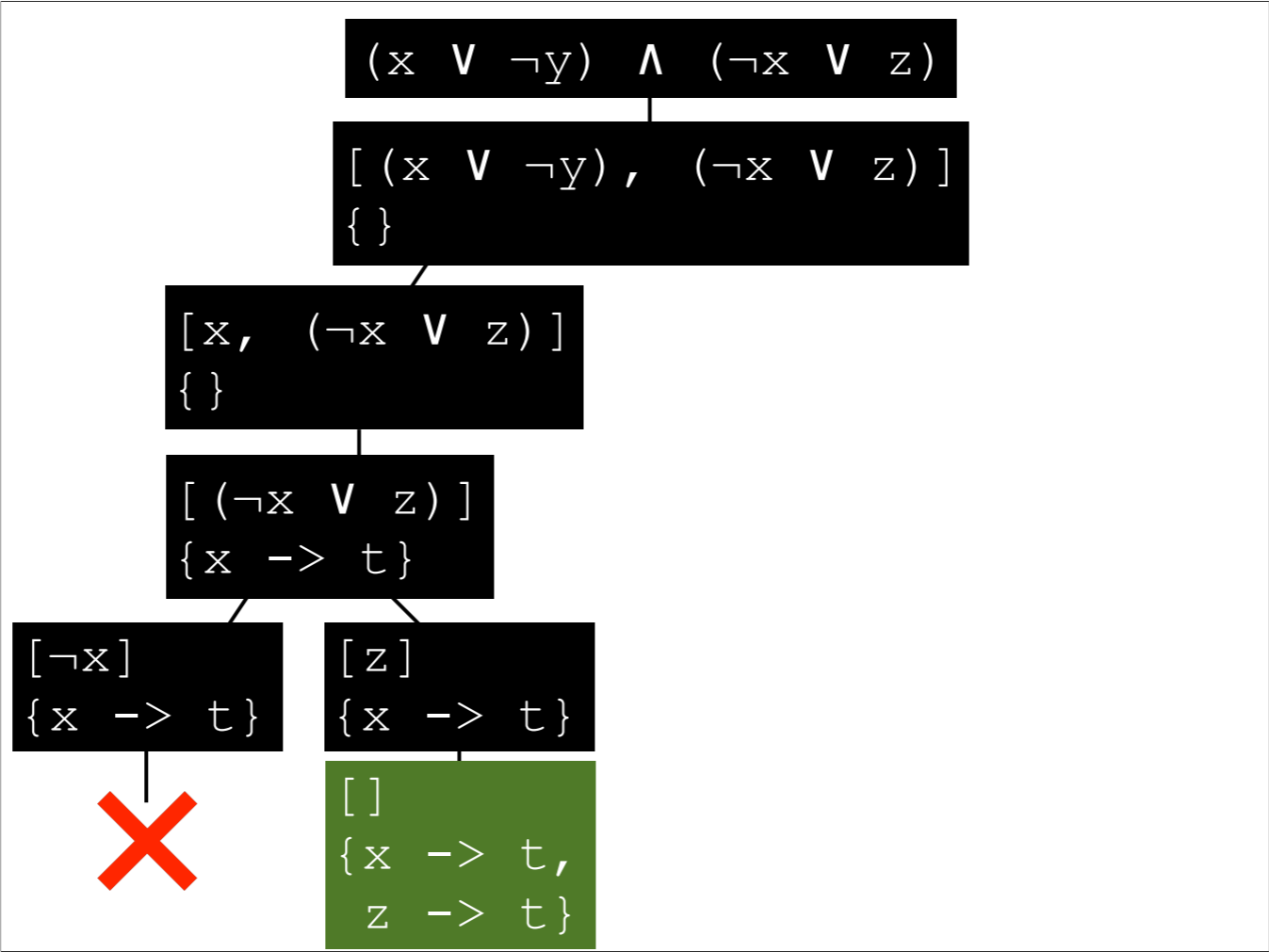
$[x, (\neg x \vee z)]$
 $\{\}$

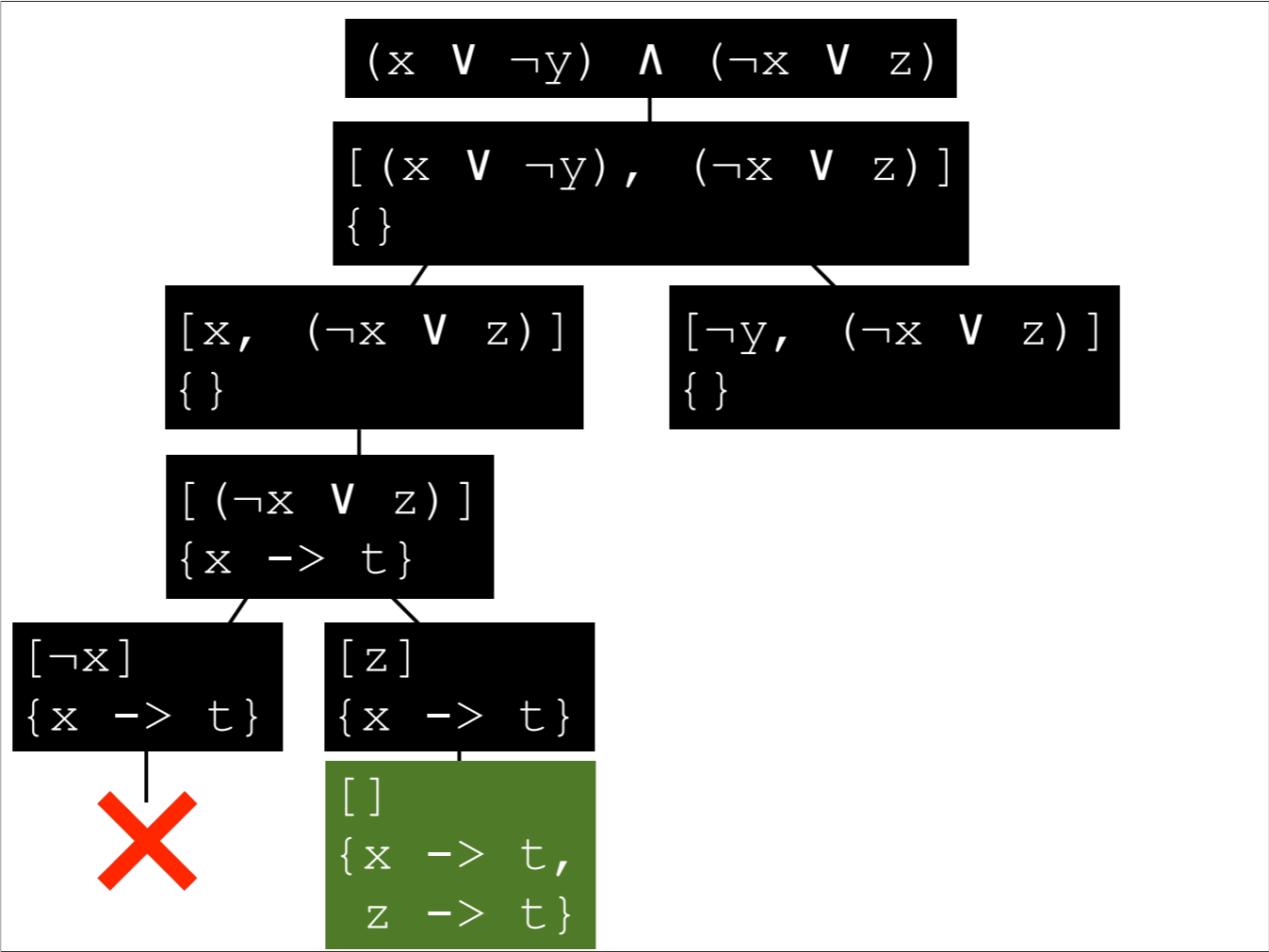
$[(\neg x \vee z)]$
 $\{x \rightarrow t\}$

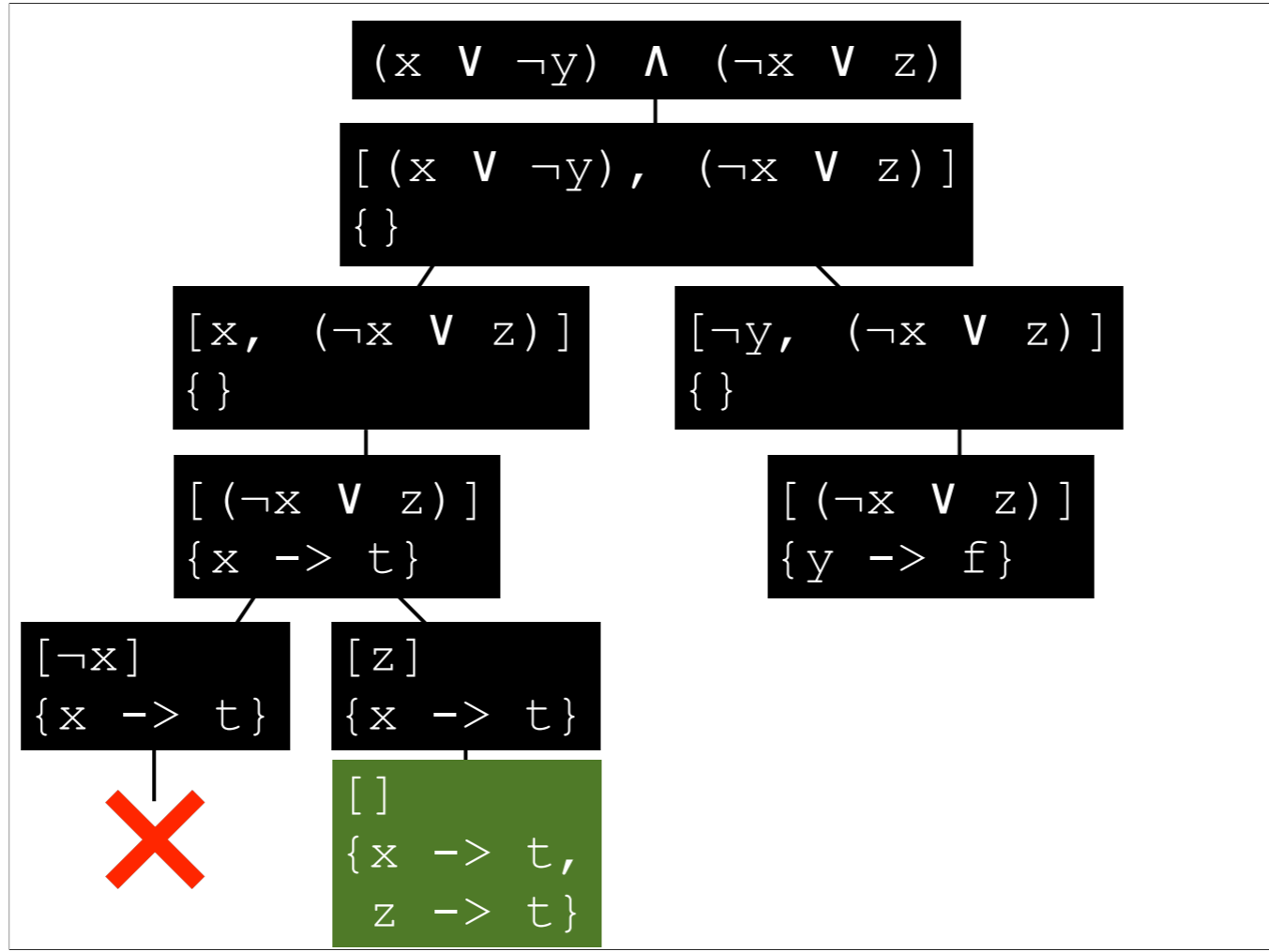
$[\neg x]$
 $\{x \rightarrow t\}$

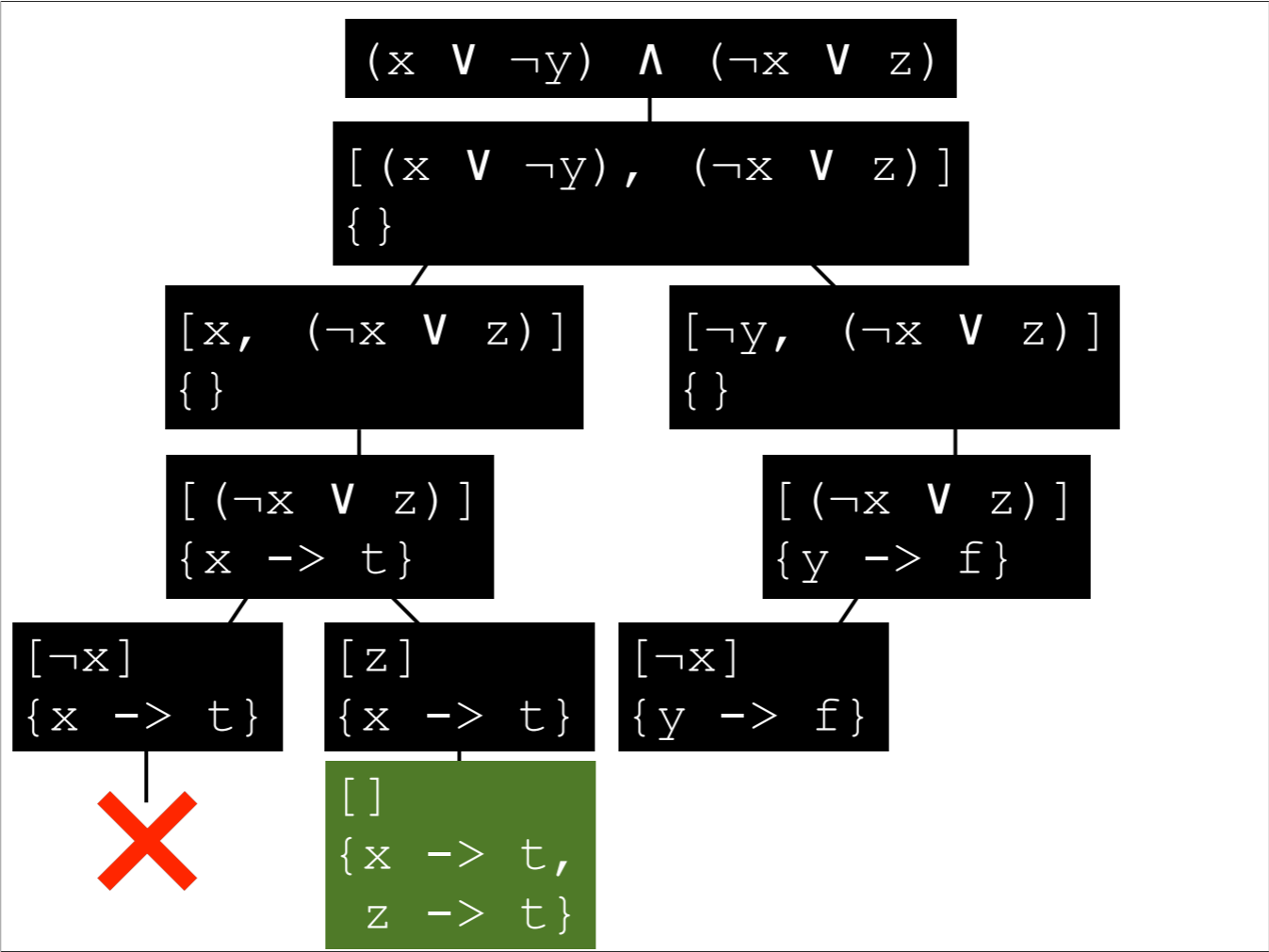


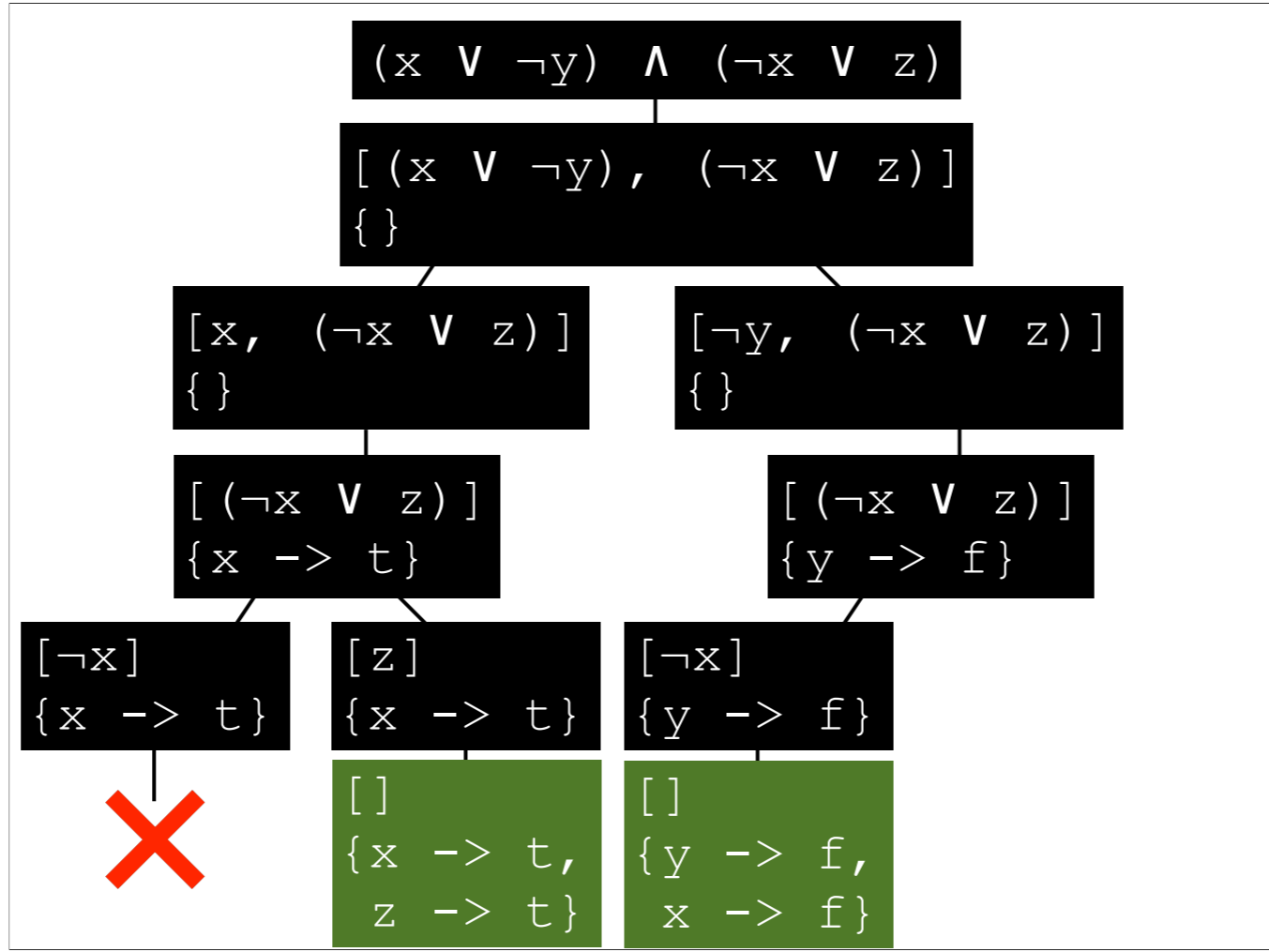


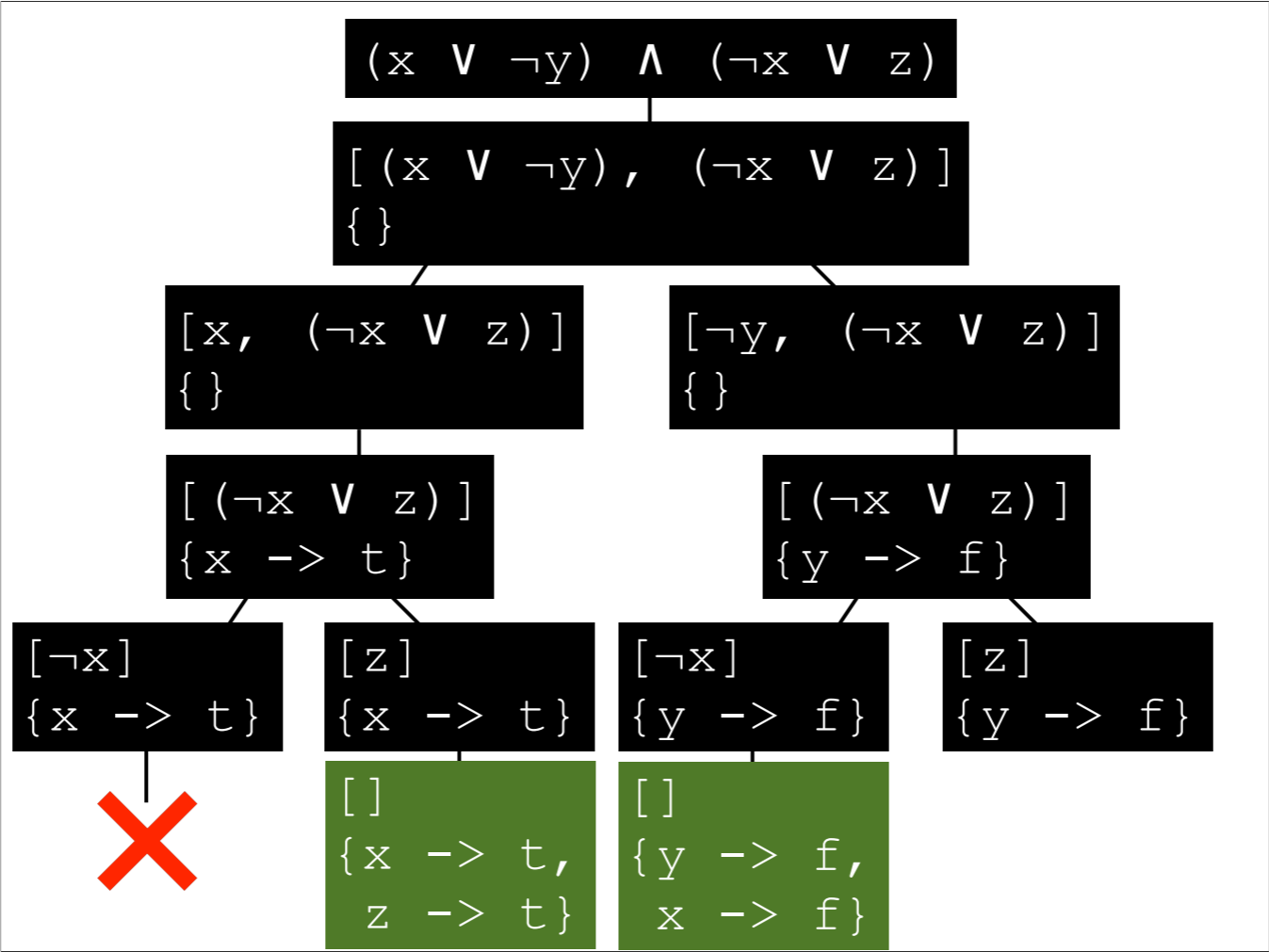


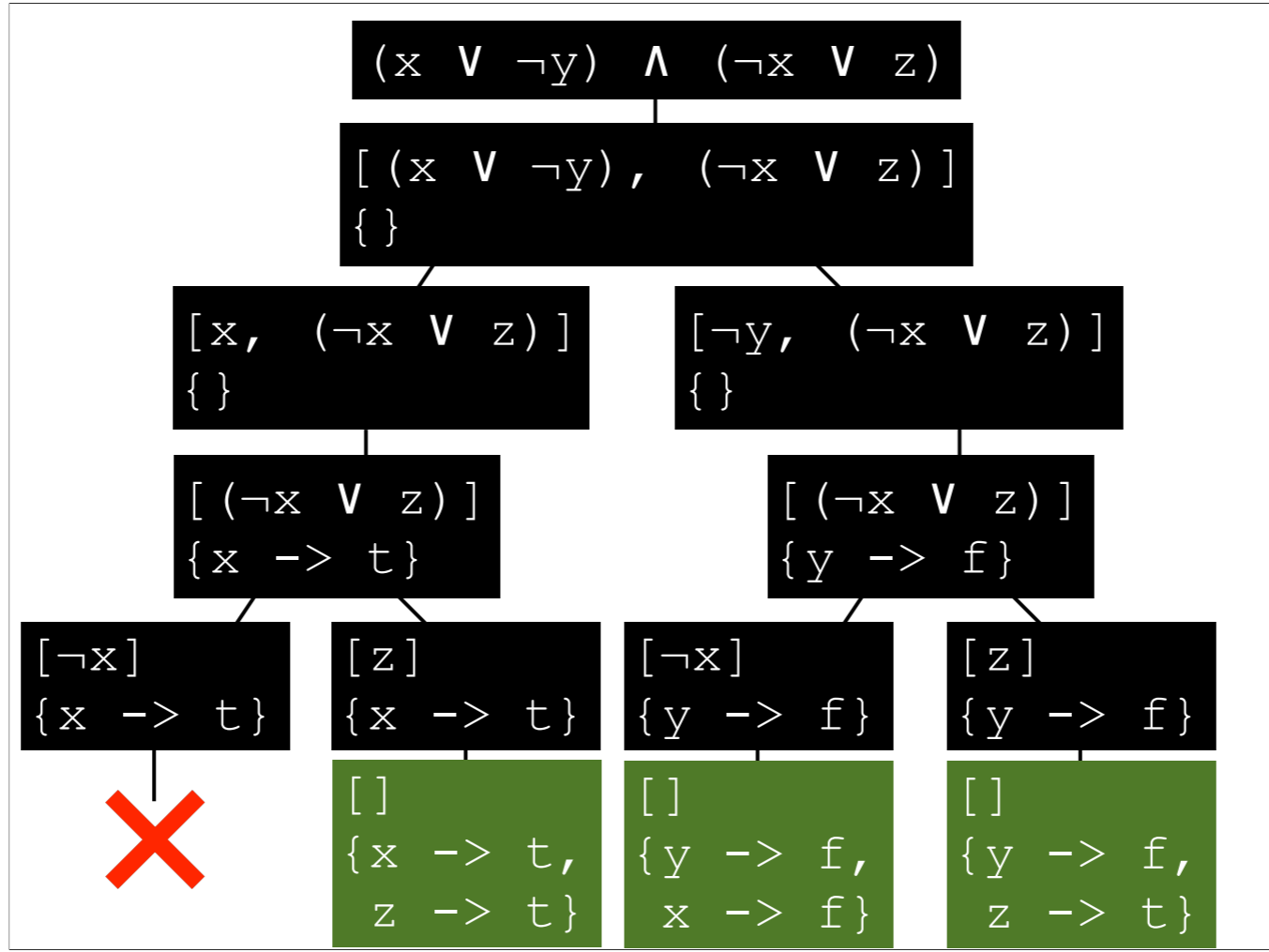












Exercise: Second Side of SAT Sheet