Warmup Problem

Translate the following sentence from English into Propositional Logic.

I want to eat ice cream even though I am on a diet.

If this is your first class, come get an index card from the front and write down your full name, preferred name and something interesting about you (or someone that shares the same name as you!)

CS 245: Logic and Computation

Carmen Bruni

Lecture 2

Based on slides by Jonathan Buss, Lila Kari, Anna Lubiw and Steve Wolfman with thanks to B. Bonakdarpour, A. Gao, D. Maftuleac, C. Roberts, R. Trefler, and P. Van Beek

Previously on CS 245

- The roadmaps of CS 245 (on the course website)
- What is logic?
- What are the applications of logic in computer science?
- What is a proposition?
- How do we translate English sentences into compound propositions?

Translating from English to Propositional Logic

Translate the following sentences to propositional logic formulas.

1. She is clever but not hard working.
2. I will eat an apple or an orange but not both.
3. If he does not study hard, then he will fail.
4. If it is sunny tomorrow, then I will play golf, provided that I do not feel stressed.
5. He will not fail only if he studies hard.

English can be ambiguous.

Give multiple translations of the following sentences into propositional logic. Are these translations logically equivalent?

1. Pigs can fly and the grass is red or the sky is blue.

2. He will fail unless he studies hard.

Learning goals — revisited

By the end of the lecture, you should be able to

- Give a high-level description of logic.
- Give examples of applications of logic in computer science.
- Define propositions.
- Classify English sentences into propositions and non-propositions.
- Give multiple translations of English sentences with ambiguity.
- Translate English sentences with no ambiguity into compound propositions.

Propositional Logic: Syntax

Learning goals

By the end of this lecture, you should be able to:

- Describe the three types of symbols in propositional logic.
- Describe the recursive definition of well-formed formulas.
- Write the parse tree for a well-formed formula.
- Determine and give reasons for whether a given formula is well formed or not.
- Identify the recursive structure in a recursive definition.
- Explain how to use structural induction to prove properties of a recursively defined concept.

Atomic and compound propositions

An *atomic* proposition (also called an atom or an atomic formula) is a statement or an assertion that must be true or false. It is represented by a single propositional variable. Such statements should be stated in the **positive**.

We construct a *compound* proposition by connecting atomic propositions using logical connectives.

Symbols and expressions

Propositions in English are represented by *formulas*.

A formula consists of a string of *symbols*.

There are three kinds of symbols.

Propositional variables: Lowercase Latin letters possibly with subscripts; e.g., p, q, r, p_1, p_2, q_{27}, etc.

Connectives: $\neg$, $\wedge$, $\vee$, $\rightarrow$ and $\leftrightarrow$.

Punctuation: '(' and ')'.

Expressions

An *expression* is a string of symbols.

Examples of expressions:

- α: $(\neg)()\vee pq\rightarrow$
- β: $a \vee b \wedge c$
- γ: $((a \rightarrow b) \vee c)$

Expressions

An *expression* is a string of symbols.

Examples of expressions:

- α: $(\neg)()\vee pq\rightarrow$
- β: $a \vee b \wedge c$
- γ: $((a \rightarrow b) \vee c)$

What does each expression mean? In how many ways can we interpret each expression?

Expressions

An *expression* is a string of symbols.

Examples of expressions:

- α: $(\neg)()\vee pq\rightarrow$
- β: $a \vee b \wedge c$
- γ: $((a \rightarrow b) \vee c)$

What does each expression mean? In how many ways can we interpret each expression?

Ideally, we would like *one and only one way* to interpret each expression.

Can we focus on a set of expressions where each expression in this set has a unique interpretation?

Definition of well-formed formulas

Let $\mathcal{P}$ be a set of propositional variables. We define the set of *well-formed formulas* over $\mathcal{P}$ inductively as follows.

1. A single symbol of $\mathcal{P}$ is well-formed.
2. If α is well-formed, then $(\neg\alpha)$ is well-formed.
3. If α and β are well-formed, then each of $(\alpha \wedge \beta), (\alpha \vee \beta), (\alpha \rightarrow \beta), (\alpha \leftrightarrow \beta)$ is well-formed.
4. Nothing else is a well-formed formula.

The parse tree of a well-formed formula

A parse tree is another way to represent a well-formed formula. The parse tree makes the structure of the formula explicit.

Write the parse tree of the following well-formed formulas:

1. $((a \vee b) \wedge (\neg(a \wedge b)))$
2. $(((\neg p) \wedge q) \rightarrow (p \wedge (q \vee (\neg r))))$.

Parse Tree Notes

1. The leaves are proposition symbols (these are valid trees).
2. All non-leaves are connectives.
3. Negation has one child (a tree)
4. Binary connectives have two children (each is another tree)
5. Can form the tree by starting with the inner most bracket, resolving the bracket then moving upward.

Unique Readability of Formulas

Does every well-formed formula have a unique meaning? Yes.

Theorem. Every well-formed formula has a unique derivation as a well-formed formula. That is, each well-formed formula has exactly one of the following forms:

(1) an atom, (2) $(\neg\alpha)$, (3) $(\alpha \wedge \beta)$, (4) $(\alpha \vee \beta)$, (5) $(\alpha \rightarrow \beta)$, or (6) $(\alpha \leftrightarrow \beta)$.

In each case, it is of that form in exactly one way.

(Why) Is the Theorem True?

As an example, consider $((p \wedge q) \rightarrow r)$. It can be formed from the two formulas $(p \wedge q)$ and r using the connective $\rightarrow$.

If we tried to form it using $\wedge$, the two parts would need to be "$(p$" and "$q) \rightarrow r$". But neither of those is a formula!

The statement holds for this example.

We will prove this theorem using structural induction (a type of mathematical induction).

Mathematical Induction

Let f_n be the sequence defined by $f_0 = 1$, $f_1 = 1$ and $f_n = f_{n-1} + f_{n-2}$ for all $n \geq 2$. Show that $f_n \leq 2^n$ for all $n \geq 0$.

Mathematical Induction

Let f_n be the sequence defined by $f_0 = 1$, $f_1 = 1$ and $f_n = f_{n-1} + f_{n-2}$ for all $n \geq 2$. Show that $f_n \leq 2^n$ for all $n \geq 0$.

Note: The Principle of Mathematical Induction is equivalent to the Principle of Strong Induction! There are lots of places online where you can find a proof of this fact.

Generalizing

How could we use induction to prove statements on well-formed formulæ?

Example

Theorem

Every well-formed formula has an equal number of opening and closing brackets.

Let's prove this by using induction on the height of a parse tree.

Proof by Induction

Let $P(n)$ be the statement that every well-formed formula with a parse tree of height n has an equal number of opening and closing brackets. We prove $P(n)$ is true for all integers $n \geq 1$.

Base Case (BC): When $n = 1$, $P(1)$ is true since a parse tree of height one must consist only of a single atom and so the associated well-formed formula has no brackets.

Induction Hypothesis (IH): Assume $P(i)$ is true **for all** $1 \leq i \leq k$ and **for some** positive integer k.

Induction Conclusion (IC): For $k + 1$, Notice that the root of a parse tree can only be one of $\neg$, $\wedge$, $\vee$, $\rightarrow$ or $\leftrightarrow$. We break this into two cases.

Case 1

For the unary connective $\neg$, notice that the tree without the root still forms a well-formed formula α and this is a parse tree of height $k-1$. Including the negative gives $(\neg\alpha)$. Let $\mathsf{op}(\varphi)$ be the number of open brackets in a formula φ and let $\mathsf{cl}(\varphi)$ be the number of closing brackets. The induction hypothesis states that

$$\mathsf{op}(\alpha) = \mathsf{cl}(\alpha)$$

and so, by the induction hypothesis at step 2, we have

$$\mathsf{op}(\ (\neg\alpha)\ ) = 1 + \mathsf{op}(\alpha) = 1 + \mathsf{cl}(\alpha) = \mathsf{cl}(\ (\neg\alpha)\ )$$

completing the claim.

Case 2

For any of the four binary connectives, say $\star$, notice that the root will have two children each of which forms a well-formed formula. Let α be the well-formed formula corresponding to one and β be the other. Notice that both of the parse trees for α and β have heights that are at least one less than the height of the original tree (which is $k+1$). Thus by the induction hypothesis.

$$\mathsf{op}(\alpha) = \mathsf{cl}(\alpha) \qquad \text{and} \qquad \mathsf{op}(\beta) = \mathsf{cl}(\beta)$$

Then by the induction hypothesis at step 2, we have

$$\mathsf{op}(\ (\alpha \star \beta)\ ) = 1 + \mathsf{op}(\alpha) + \mathsf{op}(\beta) = 1 + \mathsf{cl}(\alpha) + \mathsf{cl}(\beta) = \mathsf{cl}(\ (\alpha \star \beta)\ )$$

completing the claim

Proof Using Structural Induction

Proof.

We use structural induction. The property to prove is

$R(\alpha)$: α has an equal number of left and right parentheses

for every formula α.

Base case: α is an atom.

α has no parentheses—only a propositional variable. Thus $R(\alpha)$ holds.

This completes the proof of the base case.

Example, cont'd — the Inductive Step

Inductive step:

Hypothesis: Assume that for formulæ α and β, both of $R(\alpha)$ and $R(\beta)$ are true.

To prove: Each of the formulæ $(\neg\alpha)$, $(\alpha \wedge \beta)$, $(\alpha \vee \beta)$, $(\alpha \to \beta)$, and $(\alpha \leftrightarrow \beta)$ has property R.

WLOG, we consider $(\alpha \wedge \beta)$. (Note you should also [at least] do negation!) We calculate $op(\ (\alpha \wedge \beta)\ )$:

$$\begin{aligned} op(\ (\alpha \wedge \beta)\ ) &= 1 + op(\alpha) + op(\beta) && \text{inspection} \\ &= 1 + cl(\alpha) + cl(\beta) && R(\alpha) \text{ and } R(\beta) \text{ (IH)} \\ &= cl(\ (\alpha \wedge \beta)\ ) && \text{inspection .} \end{aligned}$$

Structural induction (1)

Step 1: Identify the recursive structure in the problem.

Theorem 1: Every *well-formed formula* has an equal number of opening and closing brackets.

Structural induction (1)

Step 1: Identify the recursive structure in the problem.

Theorem 1: Every *well-formed formula* has an equal number of opening and closing brackets. Notes:

- "Well-formed formulas" are the recursive structures that we are dealing with.
- "Has an equal number of opening and closing brackets" is the property that we are going to prove that well formed formulas have. Some examples of other properties that we could prove: (1) contains at least one propositional variable, (2) has an even number of brackets, etc.

Structural induction (2)

Step 2: Identify each recursive appearance of the structure inside its definition. (A recursive structure is self-referential. Where in the definition of the object does the object reference itself?

Let $\mathcal{P}$ be a set of propositional variables. A *well-formed formula* over $\mathcal{P}$ has exactly one of the following forms:

1. A single symbol of $\mathcal{P}$,
2. $(\neg\alpha)$ if α is well-formed,
3. One of $(\alpha \wedge \beta), (\alpha \vee \beta), (\alpha \rightarrow \beta), (\alpha \leftrightarrow \beta)$ if α and β are well-formed.

Structural induction (2)

Step 2: Identify each recursive appearance of the structure inside its definition. (A recursive structure is self-referential. Where in the definition of the object does it the object reference itself?

There are 3 cases in the following definition. The definition references itself in cases 2 and 3, as highlighted below.

Let $\mathcal{P}$ be a set of propositional variables. A *well-formed formula* over $\mathcal{P}$ has exactly one of the following forms:

1. A single symbol of $\mathcal{P}$,
2. $(\neg\alpha)$ if α *is well-formed*,
3. One of $(\alpha \wedge \beta), (\alpha \vee \beta), (\alpha \rightarrow \beta), (\alpha \leftrightarrow \beta)$ if α *and* β *are well-formed*.

Structural induction (3)

Step 3: Divide the cases into: those without recursive appearances (“ base cases”) and those with (“inductive” cases).

Case 1 has no recursive appearance of the definition. Cases 2 and 3 have recursive appearances of the definition.

Let $\mathcal{P}$ be a set of propositional variables. A *well-formed formula* over $\mathcal{P}$ has exactly one of the following forms:

1. A single symbol of $\mathcal{P}$, (base case)
2. $(\neg\alpha)$ if α *is well-formed*, (inductive case)
3. One of $(\alpha \wedge \beta), (\alpha \vee \beta), (\alpha \rightarrow \beta), (\alpha \leftrightarrow \beta)$ if α *and* β *are well-formed*. (inductive case)

A structural induction template

Problem: Prove that every recursive structure φ has property P.

Define $P(\varphi)$ to be the property in the problem.

Theorem: For every *recursive structure* φ, $P(\varphi)$ holds.

Proof by structural induction:

> Base case: *For every base case you identified, prove that the recursive structure φ has property P.*

(Continued on the next slide)

A structural induction template

Induction step: *For each recursive case you identified, write an induction step*

Recursive case 1:

Induction hypothesis: Assume *each recursive appearance of the structure in this case* has property P.
Prove that the recursive structure φ has property P using the induction hypothesis.

Recursive case 2:

(State the induction hypothesis and use it to prove the theorem.)

(Possibly more recursive cases)

By the principle of structural induction, every recursive structure φ has property P. QED