COSC 290 Discrete Structures

Lecture 17: Structural induction on logic

Prof. Michael Hay

Friday, Mar. 9, 2018

Colgate University

Plan for today

1. Practice with structural induction on trees
2. Structural induction on propositions

Practice with structural induction on trees

Recall: structural induction

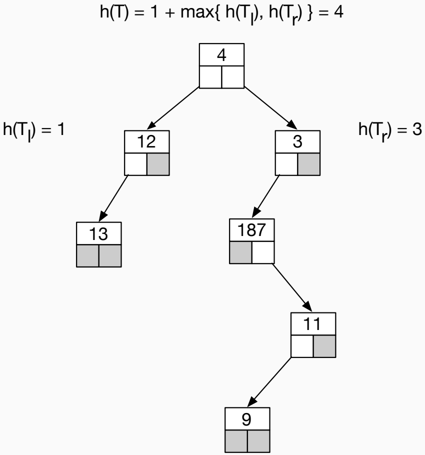
If you have a recursively defined structure – a structure defined in terms of one or more *base* cases and one or more *inductive* cases – you can prove properties about it using structural induction.

With structural induction, proof components should align with components of recursive definition.

Height of a tree, defined recursively

We can also define height *recursively*: let $h(T)$ denote the height of tree T.

- Base case: tree T is empty, $h(T) = -1$.
- Inductive case: T is non-empty, thus it consists a root node x, a left subtree T_ℓ, and a right subtree T_r. Then, $h(T) = 1 + \max\{\, h(T_\ell), h(T_r)\,\}$.



Recall: exercise from last class

Claim: $nodes(T) \leq 2^{h(T)+1} - 1$

For reference: let $h(T)$ denote the height of tree T.

- Base case: tree T is empty, $h(T) = -1$.
- Inductive case: T is non-empty, thus it consists a root node x, a left subtree T_ℓ, and a right subtree T_r. Then, $h(T) = 1 + \max\{\, h(T_\ell), h(T_r) \,\}$.

Proof of base case

Claim: $nodes(T) \leq 2^{h(T)+1} - 1$

Base case: T is empty so $nodes(T) = 0$ and $h(T) = -1$.

Indeed $nodes(T) \leq 2^{h(T)+1} - 1 = 2^{-1+1} - 1 = 0$.

Proof of inductive case

Claim: $nodes(T) \leq 2^{h(T)+1} - 1$

Inductive case: T is a non-empty tree of height $h(T)$, consisting of node x and left and right subtrees T_ℓ and T_r.

$$\begin{aligned}
nodes(T) &= 1 + nodes(T_\ell) + nodes(T_r) && \text{(b. +1 for root)} \\
&\leq 1 + (2^{h(T_\ell)+1} - 1) + (2^{h(T_r)+1} - 1) && \text{(c. ind. hypothesis)} \\
&\leq 1 + (2^{(h(T)-1)+1} - 1) + (2^{(h(T)-1)+1} - 1) && \text{(d. why?)} \\
&= 2^{h+1} - 1 = 2^{h(T)+1} - 1 && \text{(e. algebra)}
\end{aligned}$$

Explanation for line d:

Recall definition of height: $h(T) := 1 + \max\{\, h(T_\ell), h(T_r) \,\}$.

So $h(T_\ell) \leq h(T) - 1$ and $2^{h(T_\ell)+1} \leq 2^{(h(T)-1)+1}$.

Same idea for T_r.

Lower bound?

- **False Claim**: $nodes(T) \geq 2^{h(T)+1} - 1$

Lower bound?

- **False Claim**: $nodes(T) \geq 2^{h(T)+1} - 1$
- **Faulty proof by structural induction**:
  - **Base cases:** T is empty, height is -1 and $nodes(T) \geq 2^{-1+1} - 1 = 0$.

Lower bound?

- **False Claim**: $nodes(T) \geq 2^{h(T)+1} - 1$
- **Faulty proof by structural induction**:
  - **Base cases:** T is empty, height is -1 and $nodes(T) \geq 2^{-1+1} - 1 = 0$.
  - **Inductive case:** T is a non-empty tree of height $h(T)$, consisting of node x and left and right subtrees T_ℓ and T_r.

$$\begin{aligned} nodes(T) &= 1 + nodes(T_\ell) + nodes(T_r) && \text{(b. +1 for root)} \\ &\geq 1 + (2^{h(T_\ell)+1} - 1) + (2^{h(T_r)+1} - 1) && \text{(c. ind. hypothesis)} \\ &\geq 1 + (2^{(h(T)-1)+1} - 1) + (2^{(h(T)-1)+1} - 1) && \text{(d. subtree heights)} \\ &= 2^{h+1} - 1 = 2^{h(T)+1} - 1 && \text{(e. algebra)} \end{aligned}$$

Poll: Lower bound?

- **False Claim**: $nodes(T) \geq 2^{h(T)+1} - 1$
- **Faulty proof by structural induction**:
  - **Base cases:** T is empty, height is -1 and $nodes(T) \geq 2^{-1+1} - 1 = 0$.
  - **Inductive case:** T is a non-empty tree of height $h(T)$, consisting of node x and left and right subtrees T_ℓ and T_r.

$$\begin{aligned} nodes(T) &= 1 + nodes(T_\ell) + nodes(T_r) && \text{(b. +1 for root)} \\ &\geq 1 + (2^{h(T_\ell)+1} - 1) + (2^{h(T_r)+1} - 1) && \text{(c. ind. hypothesis)} \\ &\geq 1 + (2^{(h(T)-1)+1} - 1) + (2^{(h(T)-1)+1} - 1) && \text{(d. subtree heights)} \\ &= 2^{h+1} - 1 = 2^{h(T)+1} - 1 && \text{(e. algebra)} \end{aligned}$$

Where's the flaw? A) first sentence of inductive case; B) line b; C) line c; D) line d; E) line e.

Structural induction on propositions

Propositions, recursively defined

A proposition φ is a well-formed formula (wff) over the variables in the set $P \coloneqq \{ p_1, \ldots, p_n \}$, is one of the following:

- (base case) $\varphi \coloneqq p$ for some $p \in P$
- (inductive cases)
  - $\varphi \coloneqq \alpha \vee \beta$
  - $\varphi \coloneqq \alpha \wedge \beta$
  - $\varphi \coloneqq \alpha \implies \beta$
  - $\varphi \coloneqq \neg\alpha$

  where α and β are well-formed formulas.

Negation Normal Form

Definition (Negation Normal Form (NNF))

A proposition φ is in negation normal form if the negation connective is applied only to variables and not to more complex expressions, and furthermore, the only connectives allowed are in the set $\{\wedge, \vee, \neg\}$.

Exercise: Negation Normal Form

Given

$$\varphi := \neg(p \wedge (\neg q \vee r)) \vee s$$

let's write an equivalent proposition that is in NNF by "pushing negations down."

Hint: double negation and De Morgan's laws are useful.

$$\begin{aligned} \neg(\neg\alpha) &\equiv \alpha && \text{double-negation elimination} \\ \neg(\alpha \wedge \beta) &\equiv (\neg\alpha \vee \neg\beta) && \text{De Morgan's law \#1} \\ \neg(\alpha \vee \beta) &\equiv (\neg\alpha \wedge \neg\beta) && \text{De Morgan's law \#2} \end{aligned}$$

All propositions can be expressed in NNF

Claim: For any wff φ, there exists a proposition φ' that is in NNF and is logically equivalent to φ.

Restating claim

Claim: For any well-formed formula φ, there exists a proposition φ' that is in negation normal form and is logically equivalent to φ.

Notation:

- $\mathit{isNNF}(\varphi)$ denotes the predicate: φ is in NNF.
- $\mathit{hasNNF}(\varphi)$ denotes the predicate: there exists a proposition φ' that is in NNF and $\varphi' \equiv \varphi$.
- $\mathcal{W}$ denotes the set of all well-formed formulas.

Thus, our claim can be restated as $\forall \varphi \in \mathcal{W} : \mathit{hasNNF}(\varphi)$.

Proof

Claim A: $\forall \varphi \in \mathcal{W} : \mathit{hasNNF}(\varphi)$.

We will instead prove the *stronger* claim:

Claim B: $\forall \varphi \in \mathcal{W} : \mathit{hasNNF}(\varphi) \wedge \mathit{hasNNF}(\neg\varphi)$.

How is this “stronger?” How does this help?

(See also book discussion on p. 540.)

Poll: what is the base case?

Claim B: $\forall \varphi \in \mathcal{W} : \mathit{hasNNF}(\varphi) \wedge \mathit{hasNNF}(\neg\varphi)$.

We will do a proof by structural induction.

How should we structure the base case(s)?

A) Two base cases: $\varphi := p$ and $\varphi := \neg p$. In each, want to show $\mathit{hasNNF}(\varphi)$.

B) One base cases: $\varphi := p$, want to show: $\mathit{hasNNF}(p) \wedge \mathit{hasNNF}(\neg p)$

C) Either of above is acceptable.

D) Structural induction proofs do not have base cases

Inductive cases

Claim B: $\forall \varphi \in \mathcal{W} : \mathit{hasNNF}(\varphi) \wedge \mathit{hasNNF}(\neg\varphi)$.

We will do a proof by structural induction. How many inductive cases?

Inductive cases

Claim B: $\forall \varphi \in \mathcal{W} : \mathit{hasNNF}(\varphi) \wedge \mathit{hasNNF}(\neg\varphi)$.

We will do a proof by structural induction. How many inductive cases? One case for each case in the recursive definition of WFF:

1. AND: $\varphi \coloneqq \alpha \wedge \beta$
2. OR: $\varphi \coloneqq \alpha \vee \beta$
3. NOT: $\varphi \coloneqq \neg\alpha$
4. IMPLIES: $\varphi \coloneqq \alpha \implies \beta$.

Poll: Inductive case 1, what to show?

Claim B: $\forall \varphi \in \mathcal{W}$: *hasNNF*$(\varphi) \land$ *hasNNF*$(\neg\varphi)$.

Proof continued...

Inductive cases: We focus on case 1: $\varphi := \alpha \land \beta$. What do we want to show?

A) *hasNNF*(α)

B) *hasNNF*$(\alpha \land \beta)$

C) *hasNNF*$(\neg(\alpha \land \beta))$

D) *hasNNF*$(\neg\alpha \lor \neg\beta)$

E) More than one / None of the above

Poll: Inductive case 1, what can we assume?

Claim B: $\forall \varphi \in \mathcal{W}$: *hasNNF*(φ) $\land$ *hasNNF*($\neg\varphi$).

Proof continued...

Inductive cases: We focus on case 1: $\varphi := \alpha \land \beta$.

Want to show: *hasNNF*($\alpha \land \beta$) $\land$ *hasNNF*($\neg(\alpha \land \beta)$). Which of the following can we assume is true (by the inductive hypothesis)?

A) *hasNNF*(α) ... recall this means that α is logically equivalent to some NNF proposition.

B) *hasNNF*($\neg\alpha$)

C) *isNNF*(α) ... recall this means that α is an NNF.

D) A and B

E) A, B, and C

Proof for inductive case 1

Claim B: $\forall \varphi \in \mathcal{W}$: *hasNNF*$(\varphi) \wedge$ *hasNNF*$(\neg\varphi)$.

Inductive cases: We focus on case 1: $\varphi := \alpha \wedge \beta$.

Want to show: *hasNNF*$(\alpha \wedge \beta) \wedge$ *hasNNF*$(\neg(\alpha \wedge \beta))$.

Assume by inductive hypothesis:

- *hasNNF*(α), *hasNNF*(β), *hasNNF*$(\neg\alpha)$, *hasNNF*$(\neg\beta)$

Part 1: Since *hasNNF*(α), there exists α' such that $\alpha' \equiv \alpha$ and *isNNF*(α'). Similarly for β. Let $\varphi' := \alpha' \wedge \beta'$. We have *isNNF*$(\varphi')$ and $\varphi' \equiv \alpha \wedge \beta$. Thus *hasNNF*$(\alpha \wedge \beta)$.

Proof for inductive case 1

Claim B: $\forall \varphi \in \mathcal{W} :$ *hasNNF*$(\varphi) \wedge$ *hasNNF*$(\neg\varphi)$.

Inductive cases: We focus on case 1: $\varphi := \alpha \wedge \beta$.

Want to show: *hasNNF*$(\alpha \wedge \beta) \wedge$ *hasNNF*$(\neg(\alpha \wedge \beta))$.

Assume by inductive hypothesis:

- *hasNNF*(α), *hasNNF*(β), *hasNNF*$(\neg\alpha)$, *hasNNF*$(\neg\beta)$

Part 1: Since *hasNNF*(α), there exists α' such that $\alpha' \equiv \alpha$ and *isNNF*(α'). Similarly for β. Let $\varphi' := \alpha' \wedge \beta'$. We have *isNNF*$(\varphi')$ and $\varphi' \equiv \alpha \wedge \beta$. Thus *hasNNF*$(\alpha \wedge \beta)$.

Part 2: $\neg\varphi = \neg(\alpha \wedge \beta) \equiv \neg\alpha \vee \neg\beta$ by DeMorgan's law. Since *hasNNF*$(\neg\alpha)$, there exists $\bar{\alpha}$ such that $\bar{\alpha} \equiv \neg\alpha$ and *isNNF*$(\bar{\alpha})$. Similarly for β. Thus, let $\bar{\varphi} := \bar{\alpha} \vee \bar{\beta}$. We have *isNNF*$(\bar{\varphi})$ and $\bar{\varphi} \equiv \neg(\alpha \wedge \beta)$. Thus *hasNNF*$(\neg(\alpha \wedge \beta))$.

Proof for inductive case 2

Claim B: $\forall \varphi \in \mathcal{W} : \mathit{hasNNF}(\varphi) \land \mathit{hasNNF}(\neg\varphi)$.

Inductive cases: Case 2: $\varphi := \alpha \lor \beta$.

Proof is identical to case 1, just replace ANDs with ORs and vice versa.

Poll: Inductive case 3, what to show?

Claim B: $\forall \varphi \in \mathcal{W} : \mathit{hasNNF}(\varphi) \wedge \mathit{hasNNF}(\neg\varphi)$.

Inductive cases: Case 3: $\varphi := \neg\alpha$.

What do we want to show?

A) $\mathit{hasNNF}(\alpha)$

B) $\mathit{hasNNF}(\neg\alpha)$

C) $\mathit{hasNNF}(\neg\neg\alpha)$

D) B and C

E) A, B, and C

Proof for inductive case 3

Claim B: $\forall \varphi \in \mathcal{W}$: *hasNNF*$(\varphi) \wedge$ *hasNNF*$(\neg\varphi)$.

Inductive cases: Case 3: $\varphi \coloneqq \neg\alpha$.

Want to show: *hasNNF*$(\neg\alpha) \wedge$ *hasNNF*$(\neg\neg\alpha)$.

Assume by inductive hypothesis:

- *hasNNF*(α), *hasNNF*$(\neg\alpha)$

Still need to show: *hasNNF*$(\neg\neg\alpha)$.

Since $\neg\neg\alpha \equiv \alpha$ and *hasNNF*(α), then let α' be such that $\alpha' \equiv \alpha$ and *isNNF*(α'). Let $\bar{\varphi} \coloneqq \alpha'$. Since $\bar{\varphi} \equiv \neg\neg\alpha$ and *isNNF*$(\bar{\varphi})$, thus *hasNNF*$(\neg\neg\alpha)$.

Proof for inductive case 4

Claim B: $\forall \varphi \in \mathcal{W} : \mathit{hasNNF}(\varphi) \wedge \mathit{hasNNF}(\neg\varphi)$.

Inductive cases: Case 2: $\varphi := \alpha \implies \beta$.

The cleanest way to handle this is to have a lemma that states for any φ there is an equivalent φ' that contains only the connectives $\{\neg, \vee, \wedge\}$. This lemma can be proven using structural induction.