Lecture 7

The Compactness Theorem

Introduction to Logic for Computer Science

Prof Hongseok Yang
KAIST

These slides are minor variants of those made by Prof Worrell and Dr Haase for their logic course at Oxford.

Overview

- So far we studied propositional logic.
- Soon we will look at predicate logic.
- Later: reduce reasoning about **predicate formulas** to reasoning about **infinite sets of propositional formulas**.
- Today: reduce reasoning about **infinite sets of propositional formulas** to reasoning about **finite sets of prop. formulas**.

Partial assignments

A **partial assignment** is a function $\mathcal{A}\colon D \to \{0, 1\}$, whose **domain** $D \subseteq \{p_1, p_2, \ldots\}$ is a set of variables, denoted by $\mathrm{dom}(\mathcal{A})$.

A partial assignment $\mathcal{A}'$ **extends** another one $\mathcal{A}$ when $\mathrm{dom}(\mathcal{A}) \subseteq \mathrm{dom}(\mathcal{A}')$ and $\mathcal{A}(p_i) = \mathcal{A}'(p_i)$ for all $p_i \in \mathrm{dom}(\mathcal{A})$.

Satisfiability of sets

A *set* $\mathcal{S}$ of formulas is **satisfiable** when there is an assignment that makes every $F \in \mathcal{S}$ true.

Ex: Find a satisfying assignment $\mathcal{A}$ of the following $\mathcal{S}$:

$$\mathcal{S} = \{p_1 \vee p_2,\ \neg p_2 \vee \neg p_3,\ p_3 \vee p_4,\ \neg p_4 \vee \neg p_5,\ \ldots\}.$$

Satisfiability of sets

A *set* $\mathcal{S}$ of formulas is **satisfiable** when there is an assignment that makes every $F \in \mathcal{S}$ true.

Ex: Find a satisfying assignment $\mathcal{A}$ of the following $\mathcal{S}$:

$$\mathcal{S} = \{p_1 \vee p_2,\ \neg p_2 \vee \neg p_3,\ p_3 \vee p_4,\ \neg p_4 \vee \neg p_5,\ \ldots\}.$$

One answer:

$$\mathcal{A}(p_i) = \begin{cases} 1 & \text{if } i \text{ is odd,} \\ 0 & \text{if } i \text{ is even.} \end{cases}$$

The Compactness Theorem

Theorem (Compactness Theorem)

A set of formulas $\mathcal{S}$ is satisfiable if and only if each finite subset of $\mathcal{S}$ is satisfiable.

The Compactness Theorem

Theorem (Compactness Theorem)

A set of formulas $\mathcal{S}$ is satisfiable if and only if each finite subset of $\mathcal{S}$ is satisfiable.

Ex1: Which is an obvious direction? If ($\Leftarrow$) or only-if ($\Rightarrow$)?

The Compactness Theorem

Theorem (Compactness Theorem)

A set of formulas $\mathcal{S}$ is satisfiable if and only if each finite subset of $\mathcal{S}$ is satisfiable.

Ex1: Which is an obvious direction? If ($\Leftarrow$) or only-if ($\Rightarrow$)?

Ex2: Using this theorem, develop a semi-algorithm for checking the unsatisfiability of a given countably-infinite set of formulas $\mathcal{S}$.

Compactness Theorem: contrapositive

Compact Theorem, contrapositive: if a set of formulas is unsatisfiable, then some finite subset is already unsatisfiable.

Procedure to show that an infinite set of formulas is unsatisfiable:

1. Enumerate $\mathcal{S} = \{F_1, F_2, \ldots\}$ by some algorithm.
2. For each n, test whether $\{F_1, \ldots, F_n\}$ is unsatisfiable.
3. If $\mathcal{S}$ is unsatisfiable, we will detect this after finite amount of time.

The Compactness Theorem

Theorem (Compactness Theorem)

A set of formulas $\mathcal{S}$ is satisfiable if and only if each finite subset of $\mathcal{S}$ is satisfiable.

We will now prove the non-obvious if direction.

Compactness Theorem: proof

Let $\mathcal{S}$ be a set of formulas with every finite subset satisfiable.

Compactness Theorem: proof

Let $\mathcal{S}$ be a set of formulas with every finite subset satisfiable.

- Call a partial assignment $\mathcal{A}$ **good** if it satisfies any $F \in \mathcal{S}$ whose variables are in the domain of $\mathcal{A}$.

Compactness Theorem: proof

Let $\mathcal{S}$ be a set of formulas with every finite subset satisfiable.

- Call a partial assignment $\mathcal{A}$ **good** if it satisfies any $F \in \mathcal{S}$ whose variables are in the domain of $\mathcal{A}$.
- Step 1: construct $\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2, \ldots$ of **good** partial assignments such that $\mathrm{dom}(\mathcal{A}_n) = \{p_1, \ldots, p_n\}$ and each $\mathcal{A}_{n+1}$ **extends** $\mathcal{A}_n$.
- Step 2: define $\mathcal{A}$ by $\mathcal{A}(p_n) = \mathcal{A}_n(p_n)$ for every p_n.

Compactness Theorem: proof

Let $\mathcal{S}$ be a set of formulas with every finite subset satisfiable.

- Call a partial assignment $\mathcal{A}$ **good** if it satisfies any $F \in \mathcal{S}$ whose variables are in the domain of $\mathcal{A}$.
- Step 1: construct $\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2, \ldots$ of **good** partial assignments such that $\mathrm{dom}(\mathcal{A}_n) = \{p_1, \ldots, p_n\}$ and each $\mathcal{A}_{n+1}$ **extends** $\mathcal{A}_n$.
- Step 2: define $\mathcal{A}$ by $\mathcal{A}(p_n) = \mathcal{A}_n(p_n)$ for every p_n.

Ex: Explain why $\mathcal{A}$ satisfies $\mathcal{S}$.

Compactness Theorem: proof

Let $\mathcal{S}$ be a set of formulas with every finite subset satisfiable.

- Call a partial assignment $\mathcal{A}$ **good** if it satisfies any $F \in \mathcal{S}$ whose variables are in the domain of $\mathcal{A}$.
- Step 1: construct $\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2, \ldots$ of **good** partial assignments such that $\mathrm{dom}(\mathcal{A}_n) = \{p_1, \ldots, p_n\}$ and each $\mathcal{A}_{n+1}$ **extends** $\mathcal{A}_n$.
- Step 2: define $\mathcal{A}$ by $\mathcal{A}(p_n) = \mathcal{A}_n(p_n)$ for every p_n.

Ex: Explain why $\mathcal{A}$ satisfies $\mathcal{S}$.

Answer: for every formula F in $\mathcal{S}$,

- if F uses variables $\{p_1, \ldots, p_n\}$, then $\mathcal{A}_n \models F$ and so $\mathcal{A} \models F$.

Compactness Theorem: proof

Let $\mathcal{S}$ be a set of formulas with every finite subset satisfiable.

- Call a partial assignment $\mathcal{A}$ **good** if it satisfies any $F \in \mathcal{S}$ whose variables are in the domain of $\mathcal{A}$.
- Step 1: construct $\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2, \ldots$ of **good** partial assignments such that $\mathrm{dom}(\mathcal{A}_n) = \{p_1, \ldots, p_n\}$ and each $\mathcal{A}_{n+1}$ **extends** $\mathcal{A}_n$.
- Step 2: define $\mathcal{A}$ by $\mathcal{A}(p_n) = \mathcal{A}_n(p_n)$ for every p_n.

Ex: Explain why $\mathcal{A}$ satisfies $\mathcal{S}$.

Answer: for every formula F in $\mathcal{S}$,

- if F uses variables $\{p_1, \ldots, p_n\}$, then $\mathcal{A}_n \models F$ and so $\mathcal{A} \models F$.

Compactness Theorem: proof

Step 1: construct $\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2, \ldots$ of **good** partial assignments such that $\mathrm{dom}(\mathcal{A}_n) = \{p_1, \ldots, p_n\}$ and each $\mathcal{A}_{n+1}$ **extends** $\mathcal{A}_n$.

How? In particular, how to ensure the extension part?

Compactness Theorem: proof

Step 1: construct $\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2, \ldots$ of **good** partial assignments such that $\mathrm{dom}(\mathcal{A}_n) = \{p_1, \ldots, p_n\}$ and each $\mathcal{A}_{n+1}$ **extends** $\mathcal{A}_n$.

How? In particular, how to ensure the extension part?

- Construct inductively.
- Maintain the invariant:

  There are infinitely many good extensions of $\mathcal{A}_n$ whose domains are $\{p_1, \ldots, p_m\}$ for some $m \geq n$.

Compactness Theorem: proof

Step 1: construct $\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2, \ldots$ of **good** partial assignments such that $\mathrm{dom}(\mathcal{A}_n) = \{p_1, \ldots, p_n\}$ and each $\mathcal{A}_{n+1}$ **extends** $\mathcal{A}_n$.

How? In particular, how to ensure the extension part?

- Construct inductively.
- Maintain the invariant:

  There are infinitely many good extensions of $\mathcal{A}_n$ whose domains are $\{p_1, \ldots, p_m\}$ for some $m \geq n$.

Base case: $\mathrm{dom}(\mathcal{A}_0) = \emptyset$.

Compactness Theorem: proof

Step 1: construct $\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2, \ldots$ of **good** partial assignments such that $\mathrm{dom}(\mathcal{A}_n) = \{p_1, \ldots, p_n\}$ and each $\mathcal{A}_{n+1}$ **extends** $\mathcal{A}_n$.

How? In particular, how to ensure the extension part?

- Construct inductively.
- Maintain the invariant:

  There are infinitely many good extensions of $\mathcal{A}_n$ whose domains are $\{p_1, \ldots, p_m\}$ for some $m \geq n$.

Base case: $\mathrm{dom}(\mathcal{A}_0) = \emptyset$.

Ex: Why does the invariant hold? Hint: Use the assumption.

Compactness Theorem: proof

Step 1: construct $\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2, \ldots$ of **good** partial assignments such that $\mathrm{dom}(\mathcal{A}_n) = \{p_1, \ldots, p_n\}$ and each $\mathcal{A}_{n+1}$ **extends** $\mathcal{A}_n$.

How? In particular, how to ensure the extension part?

- Construct inductively.
- Maintain the invariant:

  There are infinitely many good extensions of $\mathcal{A}_n$ whose domains are $\{p_1, \ldots, p_m\}$ for some $m \geq n$.

Base case: $\mathrm{dom}(\mathcal{A}_0) = \emptyset$.

Ex: Why does the invariant hold? Hint: Use the assumption.

There is a good partial assignment on $\{p_1, \ldots, p_n\}$ for any n, because **up to equivalence**, $\{F \in \mathcal{S} \mid F \text{ uses only } p_1, \ldots, p_n\}$ is finite.

Compactness Theorem: proof

Induction step: suppose $\mathcal{A}_0, \ldots, \mathcal{A}_n$ satisfy the invariant.

Compactness Theorem: proof

Induction step: suppose $\mathcal{A}_0, \ldots, \mathcal{A}_n$ satisfy the invariant.

- Consider assignments extending $\mathcal{A}_n$:

$$\mathcal{B}_0 = (\mathcal{A}_n)_{[p_{n+1} \mapsto 0]}, \qquad \mathcal{B}_1 = (\mathcal{A}_n)_{[p_{n+1} \mapsto 1]}.$$

  Any proper extension of $\mathcal{A}_n$ with domain $\supseteq \{p_1, \ldots, p_{n+1}\}$ extends $\mathcal{B}_0$ or $\mathcal{B}_1$.

Compactness Theorem: proof

Induction step: suppose $\mathcal{A}_0, \ldots, \mathcal{A}_n$ satisfy the invariant.

- Consider assignments extending $\mathcal{A}_n$:

  $$\mathcal{B}_0 = (\mathcal{A}_n)_{[p_{n+1} \mapsto 0]}, \qquad \mathcal{B}_1 = (\mathcal{A}_n)_{[p_{n+1} \mapsto 1]}.$$

  Any proper extension of $\mathcal{A}_n$ with domain $\supseteq \{p_1, \ldots, p_{n+1}\}$ extends $\mathcal{B}_0$ or $\mathcal{B}_1$.

- So one of $\mathcal{B}_0$ or $\mathcal{B}_1$ has infinitely many good extensions. Take that one to be $\mathcal{A}_{n+1}$.

Compactness Theorem: proof

Induction step: suppose $\mathcal{A}_0, \ldots, \mathcal{A}_n$ satisfy the invariant.

- Consider assignments extending $\mathcal{A}_n$:

  $$\mathcal{B}_0 = (\mathcal{A}_n)_{[p_{n+1} \mapsto 0]}, \qquad \mathcal{B}_1 = (\mathcal{A}_n)_{[p_{n+1} \mapsto 1]}.$$

  Any proper extension of $\mathcal{A}_n$ with domain $\supseteq \{p_1, \ldots, p_{n+1}\}$ extends $\mathcal{B}_0$ or $\mathcal{B}_1$.

- So one of $\mathcal{B}_0$ or $\mathcal{B}_1$ has infinitely many good extensions. Take that one to be $\mathcal{A}_{n+1}$.

- Ex: Show that $\mathcal{A}_{n+1}$ is good.

Compactness Theorem: comments

Has a **nonconstructive** proof, which does not give an algorithm to build a satisfying assignment. It merely guarantees that one exists.

Compactness Theorem: comments

Has a **nonconstructive** proof, which does not give an algorithm to build a satisfying assignment. It merely guarantees that one exists.

Contrast it with the proofs of the following statements:

- Satisfiability is polytime decidable for every Horn formula.
- A 2-CNF formula is satisfiable iff its implication graph is consistent.
- Every formula has an equisatisfiable 3-CNF formula.
- SAT is decidable (DP and DPLL).

Compactness Theorem: contrapositive

Compact Theorem, contrapositive: if a set of formulas is unsatisfiable, then some finite subset is already unsatisfiable.

Procedure to show that an infinite set of formulas is unsatisfiable:

1. Enumerate $\mathcal{S} = \{F_1, F_2, \ldots\}$ by some algorithm.
2. For each n, test whether $\{F_1, \ldots, F_n\}$ is unsatisfiable.
3. If $\mathcal{S}$ is unsatisfiable, we will detect this after finite amount of time.

The theorem ensures one-side correctness of this procedure.

Compactness: application

[**Exam question by Prof Worrell**] Suppose $\{F_n \mid n \in \mathbb{N}\}$ is an infinite set of formulas such that $\{\neg F_n \mid n \in \mathbb{N}\}$ is unsatisfiable and $F_n \rightarrow F_{n+1}$ is valid for all $n \in \mathbb{N}$. Show that some F_n is valid.

Compactness: application

[**Exam question by Prof Worrell**] Suppose $\{F_n \mid n \in \mathbb{N}\}$ is an infinite set of formulas such that $\{\neg F_n \mid n \in \mathbb{N}\}$ is unsatisfiable and $F_n \to F_{n+1}$ is valid for all $n \in \mathbb{N}$. Show that some F_n is valid.

Ex: Solve it using the Compactness Theorem.

Compactness: application

[Exam question by Prof Worrell] Suppose $\{F_n \mid n \in \mathbb{N}\}$ is an infinite set of formulas such that $\{\neg F_n \mid n \in \mathbb{N}\}$ is unsatisfiable and $F_n \to F_{n+1}$ is valid for all $n \in \mathbb{N}$. Show that some F_n is valid.

Ex: Solve it using the Compactness Theorem.

1. **Compactness**: n with $\neg F_1 \wedge \neg F_2 \wedge \ldots \wedge \neg F_n$ unsatisfiable.
2. **De Morgan**: $F_1 \vee F_2 \vee \ldots \vee F_n$ is valid.
3. **Resolve** $F_1 \vee F_2 \vee \ldots \vee F_n$ and $F_1 \to F_2$, and get $F_2 \vee \ldots \vee F_n$. Thus, $F_2 \vee \ldots \vee F_n$ is valid.
4. **Induction**: F_n is valid.

Graph colouring

A graph $G = (V, E)$ is k**-colourable** if we can colour each vertex with $\{1, \ldots, k\}$ such that neighbours get different colours.

Graph colouring

A graph $G = (V, E)$ is k**-colourable** if we can colour each vertex with $\{1, \ldots, k\}$ such that neighbours get different colours.

Theorem

If every finite subgraph of $\mathcal{G}$ is k-colourable, so is $\mathcal{G}$ itself.

We can prove it using the Compactness Theorem.

Graph colouring: proof

Suppose all finite subgraphs of $\mathcal{G}$ are k-colourable.

Graph colouring: proof

Suppose all finite subgraphs of $\mathcal{G}$ are k-colourable.

Variable $p_{v,i}$: "vertex v has colour i."

Graph colouring: proof

Suppose all finite subgraphs of $\mathcal{G}$ are k-colourable.

Variable $p_{v,i}$: "vertex v has colour i."

Constraints $\mathcal{S} := \{F_v, G_v \mid v \in V\} \cup \{H_{u,v} \mid (u,v) \in E\}$:

- Vertex v has ≥ 1 colour: $F_v := \bigvee_{1 \leq i \leq k} p_{v,i}$.
- Vertex v has ≤ 1 colour: $G_v := \bigwedge_{1 \leq i < j \leq k} (\neg p_{v,i} \vee \neg p_{v,j})$.
- Neighbours u, v different colour: $H_{u,v} := \bigwedge_{1 \leq i \leq k} (\neg p_{u,i} \vee \neg p_{v,i})$.

Graph colouring: proof

Suppose all finite subgraphs of $\mathcal{G}$ are k-colourable.

Variable $p_{v,i}$: "vertex v has colour i."

Constraints $\mathcal{S} := \{F_v, G_v \mid v \in V\} \cup \{H_{u,v} \mid (u, v) \in E\}$:

- Vertex v has ≥ 1 colour: $F_v := \bigvee_{1 \leq i \leq k} p_{v,i}$.
- Vertex v has ≤ 1 colour: $G_v := \bigwedge_{1 \leq i < j \leq k} (\neg p_{v,i} \vee \neg p_{v,j})$.
- Neighbours u, v different colour: $H_{u,v} := \bigwedge_{1 \leq i \leq k} (\neg p_{u,i} \vee \neg p_{v,i})$.

$\mathcal{S}$ is satisfiable iff $\mathcal{G}$ is k-colourable.

Graph colouring: proof

Suppose all finite subgraphs of $\mathcal{G}$ are k-colourable.

Variable $p_{v,i}$: "vertex v has colour i."

Constraints $\mathcal{S} := \{F_v, G_v \mid v \in V\} \cup \{H_{u,v} \mid (u,v) \in E\}$:

- Vertex v has ≥ 1 colour: $F_v := \bigvee_{1 \leq i \leq k} p_{v,i}$.
- Vertex v has ≤ 1 colour: $G_v := \bigwedge_{1 \leq i < j \leq k} (\neg p_{v,i} \vee \neg p_{v,j})$.
- Neighbours u, v different colour: $H_{u,v} := \bigwedge_{1 \leq i \leq k} (\neg p_{u,i} \vee \neg p_{v,i})$.

$\mathcal{S}$ is satisfiable iff $\mathcal{G}$ is k-colourable.

Ex: Complete the proof using the Compactness Theorem.

Compactness Theorem and topology

The Compactness Theorem is equivalent to the compactness of $\{0,1\}^{\mathbb{N}}$ under the product topology, where $\{0,1\}$ is given the discrete topology.

Compactness Theorem and topology

The Compactness Theorem is equivalent to the compactness of $\{0, 1\}^{\mathbb{N}}$ under the product topology, where $\{0, 1\}$ is given the discrete topology.

Ex1: Prove that the compactness of $\{0, 1\}^{\mathbb{N}}$ implies the Compactness Theorem for propositional logic.

Ex2: Prove the other implication.

Ex3: Do you know the name of the theorem in topology that gives the compactness of $\{0, 1\}^{\mathbb{N}}$?

Summary: propositional logic

Syntax. DNF, CNF, Horn formulas.

Semantics. Assignments and truth tables.

Validity, satisfiability, and constraint problems.

Equational reasoning with Boolean algebra and substitution.

Polynomial-time algorithms for Horn, 2-CNF, X-CNF. WalkSAT.

Resolution and DPLL algorithm.

Compactness Theorem.