Lecture 05: Rules of Inference

Jingjin Yu | Computer Science @ Rutgers

RUTGERS
THE STATE UNIVERSITY OF NEW JERSEY

September 18, 2020

Algorithmic Robotics & Control Lab

Outline

- A brief review of lecture 04
- A word about proofs
- Valid arguments in propositional logic
- Rules of inference
- Building argument

- A repeating note: **make sure you read the textbook**

L04: Nested Quantifiers

Quantification	When true	When false
$\forall x \forall y\ P(x, y)$	$P(x, y) = true$ for all possible combinations of x and y	$P(x, y)$ is false for one or more pairs of x, y
$\forall x \exists y\ P(x, y)$	For every x, there is some y where $P(x, y)$ is true	For some $x = a$, $P(a, y)$ is never true for any y
$\exists x \forall y\ P(x, y)$	For some $x = a$, $P(a, y)$ is always true	For every $x = a$, there is some y such that $P(a, y)$ is false
$\exists x \exists y\ P(x, y)$	$P(x, y)$ is at least true for one pair of x, y	$P(x, y)$ is never true

- Nested quantification may be thought as nested loops
- Changing ordering of quantifiers may change the meaning
  - $\forall x \forall y\ P(x, y) \equiv \forall y \forall x\ P(x, y)$,
  - $\exists x \exists y\ P(x, y) \equiv \exists y \exists x\ P(x, y)$
  - But in general, $\forall x \exists y\ P(x, y) \neq \exists y \forall x\ P(x, y)$

L04: Negating Nested Quantifiers

- Recall: $\neg\forall x P(x) = \exists x(\neg P(x))$ and $\neg\exists x P(x) = \forall x(\neg P(x))$
- Negating nested quantifiers is just doing this one at a time
  - $\neg\forall x\forall y P(x,y) = \exists x\neg(\forall y P(x,y)) = \exists x\exists y(\neg P(x,y))$
  - $\neg\exists x\exists y P(x,y) = \forall x\neg(\exists y P(x,y)) = \forall x\forall y(\neg P(x,y))$
  - $\neg\forall x\exists y P(x,y) = \exists x\neg(\exists y P(x,y)) = \exists x\forall y(\neg P(x,y))$
  - $\neg\exists x\forall y P(x,y) = \forall x\neg(\forall y P(x,y)) = \forall x\exists y(\neg P(x,y))$

A Word about Proofs

Logical system

Assumptions (premises) $\mathcal{A} = \{A_1, \ldots, A_n\}$

Conclusion P

Proof:

$$\mathcal{A} \xRightarrow{rules\ of\ inference} C_1 \text{ (new conclusion)}$$

$$\mathcal{A} \cup \{C_1\} \xRightarrow{rules\ of\ inference} C_2$$

...

$$\mathcal{A} \cup \{C_1, C_2, \ldots\} \xRightarrow{rules\ of\ inference} P$$

Valid Arguments

- Some terminologies
  - **Argument**: a sequence of statements ending with a conclusion
  - An argument is **valid** if the conclusion follows the truth of earlier statements or **premises**
  - Incorrect reasoning, or **fallacies**, leads to invalid arguments

- **Ex:**
  - S_1: "If you have a current password, then you can log onto the network"
  - S_2: "You have a current password."

Conclusion:
  - S_3: "You can log onto the network."

Valid Arguments, cont.

- Does the argument make sense? How do we check?
  - $p =$ "you have a current password"
  - $q =$ "you can log onto the network"
  - $S_1 = p \rightarrow q,\ S_2 = p$
  - $S_3 = q$
  - Question: if $(\boldsymbol{p} \rightarrow \boldsymbol{q})$ and $\boldsymbol{p}$ are both true, is $\boldsymbol{q}$ also true?
  - How do we check? Use truth table!

S_1: "If you have a current password, then you can log onto the network"
S_2: "You have a current password."
Conclusion:
S_3: "You can log onto the network."

it is a tautology

p	q	$p \rightarrow q$	$(p \rightarrow q) \wedge p$	$((p \rightarrow q) \wedge p) \rightarrow q$
T	T	T	T	T
T	F	F	F	T
F	T	T	F	T
F	F	T	F	T

  - Note that $((p \rightarrow q) \wedge p) \rightarrow q$ is a tautology

Valid Arguments, cont.

- What we have "argued" so far?
  - Abstracted an argument using propositional variables
  - Show that the argument is true
    - Logically, $((p \rightarrow q) \land p) \rightarrow q$
    - Interpretation: when the premises are true, so is the conclusion

- Note that if $p \rightarrow q = false, q = false$
  - S_1: If you have access to the network, you can change your grades
  - S_2: You have network access
  - S_3: You can change your grades

Valid Arguments, cont.

- $\mathbf{Def}^{n}$: An **argument** in propositional logic is a sequence of $n+1$ propositions. The first n propositions are **premises** and the last one the conclusion. An argument is **valid** if the premises imply the conclusion. An **argument form** is an **abstraction** of an argument in which particular propositions are replaced with propositional variables.

- Essential points:
  - We may abstract arguments as $p_1, \dots, p_n, p_{n+1} = q$
  - Valid argument $\Leftrightarrow$ $(p_1 \wedge \cdots \wedge p_n) \rightarrow q$ is a tautology
  - Valid abstract **argument form** $(p_1 \wedge \cdots \wedge p_n) \rightarrow q$ has general applicability. For example, $((p \rightarrow q) \wedge p) \rightarrow q$
  - These valid argument forms are the **rules of inference**

Rule of Inference: Modus Ponens/Tollens

- The rule that we have just looked at
  - This form is easier to see the statements & conclusion
  - Got a fancy Latin name "**modus ponens**"
  - Meaning "mode that affirming affirms"
  - In English, this mean **implication elimination**

$$\begin{array}{l} p \rightarrow q \\ p \\ \text{-----------} \\ q \end{array}$$

- Related: modus tollens
  - How do we prove this?
  - We can use good old truth table
  - But can also use modus ponens

$$\begin{array}{l} \text{If } p \xrightarrow{\text{then}} q \\ \neg q \\ \text{-----------} \\ \text{Therefore, } \neg p \end{array}$$

Examples of Modus Ponens/Tollens

- **Ex**: modus ponens
  - S_1: If it snows today, then we go skiing
  - S_2: It is snowing today
  - S_3: We go skiing

$$\begin{array}{l} p \rightarrow q \\ p \\ \hline q \end{array}$$

- **Ex**: modus tollens
  - S_1: If it snows today, then we go skiing
  - S_2: We do not go skiing
  - S_3: It is not snowing today

$$\begin{array}{l} p \rightarrow q \\ \neg q \\ \hline \neg p \end{array}$$

A Useful Set of Rules

Jerry Harris

Rule	Tautology	Name
$p \rightarrow q$ $q \rightarrow r$ ――――― $p \rightarrow r$	$((p \rightarrow q) \wedge (q \rightarrow r)) \rightarrow (p \rightarrow r)$	Hypothetical syllogism
$p \vee q$ $\neg p$ ――――― q	$((p \vee q) \wedge \neg p) \rightarrow q$	Disjunctive syllogism
p ――――― $p \vee q$	$p \rightarrow (p \vee q)$	Addition
$p \wedge q$ ――――― p	$(p \wedge q) \rightarrow p$	Simplification
p q ――――― $p \wedge q$	$((p) \wedge (q)) \rightarrow (p \wedge q)$	Conjunction
$p \vee q$ $\neg p \vee r$ ――――― $q \vee r$	$((p \vee q) \wedge (\neg p \vee r)) \rightarrow (q \vee r)$	Resolution

Some Examples

$p \to (p \vee q)$

- **Ex:**
  - S_1: It is below freezing now. p
  - S_2: It is below freezing or raining. q

$(p \wedge q) \to p$

- **Ex:**
  - S_1: It is below freezing and raining now (p over "freezing", q over "raining")
  - S_2: It is raining now

$p \to \neg q$

- **Ex:**
  - S_1: If it rains today, we will not have BBQ today (p over "rains today", q over "have BBQ")
  - S_2: If we do not have BBQ today, then we will have BBQ tomorrow $\neg q \to r$
  - S_3: If it rains today, we will have BBQ tomorrow $(p \vee q)$

Rule
$p \to q$ $q \to r$ $p \to r$
$p \vee q$ $\neg p$ q
p $p \vee q$
$p \wedge q$ p
p q $p \wedge q$
$p \vee q$ $\neg p \vee r$ $q \vee r$

Building Arguments : Example 1

- **Ex:**
  - “It is not sunny this afternoon and it is colder than yesterday.” $\neg p \wedge q$
  - “We will go swimming only if it is sunny.” $r \rightarrow p$
  - “If we do not go swimming, then we will take a canoe trip.” $\neg r \rightarrow s$
  - “If we take a canoe trip, then we will be home by sunset.” $s \rightarrow t$
  - **Derive**: “We will be home by sunset.” t

① $\neg p \wedge q \rightarrow \neg p$ Simplification

② $((r \rightarrow p) \wedge \neg p) \rightarrow \neg r$ modus tollens

③ $((\neg r \rightarrow s) \wedge \neg r) \rightarrow s$ modus ponens

④ $((s \rightarrow t) \wedge s) \rightarrow t$ modus ponens

Building Arguments : Example 2

- **Ex:**
  - "If you send me an email, then I will finish writing the program." $p \to q$
  - "If you do not send me an email, then I will go to sleep early." $\neg p \to r$
  - "If I go to sleep early, then I will wake up feeling refreshed." $r \to s$
  - **Derive**: "If I do not finish writing the program, then I will wake up feeling refreshed.

$\neg q$ s

$\neg q \to r$

$r \to s$

$\neg q \to s$

(1) $p \to q \equiv \neg q \to \neg p$

(2) $((\neg q \to \neg p) \wedge (\neg p \to r)) \to \neg p \to r$

(3) $((\neg q \to r) \wedge (r \to s)) \to \neg q \to s$

Building Arguments : Example 3

- **Ex**: Show $(p \wedge q) \vee r$ and $r \rightarrow s$ imply $p \vee s$.

① $r \rightarrow s \equiv \neg r \vee s$

resolution

② $((p \wedge q) \vee r) \wedge (\neg r \vee s)) \rightarrow (p \wedge q) \vee s$

③ $(p \wedge q) \vee s \equiv (p \vee s) \wedge (q \vee s)$ distributive law

④ $((p \vee s) \wedge (q \vee s)) \rightarrow p \vee s$ simplification

Negating Nested Quantifiers

- Recall: $\neg\forall x P(x) \equiv \exists x(\neg P(x))$ and $\neg\exists x P(x) \equiv \forall x(\neg P(x))$
  - Negating nested quantifiers is just doing this one at a time

$$\neg(\forall x \forall y\, P(x,y)) = \exists x(\neg\forall y\, P(x,y)) = \exists x\, \exists y\, (\neg P(x,y))$$

Negating Nested Quantifiers

- Recall: $\underline{\neg\forall x P(x) = \exists x(\neg P(x))}$ and $\neg\exists x P(x) = \forall x(\neg P(x))$
  - Negating nested quantifiers is just doing this one at a time
  - $\neg\forall x\forall y P(x,y) = \exists x\neg(\forall y P(x,y)) = \exists x\exists y(\neg P(x,y))$
  - $\neg\exists x\exists y P(x,y) = \forall x\neg(\exists y P(x,y)) = \forall x\forall y(\neg P(x,y))$
  - $\neg\forall x\exists y P(x,y) = \exists x\neg(\exists y P(x,y)) = \exists x\forall y(\neg P(x,y))$
  - $\neg\exists x\forall y P(x,y) = \forall x\neg(\forall y P(x,y)) = \forall x\exists y(\neg P(x,y))$

- We can now more easily state when a nested quantification is a true or false

$$\neg(\underline{\forall x}\,\underline{\forall y\, P(x,y)}) = \exists x(\underline{\neg\forall y\, P(x,y)}) = \exists x\exists y(\neg P(x,y))$$

Quantified Statements

- Common sense rules adding/removing quantifiers
  - Universal instantiation: $\forall x P(x) \rightarrow P(c)$ for any $x = c$.
  - Universal generalization: $P(c)$ for arbitrary $c \rightarrow \forall x P(x)$.
  - Existential instantiation: $\exists x P(x) \rightarrow P(c)$ for some $x = c$.
  - Existential generalization: $P(c)$ for some $c \rightarrow \exists x P(x)$.

- **Ex:**
  - "Everyone in this course has taken a course in computer science."
  - "Marla is in this class."
  - **Derive**: Marla has taken a course is computer science.

Quantified Statements

- Common sense rules adding/removing quantifiers
  - Universal instantiation: $\forall x P(x) \to P(c)$ for any $x = c$.
  - Universal generalization: $P(c)$ for arbitrary $c \to \forall x P(x)$.
  - Existential instantiation: $\exists x P(x) \to P(c)$ for some $x = c$.
  - Existential generalization: $P(c)$ for some $c \to \exists x P(x)$.

→ x in this course

$\forall x\, C(x)$ $\qquad$ $D(x)$ $\qquad$ $\forall x\,(C(x) \to D(x))$

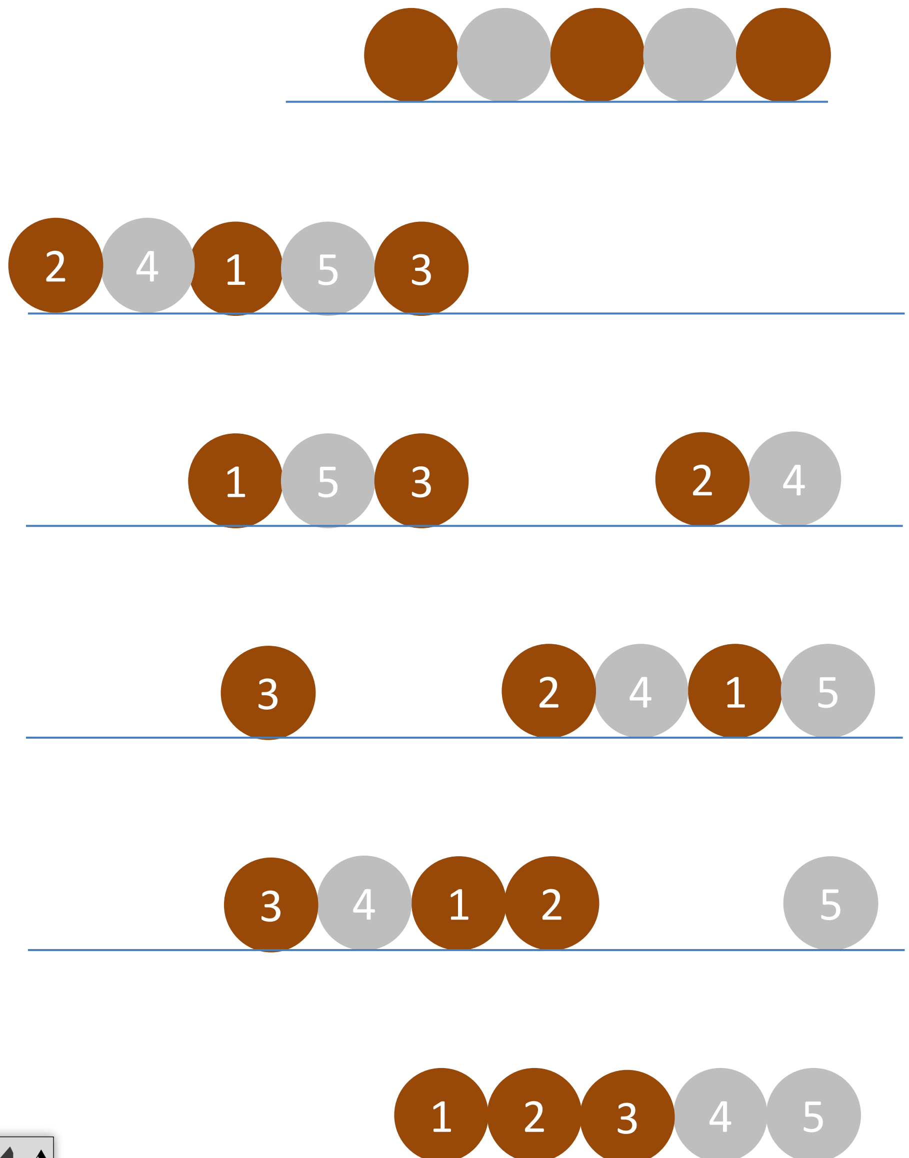
- **Ex:**
  - "Everyone in this course has taken a course in computer science."
  - "Marla is in this class." $C(\text{Marla})$ $\qquad$ $D(\text{Marla})$
  - **Derive**: Marla has taken a course is computer science.

① $(\forall x(C(x) \to D(x))) \to (C(\text{Marla}) \to D(\text{Marla}))$

② $((C(\text{Marla}) \to D(\text{Marla})) \wedge C(\text{Marla})) \to D(\text{Marla})$

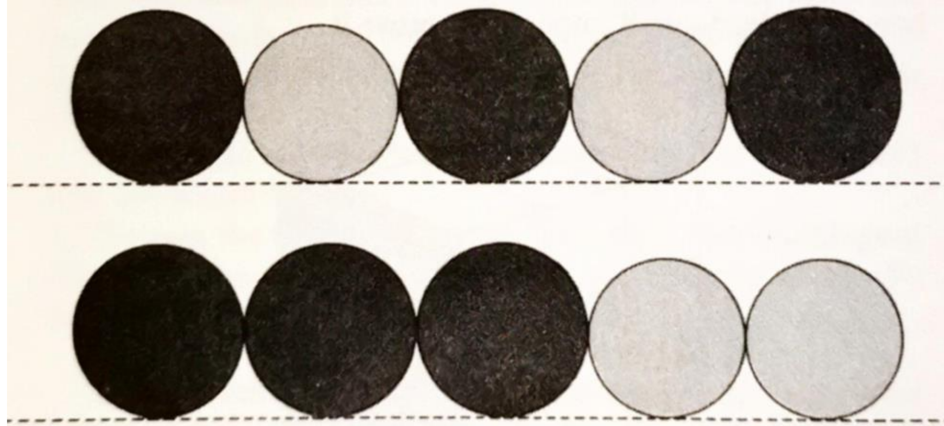
Modus ponens

Coin Rearrangement



Arrange three pennies and two dimes in a row, alternating the coins as shown. The problem is to change their positions to those shown at the bottom of the illustration in the shortest possible number of moves.

A move consists of placing the tips of the first and second fingers on any two touching coins, *one of which must be a penny and the other a dime,* then sliding the pair to another spot along the imaginary line shown in the illustration. The two coins in the pair must touch at all times. The coin at left in the pair must remain at left; the coin at right must remain at right. Gaps in the chain are allowed at the end of any move except the final one. After the last move the coins need not be at the same spot on the imaginary line that they occupied at the start.



If it were permissible to shift two coins of the same kind, the puzzle could be solved easily in three moves: slide 1, 2 to left, fill the gap with 4, 5, then move 5, 3 from right to left end. But with the proviso that each shifted pair must include a dime and penny it is a baffling and pretty problem.