Propositional Calculus
- *Semantics* (2/3)

Moonzoo Kim
CS Division of EECS Dept.
KAIST

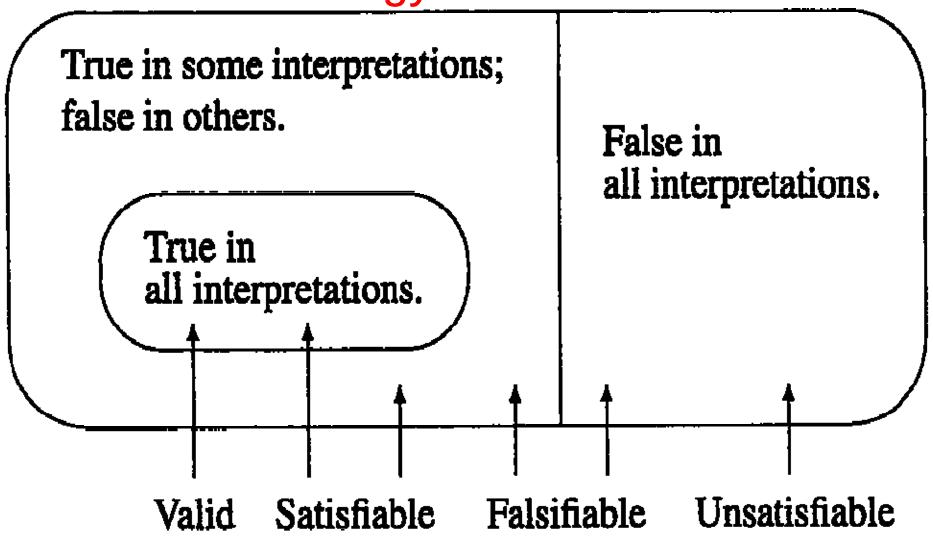
moonzoo@cs.kaist.ac.kr
http://pswlab.kaist.ac.kr/courses/cs402-07

Overview

- 2.1 Boolean operators
- 2.2 Propositional formulas
- 2.3 Interpretations
- 2.4 Logical equivalence and substitution
- 2.5 Satisfiability, validity, and consequence
- 2.6 Semantic tableaux
- 2.7 Soundness and completeness

Satisfiability v.s. validity

- Definition 2.24
  - A propositional formula *A* is satisfiable iff $\nu(A)=T$ for some interpretation ν.
    - A satisfying interpretation is called a model for *A*.
  - *A* is valid, denoted $\vDash A$, iff $\nu(A) = T$ for all interpretation ν.
    - A valid propositional formula is also called a tautology.
- Theorem 2.25
  - *A* is valid iff $\neg A$ is unsatisfiable.
  - *A* is satisfiable iff $\neg A$ is falsifiable.



Satisfiability v.s. validity

Definition 2.26

- Let $\mathcal{V}$ be a set of formulas. An algorithm is a **decision procedure** for $\mathcal{V}$ if given an arbitrary formula A $\in \mathcal{F}$, it terminates and return the answer 'yes' if A $\in \mathcal{V}$ and the answer 'no' if A $\notin \mathcal{V}$
- By theorem 2.25, a decision procedure for satisfiability can be used as a decision procedure for validity.
  - Suppose $\mathcal{V}$ is a set of all satisfiable formulas
  - To decide if A is valid, apply the decision procedure for satisfiability to $\neg$A
    - This decision procedure is called a refutation procedure

Satisfiability v.s. validity

- Example 2.27 Is $(p \rightarrow q) \rightarrow (\neg q \rightarrow \neg p)$ valid?

p	q	$p \rightarrow q$	$\neg q \rightarrow \neg p$	$(p \rightarrow q) \rightarrow (\neg q \rightarrow \neg p)$
T	*T*	*T*	*T*	*T*
T	*F*	*F*	*F*	*T*
F	*T*	*T*	*T*	*T*
F	*F*	*T*	*T*	*T*

- Example 2.28 $p \vee q$ is satisfiable but not valid

Logical consequence

- Definition 2.30 (extension of satisfiability from a single formula to a set of formulas)
  - A set of formulas $U = \{A_1 , \ldots A_n\}$ is (simultaneously) satisfiable iff there exists an interpretation ν such that $\nu (A_1) = \ldots = \nu (A_n) = T$.
  - The satisfying interpretation is called a model of U.
  - U is unsatisfiable iff for every interpretation ν, there exists an i such that $\nu (A_i) = F$.

KAIST

Logical consequence

- Let U be a set of formulas and A a formula. If A is true in every model of U, then A is a logical consequence of U.
  - Notation: $U \models A$
  - If U is empty, logical consequence is the same as validity
- Theoem 2.38
  - $U \models A$ iff $\models A_1 \wedge \ldots \wedge A_n \rightarrow A$ where $U=\{A_1 \ldots A_n\}$
  - Note Theorem 2.16
    - $A_1 \equiv A_2$ if and only if $A_1 \leftrightarrow A_2$ is true in every interpretation

Theories

- Logical consequence is the central concept in the foundations of mathematics
  - Valid formulas such as $p \vee q \leftrightarrow q \vee p$ are trivial and not interesting
  - Ex. Euclid assumed five formulas about geometry and deduced an extensive set of logical consequences.
- Definition 2.41
  - A set of formulas $\mathcal{T}$ is a theory if and only if it is **closed** under logical consequence.
    - $\mathcal{T}$ is closed under logical consequence if and only if for all formula A, if $\mathcal{T} \vDash A$ then $A \in \mathcal{T}$.
  - The elements of $\mathcal{T}$ are called theorems
- Let U be a set of formulas. $\mathcal{T}(U) = \{A \mid U \vDash A\}$ is called the theory of U. The formulas of U are called axioms and the theory $\mathcal{T}(U)$ is axiomatizable.
  - Is $\mathcal{T}(U)$ theory?

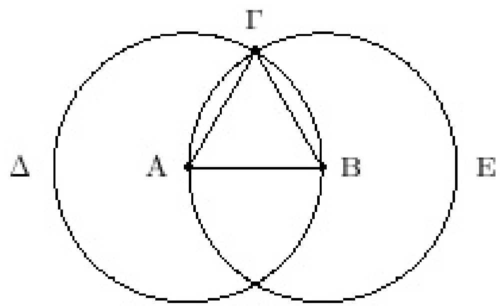
Examples of theory

- $U = \{ p \lor q \lor r, q \to r, r \to p \}$
- Interpretation v_1, v_3 and v_4 are models of U
- Which of the followings are true?
  - $U \models p$
  - $U \models q \to r$
  - $U \models r \lor \neg q$
  - $U \models p \land \neg q$
- Theory of U, i.e, $\mathcal{T}(U)$
  - $U \subseteq \mathcal{T}(U)$
    - because for all formula $A \in U$, $A \models A$
  - $p \in \mathcal{T}(U)$
    - because $U \models p$
  - $q \to r \in \mathcal{T}(U)$
    - because $U \models q \to r$
  - $p \land (q \to r) \in \mathcal{T}(U)$
    - because $U \models p \land (q \to r)$
      - since $U \models p$ and $U \models q \to r$ $\therefore$
  - …

	p	q	r	$p \lor q \lor r$	$q \to r$	$r \to p$
v_1	T	T	T	T	T	T
v_2	T	T	F	T	F	T
v_3	T	F	T	T	T	T
v_4	T	F	F	T	T	T
v_5	F	T	T	T	T	F
v_6	F	T	F	T	F	T
v_7	F	F	T	T	T	F
v_8	F	F	F	F	T	T

Ex. Theory of Euclidean geometry

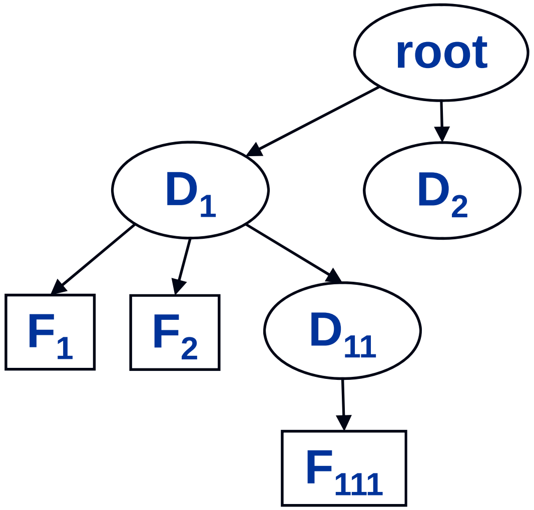
- A set of 5 axioms $U = \{A_1, A_2, A_3, A_4, A_5\}$ such that
  - A_1:Any two points can be joined by a unique straight line.
  - A_2:Any straight line segment can be extended indefinitely in a straight line.
  - A_3:Given any straight line segment, a circle can be drawn having the segment as radius and one endpoint as center.
  - A_4:All right angles are congruent.
  - A_5:For every line *l* and for every point P that does not lie on *l* there exists a unique line *m* through P that is parallel to *l*.
- Euclidean theory $\mathcal{T}_{\text{Euclid}} = \mathcal{T}(U) = \{ A \mid U \vDash A\}$
  - I.e., $\mathcal{T}_{\text{euclid}}$ is axiomatizable by the above 5 axioms
  - Ex. one logical consequence of the axioms
    - given a line segment AB, an equilateral triangle exists that includes the segment as one of its sides.



KAIST

Ex2. Model checking (formal verification)

- A file system M can be specified by the following 7 formulas (i.e., a file system model M = { $A_1,A_2,A_3,A_4,A_5,A_6,A_7$})
  - A_1:A file system object has one or no parent.
    - sig FSObject { parent: lone Dir }
  - A_2:A directory has a set of file system objects
    - sig Dir extends FSObject { contents: set FSObject }
  - A_3:A directory is the parent of its contents
    - fact defineContents { all d: Dir, o: d.contents | o.parent = d }
  - A_4: A file in the file system is a file system object
    - sig File extends FSObject {}
  - A_5: All file system objects are either files or directories
    - fact fileDirPartition { File + Dir = FSObject }
  - A_6: There exists only one root
    - one sig Root extends Dir { }{ no parent }
  - A_7: File system is connected
    - fact fileSystemConnected { FSObject in Root.*contents }



- We can prove that this file system does not have a cyclic path
  - A: No cyclic path exists
    - assert acyclic { no d: Dir | d in d.^contents }
  - $M \models A$ (i.e., this file system M does not have cyclic path)