

ソフトウェアモデル論(2010年度) 第10回・2010/11/29

桑原 寛明
情報理工学部 情報システム学科

命題

- 内容の真偽が確定できる文
 - 加算はチューリング機械計算可能である
 - 1と10は等しい
 - 情報理工学部の学生数は2233人である
- 命題同士が関連することもある
 - 風が吹く
 - 風が吹くと桶屋が儲かる
 - ⇒ 桶屋が儲かる

ソフトウェアモデル論(2010/11/29)

2

論理学

- 命題の集まりについて、ある命題の真偽が他の命題の真偽にどのように影響するか、命題間の関連を系統的に調べる学問
- 数理論理学
 - 数学における形式手法、記号的手法を用いて行う論理学

ソフトウェアモデル論(2010/11/29)

3

命題変数

- 命題を表す変数
 - 命題が述べる内容には興味がない
 - 興味があるのは命題の真偽と命題間の関係
- 特に、原子命題を表す記号
 - 原子命題: 最も基本的な命題

ソフトウェアモデル論(2010/11/29)

4

論理式

- 命題を表す記号列
1. 命題変数は論理式である
 2. P, Q が論理式であれば
 - $(\neg P)$ 否定、～でない
 - $(P \wedge Q)$ 連言、かつ
 - $(P \vee Q)$ 選言、または
 - $(P \rightarrow Q)$ 含意、ならば
- は論理式である
1. 1. と 2. から作られるものだけが論理式である

ソフトウェアモデル論(2010/11/29)

5

意味論

- 論理式の意味とは論理式の真理値
 - 真 or 偽
- 以下の2つから決まる
 - 命題変数の真理値
 - 論理演算子($\neg, \wedge, \vee, \rightarrow$)の意味

ソフトウェアモデル論(2010/11/29)

6

解釈

- 命題変数の真理値を定義する関数
 - true : 真、false : 偽
- 解釈を I 、命題変数の集合を Σ とすると

$$I : \Sigma \rightarrow \{ \text{true}, \text{false} \}$$
- すべての $p \in \Sigma$ に対して $I(p) = \text{true}$ または $I(p) = \text{false}$

ソフトウェアモデル論(2010/11/29)

7

解釈の例

- $\Sigma = \{ p, q, r \}$ とすると解釈は8通りあり得る

	p	q	r
I_1	true	true	true
I_2	true	true	false
I_3	true	false	true
I_4	true	false	false
I_5	false	true	true
I_6	false	true	false
I_7	false	false	true
I_8	false	false	false

ソフトウェアモデル論(2010/11/29)

8

論理演算子の意味

- 真理値関数によって定義
 - 以下の Not, And, Or, Imp がそれぞれ否定、連言、選言、含意の意味を定義

P	Not(P)	P	Q	And(P,Q)	Or(P,Q)	Imp(P,Q)
true	false	true	true	true	true	true
false	true	true	false	false	true	false
		false	true	false	true	true
		false	false	false	false	true

ソフトウェアモデル論(2010/11/29)

9

論理式の意味

- 命題変数の集合 Σ
- 解釈 I のもとでの論理式 P の真理値 $V_I(P)$
 - 解釈は命題変数の真理値を決める

$$V_I(P) = \begin{cases} I(p) & P \text{ が命題変数 } p \in \Sigma \text{ の場合} \\ \text{Not}(V_I(Q)) & P \text{ が } \neg Q \text{ の場合} \\ \text{And}(V_I(Q), V_I(R)) & P \text{ が } Q \wedge R \text{ の場合} \\ \text{Or}(V_I(Q), V_I(R)) & P \text{ が } Q \vee R \text{ の場合} \\ \text{Imp}(V_I(Q), V_I(R)) & P \text{ が } Q \rightarrow R \text{ の場合} \end{cases}$$

ソフトウェアモデル論(2010/11/29)

10

モデル

- 論理式集合 Φ
- 解釈 I
- I が Φ のモデルである
 - iff
 - Φ に含まれるすべての論理式 $P \in \Phi$ について $I(P) = \text{true}$
- $\models \Phi$ あるいは $I \models \Phi$ と書く

ソフトウェアモデル論(2010/11/29)

11

論理的帰結

- 論理式集合 Φ
- 論理式 P
- P は Φ の論理的帰結である
 - iff
 - すべての解釈 I について、 I が Φ のモデルならば I は P モデルでもある
- $\Phi \models P$ と書く

ソフトウェアモデル論(2010/11/29)

12

トートロジー(恒真式)

- 論理式 P
- P はトートロジーである
iff
すべての解釈 I に対して $I \models P$
- 任意の解釈のもとで真になる
- 例えば、 $p \vee \neg p$ はトートロジー

ソフトウェアモデル論(2010/11/29)

13

充足可能

- 論理式集合 Φ に対してモデルが存在するならば Φ は充足可能である
- 論理式 P に対してモデルが存在するならば P は充足可能である
- 充足可能でなければ充足不能

ソフトウェアモデル論(2010/11/29)

14

恒偽式

- 充足不能な論理式
- どのような解釈のもとでも偽である
- 例えば、 $p \wedge \neg p$ は恒偽式

ソフトウェアモデル論(2010/11/29)

15

論理的帰結の判定

- 論理式 P が論理式集合 Φ の論理的帰結であるか判定したい
- 手順？
 1. すべての解釈について Φ のモデルであるか調べる
 2. Φ のモデルであるすべての解釈のもとで P が真であるか調べる
- Φ と P に含まれる命題変数が n 種類ならば 2^n 通りの解釈について調べなければならない
⇒ 効率が悪い

ソフトウェアモデル論(2010/11/29)

16

証明系

- 論理式が論理式集合の論理的帰結であることを、論理式(の列)に対する機械的な操作のみによって調べる方法
 - 論理式の意味(真理値)を考えない
 - 判定アルゴリズムの一種とみなしてもよい
- 証明系は、論理式(の集合)から別の論理式を導出する推論規則の集合として定義される

ソフトウェアモデル論(2010/11/29)

17

証明系の種類

- 演繹系
 - 前提の論理式集合から結論の論理式が導出されるまで推論を繰り返す証明系
 - 例えば、LK や自然演繹など
- 反駁系
 - 前提の論理式集合に結論の否定を加えて推論し、否定的な結果が得られたら成功とする証明系
 - 例えば、分解証明系など

ソフトウェアモデル論(2010/11/29)

18

シーケント

- $P_1, \dots, P_n \vdash Q$
- 論理式集合 $\{P_1, \dots, P_n\}$ から推論を開始し、論理式 Q が得られることを表す
 - $\{P_1, \dots, P_n\}$: 前提、前件
 - Q : 結論、後件
- 推論を繰り返す(推論規則を繰り返し適用する)過程が証明

ソフトウェアモデル論(2010/11/29) 19

推論規則の形式

$$\frac{\text{前提1} \quad \dots \quad \text{前提n}}{\text{結論}} \text{規則名}$$

- 各前提と結論は論理式
- 前提1から前提nまでのn個の論理式から結論の論理式を推論(導出)する

$$\frac{P \quad Q}{P \wedge Q} \wedge i$$

論理式 P と Q から論理式 $P \wedge Q$ を推論(導出)してよい

ソフトウェアモデル論(2010/11/29) 20

証明系の健全性

- 論理式集合 Φ から論理式 P が導出(証明)できるならば P は Φ の論理的帰結である
 - $\Phi \vdash P$ ならば $\Phi \models P$
- 証明系が健全でない場合、証明できたことを信じてよいかわからない
 - 証明できても論理的帰結でないことがある
 - ⇒ 証明になっていない

ソフトウェアモデル論(2010/11/29) 21

証明系の完全性

- 論理式 P が論理式集合 Φ の論理的帰結ならば Φ から P を導出(証明)できる
 - $\Phi \models P$ ならば $\Phi \vdash P$
- 証明系が完全であれば、すべての論理的帰結を証明できる
 - 完全でなければ証明できないものがある

ソフトウェアモデル論(2010/11/29) 22

自然演繹

- 以下の推論規則からなる証明系

$$\frac{P \quad Q}{P \wedge Q} \wedge i \quad \frac{P \wedge Q}{P} \wedge e_1 \quad \frac{P \wedge Q}{Q} \wedge e_2 \quad \frac{P}{P \vee Q} \vee i_1 \quad \frac{Q}{P \vee Q} \vee i_2$$

$$\frac{P \quad P \rightarrow Q}{Q} \rightarrow e \quad \frac{\neg P}{P} \neg e \quad \frac{\perp}{P} \perp e \quad \frac{P \quad \neg P}{\perp} \neg e$$

$$\frac{[P] \dots Q}{P \rightarrow Q} \rightarrow i \quad \frac{[P] \dots P \vee Q \quad [Q] \dots R}{R} \vee e \quad \frac{[P] \dots \perp}{\neg P} \neg i$$

ソフトウェアモデル論(2010/11/29) 23

選言に関する推論規則

$$\frac{P \quad Q}{P \wedge Q} \wedge i \quad \frac{P \wedge Q}{P} \wedge e_1 \quad \frac{P \wedge Q}{Q} \wedge e_2$$

- i
 - 演算子を導入
- e
 - 演算子を除去

ソフトウェアモデル論(2010/11/29) 24

p∧q ⊢ q∧p の証明

$$\frac{\frac{p \wedge q}{q} \wedge e_2 \quad \frac{p \wedge q}{p} \wedge e_1}{q \wedge p} \wedge i$$

ソフトウェアモデル論(2010/11/29)

25

証明木(導出木)

- シーケントの前提から結論を推論する過程を木構造で図示したもの(ただし根が下)
- 木構造の節点は論理式
 - 葉が前提
 - 根が結論
- 葉を除く各節点は、子節点の論理式に推論規則を適用して得られる論理式

ソフトウェアモデル論(2010/11/29)

26

二重否定に関する推論規則

$$\frac{P}{\neg\neg P} \neg\neg i \quad \frac{\neg\neg P}{P} \neg\neg e$$

- $\neg\neg i$ は他の規則を用いて導出できる(例5.20)
 - なくても大丈夫

ソフトウェアモデル論(2010/11/29)

27

含意に関する推論規則

$$\frac{\begin{array}{c} [P] \\ \vdots \\ Q \end{array}}{P \rightarrow Q} \rightarrow i \quad \frac{P \quad P \rightarrow Q}{Q} \rightarrow e$$

- [P] は論理式 P の一時的な仮定を表す
 - 仮定は推論規則(この場合は $\rightarrow i$)の適用で消費される
 - 仮定は証明木の葉に出現する

ソフトウェアモデル論(2010/11/29)

28

p → (q → r) ⊢ p ∧ q → r の証明

$$\frac{\frac{\frac{[p \wedge q]}{q} \wedge e_2 \quad \frac{\frac{[p \wedge q]}{p} \wedge e_1 \quad p \rightarrow (q \rightarrow r)}{q \rightarrow r} \rightarrow e}{r} \rightarrow e}{p \wedge q \rightarrow r} \rightarrow i$$

ソフトウェアモデル論(2010/11/29)

29

p ∧ q → r ⊢ p → (q → r) の証明

$$\frac{\frac{\frac{[p]_1 \quad [q]_2}{p \wedge q} \wedge i \quad p \wedge q \rightarrow r}{r} \rightarrow e}{q \rightarrow r} \rightarrow i, 2}{p \rightarrow (q \rightarrow r)} \rightarrow i, 1$$

ソフトウェアモデル論(2010/11/29)

30

選言に関する推論規則

$$\frac{P}{P \vee Q} \vee i_1 \quad \frac{Q}{P \vee Q} \vee i_2 \quad \frac{P \vee Q \quad \begin{array}{c} [P] \\ \vdots \\ R \end{array} \quad \begin{array}{c} [Q] \\ \vdots \\ R \end{array}}{R} \vee e$$

ソフトウェアモデル論(2010/11/29)

31

否定に関する推論規則

$$\frac{\perp}{P} \perp e \quad \frac{\begin{array}{c} [P] \\ \vdots \\ \perp \end{array}}{\neg P} \neg i \quad \frac{P \quad \neg P}{\perp} \neg e$$

- \perp は矛盾を表す
- 矛盾からはどのようなことでも推論できる
- 矛盾が導出された場合は前提が間違っている

ソフトウェアモデル論(2010/11/29)

32

$\neg\neg i$ の導出

- $P \vdash \neg\neg P$ は $\neg\neg i$ ではなく別の推論規則を使って以下のように導出可能

$$\frac{\frac{P \quad [\neg P]}{\perp} \neg e}{\neg\neg P} \neg i$$

ソフトウェアモデル論(2010/11/29)

33

派生規則

- 他の推論規則を使って導出可能な推論規則
 - 証明済みの派生規則は推論規則の一つとして使ってもよい
- 例えば
 - $\neg\neg i$
 - MT(modus tollens: 後件否定)
 - PBC(proof by contradiction: 背理法)
 - LEM(law of excluded middle: 排中律)

ソフトウェアモデル論(2010/11/29)

34

MT

$$\frac{P \rightarrow Q \quad \neg Q}{\neg P} \text{MT}$$

- 証明

$$\frac{\frac{\begin{array}{c} [P] \\ \vdots \\ P \rightarrow Q \end{array} \rightarrow e \quad Q}{Q} \rightarrow e \quad \neg Q}{\perp} \neg e}{\neg\neg P} \neg i$$

ソフトウェアモデル論(2010/11/29)

35

PBC

$$\frac{\begin{array}{c} [\neg P] \\ \vdots \\ \perp \end{array}}{P} \text{PBC}$$

- 証明

$$\frac{\frac{\begin{array}{c} [\neg P] \\ \vdots \\ \perp \end{array}}{\neg\neg P} \neg i}{P} \neg\neg e$$

ソフトウェアモデル論(2010/11/29)

36

LEM

$$\frac{}{P \vee \neg P} \text{LEM}$$

- 証明

$$\frac{\frac{\frac{[P]_1}{P \vee \neg P} \vee i_1}{\neg(P \vee \neg P)} \neg e}{\perp} \neg i, 1 \quad \frac{\frac{[\neg P]_2}{P \vee \neg P} \vee i_2}{\neg(P \vee \neg P)} \neg e}{P} \text{PBC, 2}}{P \vee \neg P} \text{PBC, 3}$$

ソフトウェアモデル論(2010/11/29) 37

矛盾

- 論理式集合 Φ から \perp が導出できる場合、 Φ は矛盾
 - $\Phi \vdash \perp$
 - 任意の解釈について、 Φ に含まれるすべての論理式が真にならない
- 矛盾でない場合、無矛盾

ソフトウェアモデル論(2010/11/29) 38

矛盾の性質

- Φ は矛盾
- 任意の論理式 P に対して $\Phi \vdash P$
- $\Phi \vdash P$ かつ $\Phi \vdash \neg P$ なる論理式 P が存在する

ソフトウェアモデル論(2010/11/29) 39