谓词的逻辑

对应《问题求解》
表达更丰富的内容: 谓词 部分与整体的关系;尝试解释谓词公式;
谓词公式的标准形;谓词逻辑的推理要求

语言的表达能力

命题逻辑的缺陷

(1) 张三是个法外狂徒;

(2) 李四与张三是好朋友;

(3) 李四也是一个法外狂徒;

(4) 王五站在张三与李四中间;

(5) 王五长得比张三与李四都高;

(6) 所有的法外狂徒终将绳之以法;

(7) 存在法外狂徒改过自新.

无法表达部分和整体的关系!

一. 个体, 谓词和量词

Eg. 在所有的算法中, 有一个算法优于其他的每一个算法

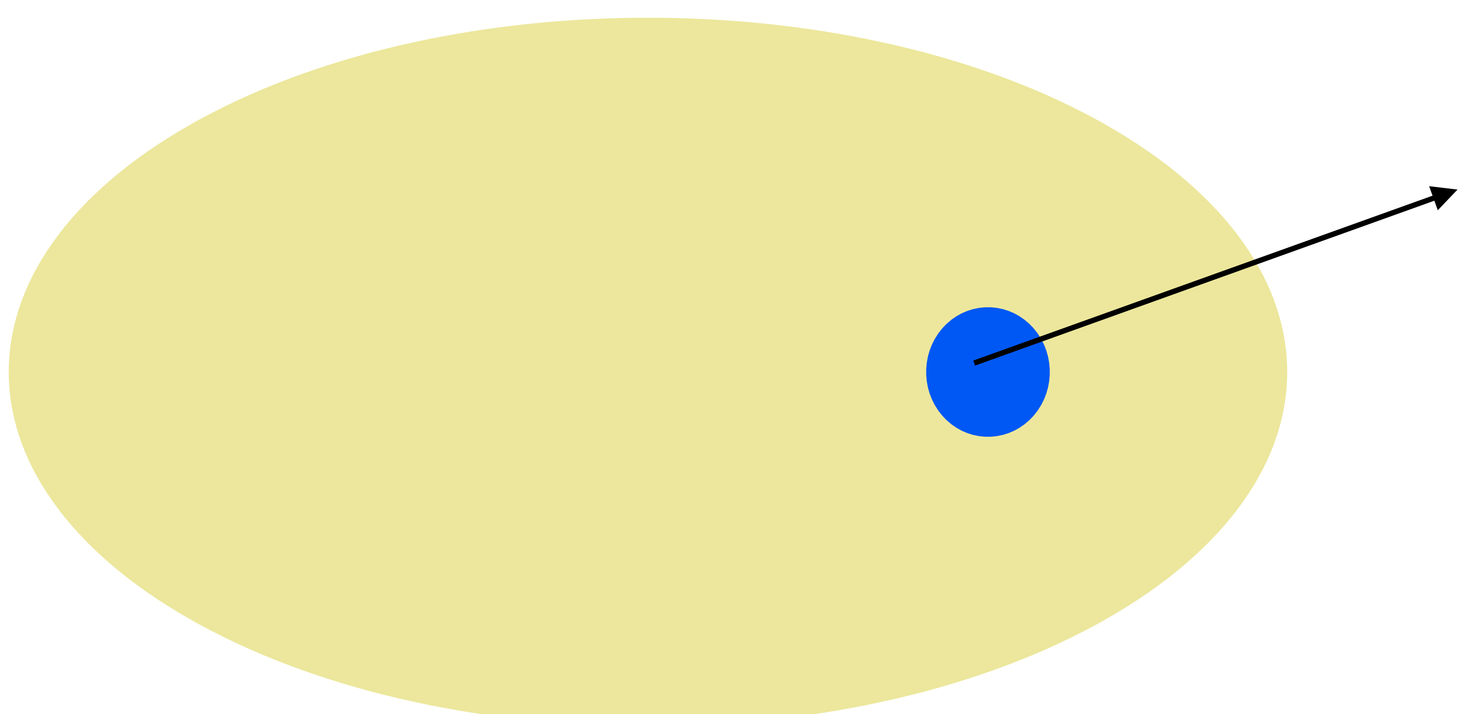
1. 构成
    - 圈定讨论的范围 论域(domain)

所有的算法

一. 个体, 谓词和量词

Eg. 在所有的算法中, 有一个算法优于其他的每一个算法

1. 构成
   - 圈定讨论的范围 论域(domain)
   - 所有的都是? 还是有一部分存在? 量词(quantifier)



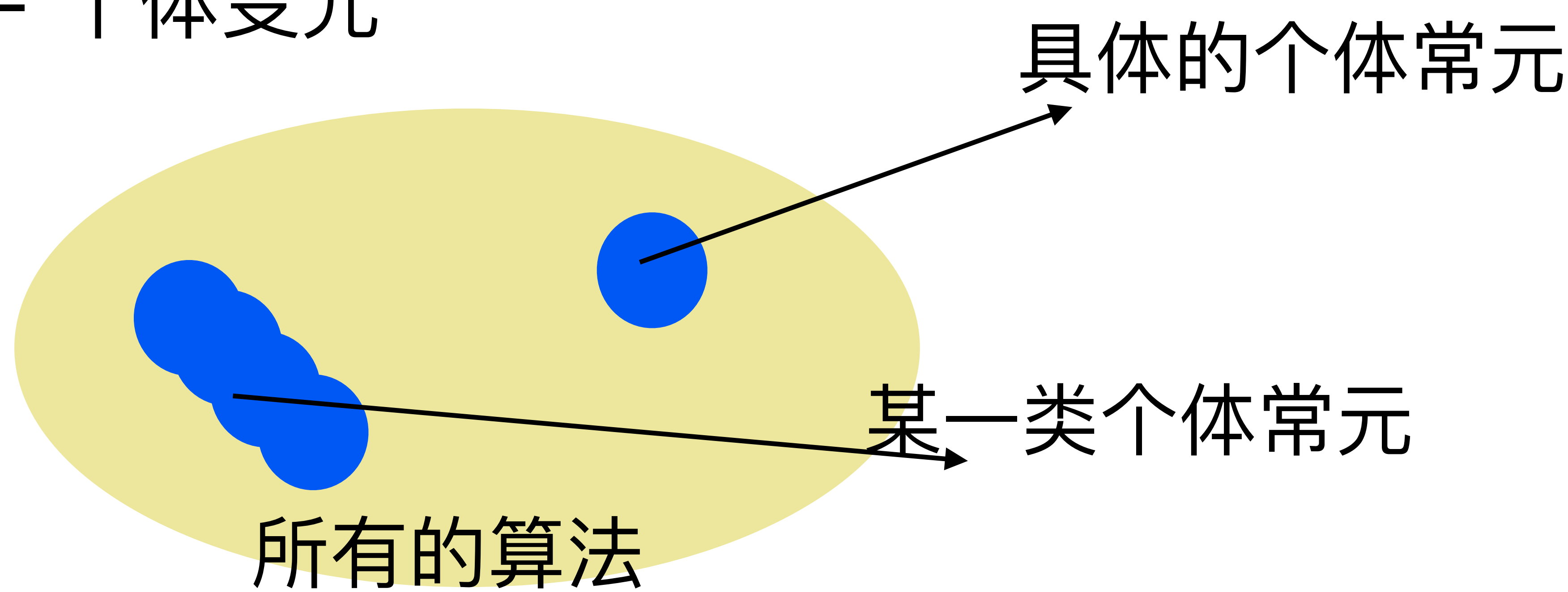
这个算法很特

所有的算法

一. 个体, 谓词和量词

Eg. 在所有的算法中, 有一个算法优于其他的每一个算法

1. 构成
    - 圈定讨论的范围 论域(domain)
    - 所有的都是? 还是有一部分存在? 量词(quantifier)
    - 有一个 “个体”
        - 具体, 不变的 – 个体常元
        - 泛指的 – 个体变元



一. 个体, 谓词和量词

Eg. 在所有的算法中, 有一个算法优于其他的每一个算法

1. 构成
   - 圈定讨论的范围　论域(domain)
   - 所有的都是? 还是有一部分存在? 量词(quantifier)
   - 有一个 “个体”
     - 具体, 不变的　–　个体常元
     - 泛指的 – 个体变元
   - 谓词: 做判断的基础

一. 个体, 谓词和量词

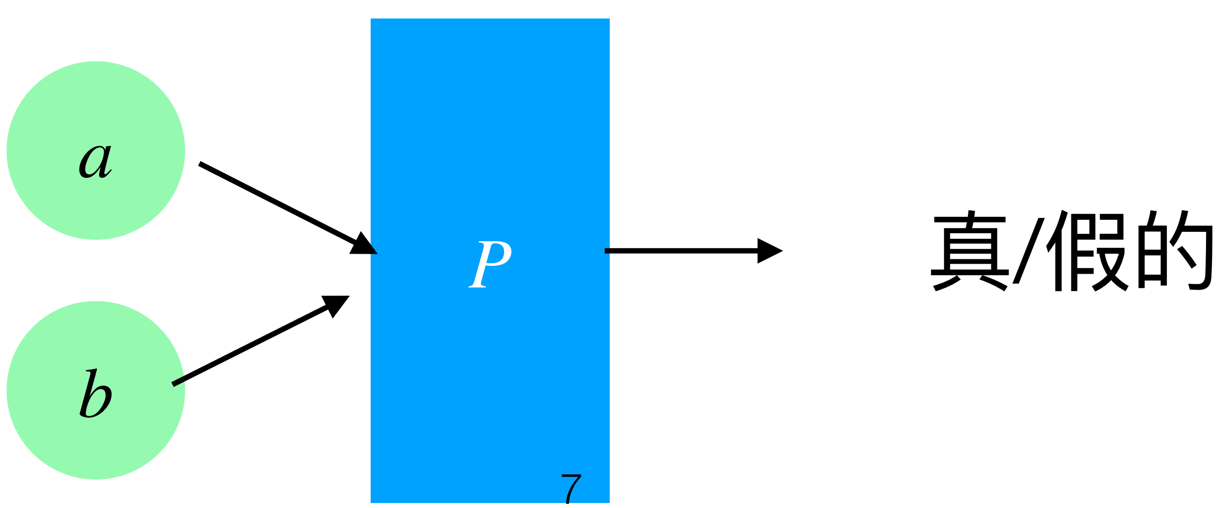
[注]

(1) 如果讨论问题的论域是有限的时候(比如是$\{d_1, d_2, \cdots, d_n\}$):

$$\forall x . P(x) \Leftrightarrow (P(d_0) \wedge P(d_1) \wedge \cdots \wedge P(d_n) := \bigwedge_{i=0}^{n} P(d_i);$$

$$\exists x . P(x) \Leftrightarrow (P(d_0) \vee P(d_1) \vee \cdots \vee P(d_n) := \bigvee_{i=0}^{n} P(d_i).$$

(2) 谓词的 “元数”: 左边的 “输入” 个数.



一. 个体, 谓词和量词

2. 一些记录方法

- 所有算法的集合 $\exists x \,.\, (\forall y \,.\, \text{Better}(x, y))$
- 全总域——切的东西

  $\exists x \,.\, (\text{isAlgorithm}(x) \land \forall y \,.\, \text{isAlgorithm}(y) \rightarrow \text{Better}(x, y))$

二. 谓词公式

$$\exists x\,.\,(\text{isAlgorithm}(x) \wedge \forall y\,.\,\text{isAlgorithm}(y) \rightarrow \text{Better}(x, y))$$

Syntax	Schematics

字母表
Alphabet

a, b, c

1. 谓词公式的构成

(1) 构成字母的元素

- 逻辑连接词: $\neg, \vee, \wedge, \rightarrow, \leftrightarrow$.
- 量词符号: $\forall, \exists$
- 变元符号: $x, y, z, \cdots$
- 左右括号: (,)
- 常数符号: 表达特殊的个体
- 函数符号: n元函数符号$f, g, h, \cdots$(个体上的运算)
- 谓词符号: n元谓词符号P, Q, R(个体之间的关系)

二. 谓词公式

$$\exists x\,.\,(\text{isAlgorithm}(x) \land \forall y\,.\,\text{isAlgorithm}(y) \rightarrow \text{Better}(x, y))$$

Syntax	Schematics

单词
Words

apple

1. 谓词公式的构成

(2) 项

- 每个变元x, y, z是一个项
- 每个常数记号是一个项
- 如果有$t_1, t_2, \cdots, t_n$是, 并且f是n元函数的记号, 那么$f(t_1, t_2, \cdots, t_n)$是项.
- 除此之外, 别无其他

二. 谓词公式

$\exists x . (\text{isAlgorithm}(x) \wedge \forall y . \text{isAlgorithm}(y) \to \text{Better}(x, y))$

Syntax | Schematics

I like eating apples

句子
Sentences

1. 谓词公式的构成

(3) 公式

- 若$t_1, \cdots, t_n$是项, P是一个n元函数符号, 那么 $P(t_1, t_2, \cdots, t_n)$是原子公式
- 如果α, β是公式, 那么$\neg\alpha, \alpha * \beta$是一个项
- 如果α是一个公式, 那么$\forall x . \alpha, \exists x . \alpha$是公式.
- 除此之外, 别无其他

二. 谓词公式

$$\exists x\,.\,(\text{isAlgorithm}(x) \wedge \forall y\,.\,\text{isAlgorithm}(y) \to \text{Better}(x, y))$$

Syntax	Schematics

忘记刚刚说的吧!

只要正常写过数学表达式, 都是可以胜任的.
但是还是给出一些[小提示]

- 不要起冲突的名字, 注意名字的 “存活区间”

[注]①变元的“自由”与“约束” — 类比程序设计语言的变元

$\forall x(\beta(x))$ 约束变元

$\forall x(\beta(x)) \wedge x \ldots$

②连续出现 $\forall x.\exists y.\forall z(\cdots)$

for i in range ...
for j in range ...
for k in range ...

二. 谓词公式

这方面还是要回忆一下我们写过的代码(很不巧, python没有全局变量的概念)

```
In [1]: a=1

In [2]: b=2

In [3]: for i in range(1):
   ...:     a = 3
   ...:     a = a+1
   ...:     print(a)
   ...: for i in range(1):
   ...:     a = a+1
   ...:     print(a)
   ...:
4
5
```

二. 谓词公式

这方面还是要回忆一下我们写过的代码(不过C有!)

nvim a.cpp

```
#include <bits/stdc++.h>
using namespace std;

int a=5;// <-----------------------------------------+
int main(){                                        //|
    for(int i=1; i<2; i++){                        //|
        int a=3; // <---Opened a new LOCAL variable--+ |
        a = a+1; //----------------------------------+ //|
        printf("%d\n", a);                         //|
    }                                              //|
     for(int i=1; i<2; i++){                       //|
        a = a+1;// Still using GLOBAL variable ------------+
        printf("%d", a);
    }
    return 0;
}
~
~
~
:!g++ a.cpp && ./a.out
4
6
Press ENTER or type command to continue
```

二. 谓词公式

$$\exists x\,.\,(\text{isAlgorithm}(x) \wedge \forall y\,.\,\text{isAlgorithm}(y) \rightarrow \text{Better}(x, y))$$

Syntax	Schematics

free bound

2、变元 ①自由与约束(有指令时告诉着"什么是可以的")

② 对变元的操作.

(I) 改名: A中自由/约束出现 ①不自由、约束出现 ②量词后约束不同

(II) 代入: 不重名… 混淆.

Eg. $\forall x(P(x,y) \wedge \exists y\, Q(y) \wedge M(x,y)) \wedge (\forall x R(x) \rightarrow Q(x))$

z $\downarrow$ ℓ

— 得到的5万公式节例.

二. 谓词公式

当时高中我同桌(数学成绩常年~130+)的小插曲…

一道圆锥曲线题: (1) $x^2/a^2 + y^2/b^2 = \ldots$, 求a,b
(2)关于a的一个问题….

某同学: 这a不是具体的数, 我不会做
数学老师: 同一道题里面一个字母有一个意思.

二. 谓词公式

$$\exists x\,.\,(\text{isAlgorithm}(x) \land \forall y\,.\,\text{isAlgorithm}(y) \rightarrow \text{Better}(x, y))$$

Syntax	Schematics

3. 公式的解释：意义是什么？　　抽象符号 → 什么意义？

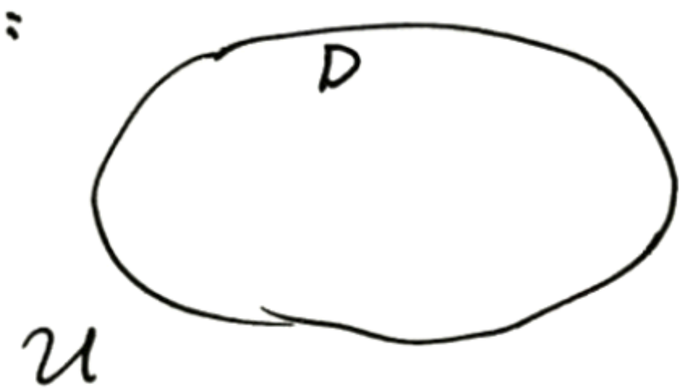
Schematics

例子：$\alpha = \forall x.(x \cdot x \neq 1+1)$,

α 在数字结构 $\mathcal{U}=\mathbb{Q}$ 中为真，但在 $\mathcal{U}=\mathbb{R}$ 为假.

①前提：
- 论域是什么？　解释论域　限定个体范围.
- 对常数符号、函数符号、谓词符号解释
- 对自由变元解释

②解释：



predicate
- 谓词

function
- 函数

free variable
- 自由变元

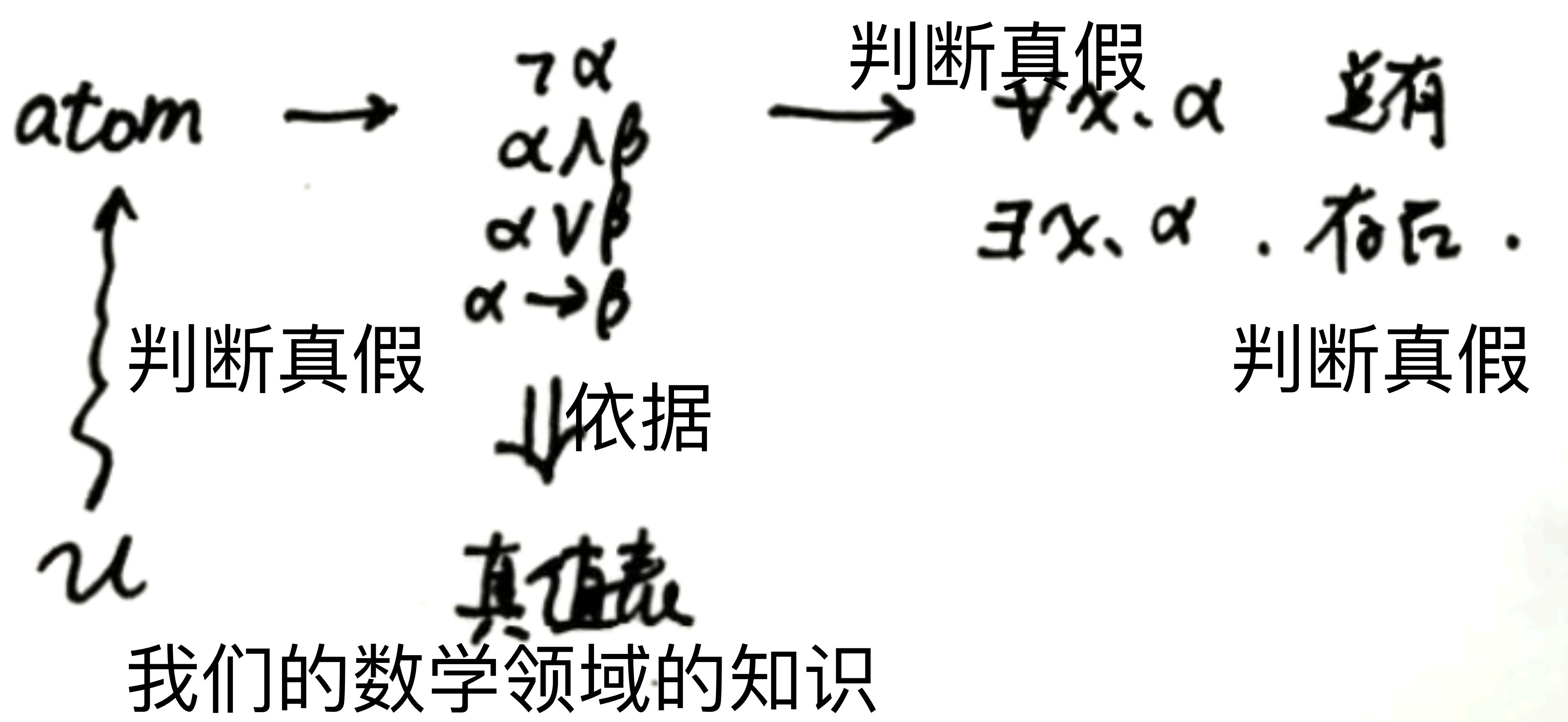
individual const.
- 个体变元

propositional var.
- 命题变元　$f \rightarrow \{T, F\}$

二. 谓词公式

$$\exists x\,.\,(\mathrm{isAlgorithm}(x) \wedge \forall y\,.\,\mathrm{isAlgorithm}(y) \rightarrow \mathrm{Better}(x, y))$$

Syntax	Schematics



回忆: 在命题逻辑里面学习了真值表技术, 这里能不能用?

二. 谓词公式

$$\exists x\,.\,(\text{isAlgorithm}(x) \wedge \forall y\,.\,\text{isAlgorithm}(y) \rightarrow \text{Better}(x, y))$$

Syntax	Schematics

4、公式的真值.

① Models ：一个可以让公式为真的解释

α	β	$\alpha\wedge\beta$
T	T	T

② 逻辑有效： true for all valid interpretations

问：是否有机械的方法判定 validity？ 不存在、

Turing停机问题

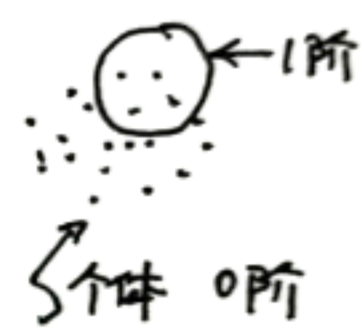
5、一阶谓词逻辑：引入了变量，但未完全引入.

①"一阶" 变量仅仅代表取值于个体.

Eg、 $\forall P((0\in P\wedge\forall i(i\in P\rightarrow i+1\in P))\rightarrow\forall n(n\in P))$.

② 不可以在谓词集合上定义量词

张三具有所有性质

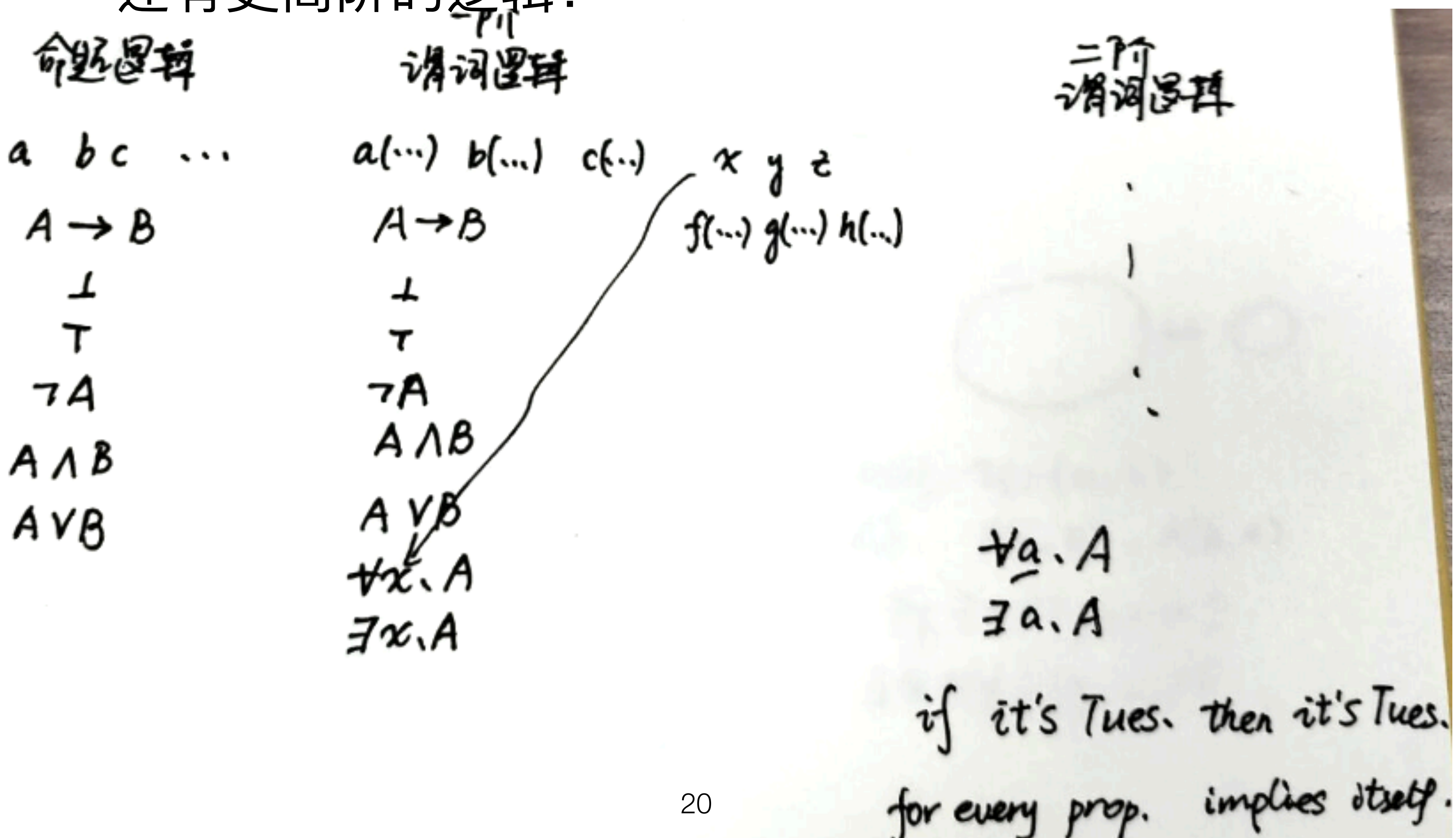


二. 谓词公式

$$\exists x\,.\,(\text{isAlgorithm}(x) \land \forall y\,.\,\text{isAlgorithm}(y) \rightarrow \text{Better}(x, y))$$

Syntax	Schematics

还有更高阶的逻辑?



三. 等值演算(calculus)

1、逻辑等价： α与β在论域上"一致"（等值）.

来源 命题逻辑 一阶逻辑 → 量词

2、一阶逻辑中的等值式.

① 有限个体中的消去 $\{a_1, \cdots, a_n\}$

$$\forall x. A(x) \Leftrightarrow A(a_1) \wedge A(a_2) \wedge \cdots \wedge A(a_n)$$

$$\exists x. A(x) \Leftrightarrow A(a_1) \vee A(a_2) \vee \cdots \vee A(a_n).$$

$\Sigma \quad \Pi$

② 量词否定等值式.

$$\neg(\forall x A(x)) \Leftrightarrow \exists x(\neg A(x))$$

$$\neg(\exists x A(x)) \Leftrightarrow \forall x(\neg A(x))$$

"对偶原理"

③ 量词分配

$$\forall x(A(x) \wedge B(x)) \Leftrightarrow \forall x A(x) \wedge \forall x B(x)$$

$$\exists x(A(x) \vee B(x)) \Leftrightarrow \exists x A(x) \vee \exists x B(x).$$

④ 量词顺序变换

$$\forall x \forall y(A(x,y)) \Leftrightarrow \forall y \forall x(A(x,y))$$

$\exists$

三. 等值演算(calculus)

评论: 这些应该在上学期就学习了

好处: 很方便的否定一些复杂的式子

- 线性相关(任意…存在不…)
- 极限的定义(任意…存在…)

三. 等值演算(calculus)

3、一些推导式

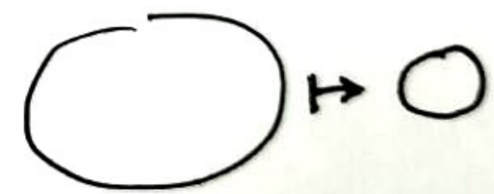
$\forall x.\alpha \to \exists x.\alpha$

$\exists x \forall y.\alpha \to \forall y \exists x.\alpha$

$\forall y \exists x.\alpha \not\to \exists x \forall y.\alpha$

$\forall x \alpha \vee \forall x \beta \to \forall x(\alpha \vee \beta)$

$\exists(\alpha \wedge \beta) \to \exists x \alpha \wedge \exists x \beta$



反例：$U=\{a, b\}$

关系：$P(a, b)$，$P(b, a)$

$\forall y \exists x P(y, x) \equiv T$

$\exists x \forall y P(y, x) \equiv F$.

四. 范式

Eg. $\forall x \forall y\, (\exists z (P(x,z) \wedge P(y,z)) \to \exists z.\, Q(x,y,z))$

$\Leftrightarrow \forall x \forall y (\neg \exists z (P(x,z) \wedge P(y,z)) \vee \exists z\, Q(x,y,z))$

$\Leftrightarrow \forall x \forall y\, (\forall z\, (\neg P(x,y) \vee \neg P(y,z)) \vee \exists z\, Q(x,y,z))$

$\Leftrightarrow$

$u \qquad u$

$\Leftrightarrow \forall x \forall y \forall z \exists u\, (\qquad\qquad)$.

只是一个例子… 不用太在意.

其实有时候也可以把它写成分数线的形式:

$$\frac{\text{前提的条件1, 前提的条件2,...}}{\text{得到的结论}}\text{(使用的推理规则)}$$

把它横着写, 就有

{前提的条件1, 前提的条件2,...} ⊢ 得到的结论 (使用的推理规则)

五. 形式化的推理理论

把这些搞定了之后就转化为了命题逻辑里面的东西了

1、 $\forall$-elim　　$\forall x.\alpha \vdash \alpha[t/x]$.　($t$ is free for x in α)

$\forall x. P(x) \vdash P(c)$　(c为常元)

$\forall x. \exists y. (x<y) \vdash \exists y. (c<y)$

$\forall x. \exists y. (x<y) \vdash \exists y. (z<y)$　($z \neq y$ 是任意变元符号)

$\forall x. \exists y. (x<y) \nvdash \exists y. (y<y)$.

$$\frac{\forall x. (H(x) \to M(x)) \qquad H(s)}{M(s)}$$

2、 $\forall$-intro

$$\frac{\begin{array}{c}[t] \\ \vdots \\ \alpha[t/x]\end{array}}{\forall x. \alpha}$$

引入变量t

得到的表达式里面所有的x都是t

然后就可以推出来这个

任取t，都证得α对t成立，那么α对x成立.

五. 形式化的推理理论

任意推理规则的应用

$$\{P(t), \forall x(P(x) \rightarrow \neg Q(x)\} \vdash \neg Q(t)$$

这一步解决了就可以转化到第一节的内容了

$P(t)$	P
$\forall x(P(x) \rightarrow \neg Q(x))$	P
$P(t) \rightarrow \neg Q(t)$	Universal Specification
$\neg Q(t)$	T

标号有点麻烦...就先不标示了...

五. 形式化的推理理论

任意推理规则的应用

$$\{\forall x(P(x) \to Q(x)), \forall x\,.\,P(x)\} \vdash \forall x\,.\,Q(x)$$

$\forall x\,.\,(P(x) \to Q(x))$	(P)
$\forall x\,.\,P(x)$	(P)
$\left[x_0\right]$	(引入变量)
$P\left(x_0\right) \to Q\left(x_0\right)$	($\forall$-elim , (1), (3))
$P\left(x_0\right)$	($\forall$-elim, , (2), (3))
$Q\left(x_0\right)$	($\to$ -elim , (4), (5))
$\forall x\,.\,Q(x)$	($\forall$-intro , (3) − (6))

五. 形式化的推理理论

3. $\exists$ – intro: $\dfrac{\alpha[t/x]}{\exists x\,.\,\alpha}$ $(\exists x - \text{intro})$

Where t is free for x in α

"如果 α 对某个项 t 成立, 则 ∃x. α 成立."

例如: $P(c) \vdash \exists x\,.\,P(x)$, c是任意的常元符号

但是 $\forall y\,.\,(y = y) \nvdash \exists x\,.\,\forall y\,.\,(x = y)$(y is not free for x in α)

五. 形式化的推理理论

4. $\exists$ – elim: $\dfrac{\exists x\,.\,\alpha}{\alpha[x_0/x]}$ $(\exists x - \text{elim})$

Where t is free for x in α

就是上面的反过来的过程.

5. $\exists$ – elim: $\dfrac{\exists x\,.\,\alpha \qquad [x_0] \qquad \begin{array}{c}[\alpha[x_0/x]]\\ \vdots \\ \beta\end{array}}{\beta}$ $(\exists x - \text{elim})$

Where x_0 is free for x in α

假设x_0使得α成立, 如果从$\alpha[x_0/x]$可以推导出β, 那么从$\exists x\,.\,\alpha$可以推导出β.

五. 形式化的推理理论

存在推理规则的应用

$$\{\forall x\,.\,P(x)\} \vdash \exists x\,.\,P(x)$$

五. 形式化的推理理论

Level up!

$$\{\forall x.(P(x) \to Q(x)), \exists x.P(x)\} \vdash \exists x.Q(x)$$

$\forall x.(P(x) \to Q(x))$	（前提）
$\exists x.P(x)$	(前提）
$[x_0] \quad [P(x_0)]$	（引入变量与假设）
$P(x_0) \to Q(x_0)$	(∀-elim, $(1),(3)$)
$Q(x_0)$	($\to$ -elim, $(3),(4)$)
$\exists x.Q(x)$	(∃-intro , (5))
$\exists x.Q(x)$	(∃-elim, $(2),(3)-(6)$)

五. 形式化的推理理论

Level up!

$$\{\exists x\,.\,P(x), \forall x\,.\,\forall y\,.\,(P(x) \to Q(y)\} \vdash \forall y\,.\,Q(y)$$

$\exists x\,.\,P(x)$	（前提）
$\forall x\,.\,\forall y\,.\,(P(x) \to Q(y))$	（前提）
$\left[x_0\right] \quad \left[P\left(x_0\right)\right]$	（引入变量与假设）
$\forall y\,.\,\left(P\left(x_0\right) \to Q(y)\right)$	($\forall$-elim, (2), (3))
$\left[y_0\right]$	（引入变量）
$P\left(x_0\right) \to Q\left(y_0\right)$	($\forall$-elim, (4), (5))
$Q\left(y_0\right)$	($\to$ -elim, (3),(6))
$Q\left(y_0\right)$	($\exists$-elim, (1), (3) – (7))
$\forall y\,.\,Q(y)$	($\forall$-intro, (5) – (8))

五. 形式化的推理理论

一个应用问题

前提:

- 每个人或者喜欢美剧, 或者喜欢韩剧(可以同时都喜欢)
- 任何人如果喜欢抗日神剧, 他就不会喜欢美剧
- 有的人不喜欢韩剧

结论: 有的人不喜欢抗日神剧

这是一个有点难度的问题…

五. 形式化的推理理论

一个应用问题

前提:

- 每个人或者喜欢美剧, 或者喜欢韩剧(可以同时都喜欢)
- 任何人如果喜欢抗日神剧, 他就不会喜欢美剧
- 有的人不喜欢韩剧

结论: 有的人不喜欢抗日神剧

这是一个有点难度的问题...

$$\frac{\forall x.\ A(x) \vee K(x) \qquad \forall x.\ J(x) \to \neg A(x) \qquad \exists x.\ \neg K(x)}{\exists x.\ \neg J(x)}$$

五. 形式化的推理理论

一个应用问题

$$\frac{\forall x.\ A(x) \vee K(x) \qquad \forall x.\ J(x) \rightarrow \neg A(x) \qquad \exists x.\ \neg K(x)}{\exists x.\ \neg J(x)}$$

$$\begin{array}{rll}
\forall x.\ A(x) \vee K(x) & (\text{前提}) & (1) \\
\forall x.\ J(x) \rightarrow \neg A(x) & (\text{前提}) & (2) \\
\exists x.\ \neg K(x) & (\text{前提}) & (3) \\
{\color{red}[x_0] \quad [\neg K(x_0)]} & (\text{引入变量与假设}) & (4) \\
A(x_0) \vee K(x_0) & (\forall\text{-elim}, (1), (4)) & (5) \\
A(x_0) & ((4), (5)) & (6) \\
J(x_0) \rightarrow \neg A(x_0) & (\forall\text{-elim}, (2), (4)) & (7) \\
\neg J(x_0) & ((6), (7)) & (8) \\
{\color{red}\exists x.\ \neg J(x)} & (\exists\text{-intro}, (8)) & (9) \\
\exists x.\ \neg J(x) & (\exists\text{-elim}, (3) - (8)) & (10)
\end{array}$$

五. 形式化的推理理论

一个应用问题

$$\frac{\forall x.\ A(x) \lor K(x) \qquad \forall x.\ J(x) \to \neg A(x) \qquad \exists x.\ \neg K(x)}{\exists x.\ \neg J(x)}$$

我们还可以分类讨论!

$$\forall x.\ A(x) \lor K(x) \qquad (1)$$

$$\forall x.\ J(x) \to \neg A(x) \qquad (2)$$

$$\exists x.\ \neg K(x) \qquad (3)$$

根据 (3), 不妨设 $\neg K(x)$ 对 x_0 成立:

$$\neg K(x_0) \qquad (4)$$

根据 (1), 有

$$A(x_0) \lor K(x_0) \qquad (5)$$

根据 (4) 与 (5), 有

$$A(x_0) \qquad (6)$$

根据 (2), 有

$$J(x_0) \to \neg A(x_0) \qquad (7)$$

根据 (6) 与 (7), 有

$$\neg J(x_0) \qquad (8)$$

因此, $\exists x.\ \neg J(x)$

总结: 学到了什么

参考的课件与书本

1. Maths for Computer Science
2. 魏恒峰《离散数学》(2020) 谓词逻辑
3. 薛思清《离散数学》(2022) 谓词逻辑(1)(2)(3)

Thank You!

Your opinon
Matters

QQ: 2095728218
Email: micoael@qq.com
(学校) gwzhang@cug.edu.cn