关于描述计算机科学数学对象的形式系统的构造

侯广坤
(计算机科学系)

关于对计算机科学对象进行形式描述，以建立计算机科学的形式理论，是理论计算机科学中一个重要的问题。本文的宗旨是：在计算机科学的对象集合的总体中，排除不可判定的部分，从而构成计算机科学数学对象的形式系统，此系统用S来表示。它是满足一定条件的受约字公理数论系统的子类。

定义1 D为计算机科学的对象集合，如果存在着算法在有穷步内判定D中的对象是否具有某些性质$\Gamma_1,\Gamma_2,\cdots,\Gamma_k$和它们之间是否具有某些关系$R_1,R_2,\cdots,R_r$则$D$称为计算机科学数学对象集合。以下简称为对象集合。

从能行性理论中[2,3]看到，如果D为某抽象机所接受，那么它由所有抽象机所接受[4]。因此某集合是否是对象集合，它只与$\Gamma_1,\Gamma_2,\cdots,\Gamma_k$，$R_1,R_2,\cdots,R_r$有关。

在谓词演算中可知，性质和关系可以由公式来确定。因此具有使用公理方法的基础。

计算机科学数学对象的公理方法直观地理解为：在应用这个方法时不具体指出某对象集合；也不给出具体对象的具体性质和对象之间的具体关系，而只指出表示性质和关系的术语，往往用符号来表示；同时给出一系列的断言，从一切可能的对象系统中区分出满足这些断言的那部分对象系统来。

由公理方法所建立起来的逻辑体系，并不是只讨论一个对象集合。虽然表示性质和关系的术语在不同的集合中有着不同的意义，但它们都满足共同的断言系统。

定义2 若对于给出的断言系统存在着解释，即存在着具体的对象集合，其中定义具体的计算机科学中的性质和关系，使断言系统为真，则此断言系统称为计算机科学的公理系统。

是否存在没有解释的公理系统呢？事实上，内在矛盾的公理系统是不存在解释的。本文只讨论S的构造。

设$\Sigma=\{\sigma_1,\sigma_2,\cdots,\sigma_t\}$为有穷非空字符表，$\Sigma^*$是包含所有$\Sigma$上的字的集合。

定义3 任意$A\subseteq\Sigma^*$，称为语言。

引理 任意对象集合，都存在着某字符表上的语言与它成一一对应关系。

根据引理，S是以Σ^*作为基本构形的，满足一定性质的字谓词演算作为基础。

定义4 扩充字谓词演算是以一般狭义字谓词演算加进字函数概念；及相等公理

Ⅵ　(1) $x=x$;

(2) $x=y\rightarrow(A(x)\rightarrow A(y))$。

与原有狭义字谓词演算的五组公理[6]，所构成的公理系统；推理规则是字谓词演算的推理规则，对于代入规则作如下调整：

(a)设**A**是扩充字谓词演算的可推出公式，如果将包含在**A**中的自由变量x，用只包括不包含在**A**中作为约束变量的那些变量的项代替，所得到的公式也是扩充字谓词演算的可推出公式。

(b)在字谓词演算中的谓词变元的代入规则 $R^{B(t_1,\cdots,t_n)}_{F(\cdots)}(\mathbf{A})$在扩充字谓词演算中有如下意义：包含在**A**中的每一个形如$F(x_1,x_2,\cdots,x_n)$的表达式，用公式$\mathbf{B}(x_1,x_2\cdots,x_n)$代替，其中$F(\cdots)$是扩充字谓词演算中的谓词变元，$x_1,x_2\cdots,x_n$是它的项，公式$\mathbf{B}(x_1,x_2,\cdots,x_n)$是公式$\mathbf{B}(t_1,t_2,\cdots,t_n)$用相应的项$x_1,x_2,\cdots,x_n$代替变量$t_1,t_2,\cdots,t_n$而得到的公式。那么若**A**是扩充字谓词演算的可推出公式，则 $R^{B(t_1,\cdots t_n)}_{F(\cdots)}(\mathbf{A})$也是其中可推出公式，可知扩充字谓词演算的推理规则只是将狭义字谓词演算的变量的代换扩充为项的代换。

定义5　(1)　$\lambda_0(x_1,x_2,\cdots,x_n)=\Lambda$。

(2)　$\lambda_i(x)$由如下递归定义，

(a)　$\lambda_i(\Lambda)=\sigma_i$　$(i=1,2,\cdots,t)$

(b)　若$x\in\Sigma^*$，则$\lambda_i(x)=x\sigma_i$，　$i=1,2,\cdots,t$　〔7〕。

定理1　设D为Σ上的语言，则对于D中的任一元素a，都存在着λ—函数序列

$\lambda_{i1}(x)$，$\lambda_{i2}(x)$，$\cdots$，$\lambda_{is}(x)$，$i_j\in\{1,2,\cdots,t\}$，$j=1,2,\cdots,s$，。

使得　$a=\lambda_{is}(\lambda_{is-1}(\cdots(\lambda_{i1}(\Lambda))\cdots))$。

证明按字a的长施归纳法即可。

推论1　对于Σ^*中的任意字a，都可以根据λ—函数的性质列成一个由Λ开始到字a结束的一个链，并且在链中的节点对应着Σ^*中的一个字；如果在链中某节点的字δ_1前于字δ_2，则δ_1是δ_2中的一部分。

推论2　在Σ^*中的所有字可以与附图中的各顶点成一一对应关系：Λ为向下生长顶点，每一个顶点向下生长t个分支。若b是a的后继，则$b=\lambda_i(a)$，$i\in\{1,2,\cdots,t\}$。如附图所示。

定义6　若字$a,b\in\Sigma^*$位于附图结构中的同一分支，则它们称为可比较的。

在Σ^*中的编序关系，可以用扩充谓词演算的公式来确定，这些公式称为编序公理

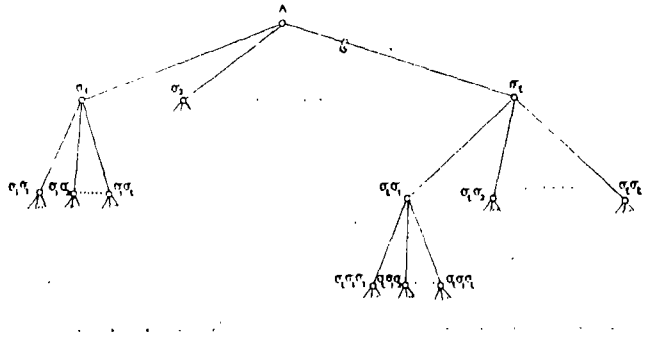
Ⅶ　(1)　$\overline{x<x}$，

(2)　$x<y\longrightarrow(y<z\longrightarrow x<z)$，

(3)　$x<\lambda_i(x)$，$i=1,2,\cdots\cdots,t$。

从扩充字谓词演算的基础上建立受约字公理算术和字公理算术，除将扩充字谓词演算的某些不可推出公式加进公理系统中外，后者只讨论包括在前者中的一部分对象。为

此我们建立具有一些特殊性质的对象。



集Σ^*的结构

在定义5中我们定义了两个函数(1),(2),连同函数(3) $I^n_m(x_1,x_2,\cdots,x_n)=x_m$, $(1\leqslant m\leqslant n)$。统称为基本函数。

定义7 (1) 任意基本函数是递归函数。

(2) 设$h(y_1,y_2,\cdots,y_m)$和$g_1(x_1,x_2,\cdots,x_n),\cdots,g_m(x_1,x_2,\cdots,x_n)$是字递归函数,则

$$f(x_1,x_2,\cdots,x_n)=h(g_1(x_1,x_2,\cdots,x_n),\cdots,g_m(x_1,x_2,\cdots,x_n))$$

也是字递归函数。

(3)设$g(x_1,x_2,\cdots,x_n)$和$h_i(x_1,x_2,\cdots x_n,y,z)(i=1,2,\cdots t)$是字递归函数,则由如下(递归)等式确定的函数$f_i(x_1,x_2,\cdots x_n,y)$也是递归函数。

$$\begin{cases} f_i(x_1,x_2,\cdots,x_n,\Lambda)=g(x_1,x_2,\cdots,x_n), \\ f_i(x_1,x_2,\cdots,x_n,\lambda_i(y))=h_i(x_1,x_2,\cdots,x_n,y,f_i(x_1,\cdots x_n,y)), \quad i=1,2,\cdots,t. \end{cases}$$

(4) 令$\sigma_i^m=\overbrace{\sigma_i\sigma_i\cdots\sigma_i}^{m}$。$\mu\sigma_i^m(h(x,\sigma_i^m)=\Lambda)$是如下定义的函数《它依赖于$x$接受$\sigma_i^m$,使得$h(x,\sigma_i^m)=\Lambda$的函数,并且对于任意$s<m$,$h(x,\sigma_i^s)$有定义且不等于$\Lambda$》。若$h(x,y)$是递归函数,则$f_i(x)=\mu\sigma_i^m(h(x,\sigma_i^m)=\Lambda)(i=1,2,\cdots,t)$也是部分递归函数。若

$$f_i(x)=\begin{cases} \sigma_i^m,\text{若存在着}m,\text{使}h(x,\sigma_i^m)=\Lambda, \\ \Lambda,\text{否则}, \end{cases}$$

则$f_i(x)$是全递归函数,今后只讨论后者。

定义8 (1)变量符号和常量符号是字公理算术的递归项;

(2)若$f(x_1,x_2,\cdots,x_n)$为递归函数,$t_1,t_2,\cdots,t_n$为递归项,则$f(t_1,t_2,\cdots,t_n)$也是字公理算术的递归项。

定义9 在扩充字谓词演算的基础上,只考虑递归项,在扩充字谓词演算的公理系统中加进编序公理,保持原有扩充字谓词演算的推理规则,所构成的形式系统称为受约字公理算术。

Σ^*上的可比较性,用一谓词$Q(x,y)$来表示《$Q(x,y)$为T,当且仅当x,y是比较的》。

定义10 如果在受约字公理算术中加进完全归纳公理 $Q(x,y)\&A(\Lambda)\&(\forall x)(A(x)\rightarrow A(\lambda_i(x)))\rightarrow A(y)$所构成的形式系统称为字公理算术系统或字公理数论系统。

用G来表示受约字公理算术，用G'来表示字公理算术。

定义11 设$F(x_1,x_2,\cdots,x_n)$为定义在语言$D\subseteq\Sigma^*$上的n元字谓词。对于谓词$F(x_1,x_2,\cdots,x_n)$建立函数$f_F(x_1,x_2,\cdots,x_n)$它定义在语言D上，对于所有使得$F(x_1^\circ,x_2^\circ,\cdots,x_n^\circ)=T$的$x_1^\circ,x_2^\circ,\cdots,x_n^\circ$上定义$f_F=\sigma_1$；对于所有使得$F(x_1',x_2',\cdots,x_n')=F$的$x_1',x_2',\cdots,x_n'$上定义$f_F=\Lambda$。$f(x_1,x_2,\cdots,x_n)$称为谓词$F(x_1,x_2,\cdots,x_n)$的特征函数。特征函数$f_F(x_1,x_2,\cdots,x_n)$是递归函数的字谓词$F(x_1,x_2,\cdots,x_n)$称为递归谓词。

定义12 S是只包括递归谓词的受约字公理算术系统。

定理2 设$P(x),Q(x),R(x)$是递归谓词，则谓词$P(x)\vee Q(x)$，$P(x)\&Q(x)$，$P(x)\rightarrow Q(x)$，$P(x)\sim Q(x)$，$\overline{R}(x)$也是递归谓词。

证明 只需要证明$\overline{R}(x)$和$P(x)\vee Q(x)$是递归谓词即可。

看$\overline{R}(x)$的特征函数：《若$NULL(f_{R(x)}(x))$则σ_1，否则Λ》[7]。根据定理条件$R(x)$是递归谓词则它的特征函数$f_{R(x)}(x)$是递归函数，然后根据定义7(2)，得到$f_{R(x)}(x)$也是递归函数，因而$\overline{R}(x)$是递归谓词。

同理可证$P(x)\vee Q(x)$的特征函数：《若$NULL(APPEND(f_{P(x)}(x),f_{Q(x)}(x))$则$\Lambda$，否则$\sigma_1$》[7]是递归函数，因而$P(x)\vee Q(x)$是递归谓词。〔证毕〕

推论 设$P(x)$为递归谓词，则任意受约量词：$(\underset{<a}{\exists}x)P(x)$和$(\underset{<a}{\forall}x)P(x)$也是归递谓词。

定义13 设τ为字公理算术G'的递归项，若它只包含常量符号而不包含变量符号，则称τ递归为常量。或者τ是任意递归项中的变量符号用Σ^*中的字代替所得到的项。

定理3 设h为任意递归项，而h^*是在h中的所有变量用Σ^*中的字代入所得到的递归常量，则存在着一个字$k\in\Sigma^*$，使公式$h^*=k$在G中是可推出的。

证明 对于递归项的定义施归纳法。

对于任意字常量和Λ，定理显然是正确的，因为字常量和Λ就是递归常量，并且所找到的字就是它本身。根据公理Ⅵ-1 $\vdash x=x$，用任意字常量或Λ代入x，得到$\vdash a=a$或$\vdash\Lambda=\Lambda$。

若h是变量x，用Σ^*中的任意字a代x所得到的递归常量是a，那么所找到的字是a本身，使$\vdash a=a$。

假设$f(x_1,x_2,\cdots,x_n)$为递归函数，$t_1,t_2,\cdots,t_n$为递归项。若对于递归项$t_1,t_2,\cdots,t_n$定理正确，则对于归递项$f(t_1,t_2,\cdots,t_n)$定理也正确。这里仍按$f(x_1,x_2,\cdots,x_n)$的构造施归纳法。

设$f(x_1,x_2,\cdots,x_n)$是形如$\lambda_0(x_1,x_2,\cdots,x_n)=\Lambda$，$\lambda_i(x)=x\sigma_i$和$I_m^n(x_1,x_2,\cdots,x_n)=x_m$的递归函数。对于$\lambda_0$和$I_m^n$在上面事实已证明了。对于$\lambda_i(x)$，令$t$为递归项，且满足定理条件，根据归纳假设定理正确，则用Σ^*中字代入t所得的递归常量t^*，存在着$k^*\in\Sigma^*$，使$\vdash t^*=k^*$。根据公理Ⅵ-2，先应用项的代入规则，有$\vdash t^*=k^*\rightarrow\lambda_i(t^*)=\lambda_i(k^*)$，又应用推断

规则有$\vdash\lambda_i(i^*)=\lambda_i(k^*)$。又$\lambda_i(k^*)$根据定义为$k^*\sigma i\in\Sigma^*$，所以$\vdash\lambda i(t^*)=k^*\sigma i$。

设递归函数$f_i(x_1,x_2,\cdots,x_n,y)$是由如下递归等式得到，设$g(x_1,x_2,\cdots,x_n)$和$h_i(x_1,x_2,\cdots,x_n,y,z)$为递归函数，且满足定理条件使定理成立，则

$$f_i(x_1,x_2,\cdots,x_n,y)$$

使得 $$f_i(x_1,x_2,\cdots,x_n,\Lambda)=g(x_1,x_2,\cdots,x_n)$$

$$f_i(x_1,x_2,,\cdots,x_n,\lambda_i(y))=h_i(x_1,x_2,\cdots,x_n,y,f_i(x_1,x_2,\cdots,x_n,y))。$$

如果$t_1,t_2,\cdots,t_n,s$递归项使定理成立，则递归项$f_i(i_1,t_2,\cdots,t_n,s)$也使定理成立。

设$t_1,t_2,\cdots,t_n,s$定理成立。则它在Σ^*中找到$t_1^*,t_2^*,\cdots,t_n,s^*$，将这些字代入$f_i(x_1,x_2,\cdots,x_n,y)$得到一个字递归常量的序列：

$$f_i(t_1^*,t_2^*,\cdots,t_n^*,\Lambda),f_i(t_1^*,t_2^*,\cdots,t_n^*,\lambda_i(\Lambda))\cdots\cdots$$

$f_i(t_1^*,t_2^*,\cdots,t_n^*,\lambda_i(\lambda_i(\cdots\lambda_i(\Lambda)\cdots))$。往证对于上述的序列中的每一个成员都存在$k_m^*$。$\lambda_i^{(m)}(x)$表示$\lambda_i(\lambda_i(\cdots(\lambda_i(\Lambda)\cdots))$。关于这一点我们仍用归纳法证明。

对于$\lambda_i^{(0)}$时，则$f_i(t_1^*,t_2^*,\cdots,t_n^*,\Lambda)=g(t_1^*,t_2^*,\cdots,t_n^*)$.根据归纳假设$f_i(t_1^*,t_2^*,\cdots,t^*,\Lambda)=g(t_1^*,t_2^*,\cdots,t_n^*)=k_0^*$。则有 $\vdash f_i(t_1^*,t_2^*,\cdots,t_n^*,\Lambda)=k_0^*$

设对于任意m_1都存在着k_m^*，使得

$$\vdash f_i(t_1^*,t_2^*,\cdots,t_n^*,\lambda_i^{(m)}(\Lambda))=k_m^*$$

根据前面的归纳假设，对于$h_i(t_1,t_2,\cdots,t_n^*,\lambda_i^{(m)}(\Lambda),k_m^*)=k_{m+1}^*$，又$f_i(t_1^*,t_2^*,\cdots,t_n^*,\lambda_i^{(m+1)}(\Lambda))=f_i(t_1^*,t_2^*,\cdots,t_n^*,\lambda_i(\lambda_i^{(m)}(\Lambda)))=h_i(t_1^*,t_2^*,\cdots,t_n^*,\lambda_i^{(m)}(\Lambda),f_i(t_1^*,t_2^*,\cdots,t_n^*,\lambda_i^{(m)}(\Lambda)))=h_i(t_1^*,t_2^*,\cdots,t_n^*,\lambda_i^{(m)}(\Lambda),k_m^*)=k_{m+1}^*$。

所以 $$\vdash f(t_1^*,t_2^*,\cdots,t_n^*,\lambda_i^{(m+1)}(\Lambda)=k_{m+1}^*。$$

对于形成递归函数的代入运算，只要连续地使用公理Ⅵ—2，及项的代入规则和推断规则即可得到。对于形成递归函数的μ—运算，分两种情况，一种是代入t^*时函数值为Λ的情况，另一种是代入t^*时函数值为σ_i^m的情况，这两种情况在前面已证过。〔证毕〕

推论 设τ_1和τ_2是递归项，则$\tau_1=\tau_2$和$\tau_1<\tau_2$是递归谓词。

定理4 G中的推理规则保持谓词的递归性。

证明可由定理2、3的推论得出。

在谓词演算中，任意谓词演算的公式都可以看成是一个谓词，在G中也是如此。

定义14 不成为原子的G中的谓词称为复杂谓词。

定义15 设$P(x,\cdots)$和$Q(x,\cdots)$为G中的谓词，若对于公式$P(x,\cdots)\sim Q(x,\cdots)$中的所有变量，用$\Sigma^*$中的字代入得到$G$的可推出公式，则$P(x,\cdots)$和$Q(x,\cdots)$称为等价的。

定义16 设$f(t_1,t_2,\cdots,t_n)$为递归项，所有与谓词$f(t_1,t_2,\cdots,t_n)=\Lambda$等价的谓词称为对应

于f的递归谓词。

这个定义同前面递归谓词的定义11是等价的。

定理5 设$P(x_1,x_2,\cdots,x_n)$为递归谓词，若对于$P(x_1,x_2,\cdots,x_n)$中的自由变量用Σ^*中的字$a_1,\cdots a_n$代入，则$P(a_1,a_2,\cdots,a_n)$或$P(a_1,a_2,\cdots,a_n)$在受约字公理算术中是可推出的。

证明 因为$P(x_1,x_2,\cdots,x_n)$是递归谓词，根据定义15 $\vdash P(a_1,a_2,\cdots,a_n)\sim f(a_1,a_2,\cdots,a_n)=\Lambda$。但根据定理3对于$f(a_1,a_2,\cdots,a_n)$存在着$k^*\in\Sigma^*$，使得$\vdash f(a_1,a_2,\cdots,a_n)=k^*$。$k^*$是$\Sigma^*$中的一个字。根据定理3的推论，这个字是否是$\Lambda$可以由$G$中的方法推出。因$k^*=\Lambda$，或$\overline{k^*=\Lambda}$，二者必居其一。用项的代入规则，得到$P(a_1,a_2,\cdots,a_n)=\Lambda$，或$P(a_1,a_2,\cdots,a_n)=\Lambda$在$G$中可推出。〔证毕〕

推论 根据定理4的结论，定理5的结论可以在S中得到满足。

应用上述所建立起来的系统S可以解决计算机科学的某些正确性判定问题。

参 考 文 献

〔1〕王浩，数理逻辑通俗讲话，科学出版社，(1980)。

〔2〕Kleene, *Introduction to Metamathematics*, D. Van Nostrand Company INC, New York, (1953).

〔3〕Малецев.А.И.,Алгоритмы и рекурсивные фрункции,Наука, Москва, (1965).

〔4〕侯广坤，关于Neumann自动机与Turing机相似问题，中山大学学报，(1981)，2。

〔5〕Martin Davis,(麦卓文译),计算是什么，计算机科学，(1981)，5。

〔6〕Л.С.Новинков, Элементы математической логики, Изд Наука, Москва, (1973).

〔7〕Eonar, Manna, Mathematical Theory of Computation, Mc Graw-Hill, (1974), 448.

〔8〕Blum, M., On Effective Procedures for Speeding up Algorithms, *J. ACM*, 18 (1971), 290-305.

〔9〕Luekham, D.C., Park, D.M.R.,Paterson, M.S., On for Nulised Computer Programs, *J. Computer and System Sci.*, 4(1970), 220-249.

〔10〕Grzegorczyk A., Some Classes of Recursive Functions, *Rorprawy Matemat.*, 4 (1953), 1-44.