Trace 演算*

黄 涛 钱 军 倪 彬

(中国科学院软件研究所计算机科学开放研究实验室 北京 100080)
(中国科学院软件研究所对象技术中心 北京 100080)

摘要 文章定义了基于踪迹(trace)的逻辑语言 $\mathscr{L}_{Trace}$,它是一阶线性时序逻辑语言的扩充,同时也是"对象演算"研究工作的基础. Trace 演算所述的"对象"用来刻画具有内部状态和外部行为的动态实体,语法上由对象标记表示. 对象标记 $\Omega=(S,F,A,E)$ 包含4个部分:数据类型 S、函数 F、属性 A 和动作 E. $\Sigma=(S,F)$ 构成通常代数规范意义下的标记,可将动作看成一个广义数据类型,从而得到标记 Σ 的动作扩充 ΣE. 对象标记的语义解释结构由关于标记 ΣE 的代数、映射和动作与踪迹的关系来定义. ΣE-代数给出关于数据参数的解释;映射给出属性在动作踪迹中所取的值;而动作与踪迹的关系则给出执行一有限踪迹以后该动作是否允许执行. 在定义了 Trace 演算的语法和语义之后,文章给出了 Trace 演算的公理系统及其可靠性证明.

关键词 Trace 演算,对象标记,动作,踪迹,公理化,可靠性.

中图法分类号 TP311

对象技术代表了80年代以后计算机软件技术发展的主流. 就目前状况而言,还是组合软件工程技术主宰了发展的潮流,而理论的研究滞后于工程技术的发展要求. 对象乃至对象系统的语义研究,虽然还不能对理论及工程技术的发展产生深远的影响和推动,但在澄清基本概念、增强系统可靠性等方面,其作用仍是不可替代的. 我们早在文献[1]中就已开始使用代数规范的方法研究对象行为约束. 而在文献[2,3]中,我们进一步给出对象语义模型及其指称描述. 本文将在文献[2,3]的基础上,结合代数规范[4,5]和时序逻辑方法[6],定义基于踪迹(trace)的逻辑语言 $\mathscr{L}_{Trace}$,它是一阶线性时序逻辑语言的扩充. 以这样的逻辑演算系统为工具,我们就可以构造一个用于描述对象和特性推理的逻辑演算系统.

当前,关于形式化的面向对象的语义模型的研究,主要有基于代数模型或基于时序模型这两种手段. 对象的代数语义模型理论从数学上讲是很完美的(如抽象数据类型),但缺点是:缺乏表达对象的动态行为特征的能力. 对象除了具有静态的结构特征外,还具有动态的语义特征. 在系统的运行过程中,对象可以被生成、修改、演变和消亡. 此外,用于刻画系统动态行为的约束关系大多与时序相关,仅用代数模型是难以表达的,很自然地需要用时序模型的观点来建立对象的语义理论.

应该指出,本文所述的对象是指一些具有内部状态和外部行为的动态实体,并非面向对象方法中的对象概念. 它没有考虑对象概念中的一项重要特性——封装性,因此,Trace 演算并不是一种对象演算. 但是,我们的目的是建立一个面向对象的逻辑演算系统,所以,我们这里的讨论都是针对对象的:我们给出对象标记的定义,提供动作运算符,可以方便地描述和推理动态实体的动态行为.

对象标记包含数据类型、函数、属性和动作4个部分. 数据类型和函数构成通常代数规范意义下的标记,给出与状态无关的信息,它是对象所处的数据上下文(或环境),也可以说是一个用于描述对象改变的框架;属性部

---

* 本文研究得到国家自然科学基金和国家863高科技项目基金资助. 作者黄涛,1965年生,博士,研究员,主要研究领域为软件工程,对象技术,分布计算,程序设计语言及环境. 钱军,1966年生,博士,主要研究领域为面向对象的理论和技术,分布对象计算,形式化方法. 倪彬,1959年生,博士,主要研究领域为面向对象的理论和技术,模型检查.

本文通讯联系人:黄涛,北京 100080,中国科学院软件研究所计算机科学开放研究实验室

本文1997-12-12收到原稿,1998-09-04收到修改稿

分包含对象状态相关信息,如同通常程序设计语言中程序变量等;动作部分给出该对象可以执行的动作.

值得指出是,对象标记仅给出相应对象的(行为)特征的(语法)描述,同其他对象的关系是通过指定一些相关动作的同步而建立的.

本文第1节给出 Trace 演算的语法.第2节给出 Trace 演算的语义.第3~5节分别讨论 Trace 演算的法则和公理化.最后给出总结.

1 Trace 语法

1.1 对象标记

定义 1. 对象标记 Ω 是一个四元组(S,F,A,E).

· S 是类型(sorts)的集合.

· F 是一函数簇,且存在一映射 $type:F\to S^*\times S$. 对任意 $f\in F$,$type(f)$称为 f 的类型. 常数可表示为零元函数符号.

· A 是以 $S^*\times S$ 标识的属性符号簇.

· E 是以 S^* 标识的动作符号簇.

其中 F,A 和 E 是有限集且不相交,S^* 代表零个或任意多个 S. 零元属性符号相应于通常程序设计语言中的程序变量,而非零元属性符号则用于处理复杂数据结构. 显然,S 和 F 构成通常代数规范意义下的标记 $\Sigma=(S,F)$. 第2节中我们给出函数和动作以严格的解释,即函数和动作的解释是状态无关的,而属性则相反,是状态相关的.

我们可以将动作看做广义数据,以 E' 表示这样的广义数据的集合. 对于动作符号簇 E 中的每个动作符号 $f:s_1\times\ldots\times s_n$,定义函数 $f':s_1\times\ldots\times s_n\to E'$ 来得到一个广义数据类型 E',其代数规范的标记为

Signature $E'=$

Sort E'

Functions

$f'_1:s_1\times\ldots\times s_{n_1}\to E'$　　/* $f_1:s_1\times\ldots\times s_{n_1}\in E$ */

$f'_2:s_1\times\ldots\times s_{n_2}\to E'$　　/* $f_2:s_1\times\ldots\times s_{n_2}\in E$ */

⋮

End

为方便以后的处理,我们用 E 表示 E',并以其表示广义数据类型 E'.

定义 2. 给定一对象标记 $\Omega=(S,F,A,E)$,ΣE-标记为 $\Sigma E=(SE,FE)$,这里,

· $SE=S\cup\{E\}$,即我们以一新类型 E 扩充 S.

· FE 是以 $S^*\times SE$ 标识的符号簇,对$\forall\ \omega\in S^*$,$s\in SE$,若 $s\in S$,则 $FE_{\omega,s}=F_{\omega,s}$,反之,$FE_{\omega,s}=E_\omega$,即我们以动作符号扩充函数簇.

1.2 项

令 X 是以 S 标识的变量簇,即 $X=\{X_s|s\in S\}$,且 X 与 A,F,E 不相交.

定义 3(项). 给定对象标记 Ω 和变量簇 X,对$\forall\ s\in SE$,项集 $T(\Omega,X)_s$ 归纳定义如下:

1. $\forall\ x\in X_s$,$x\in T(\Omega,X)_s$,
2. $Init\in T(\Omega,X)_E$,
3. $\forall\ f\in FE_{(s_1\times\ldots\times s_n),s}\cup A_{(s_1\times\ldots\times s_n),s}$,$t_i\in T(\Omega,X)_{s_i}$,$0\leqslant i\leqslant n$,则 $f(t_1,\ldots,t_n)\in T(\Omega,X)_s$,
4. $a\in T(\Omega,X)_E$,$t\in T(\Omega,X)_s$,则$[a]t\in T(\Omega,X)_s$,
5. $t\in T(\Omega,X)_s$,则 $\boldsymbol{X}t$,$\boldsymbol{X}^-t\in T(\Omega,X)_s$.

我们称 $T(\Omega,X)_E$ 中的项为动作项,$Init$ 为初始动作项.

1.3 公　式

定义 4(原子公式). 给定对象标记 Ω,变量簇 X,原子公式集 $AF(\Omega,X)$可归纳定义如下:

1. $t_1,t_2\in T(\Omega,X)_s, s\in SE$,则 $t_1=t_2\in AF(\Omega,X)$,

2. $a\in T(\Omega,X)_E$,则 $enabled(a)\in AF(\Omega,X)$.

定义 5(公式). 给定一个对象标记 Ω,变量簇 X,公式集 $F(\Omega,X)$可由以下规则归纳定义:

1. $p\in AF(\Omega,X)$,则 $p\in F(\Omega,X)$(每个原子公式是公式);

2. $a\in T(\Omega,X)_E, p\in F(\Omega,X)$,则$\bar{}a]p\in F(\Omega,X)$;

3. 通常时序运算:$p,q\in F(\Omega,X)$,则 $\boldsymbol{X}p, p\cup q, \boldsymbol{X}^{\bar{}}p, p\cup^{\bar{}}q\in F(\Omega,X)$;

4. 通常一阶运算:

(1) $p,q\in F(\Omega,X)$,则$\neg p, p\rightarrow q\in F(\Omega,X)$;

(2) $p\in F(\Omega,X), x\in X_s$,且 x 在 p 中自由出现,则$\forall x:s\ p\in F(\Omega,X)$,我们称 x 在$\forall x:s\ p\in F(\Omega,X)$约束出现.

另外,我们可按时序逻辑通常采用的方法引入以下等公式的缩写:$p\wedge q, p\vee q, p\leftrightarrow q, t, f, \boldsymbol{F}p, \boldsymbol{G}p, p\boldsymbol{B}q, \boldsymbol{F}^{\bar{}}p, \boldsymbol{G}^{\bar{}}p, \exists x:s\ p$.

在公式 p 中,变元 $x\in X_s$ 的出现若不是在$\forall x:s$ 或$\exists x:s$的范围中,则称 x 在 p 中自由出现;反之,则 x 在 p 中称约束出现.

如果 $t\in T(\Omega,X)_s, p\in F(\Omega,X)$,则 $p_{x:s}(t)$表示以 t 替换所有在 p 中自由出现的 x,且 $x\in X_s$. 我们假设 t 中不含在 p 中约束出现的自由变元.

值得指出的是,我们所涉及的项和变量均是属于某一特定的类型(sort)的. 为简明起见,在不引起混淆的情况下,我们略去附加的类型信息,如$\forall x:s\ p$ 有时我们简写为$\forall x\ p$.

2 Trace 语义

$\mathscr{L}_{\text{Trace}}$的语义可由扩充的一阶线性时序结构给出.

2.1 语义域

2.1.1 动作和踪迹

踪迹是动作的执行序列. 我们将动作扩充为广义数据类型,由 ΣE 的标记代数 $\mathscr{A}$ 中的 $\mathscr{A}_E$ 给出 E 中动作符号的解释,而 $\mathscr{A}_E^*$ 则给出动作符号序列的解释. 我们称 $\mathscr{A}_E$ 中的元素为动作,$\mathscr{A}_E^*$ 中的元素为踪迹. 为表示方便,我们引入一些有关踪迹的符号和运算.

定义 6.

- 空踪迹表示为 ε;
- 若 $\omega\in\mathscr{A}_E^*$,则 $\omega\cdot a\in\mathscr{A}_E$ 表示踪迹 ω 后串接一动作 a;
- $|\omega|$表示踪迹 ω 的长度;满足:$|\varepsilon|=0, |\omega\cdot a|=|\omega|+1$;
- 若 $1\leqslant i\leqslant|\omega|$,则 ω_i 表示踪迹 ω 中的第 i 个动作,满足:

$$(\omega\cdot a)_i=\omega_i, \quad \text{若 } 0\leqslant i\leqslant|\omega|,$$
$$(\omega\cdot a)_i=a, \quad \text{若 } i=|\omega|+1;$$

- 若 $0\leqslant i\leqslant|\omega|$,则$^i\omega$表示踪迹 ω 的长为 i 的前缀,满足:

$$^0\omega=\varepsilon,$$
$$^{|\omega|}(\omega\cdot a)=\omega.$$

通常我们省略运算符·.

2.1.2 Ω-语义解释结构

对象标记的语义解释结构由一个关于标记 ΣE 的代数、一个映射和动作与踪迹的关系来定义. ΣE-代数给出关于数据参数的解释;映射给出属性在动作踪迹中所取的值;而动作与踪迹的关系则给出执行一条有限踪迹以后该动作是否允许执行.

定义 7. 一个对象标记 $\Omega=(S,F,A,E)$的 Ω-语义解释结构是一个二元组 $\mathscr{I}=(\mathscr{A},\mathscr{T},\mathscr{E})$,其中

- $\mathscr{A}$ 是一 ΣE-代数;
- $\forall f\in\mathscr{A}_{(s_1\times\ldots\times s_n),s},n\geqslant 0$ 有:$\mathscr{F}(f):\mathscr{A}_E^*\to(\mathscr{A}_{s_1}\times\ldots\times\mathscr{A}_{s_n}\to\mathscr{A}_s)$;
- $\varepsilon\subset\mathscr{A}_E^*\times\mathscr{A}_E$.

这里,ΣE-代数是 ΣE-初始语义代数.ε 给出动作执行的前条件(我们以动作和踪迹的关系来表示,以指出在某一状态下该动作对在它执行以前所执行的动作序列的限制).

2.1.3　Trace-结构

定义 8. 给定一个对象标记 $\Omega=(S,F,A,E)$,一个 $Trace_\Omega$-结构 $\mathscr{X}=(\mathscr{A},v,\omega)$ 包含:

- 一个 Ω-语义解释结构 $\mathscr{A}=(\mathscr{A},\mathscr{F},\mathscr{E})$;
- 一个赋值函数簇 $v=\{v_s\}_{s\in SE},v_s:X_s\to\mathscr{A}_s$;
- 一个踪迹 ω.

注意,这里我们以踪迹 ω 替换通常时序结构中的状态序列.设 s_i 为第 i 个状态,a_i 为在 s_i 状态下执行的动作,该动作的执行将导致状态转换到 s_{i+1},因此,对 $S_0\xrightarrow{a_0}s_1,\ldots,s_i\xrightarrow{a_i}s_{i+1},\ldots$,我们可以踪迹 $a_0a_1\ldots$ 代替时序演算意义下的状态序列 $s_0s_1\ldots$.这样,对 $\mathscr{F},\mathscr{F}_i$ 给出在状态 s_i 下属性值映射函数,以下用到的 $v_i^*,\mathscr{X}_i$ 等均是在状态 s_i 意义下的.

2.2　项解释

我们可模仿经典逻辑的方法为每个项定义一个值.首先,我们定义有关变量的赋值.

定义 9(赋值函数). 给定一个对象标记 $\Omega=(S,F,A,E)$,Ω-语义解释结构 $\mathscr{A}=(\mathscr{A},\mathscr{F},\mathscr{E})$,则关于变量簇 X 的赋值函数簇记为:$v=\{v_s|s\in SE\}$,其中 v_s 是变量集 X_s 到语义域 $\mathscr{A}_s$ 的映射.

我们可如下定义赋值函数簇 v 在一踪迹 ω 上的自然扩充 $v_i^*,i\in N$:

- $\forall x\in X_s,v_i^*=v(x)$;
- $\forall f\in FE_{(s_1\times\ldots\times s_n),s},t_i\in T(\Omega,X)_{s_i},\forall i,1\leqslant i\leqslant n$,则 $v_i^*(f(t_1,\ldots,t_n))=f^{\mathscr{A}}(v_i^*(t_1),\ldots,v_i^*(t_n))$;
- $\forall f\in A_{(s_1\times\ldots\times s_n),s},t_i\in T(\Omega,X)_{s_i},\forall i,1\leqslant i\leqslant n$,则 $v_i^*(f(t_1,\ldots,t_n))=\mathscr{F}_i(f)(v_i^*(t_1),\ldots,v_i^*(t_n))$;
- $a\in T(\Omega,X)_E,a\neq Init,t\in T(\Omega,X)_s$,则 $v_i^*([a]t)=\begin{cases}v_{i-1}^*(t), & \text{若 } v_i^*(a)=\omega_{i+1}\\ v_i^*(t), & \text{否则}\end{cases}$;
- $t\in T(\Omega,X)_s$,则 $v_i^*([Init]t)=v_0^*t$;$v_i^*(Xt)=v_{i+1}^*t$;$v_i^*(X^-t)=v_{i-1}^*t,i>0$.

这里,我们并没有给初始动作项 $Init$ 的语义解释,对 $Init$,我们主要是用于运算符 $[Init]$.

现在,我们可以定义公式的真假值.

对任意一个 $Trace_\Omega$-结构,对每个原子公式 F,我们可如下定义 $\mathscr{X}_i(F)\in\{t,f\}$,表示公式 F 在状态 s_i 下的真假值:

1. $\mathscr{X}_i(t_1=t_2)=t$　　当且仅当 $v_i^*(t_1)=v_i^*(t_2)$;
2. $\mathscr{X}_i(enabled(a))=t$　　当且仅当$(^i\omega,v_i^*(a))\in\varepsilon$.

对任意一个 $Trace_\Omega$-结构,对每个公式 F,我们可如下归纳定义 $\mathscr{X}_i(F)\in\{t,f\}$,表示公式 F 在状态 s_i 下的真假值:

1. $\mathscr{X}_i(\neg p)=t$　　当且仅当 $\mathscr{X}_i(p)=f$;
2. $\mathscr{X}_i(p\to q)=t$　　当且仅当 $\mathscr{X}_i(p)=f$ 或 $K_i(q)=t$;
3. $\mathscr{X}_i([a]p)=t$　　当且仅当 $\begin{cases}\mathscr{X}_{i+1}(p)=t, & \text{若 } v_i^*(a)=w_{i+1}\\ \mathscr{X}_i(p)=t, & \text{否则}\end{cases}$;
4. $\mathscr{X}_i([Init]p)=t$　　当且仅当 $\mathscr{X}_0(p)=t$;
5. $\mathscr{X}_i(Xp)=t$　　当且仅当 $\mathscr{X}_{i+1}(p)=t$;
6. $\mathscr{X}_i(p\cup q)=t$　　当且仅当$\exists\ j\geqslant i(\mathscr{X}_j(q)=t$ 且$\forall\ k(i\leqslant k<j)\mathscr{X}_i(p)=t)$;
7. $\mathscr{X}_i(X^-p)=t$　　当且仅当 $i>0,\mathscr{X}_{i-1}(p)=t$;
8. $\mathscr{X}_i(p\cup^-q)=t$　　当且仅当$\exists\ j\leqslant i(\mathscr{X}_j(q)=t$ 且$\forall\ k(i<k\leqslant j)\mathscr{X}_i(p)=t)$;
9. $\mathscr{X}_i(\forall\ x\ p)=t$　　当且仅当 $\mathscr{X}'_i(q)=t$ 对任意 $\mathscr{X}'=(\mathscr{A},v,\omega)$ 满足:$v'(y)=v(y),\forall\ y,y\neq x$.

也就是说,$\mathscr{X}_i(p)$ 给出公式 p 在状态 s_i 下的真假值.

定义 10(模型和有效式). 给定一个公式 p 和一个 Trace$_n$-结构 $\mathscr{N}$,若对任意 $i\geqslant 0$,$\mathscr{N},(p)=t$,则称 p 对 $\mathscr{N}$ 是有效的(记为$\models_{\mathscr{N}}p$),称 $\mathscr{N}$ 为 p 的模型. 若对所有 Trace$_n$-结构 $\mathscr{N}$,$\models_{\mathscr{N}}p$,则称 p 为有效式(记为$\models p$). 给定一公式集 Γ,若对所有 Γ 中的公式的全称闭式 p,$\models_{\mathscr{N}}p$,则称 $\mathscr{N}$ 为 Γ 的模型.

定义 11. 若存在一个 Trace$_n$-结构 $\mathscr{N}$ 为公式集 Γ 中所有公式的模型,则称 Γ 为可满足的.

定义 12. 给定一公式集 Γ 和一公式 p,若对所有 Γ 的模型 $\mathscr{N}$ 有$\models_{\mathscr{N}}p$ 成立,则称 p 为 Γ 的语义推论,记为 $\Gamma\models p$.

定理 1. 若$\forall\ q\in\Gamma,\models q$ 且 $\Gamma\models p$,则$\models p$.

定理 2. $p_1,\ldots,p_n\models$当且仅当$(\models Gp_1\wedge\ldots\wedge Gp_n)\rightarrow q$.

定理 3. 若 $\Gamma\models p$,且 $\Gamma\models p\rightarrow q$,则 $\Gamma\models q$.

定理 4. 若 $\Gamma\models p$,则

1. $\Gamma\models[a]p$;

2. $\Gamma\models[Init]p$.

定理 5. $\models p$ 当且仅当$\{\neg p\}$不是可满足的.

3 Trace 法则

现在,我们给出一些 Trace 演算的逻辑法则. 首先我们可以证明,一阶时序演算的重言式在新的逻辑框架下仍然适用.

定义 13(时序重言式). 设 $x_1,\ldots,x_n$ 是一阶时序演算的原子公式,$p(x_1,\ldots,x_n)\in\mathscr{L}_{\mathrm{FLTLB}}$是一阶时序演算的重言式,则对任意的 $p_1,\ldots,p_n\in\mathscr{L}_{\mathrm{Trace}}$,$p(p_1,\ldots,p_n)$称为 Trace 演算的时序演算型重言式,或简称时序重言式. 这里,$p(p_1,\ldots,p_n)$是用 $p_1,\ldots,p_n$ 分别替换 $p(x_1,\ldots,x_n)$中的 $x_1,\ldots,x_n$ 所得结果.

定理 6. 时序重言式是有效式.

证明省略,详见文献[7]. 根据这一定理,我们可以将所有一阶时序演算的重言式用作 Trace 演算的"逻辑法则". 进一步地,我们可以证明如下断言.

若在一阶时序演算中,公式 q 是公式 $p_1,\ldots,p_n$ 的语义推论;则我们用 Trace 公式一致替换 $p_1,\ldots,p_n$ 和 q,将不会破坏逻辑关系.

定义 14. 令 $p_1,\ldots,p_n,q\in\mathscr{L}_{\mathrm{Trace}}$,若 $p_1\wedge\ldots\wedge p_n\rightarrow q$ 是时序重言式,则称 q 是 $p_1,\ldots,p_n$ 的时序重言推论.

定理 7. 若 q 是 $p_1,\ldots,p_n$ 的时序重言推论,则 $p_1,\ldots,p_n\models q$.

为节省篇幅,我们列出有关 Trace 运算符的逻辑法则,证明则从略.

对偶律

(T1) $\neg[a]p\leftrightarrow[a]\neg p$

同一律

(T2) $[Init][Init]p\leftrightarrow[Init]p$

分配律

(T3) $[a](p\rightarrow q)\leftrightarrow[a]p\rightarrow[a]q$

(T4) $[a](p\wedge q)\leftrightarrow[a]p\wedge[a]q$

(T5) $[a](p\vee q)\leftrightarrow[a]p\vee[a]q$

单调律

(T6) $p\rightarrow q\models[a]p\rightarrow[a]q$

吸收律

(T7) $[a][Init]p\leftrightarrow[Init]p$

(T8) $X[Init]p\leftrightarrow[Init]p$

(T9) $G[Init]p\leftrightarrow[Init]p$

(T10) $F[Init]p\leftrightarrow[Init]p$

(T11) $X^-[Init]p\leftrightarrow[Init]p$

(T12) $G^-[Init]p\leftrightarrow[Init]p$

(T13) $F^{-}[Init]p \leftrightarrow [Init]p$

(T14) $[Init]Gp \leftrightarrow (Gp \wedge G^{-}p)$

定理 8. 若 p 中不含属性符号

1. $p \leftrightarrow [Init]p$;

2. $p \leftrightarrow [a]p$.

定义 15. 一项 $t \in T(\Omega,X)_s$,变量 $x \in X_s$,如果以 t 替换公式 p 中的变量 x 不产生属性符号在时序运算符 $(X,F,\ldots)$ 和动作运算符($[Init]$,$[a]$)作用域内的新出现,且 t 中的变量不在替换出时受约束,则称 t 对于 p 中的 x 是可替换的.

可替换的要求来自于变量 x 是全局的,其解释是时序/踪迹无关的.

定理 9. 如果 $t \in T(\Omega,X)_s$ 对于 p 中的 $x \in X_s$ 是可替换的,则 $\models \forall\, x\ p \rightarrow p_x(t)$.

定理 10. 若 a 不在 p 中自由出现,则 $\models \forall\, a\ [a]p \rightarrow (Xp)$.

定理 11. $a \in T(\Omega,X)_E$,$x \in X_s$,若 x 不在 a 中自由出现,则 $\models \forall\, x [a]p \leftrightarrow [a](\forall\, x\ p)$.

定理 12. $\models \forall\, x [Init]p \leftrightarrow [Init](\forall\, x\ p)$.

若 $a \neq Init$,我们有:

(T15) $[a]p \rightarrow (p \vee Xp)$

(T16) $(p \wedge Xp) \rightarrow [a]p$

4 Trace 公理化

给出 Trace 演算的语义以后,现在,我们给出一个形式系统来进行公式推导.

公　理

(taut)　所有一阶线性时序逻辑的公理和重言式

(a1)　$p \rightarrow [a]p$　　若 p 中不含属性符号

(a2)　$[a][Init]p \leftrightarrow [Init]p$

(a3)　$X[Init]p \leftrightarrow [Init]p$

(a4)　$X^{-}[Init]p \leftrightarrow [Init]p$

(a5)　$\neg [a]p \leftrightarrow [a]\neg p$

(a6)　$[a](p \rightarrow q) \leftrightarrow [a]p \rightarrow [a]q$

(a7)　$\forall\, x\ p \rightarrow p_x(t)$　　若 t 对于 p 中的 x 是可替换的

(a8)　$[Init]Gp \leftrightarrow Gp \wedge G^{-}p$

(a9)　$\forall\, a [a]p \rightarrow Xp$　　若 a 不在 p 中自由出现

(a10)　$\forall\, x [a]p \rightarrow [a]\forall\, x\ p$　　若 x 不在 a 中自由出现

规　则

(mp)　$p, p \rightarrow q \vdash q$

(nex)　$p \vdash Xp$

(las)　$p \vdash X\ p$

(act)　$p \vdash [a]p$

(emp)　$p \vdash [Init]p$

(inv)　$[Init]p, p \rightarrow Xp \vdash p$

(ind)　$p \rightarrow q, p \rightarrow Xp \vdash p \rightarrow Gq$

(indp)　$p \rightarrow q, p \rightarrow X^{-}\ p \vdash p \rightarrow G^{-}q$

(gen)　$p \rightarrow q \vdash p \rightarrow \forall\, x\ q$　　若 p 中无 x 的自由出现

这里,没有讨论有关等词的公理和规则.

定理 13(**可靠性**). 令 p 为一公式,Γ 为一公式集,若 $\Gamma \vdash p$,则 $\Gamma \models p$.

证明:对从 Γ 证明 q 的推导过程进行归纳.

1. p 是 Trace 演算公理,由定理 6 和定理 8,法则(T7),(T8),(T11),(T1),(T3),定理 9,法则(T14),定理 10 和定理 12,分别可知公理(taut)和公理(a1)~(a10)均是有效式.因此,我们有 $\Gamma \models p$.

2. $p\in\Gamma$,显然 $\Gamma\models p$.

3. p 是由 q 和 $q\rightarrow p$ 经(mp)而得,则 $\Gamma\vdash q$,$\Gamma\vdash q\rightarrow p$. 由归纳假设 $\Gamma\models q$,$\Gamma\models q\rightarrow p$. 由定理 3 可得 $\Gamma\models p$.

4. $p\equiv Xq$ 是由 q 经(nex)而得,则 $\Gamma\vdash q$. 由归纳假设 $\Gamma\models q$. 令 $\mathscr{K}=(\mathscr{S},v,\omega)$为一个 Trace$_n$-结构,满足:$\models_{\mathscr{K}}r$,$r$ 为 Γ 中公式的一个全称闭式. 则$\models_{\mathscr{K}}q$,即对任意 $i\geqslant 0$,$\mathscr{K}_i(q)=t$,所以对任意 $j\geqslant 0$,$\mathscr{K}_{j+1}(q)=t$,即有 $\Gamma\models Xq$.

5. $p\equiv X^{-}q$ 是由 q 经(las)而得,则 $\Gamma\vdash q$. 由归纳假设 $\Gamma\models q$. 令 $\mathscr{K}=(\mathscr{S},v,\omega)$为一个 Trace$_n$-结构,满足:$\models_{\mathscr{K}}r$,$r$ 为 Γ 中公式的一个全称闭式. 则$\models_{\mathscr{K}}q$,即对任意 $i\geqslant 0$,$\mathscr{K}_i(q)=t$,所以对任意 $j>0$,$\mathscr{K}_{j-1}(q)=t$,即有 $\Gamma\models X^{-}q$.

6. $p\equiv[a]q$ 是由 q 经(act)而得,则 $\Gamma\vdash q$. 由归纳假设 $\Gamma\models q$. 由定理 4 可得 $\Gamma\models[a]q$.

7. $q\equiv[Init]r$ 是由 r 经(emp)而得,则 $\Gamma\vdash q$. 由归纳假设 $\Gamma\models q$. 由定理 4 可得 $\Gamma\models[Init]q$.

8. p 是由$[Init]p$ 和 $p\rightarrow Xp$ 经(inv)而得,则 $\Gamma\vdash[Init]p$,$\Gamma\vdash p\rightarrow Xp$. 由归纳假设 $\Gamma\models[Init]p$,$\Gamma\models p\rightarrow Xp$. 令 $\mathscr{K}=(\mathscr{S},v,\omega)$为一个 Trace$_n$-结构,满足:$\models_{\mathscr{K}}r$,$r$ 为 Γ 中公式的一个全称闭式. 则:

(1) $\models_{\mathscr{K}}[Init]p$;

(2) $\models_{\mathscr{K}}p\rightarrow Xp$.

由(1)可得,$\mathscr{K}_0(r)=t$,由(2)可得,$\mathscr{K}_0(p)=\mathscr{K}_2(q)=\ldots=\mathscr{K}_i(p)=\ldots=t$,即 $\Gamma\models p$.

9. $p\equiv q\rightarrow Gr$ 是由 $q\rightarrow r$ 和 $q\rightarrow Xq$ 经(ind)而得. 则 $\Gamma\vdash q\rightarrow r$,$\Gamma\vdash q\rightarrow Xq$. 由归纳假设 $\Gamma\models q\rightarrow r$,$\Gamma\models q\rightarrow Xq$. 令 $\mathscr{K}=(\mathscr{S},v,\omega)$为一个 Trace$_n$-结构,满足:$\models_{\mathscr{K}}r$,$r$ 为 Γ 中公式的一个全称闭式则:

(1) $\models_{\mathscr{K}}q\rightarrow r$;

(2) $\models_{\mathscr{K}}q\rightarrow Xq$.

对任意 i 且 $\mathscr{K}_i(q)=t$,由(2)可得 $\mathscr{K}_{i+1}(q)=\mathscr{K}_{i+2}(q)=\mathscr{K}_{i+3}(q)=\ldots=t$,即对任意 $j\geqslant i$,$\mathscr{K}_j(q)=t$. 由(1)可得 $\mathscr{K}_j(r)=t$,对任意 $j\geqslant i$. 因此我们有:$\mathscr{K}_i(q\rightarrow Gr)=t$,对任意 $i\geqslant 0$,即 $\Gamma\models p$.

10. $p\equiv q\rightarrow Gr$ 是由 $q\rightarrow r$ 和 $q\rightarrow X^{-}q$ 经(indp)而得. 与(ind)类似.

11. $p\equiv s\rightarrow\forall\ x\ r$,$x$ 不在 s 中的自由出现,是由 $s\rightarrow r$ 经(gen)而得. 则 $\Gamma\vdash s\rightarrow r$. 由归纳假设 $\Gamma\models s\rightarrow r$,令 $\mathscr{K}=(\mathscr{S},v,\omega)$为一个 Trace$_n$-结构,满足:$\models_{\mathscr{K}}q$,$q$ 为 Γ 中公式的一全称闭式. 则$\models_{\mathscr{K}}s\rightarrow r$. 假设对某个 i,$\mathscr{K}_i(s\rightarrow\forall\ x\ r)=f$,即 $\mathscr{K}_i(s)=t$ 且 $\mathscr{K}_i(\forall\ x\ r)==f$,则存在一个 $\mathscr{K}'$,使得 $\mathscr{K}'_i(r)=f$,且 $\mathscr{K}'$与 $\mathscr{K}$ 的区别仅在于对 x 的赋值不同. 这意味着对所有如上的闭式 q,有$\models_{\mathscr{K}}q$,因此$\models_{\mathscr{K}}s\rightarrow r$. 由于 s 中无 x 的自由出现,则有 $\mathscr{K}'(s)=t$,这样 $\mathscr{K}'(r)=t$,矛盾. 所以对任意 i,$\mathscr{K}_i(s\rightarrow\forall\ x\ r)=t$,如此 $\Gamma\models s\rightarrow\forall\ x\ r$. □

为简化证明,我们增加一规则来缩写经典一阶线性时序演算的推导步(仅使用一阶线性时序演算的公理和规则(mp),(nex),(ind)),即

(prep) $p_1,\ldots,p_n\vdash q$ 若 q 是 $p_1,\ldots,p_n$ 的一阶线性时序演算的语法推论

常用的 cut 规则即是(prep)的一个实例.

定理 14(演绎定理). $\Gamma\cup\{p\}\vdash q$,且证明中使用的 Gen 变元不在 p 中自由出现,则不增加新的 Gen 变元就可得 $\Gamma\vdash[Init]Gp\rightarrow q$.

我们还可弱化演绎定理的要求.

定理 15. $\Gamma\cup\{p\}\vdash q$,若证明过程中未使用 PLTLP 规则[5]和(emp),且证明中使用的 Gen 变元不在 p 中的自由出现则不增加新的 Gen 变元就可得 $\Gamma\vdash Gp\rightarrow q$.

推论 1. 若 p 是闭式,$\Gamma\cup\{p\}\vdash q$,则 $\Gamma\vdash[Init]Gp\rightarrow q$.

定理 16. 若 $\Gamma\vdash[Init]Gp\rightarrow q$,则 $\Gamma\cup\{p\}\vdash q$.

下面,我们给出一些使用 Trace 演算进行定理证明的例子.

(T17) $p\rightarrow Xp\vdash[Init]p\rightarrow p$

(T18) $Xp\rightarrow p\vdash p\rightarrow[Init]p$

证明:

(T17) (1) $[Init]G[Init]p\rightarrow p$ (inv),假设、演绎定理

(2) $[Init]p\rightarrow G[Init]p$　　(a3),(prep)
(3) $[Init]p\rightarrow[Init]G[Init]p$　　(2),(a2),(prep)
(4) $[Init]p\rightarrow p$　　(1),(2),cut
(T18) (1) $(Xp\rightarrow p)\rightarrow(\neg p\rightarrow\neg Xp)$　　prep
(2) $\neg p\rightarrow\neg Xp$　　假设,(1),mp
(3) $\neg Xp\rightarrow X\neg p$　　(prep)
(4) $\neg p\rightarrow X\neg p$　　(2),(3),cut
(5) $[Init]\neg p\rightarrow\neg p$　　(4),(T17)
(6) $\neg[Init]p\rightarrow[Init]\neg p$　　(a5)
(7) $\neg[Init]p\rightarrow\neg p$　　(5),(6),cut
(8) $(\neg[Init]p\rightarrow\neg p)\rightarrow(p\rightarrow[Init]p)$　　(prep)
(9) $p\rightarrow[Init]p$　　(7),(8),cut
(T19) $p\rightarrow\forall a[a]p\vdash Init]p\rightarrow p$
(T20) $\forall a[a]p\rightarrow p\vdash p\rightarrow[Init]p$　□
推论 2. $[Init]p,p\rightarrow\forall a[a]p\vdash p$.

5 Trace 公理化的进一步讨论

上一节的公理化讨论中未给出有关等词的公理化,我们将在这一节加以讨论.限于篇幅,我们只有将相关证明省略,详见文献[7].

对任意 $t_1,t_2\in T(\Omega,X)_s$,$a\in T(\Omega,X)_E$,有

(T21) $[a](t_1=t_2)\leftrightarrow([a]t_1=[a]t_2)$

对任意基函数 $f\in FE_{(s_1\times\ldots\times s_n),s}$,如下公式是有效式

(T22) $[a]f(t_1,\ldots,t_n)=f([a]t_1,\ldots,[a]t_n$

其中 $a\in T(\Omega,X)_E,t_i\in T(\Omega,X)_{s_i},\forall i,1\leqslant i\leqslant n$.

也就是说,"="、函数符号和动作符号的解释是严格的(rigid),即状态/踪迹无关的.

(T23) $[a][Init]t=[Init]t$
(T24) $X[Init]t=[Init]t$
(T25) $X^-[Init]t=[Init]t$
定理 17.
1. $u=v\models t_x(u)=t_x(v)$;
2. $u=v\models p_x(u)\rightarrow p_x(v)$.

定理 18. $x,y\in T(\Omega,X)_s$,若 p 中不含时序运算符和动作运算符,则$\models x=y\rightarrow(p\rightarrow p_x(y))$.

推论 3. $x,y\in T(\Omega,X)_s$,若 p 中 x 不在时序运算符和动作运算符作用域内出现,则$\models x=y\rightarrow(p\rightarrow p_x(y))$.

(T26) $u=v\rightarrow(t_x(u)=t_x(v))$
(T27) $u=v\rightarrow(p_x(u)\rightarrow p_x(v))$

其中 $u,v\in T(\Omega,X)_s,x\in X_s,t$ 为任一项,p 为任一公式,u,v 对于 p 中的 x 是可替换的.

定理 19. 若 a 不在 p 中自由出现,则$\models\forall a p_x([a]t)\rightarrow p_x(Xt)$.

其中 $x\in X_s,t\in T(\Omega,X)_s$,$p$ 中 x 不在时序运算符和动作运算符作用域内出现.

最后,我们得到如下公理.

公　理

(ae1) $[a]t=t$　　若 t 中不含属性符号
(ae2) $[a][Init]t=[Init]t$
(ae3) $X[Init]t=[Init]t$
(ae4) $X^-[Init]t=[Init]t$
(ae5) $[a](t_1=t_2)\leftrightarrow([a]t_1=[a]t_2)$
(ae6) $[a]f(t_1,\ldots,t_n)=f([a]t_1,\ldots,[a]t_n)$　　$f\in FE_{(s_1\times\ldots\times s_n),s}$

(eq1) $x=x$

(eq2) $x=y\rightarrow(p\rightarrow p_x(y))$ 若 p 中不含时序运算符和动作运算符

6 结 论

本文结合代数规范方法和一阶线性时序逻辑方法给出具有一般性的 Trace 演算.综合来说,Trace 演算具有如下特征.

(1) 对象标记是对象的语法界面的抽象.

(2) 统一时序模型和代数模型,以基于对象标记的 ΣE-代数为语义域,并建立时序逻辑在 ΣE-代数上的线性解释.

(3) 引入动作运算符并给出其在语义域上的解释.动作运算符的引入使我们可以方便地刻画动作的作用效果.

(4) 定义一系列公理和推导规则,建立一个可靠的 Trace 演算.它是一阶线性时序演算的扩充,并给出扩充部分与命题演算、一阶谓词演算以及时序演算之间的关系.

Trace 演算虽可描述和推理具有内部状态的动态实体及其行为特征.但我们的目标是建立一个对象演算系统.因此,如何在 Trace 演算的基础之上,进一步考虑对象所特有的性质,正是我们进一步的研究工作,这些将在本文的续篇,"对象演算(Ⅰ)(Ⅱ)"中加以讨论.

参考文献

1 冯玉琳,李京,黄涛.对象语义理论和行为约束推理.计算机学报,1993,16(11):889～897
(Feng Yu-lin, Li Jing, Huang Tao. Object semantics theory and behavior constraint deduction. Chinese Journal of Computers, 1993,16(11):889～897)

2 黄涛,冯玉琳,李京.对象形式语义模型.软件学报,1995,6(增刊):207～212
(Huang Tao, Feng Yu-lin, Li Jing. Model of object formal semantics. Journal of Software, 1995,6(supplement):207～212)

3 黄涛,冯玉琳,倪彬等.对象描述语言及其指称描述.软件学报,1996,7(10):577～586
(Huang Tao, Feng Yu-lin, Ni Bin *et al*. Object description language and its denotation description. Journal of Software, 1996,7(10):577～586)

4 Beeri C. Recent trends in data type specification. In: Astestiano E, Reggio G, Tarlecki A eds. Bulk Types and Query Language Design. LNCS 906, Berlin: Springer-Verlag, 1995

5 Ehrig H, Mahr B. Fundamentals of Algebraic Specifications 1: Equations and Initial Semantics. Berlin: Springer-Verlag, 1985

6 Kroger Fred. Temporal Logic of Programs. Berlin: Springer-Verlag, 1987

7 黄涛.对象形式语义理论研究[博士学位论文].合肥:中国科技大学,1994
(Huang Tao. Theoretical research on object formal semantics [Ph. D. Thesis]. Hefei: University of Science and Technology of China, 1994)

Trace Calculus

HUANG Tao QIAN Jun NI Bin

(*Laboratory of Computer Science Institute of Software The Chinese Academy of Sciences Beijing* 100080)

(*Object Technology Center Institute of Software The Chinese Academy of Sciences Beijing* 100080)

Abstract A trace-based logic language called $\mathcal{L}_{\text{Trace}}$ is defined in this paper, which is an extension of the first-order linear temporal logic and serves as cornerstone of the works——object calculus. The objects in trace calculus represent the dynamic entities endowed with a local state and external actions, and described by an object

signature in syntax. An object signature is a 4-tuple $\Omega=(S,F,A,E)$, in which S stands for a set of data sorts, F functions, A attributes and E actions. $\Sigma=(S,F)$ is nothing but a usual signature in the context of algebraic specification. It can be extended to ΣE with the action regarded as a special data sort. The semantics of trace calculus is defined by an object signature semantic interpreting structure $\mathscr{I}=(\mathscr{A},\mathscr{F},\mathscr{E})$, which consists of a ΣE-Algebra $\mathscr{A}$ giving an interpretation about data parameters, a mapping $\mathscr{F}$ evaluating the attributes on an action trace, and a relation $\mathscr{E}$ giving a relationship between actions and a trace. Finally, the authors contribute an axiom system of their trace calculus and outline its proof of soundness after defining the syntax and semantics of the trace calculus.

Key words Trace calculus, object signature, action, trace, axiomatization, soundness.