

一种不确定性语义计算模型

苗德成^{1,2}, 奚建清²

(1 韶关学院 数学与信息科学学院, 广东 韶关 512000; 2 华南理工大学 计算机科学与工程学院, 广东 广州 510000)

摘 要: 随着软件系统复杂度的持续增长, 如何保证大型复杂软件系统的健壮性与正确性逐渐成为一个热点问题, 不确定性语义计算的研究是解决这一问题的关键. 本文提出了一种不确定性语义计算模型, 并应用模型对示例小语言设计了四种不同的形式语义, 通过四种形式语义等价性的证明论证了模型的正确性与灵活性.

关键词: 不确定性; 模型; 软件系统; 语义; 程序设计

中图分类号: TP301

文献标识码: A

文章编号: 1000-7180(2012)01-0015-03

A Model of Uncertainty Semantics Computation

MIAO De-cheng^{1,2}, XI Jian-qing²

(1 College of Mathematics and Information Science, Shaoguan University, Shaoguan 512000, China

2 College of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China)

Abstract: Software system complexity has increased continuously with the promoting of users' requirements. Therefore, how to keep robustness and correctness of large-scaled complicated software system has gradually become a hot issue, and study on uncertainty semantics computation is the key solution to this question. By analyzing the relevant study work, this paper makes a model of uncertainty semantics computation, designs four kinds of different formal semantics for a small typical language, and also proves the correctness and flexibility of this model by proof of equivalences for four kinds of formal semantics.

Key words: uncertainty; model; software system; semantics; programming

1 引言

不确定性语义计算旨在处理大型软件系统程序运行时不能预言执行顺序的语义, 是保证大型复杂软件系统健壮性与正确性, 为用户提供高质量数据服务而必须解决的一个关键问题. 文献[1]提出基于熵的云模型为实现不确定性人工智能找到一种简单、有效的形式化方法, 但这种方法并不适合软件系统不确定性语义计算. 文献[2]提出了基于最大熵模型的语义角色标注方法, 该方法关键在于模型参数估计和特征选择. Rough 集方^[3]的最大优点在于无需额外假设条件, 由已知数据如实回答问题, 但它要求数据是精确或确定的, 这与实际程序运行特点不符, 使得软件系统的开发与设计难以实现. 基于贝叶

斯理论的不确定性计算模型逻辑清晰并具有坚实的理论基础, 但贝叶斯网的独立假设条件过于严格, 这在不确定性语义计算过程中难以满足, 而且贝叶斯理论严格依赖的先验概率在软件开发工程实践中难以获取. 文献[4]提出了一种考虑不确定性的移动机器人路径规划方法, 在路径规划算法中综合考虑了建模与传感数据等系统的不确定性. 当前许多并发程序设计语言(如 Java、C++)虽考虑到进程不确定执行的特性, 但实际处理效果却都比较随意, 仅有语法意义而没有定义语义操作, 已有不确定性语义计算方法难以满足当前大型软件系统开发的需要.

本文采用幂域理论和形式语义学方法研究和处理程序设计语言不确定性语义计算问题, 建立一种不确定性语义计算模型. 应用本文建立的模型对示

例小语言设计不确定性语义计算的四种语义形式：操作语义、Egli-Milner 幕域指称语义、Smyth 幕域指称语义和代数公理语义，通过四种语义形式的等价性证明论证模型的正确性和灵活性。

2 模型的建立

首先给出一个示例小语言 $\mathcal{L}$ 。 $\mathcal{L}$ 是一个不确定性语言，其数据类型^[3]只有最基本的布尔型、整型及字符型，主要语法成分有原子命令语句、空语句、赋值语句、顺序语句、条件语句及循环语句。 $\mathcal{L}$ 的形式说明采用类似 Java 的形式。

定义 1 不确定性语义计算模型的形式化描述为 $M = (\text{SYN}, \text{SEM}, \text{FUN}, \text{EQU})$ ，其中 SYN 与 SEM 分别为既定形式语言的语法域和语义域，FUN 为在 SYN 与 SEM 上建立的语义解释函数，对于复杂的形式语言可定义一些辅助形式说明 EQU 为语义方程，用来解释语法单位的语义。

3 模型的应用

3.1 模型的操作语义

本文操作语义在语句与状态的格局^[3] $\langle s, \sigma \rangle$ 上设计，格局间归约关系用 $\rightarrow$ 表示。对 $\mathcal{L}$ 应用模型 M ，其操作语义可设计为：

SYN

$\pi; \text{Prg} = \text{Stmt}^*$ 程序 // Stmt^* 表示无穷表

域，即 $\text{Stmt}^* = \left\{ \sum_{i=0}^{\infty} D^i \mid D^i \text{ 是长度为 } i \text{ 的表域} \right\}$

$s; \text{Stmt}$ 语句
 $c; \text{Cmd}$ 原子命令
 $e; \text{expr}$ 表达式
 $i; \text{Iden}$ 标识符
 $op; \text{Oper}$ 操作符
 $e ::= \text{true} \mid \text{false} \mid i \mid e_1 \text{ op } e_2 \mid (e) \mid \lambda i. e$
 $s ::= c \mid \text{skip} \mid i; = e \mid s_1; s_2 \mid \text{if } e \text{ then } s_1 \text{ else } s_2 \text{ fi} \mid \text{while } e \text{ do } s \text{ od}$

SEM

$T = \{\text{true}, \text{false}\}$ Tarski 真值集合
 I 整型
 CH 字符型
 $d; D = T + I + CH$ 指称值
 $\sigma; S = \text{Iden} \rightarrow D$ 状态

FUN

$A; \text{Cmd} \rightarrow (S \rightarrow (P[S] - \{\varphi\}))$ // $P[S]$ 表示 S 上的幕域， φ 表示空元素

$B; \text{Expr} \rightarrow (S \rightarrow T)$

EQU

(1) $\langle c, \sigma \rangle \rightarrow \langle \text{skip}, \sigma' \rangle$ if $\sigma' \in A \llbracket c \rrbracket \sigma$

(2) $\langle \text{skip}; s, \sigma \rangle \rightarrow \langle s, \sigma \rangle$ // 方程左部的；
表示语句的复合

(3) $\langle i; = e, \sigma_{i \leftarrow e} \rangle \rightarrow \langle i \leftarrow e, \sigma_{i \leftarrow e} \rangle$ // $i \leftarrow e$
表示 e 替换 i 的出现

(4) if $\langle s_1, \sigma \rangle \rightarrow \langle s_2, \sigma' \rangle$ and $\langle s_2, \sigma' \rangle \rightarrow \langle s_2', \sigma'' \rangle$ then $\langle s_1; s_2, \sigma \rangle \rightarrow \langle s_2', \sigma'' \rangle$

(5) $\langle \text{if } e \text{ then } s_1 \text{ else } s_2 \text{ fi}, \sigma \rangle \rightarrow \langle s_1, \sigma \rangle$ if $B \llbracket e \rrbracket \sigma = \text{true}$ or $\langle s_2, \sigma \rangle$ if $B \llbracket e \rrbracket \sigma = \text{false}$

(6) $\langle \text{while } e \text{ do } s \text{ od}, \sigma \rangle \rightarrow \langle s; \text{while } e \text{ do } s \text{ od}, \sigma \rangle$ if $B \llbracket e \rrbracket \sigma = \text{true}$ or $\langle \text{skip}, \sigma \rangle$ if $B \llbracket e \rrbracket \sigma = \text{false}$

引入无定义值 $\perp$, $D^\perp = D \cup \{\perp\}$ 。记 $\epsilon(D^\perp)$ 为域 D 上具有 Egli-Milner 序非空子集的集合， $\mathcal{L}$ 操作语义形式化描述为： $OP; \text{Stmt} \rightarrow S \rightarrow \epsilon(D^\perp)$ 。

定义 2 程序 π 执行可终止或不终止， π 终止的操作语义形式化描述为 $\langle \pi, \sigma \rangle \rightarrow \langle \text{skip}, \sigma' \rangle$ 。其中， σ' 为 π 终止时的状态；如果 π 不终止称 π 在 σ 上是发散的，则有 $OP \llbracket \pi \rrbracket \sigma = \{\sigma' \mid \langle \pi, \sigma \rangle \Rightarrow \langle \text{skip}, \sigma' \rangle\} \cup \{\perp \mid \pi \text{ 在 } \sigma \text{ 上是发散的}\}$ 。

3.2 模型的指称语义

对 $\mathcal{L}$ 应用模型 M 在 Egli-Milner 幕域与 Smyth 幕域上设计两个指称语义，其中 SYN、SEM 与 3.1 同，FUN 在其基础上增加两个语义解释函数：

$D_e; \text{Stmt} \rightarrow S \rightarrow \epsilon(S^\perp)$

$D_j; \text{Stmt} \rightarrow S \rightarrow J(S^\perp)$

记 $J(S^\perp)$ 为域 S 上具有 Smyth 序非空子集的集合，令 $D \in \{D_e, D_j\}$ ， $\mathcal{L}$ 指称语义的语义方程 EQU 为：

(1) $D \llbracket c \rrbracket \sigma \rightarrow A \llbracket c \rrbracket \sigma$

(2) $D \llbracket \text{skip} \rrbracket \rightarrow \text{Singleton}_e$ or Singleton_j // $\text{Singleton}_e; D \rightarrow \epsilon(D^\perp)$, $\text{Singleton}_j; D \rightarrow J(D^\perp)$ 为单类集函数

(3) $D \llbracket i; = e \rrbracket \sigma_{i \leftarrow e} \rightarrow D \llbracket i \leftarrow e \rrbracket \sigma_{i \leftarrow e}$

(4) $D \llbracket s_1; s_2 \rrbracket \rightarrow D \llbracket s_1 \rrbracket; D \llbracket s_2 \rrbracket$ // 方程右部；表示 Egli-Milner 与 Smyth 序上的函数复合

(5) $D \llbracket \text{if } e \text{ then } s_1 \text{ else } s_2 \text{ fi} \rrbracket \sigma \rightarrow \text{if } B \llbracket e \rrbracket \sigma \text{ then } D \llbracket s_1 \rrbracket \sigma \text{ else } D \llbracket s_2 \rrbracket \sigma$

(6) $D \llbracket \text{while } e \text{ do } s \text{ od} \rrbracket \sigma \rightarrow \text{if } B \llbracket e \rrbracket \sigma \text{ then } (D$

$\llbracket s \rrbracket \sigma; D \llbracket \text{while } e \text{ do } s \text{ od} \rrbracket \sigma \text{ else } D \llbracket \text{skip} \rrbracket \sigma$

对 $\mathcal{L}$ 指称语义应用泛函计算不动点,在 Egli-Milner 幂域上从 $\perp$ 开始求出的不动点集不具有 $\perp$, 但不确定性语义计算导致求解过程中存在多种路径选择,若完全确定所有求解路径则计算可转入 Smyth 幂域上去执行,因此, Egli-Milner 幂域与 Smyth 幂域对不确定性语义计算不可缺少.同时, Egli-Milner 幂域与 Smyth 幂域存在一种转换关系,定义函数 $C_S: \epsilon(S^\perp) \rightarrow J(S^\perp)$,通过对易关系 $\text{Singleton}_\epsilon C_S = \text{Singleton}_J$ 可以实现 $\mathcal{L}$ 两个指称语义形式相互转换.

$\mathcal{L}$ 指称语义的操作特性可归结为定理 1.

定理 1

(1) $D_\epsilon = OP$;

(2) 若程序 π 计算可终止,则 $\sigma' = D_J \llbracket \pi \rrbracket \sigma$ iff $\langle \pi, \sigma \rangle \Rightarrow \langle \text{skip}, \sigma' \rangle$; $\perp \in D_J \llbracket \pi \rrbracket \sigma$ iff π 在 σ 上发散

证明:(1) 显然成立;由定义 2 易证(2)成立.证毕.

定理 1 论证了 $\mathcal{L}$ 操作语义与指称语义的等价性.

3.3 模型的代数公理语义

采用最弱前置谓词方法^[4]对 $\mathcal{L}$ 应用模型 M 设计代数公理语义.其中 SEM 与 3.1 同, SYN 增加一个语法成分: $\phi: \text{Pred}$ 谓词. FUN 在 3.2 基础上增加一个语义解释函数 $R: \text{Stmt} \rightarrow PT$, 其中 PT 为谓词转换器集. 设 Smyth 状态转换器为 $m: S_1 \rightarrow J(S_2^\perp)$, ST 为谓词转换器集, ST 与 PT 都是点态序的且在偏序意义上是同构的. 取 S_2 任意子集 ϕ , 定义最弱前置谓词方法 $WP(m, \phi) = \{ \forall a \in S_1 \mid m(a) \subseteq \phi \}$, $WP(m, \phi)$ 在 m 上是单调的. 记 $C_1 = \{ WP(s, \phi) \}$, $C_2 = \{ WP(s, \neg \phi) \}$, $C_3 = \{ \neg WP(s, \text{true}) \}$, C_1, C_2, C_3 两两互斥. 记 π 确定性语义计算集为 $\text{CerSem} = \{ C_1, C_2, C_3 \}$, 则前置谓词 ϕ 满足 ${}_l \phi$ ($\phi \in \text{CerSem}$), ${}_l$ 是摹状词^[5]. π 不确定性语义计算集 UncSem 较为复杂, $\text{UncSem} = \{ P[\text{CerSem}] - \{ \phi \} \}$, 对于 π 的一次执行, ${}_l \phi$ ($\phi \in \text{UncSem}$).

代数公理语义方法定义语言的语义必须保证公理规则有实际模型^[6], 本文用指称语义作为模型, 使得 $\mathcal{L}$ 的代数公理语义一致于其指称语义模型. $\mathcal{L}$ 代数公理语义的语义方程 EQU 为:

(1) $R \llbracket c \rrbracket (\phi) \rightarrow WP(A \llbracket c \rrbracket, \phi)$

(2) $R \llbracket \text{skip} \rrbracket \rightarrow WP(\text{skip}, \phi) \rightarrow \phi$

(3) $R \llbracket i := e \rrbracket (\phi) \rightarrow WP(i := e, \phi) \rightarrow \phi_{i \leftarrow e}$

(4) $R \llbracket s_1; s_2 \rrbracket \rightarrow R \llbracket s_1 \rrbracket \circ R \llbracket s_2 \rrbracket$ // 方程右部 $\circ$ 表示谓词函数复合

(5) $R \llbracket \text{if } e \text{ then } s_1 \text{ else } s_2 \text{ fi} \rrbracket (\phi) \rightarrow R \llbracket s_1 \rrbracket (\phi)$ if $B \llbracket e \rrbracket^{-1}(\text{true})$ or $R \llbracket s_2 \rrbracket (\phi)$ if $B \llbracket e \rrbracket^{-1}(\text{false})$

(6) $R \llbracket \text{while } e \text{ do } s \text{ od} \rrbracket (\phi) \rightarrow (R \llbracket s \rrbracket (\phi); R \llbracket \text{while } e \text{ do } s \text{ od} \rrbracket (\phi))$ if $B \llbracket e \rrbracket^{-1}(\text{true})$ or ϕ if $B \llbracket e \rrbracket^{-1}(\text{false})$

$\mathcal{L}$ 代数公理语义的操作特性可归结为定理 2.

定理 2 (1) $\sigma \in R \llbracket \pi \rrbracket (\phi)$ iff π 计算可终止且

$\langle \pi, \sigma \rangle \Rightarrow \langle \text{skip}, \sigma' \rangle, \sigma' \in \phi$;

(2) π 在 σ 上发散 iff $\perp \in \phi$.

证明: 由 WP 的函数定义知 σ 是 π 计算前的状态, $\sigma' \in \phi$ 为 π 计算终止时状态, 定义 2 易证(1)显然成立; 对于程序 π , 若 π 在 σ 上发散, 计算必不终止, 则后置谓词 ϕ 必包含 $\perp$, 故(2)成立. 证毕.

定理 2 论证了 $\mathcal{L}$ 代数公理语义与操作语义的等价性.

4 结束语

本文采用幂域理论和形式语义学方法研究和处理程序设计语言不确定性语义计算问题, 建立不确定性语义计算模型 M , 并给出相关定义和示例, 同时, 在部分资助项目中对模型 M 进行了实现. 本文模型的优势在于恰当选择示例小语言 $\mathcal{L}$ 的语法域与语义域, 应用模型 M 可有效处理程序运行时不确定性语义计算问题, 四种语义形式的等价性进一步证明模型 M 的灵活性, 这种灵活性蕴涵的现实意义不仅在于给软件开发人员提供多种选择, 而且还可方便工程设计人员开发出结构清晰、层次分明、正确可靠、易于维护的软件系统. 本文模型 M 不局限于特定程序设计语言的高度抽象性是研究不确定性语义计算一个很好的思路, 对于模型 M 的极小性与完备性, 应用模型 M 构成的形式系统的协调性等问题是下一步研究的工作.

参考文献:

- [1] 李德毅, 刘常昱, 杜鹃, 等. 不确定性人工智能 [J]. 软件学报, 2004, 15(11): 1583-1594.

(下转第 22 页)

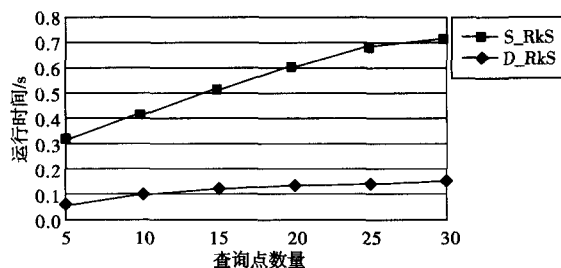


图5 k 值对候选者数量的影响

5 结束语

根据问题的实际需求提出了对于多对象综合考虑的 RkNN 查询算法. 本文对于一组查询点集合 $Q=\{q_1, q_2, \dots, q_s\}$, 先是生成覆盖查询 Q 的最小圆域 $DOM(O, R)$, 然后利用算法 1 输出 RkNN 查询搜索区域的初始化凸多边形区域, 并进一步确定其查询结果的候选区域, 然后对非法的数据点进行排除, 最后求得 RkNN 的查询结果. 本文的最后, 在测试数据集 CAR 上进行了算法 D_RkS 与算法 S_RkS 算法性能和效率测试, 结果表明, 对于一组空间中较近距离的若干个查询点的 RkNN 查询数据处理问题, D_RkS 算法的性能和效率要优于单个查询 S_RkS 算法的性能和效率.

参考文献:

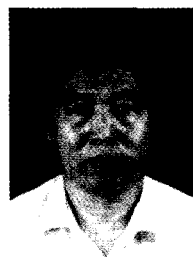
- [1] Korn F, Muthukrishnan S. Influence sets based on reverse nearest neighbor queries [C]//Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data. Dallas, New York, USA: ACM Press, 2000: 201-212.

(上接第 17 页)

- [2] 于江德, 王希杰, 余正涛. 基于最大熵模型的语义角色标注 [J]. 微电子学与计算机, 2010, 27(8): 173-180.
- [3] 周兰凤, 徐芳. 一种考虑不确定性的移动机器人路径规划方法 [J]. 微电子学与计算机, 2010, 27(7): 86-89.
- [4] 陈火旺, 罗朝晖, 马庆鸣. 程序设计方法学基础 [M]. 长沙: 湖南科学技术出版社, 1987: 18-38.
- [5] 莫绍揆, 徐永森, 沈百英. 数理逻辑 [M]. 北京: 高等教育出版社, 1984: 117-118.
- [6] 陈意云. 程序设计语言理论 [M]. 北京: 高等教育出版社, 2004: 16-60.

- [2] 陈丽, 陈静, 高新涛, 等. 基于支持向量机与 RkNN 的分类算法研究 [J]. 计算机工程与应用, 2010, 46(24): 135-137.
- [3] Wu Wei, Chee Fei Yang, Chan Yong, et al. Continuous reverse k-nearest-neighbor monitoring [C]//Proceedings of the 9th International Conference on Mobile Data management. Beijing, 2008: 132-139.
- [4] Tao Y, Papadias D, Lian X. Reverse kNN search in arbitrary dimensionality//Proceedings of the Very large Data Bases. Toronto, Canada, 2004: 744-755.
- [5] 汪卫, 王文平. 求一个包含点集所有点的最小圆的算法 [J]. 软件学报, 2000, 9(11): 1237-1240.
- [6] 余小高. P2P 环境中 k 最近邻搜索算法研究 [J]. 微电子学与计算机, 2009, 26(9): 61-63.
- [7] Census Bureau U S. Census, TIGER/Line Shapefiles [EB/OL]. [2008-07-12]. <http://www.census.gov/geo/www/tiger/tgrsh2008/tgrshp2008.html>.

作者简介:



刘金岭 男, (1958—), 教授, 硕士生导师. 研究方向为数据库、数据仓库、数据挖掘.

杨凤霞 女, (1976—), 硕士, 讲师. 研究方向为数据库、数据仓库.

刘国香 女, (1978—), 硕士研究生, 讲师. 研究方向为数据库.

作者简介:



苗德成 男, (1979—), 博士研究生, 讲师. 研究方向为软件系统形式化理论、形式语义学.

奚建清 男, (1962—), 博士后, 教授, 博士生导师. 研究方向为软件系统形式化理论.