维普资讯 http://www.cqvip.com

空间逻辑的一个定义框架及其可判定性*)

颜 锋 陈韬略 韩婷婷 吕 建
(南京大学计算机软件新技术国家重点实验室 南京 210093)

摘 要 空间逻辑作为一个模态逻辑,能很好地描述分布式系统的行为和空间属性。其中,逻辑公式的有效性、可满足性及模型检测问题的可判定性已经得到广泛的研究。本文即是关于空间逻辑可判定性的一个综述,为此首先提出一个空间逻辑的定义框架,据此可以构造各种空间逻辑,并对它们的可判定性进行考察,从而指出影响空间逻辑的可判定性的关键因素。

关键词 移动计算,分布式计算,空间逻辑,可判定性,定义框架

A Definition Framework of Spatial Logic and Decidability

YAN Feng CHEN Tao-Lue HAN Ting-Ting LU Jian
(State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093)

Abstract The introduction of spatial logics has been motivated by a recent shift of focus from monolithic concurrent systems towards distributed computing systems. The decidability of validity, satisfiability, and model checking for spatial logics is an important issue which has been studied deeply. This paper is survey of these results. For this, a definition framework of spatial logic is proposed for constructing various logics and the decidability is investigated. And the crux of the matter that influences the decidability of spatial logics is revealed.

Keywords Mobile computation, Distributed computing, Spatial logic, Decidability, Definition framework

1 引言

移动计算是随着移动通信、互联网、数据库、分布式计算等技术的发展而兴起的新的分布式计算范型。为了从理论的高度对移动计算加以研究,研究者提出了多种移动计算模型,其中π演算[1]和移动环境演算(Mobile Ambient[2])是两个最受瞩目的移动进程演算。

在对移动计算进行研究的过程中,我们需要精确地表达移动计算的某些属性,如进程的移动、系统的安全等,传统的进程代数方法如进程等价存在一些缺陷,如某些属性很难表达或者表达显得很不自然。而用模态逻辑来表达这些属性显得非常容易,并且我们可以使用已有的模型检测的方法,因此,最近许多研究针对模态逻辑展开。

空间逻辑作为一个模态逻辑被引入,以适应对分布式移动计算系统的研究。分布式系统相对于并发系统来说,系统的空间结构比较分散,它是由许多分散的、独立可观察的行为实体和计算实体组成。因此,在分布式系统中,空间和位置信息是非常重要的。已有的逻辑系统可以很好地刻画系统的行为相关属性,但并不能很好地表达出系统的空间分布属性,因此我们需要一个新的逻辑系统来描述分布式系统的空间分布属性,而空间逻辑就是我们所需要的能够对分布式系统的空间分布属性进行刻画的逻辑系统。事实上,空间逻辑可以刻画系统空间结构的能力使得它有更为广泛的应用。例如空间逻辑的模型包括有并发对象[3],堆(heap)[4,5],树(tree)[6],图[7]以及进程演算如π演算[8,9]和环境演算[10,11]等。本文选择树和环境演算作为空间逻辑模型,分别用来考察静态和动态空间逻辑的表达能力和可判定性。静态逻辑只能刻画模型在某一时刻的空间属性,而动态逻辑可以刻画模型随时间而动态变化的空间结构属性。我们以树作为静态逻辑的模型,树可以用来表达类似于XML的半结构化数据,因此我们可以利用空间逻辑管理和查询半结构化数据[12,13]。环境演算非常适合于对分布式移动计算系统建模,因此我们在此选择它作为动态空间逻辑的模型。

在关于空间逻辑的研究中,逻辑公式的有效性(validity)、可满足性(satisfiability)及模型检测的可判定性问题是非常重要的,因为它们关系到逻辑系统能否被实际应用,其中模型检测的可判定性问题最为重要。许多文献针对不同的空间逻辑进行了研究,并得到了很多结果。但他们的研究都是针对某一个或者某种特定的空间逻辑,例如静态逻辑或动态逻辑。这使得我们对逻辑中的各个模态词缺乏整体的认识,对它们与逻辑的表达能力及可判定性的关系也没有清晰的理解。本文的目的在于将各文献中所讨论的空间逻辑置于一个统一的定义框架中,由此我们可以很明显地看出各个模态词在空间逻辑中的作用以及它们对空间逻辑的表达能力和可判定性的影响。具体而言,本文提出了一个空间逻辑的定义框架,也即给出了一个构造空间逻辑的统一方式。基于这个框架我们研究了目前为止各文献中关于空间逻辑的可判定性结果,并讨论了各个模态词对空间逻辑的表达能力和可判定性的影响,由此加深了我们对各个模态词的理解。同时,基于这个组合框架,我们还可以构造更多有意义的空间逻辑进行研究。

本文第2部分介绍树和环境演算这两个模型,并介绍空

*)基金项目:国家重点基础研究发展规划973项目(2002CB312002);863项目(2002AA116010);国家自然科学基金(60273034)。颜 锋 硕士生,现从事移动计算,形式化方法研究。

维普资讯 http://www.cqvip.com

间逻辑的逻辑公式及满足关系；第3部分提出一个空间逻辑的定义框架，并在此基础上讨论各种空间逻辑的可判定性问题，指出各个模态词在空间逻辑中的作用；第4部分总结全文，并讨论相关工作。

2 模型及空间逻辑

本部分介绍树和环境演算这两个模型及空间逻辑的语法和语义。

2.1 有穷标号树

假设 N 是可数无限的名字集，以 m,n 表示名。我们所讨论的树是有穷分支，有穷深度并且是无序的。以 T,U 表示树，定义如下：

$$T,U::=0\mid n[T]\mid T\mid U\mid(\nu n)T$$

在树 $(\nu n)T$ 中，n 为约束名，否则 n 便为自由名，以 $fn(T)$ 表示树 T 中的自由名集。记 $T\{n/x\}$ 表示用名 n 替换树 T 中名 x。下面我们引入树之间的结构同余关系，一般地，以 $\equiv$ 表示。为此首先定义上下文(Context)的概念，其定义如下：

定义 1(上下文) 一个上下文(用 C 表示)是内含孔(hole)$[\cdot]$的树，以 BNF 定义如下：

$$C[\cdot]::=[\cdot]\mid T\mid C[\cdot]\mid n[C[\cdot]]\mid(\nu n)C[\cdot]$$

结构同余关系即为满足以下规则的最小二元关系。

(Struct Refl)	$T\equiv T$
(Struct Symm)	$T\equiv U\Rightarrow U\equiv T$
(Struct Trans)	$T\equiv U,U\equiv V\Rightarrow T\equiv V$
(Struct Context)	$T\equiv U\Rightarrow C[T]\equiv C[U]$
(Struct Par Comm)	$T\mid U\equiv U\mid T$
(Struct Par Assoc)	$(T\mid U)\mid V\equiv T\mid(U\mid V)$
(Struct Zero Par)	$T\mid 0\equiv T$
(Struct Zero Res)	$(\nu n)0\equiv 0$
(Struct Renaming)	$(\nu n)T\equiv(\nu m)T\{m/n\}$ if $m\notin fn(T)$
(Struct Res Res)	$(\nu m)(\nu n)T\equiv(\nu n)(\nu m)T$
(Struct Res Amb)	$(\nu n)m[T]\equiv m[(\nu n)T]$, if $m\neq n$

2.2 环境演算

假设 N 是可数无限的名字集。一般叶，$N=\{m,n,q,r,\cdots\}$。以 P,Q 表示其中的进程，定义如下：

$$P,Q::=0\mid(\nu n)P\mid P\mid Q\mid \text{cap}.P\mid n[P]$$

其中，cap 为能力(capability)集合的元素。其定义如下：

$$\text{cap}::=\text{in } n\mid\text{out } n\mid\text{open } n$$

囿于篇幅，在此不给出直觉解释，详见文[2]。

约束名和自由名的定义和树中相同，在此不赘述。记 $P\{n/x\}$ 表示用名 n 替换进程 P 中名 x。注意在进程的定义中不包含有复制操作，这是因为如果包含复制操作则会导致相应的空间逻辑不可判定。

下面以环境演算进程间的归约关系→的形式给出环境演算的语义，这种语义源自 Berry 和 Boudol 的化学抽象机[14]，是一种典型的操作语义的描述风格，在进程演算系统中被广为使用。单纯的归约语义是不能够很好地描述进程行为的，为此，必须引入结构同余关系，环境演算的结构同余关系如上述树的结构同余关系，不同之处在于上下文定义的不同，在此只给出环境演算上下文的定义。

定义 2(上下文) 一个上下文(用 C 表示)是内含孔(hole)$[\cdot]$的进程项，以 BNF 定义如下：

$$C[\cdot]::=[\cdot]\mid P\mid C[\cdot]\mid n[C[\cdot]]\mid(\nu n)C[\cdot]\mid \text{cap}.C[\cdot]$$

以下是环境演算进程的归约规则：

(R-in) $n[Q]\mid m[\text{in } n.Q_1\mid Q_2]\rightarrow n[Q\mid m[Q_1\mid Q_2]]$

(R-out) $n[m[\text{out } n.Q_1\mid Q_2]\mid P_1]\rightarrow n[P_1]\mid m[Q_1\mid Q_2]$

(R-open) $\text{open } n.P\mid n[Q]\rightarrow P\mid Q$

(R-Res) $\dfrac{P\rightarrow P'}{(\nu n)P\rightarrow(\nu n)P'}$

(R-Par) $\dfrac{P\rightarrow P'}{P\mid Q\rightarrow P'\mid Q}$

(R-Amb) $\dfrac{P\rightarrow P'}{n[P]\rightarrow n[P']}$

(R-≡) $\dfrac{P\equiv P'P'\rightarrow P''P''\equiv P'''}{P\rightarrow P'''}$

这些规则的直觉含义的解释请读者参考文[2]。一般地，以⇒表示→的自反传递闭包。

由此可以看出，树模型实际上是环境演算的一个静态子集。从另一角度来说，环境演算也可以看作是一个动态的树模型，也是就说进程的空间结构会随着时间发生变化。

2.3 空间逻辑

本部分简要介绍了空间逻辑，详细的直觉解释读者可见文[8~11]。

2.3.1 *逻辑公式* 假设 V 是可数无穷的变量集，一般地，$V=\{x,y,z,\cdots\}$。空间逻辑公式用 A,B 表示，则 A,B 以 BNF 定义如下：

$$A,B::=\mathbf{T}\mid\neg A\mid A\vee B\mid 0\mid A\mid B\mid n[A]\mid A@n\mid\Diamond A\mid A\triangleright B\mid \exists x.A\mid \mathbf{I}x.A\mid n\circledR A\mid A\circledcirc n\mid\Diamond A$$

上面一行模态词组成一个逻辑记为 L_{∞}，它包含了基本的命题逻辑和空间逻辑模态词，这是空间逻辑中最基本的部分，本文中讨论的空间逻辑都包含 L_{∞} 中的模态词。模态词 $\exists$，I 是绑定操作，因此引入了约束变量的概念，它与传统的定义是一致的。本文中，用 $fv(A)$ 和 $bv(A)$ 分别表示公式 A 中的自由变量集与约束变量集。如果一个公式的自由变量集为空集，则称这个公式是闭合的。

2.3.2 *满足关系*(Satisfaction relation) 满足关系 $P\models A$ 表示进程 P 满足闭合公式 A。对于空间模态词"◇"，我们需要定义关系 $P\downarrow P'$ 如下：

$$P\downarrow P'\Leftrightarrow\exists n,Q.\ P\equiv n[P']\mid Q$$

记 $\Downarrow$ 为 $\downarrow$ 的自反传递闭包。下面给出满足关系的定义，需要指出的是在这儿我们以环境演算作为模型，给出了逻辑公式的语义，而以树为模型的逻辑公式的语义据此可以得到，篇幅所限，不再单独给出。

定义 3(满足关系) 以环境演算作为模型。满足关系定义在闭合公式上，具体定义如下：

$P\models\mathbf{T}\overset{def}{=}always\ true$；

$P\models\neg A\overset{def}{=}\neg(P\models A)$；

$P\models A\vee B\overset{def}{=}P\models A\vee P\models B$；

$P\models 0\overset{def}{=}P\equiv 0$；

$P\models A\mid B\overset{def}{=}\exists P_1,P_2.P\equiv P_1\mid P_2\wedge P_1\models A\wedge P_2\models B$；

$P\models A\triangleright B\overset{def}{=}\forall R.R\models A\Rightarrow R\mid P\models B$；

$P\models n[A]\overset{def}{=}\exists P_1.P\equiv n[P_1]\wedge P_1\models A$；

$P\models A@n\overset{def}{=}n[P]\models A$；

维普资讯 http://www.cqvip.com

$P\models\Diamond A \overset{def}{=} \exists P_1.\ P \Downarrow P_1 \wedge P_1 \models A$;

$P\models \exists x.\ A \overset{def}{=} \exists n\in N.\ P\models A\{n/x\}$;

$P\models \boldsymbol{И}x.\ A \overset{def}{=} \exists n\notin fn(P)\cup n(A).\ P\models A\{n/x\}$;

$P\models n\ \textcircled{R}\ A \overset{def}{=} \exists P_1.\ P\equiv(\nu n)P_1 \wedge P_1\models A$;

$P\models A\ \textcircled{1} n \overset{def}{=} (\nu n)P\models A$;

$P\models\Diamond A \overset{def}{=} \exists P_1.\ P\Rightarrow P_1 \wedge P_1\models A$;

下列附属模态词经常被使用，我们也在定义框架中考虑它们，其定义如下：

$\mathbf{H}x.\ A \overset{def}{=} \boldsymbol{И}x.\ x\ \textcircled{R}\ A$;

$P\models \mathbf{H}x.\ A \overset{def}{=} \exists n\notin fn(A).\ \exists Q.$

$P\equiv(\nu n)Q, P\models A\{n/x\}.$

$\textcircled{C}\ n \overset{def}{=} \neg n\ \textcircled{C}\mathbf{T}$;

$P\models\ \textcircled{C}\ n \overset{def}{=} n\in fn(P).$

下面我们给出公式的有效性、可满足性的概念，这些概念与其他文献是类似的。

$\mathbf{vld}(A) \overset{def}{=} \forall P.\ P\models A$ （有效性）

$\mathbf{sat}(A) \overset{def}{=} \exists P.\ P\models A$ （可满足性）

“▷”模态词具有很强的表达能力，它能将公式的有效性问题和可满足性问题转化成模型检测问题。

引理 1 $\mathbf{vld}(A)\Leftrightarrow 0\models \mathbf{T}\triangleright A$.

引理 2 $\mathbf{sat}(A)\Leftrightarrow 0\models\neg(\mathbf{T}\triangleright\neg A)$.

由此还可以看出，在有“▷”模态词的情况下，模型检测问题是我们所关注的可判定性问题的核心问题。

3 定义框架及空间逻辑的可判定性

在本节中，我们将各个不同的模态词进行组合得到各种不同的空间逻辑，并分别对它们的可判定性进行研究。为此，我们首先提出了一个定义框架，在这个框架中可以很方便地将各个模态词进行组合。在此基础上，我们对所形成的各种空间逻辑的可判定性进行分析，找出影响空间逻辑的可判定性的关键因素。定义框架的作用也正在于此，它使得我们可以更方便地了解各个模态词在空间逻辑中的作用，从而可以更深刻地理解它们。

3.1 定义框架

给出定义框架的关键在于确定组合参数，我们从三个角度选取组合参数，一方面考虑是否为动态逻辑即是否包含“◇”模态词，另一方面考虑是否包含“▷”模态词，最后考虑空间逻辑中其他模态词。下面详细阐述之。

(1)是否为动态逻辑。“◇”模态词赋予了空间逻辑描述进程动态运行的能力，使得空间逻辑可以描述进程运行各个阶段的空间结构，从而可以刻画进程的运行过程。因此，“◇”模态词显著提高了空间逻辑的表达能力。不含“◇”模态词的空间逻辑称之为静态逻辑。

(2)是否包含“▷”模态词。“▷”模态词的能力非常强，它允许逻辑对进程进行相当强的观察，并且在它的作用下有效性和可满足性问题可以转化模型检测问题[10]，这样有效性判定可以通过模型检测进行。

(3)其他模态词。其他模态词包括有处理限制操作的“①”、“©”、“**И**”、“Ⓡ”、“**H**”以及传统的存在量词“∃”。其中最重要的模态词为“**И**”、“Ⓡ”和“∃”，其它模态词可以通过它们被表达出来，为完全起见，我们也把它们考虑进定义框架中。

以基本的命题逻辑及空间逻辑模态词为基础，上述三个方面构成了我们的空间逻辑定义框架。如图 1 所示，L_{00} 为最基本的空间逻辑，在此基础上增加“▷”模态词得到 L_{01}，增加“◇”模态词得到 L_{10}，同时增加“▷”和“◇”模态词得到 L_{11}。因为 L_{00} 中既没有“▷”模态词也没有“◇”模态词，所以它的表达能力非常弱，对它的研究也非常少。在此我们只关注另外三个逻辑 L_{01}，L_{10}，L_{11} 以及在它们基础上增加其他模态词所得到的空间逻辑，例如，$L_{10}\cup\{\textcircled{R}\}$ 表示在 L_{10} 的基础上增加“Ⓡ”模态词所得到的逻辑。

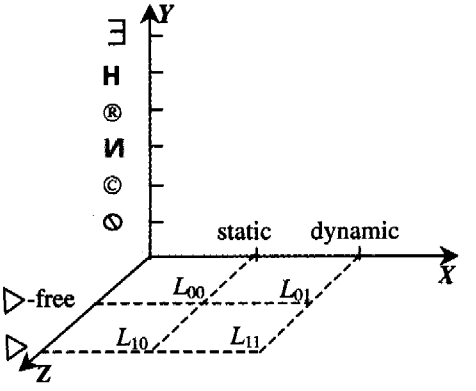

图 1 空间逻辑定义框架

3.2 空间逻辑的可判定性

上述空间逻辑定义框架给出了研究各种空间逻辑的系统方法。在本节中，我们根据上述框架对各种模态词进行组合，并给出相应空间逻辑的各种可判定性结果，据此我们讨论各种模态词对空间逻辑的可判定性的影响。

首先，我们给出逻辑 L_{10} 的可判定性结果，这个逻辑是在 L_{00} 的基础上增加“▷”而得。文[15]中选取没有限制操作的树作为模型，证明了它的可判定性。

定理 1 选取不含限制操作的树作为模型，空间逻辑 L_{10} 中的逻辑公式的有效性、可满足性及模型检测问题都是可判定的。

如果我们在 L_{00} 的基础上增加“◇”得到空间逻辑 L_{01}，选取不含复制操作的环境演算作为模型，则这个空间逻辑的模型检测问题也是可判定的[10,16,17]。

定理 2 选取不含复制操作的环境演算作为模型，空间逻辑 L_{01} 的逻辑公式的模型检测问题是可判定的。

在 L_{00} 中同时增加“▷”和“◇”模态词得到逻辑 L_{11}，选取环境演算作为模型，这个逻辑是不可判定的[18]。

定理 3 以环境演算作为模型，空间逻辑 L_{11} 中逻辑公式的有效性、可满足性及模型检测问题都是不可判定的。

上面我们讨论了三个基本的空间逻辑的可判定性问题，下面我们在这个基础上增加 Y 轴上的模态词进行讨论。

首先，对 L_{10} 来说，以树作为模型，在其基础上增加“①”、“©”、“**И**”这三个模态词仍然保持可判定性，但若增加“∃”、“Ⓡ”、“***H***”中任意一个都会导致所得到的逻辑是不可判定的[19]。

定理 4 在空间逻辑 $L_{10}\cup\{①,©,\boldsymbol{И}\}$ 中，以树作为模型，逻辑公式的有效性、可满足性及模型检测问题都是可判定的。

定理 5 在空间逻辑 $L_{10}\cup\{\exists\}$，$L_{10}\cup\{\textcircled{R}\}$，$L_{10}\cup\{\mathbf{H}\}$ 中，以树作为模型，逻辑公式的有效性、可满足性及模型检测

维普资讯 http://www.cqvip.com

问题都是不可判定的。

对 L_{01} 来说，选取不含复制操作的环境演算为模型，在其中增加除“▷”之外的模态词，其模型检测问题是可判定的[10,16,17]。但由于没有“▷”，我们不能根据模型检测问题的可判定性得到有效性及可满足性问题的可判定性。而事实上，L_{01} 中有效性和可满足性这两个问题是不可判定的。

定理 6 在空间逻辑 $L_{01}\cup\{\textcircled{1},\textcircled{c},\mathbf{И},\textcircled{R},\mathbf{H},\exists\}$ 中，以不含复制操作的环境演算为模型，逻辑公式的模型检测问题是可判定的。

最后，对 L_{11} 来说，由于其本身就是不可判定的，因此，我们在其中增加其他模态词仍然保持不可判定性。

为了能更明显地看出各个模态词对空间逻辑可判定性的影响，我们把上述关于模型检测问题的可判定性结果一起放入图 2 中。其中，标号箭头表示在原来逻辑中增加标号中的模态词而得到箭头所指。

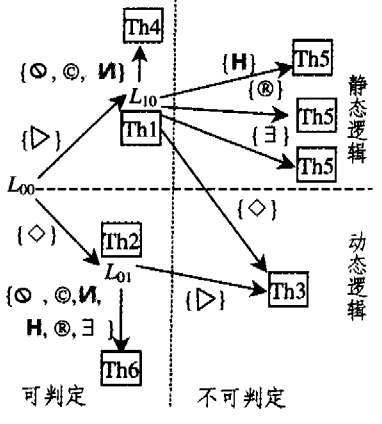

图 2 空间逻辑模型检测问题的可判定性

从图 2 中我们可以看出，在不同的环境下，各个模态词对空间逻辑的影响也不相同。下面我们从静态逻辑和动态逻辑两个角度来说明各个模态词对空间逻辑的影响。

(1)对静态逻辑来讲，在 L_{10} 中增加“①”、“©”、“И”模态词逻辑仍然是可判定的，但增加“®”，“∃”和“H”后逻辑变得不可判定。表面上看来是“®”，“∃”影响了逻辑的可判定性，但事实上这两个模态词对逻辑的影响是建立在有“▷”的基础上的，如果没有“▷”，则增加这两个模态词并不影响逻辑的可判定性。虽然“▷”使得静态逻辑变得不可判定，但它却没有增加静态逻辑的表达能力，文[20]中说明了模态词“▷”在静态逻辑中可被消除。这是一个非常令人惊奇的结果，因为“▷”使得逻辑不可判定，但其却没有提高逻辑的表达能力。又由“▷”使得逻辑不可判定可知，不可能存在一个能行的“▷”消除过程[19]。

(2)对动态逻辑来讲，如果不存在“▷”模态词，则逻辑便是可判定的，否则，逻辑便不可判定。与静态逻辑不同，在这里，“▷”模态词显著增加了逻辑的表达能力[18]。而其他模态词几乎没有影响动态逻辑的可判定性。

结论和相关工作 空间逻辑作为一个模态逻辑被引入，以适应对分布式移动计算系统的研究，其逻辑公式的可判定性问题是一个非常值得研究的论题。事实上，目前已经有许多文献对此进行了研究，也有非常多的结果，本文对这些结果进行了综合全面的考察。具体而言，首先，我们提出了一个空间逻辑的定义框架，在此基础上构造各种空间逻辑，考察这些逻辑的可判定性结果。由此，我们可以看出“▷”和“◇”是影响空间逻辑可判定性的两个关键因素。在静态逻辑中，在 L_{10} 的基础上增加“®”，“∃”和“H”后会使逻辑变得不可判定，但其深层原因在于“▷”的存在，然而令人惊奇的是“▷”并没有增加静态逻辑的表达能力。在动态逻辑中，在增加“▷”的情况下会使得逻辑变得不可判定，增加其他模态词则不影响逻辑的可判定性，在这个情况下，模态词“▷”增加了逻辑的表达能力。

进一步的工作在于考察空间逻辑的最小化问题，也即使用最少的模态词获得相同的表达能力。在这方面文[21,20,18]进行了一部分工作，我们可考虑在本文这个框架中讨论各种空间逻辑的最小化问题。另外，针对带有复制操作的环境演算，空间逻辑的可判定性问题还没有得到充分的研究，文[16,17,22]中对此进行了部分工作，这也是我们进一步的工作之一。

参 考 文 献

1 Milner R, Parrow J, Walker D. A Calculus of Mobile Process, part I/II. Journal of Information and Computation, 100: 1-77, Sept. 1992, 100: 1～77
2 Cardelli L, Gordon A D. Mobile ambient. Theoretical Computer Science, 2000, 240(1): 177～213
3 Benzaken V, Castagna G, Frisch A. Cduce: a white paper. PLAN-X: Programming Language Technologies for XML, Pittsburgh PA, Oct. 2002. http://www.cduce.org.
4 Gabbay M J, Pitts A M. A New Approach to Abstract Syntax Involving Bingders. In: Proc. LICS'1999. IEEE Computer Society Press, 1999. 214～224
5 Gabby M J. A Theory of Inductive Definitions with a-Equivalence: Semantics, Implementation, Programming Language: [Ph. D. Thesis]. University of Cambridge, 2000
6 Caires L. A Model for Declarative Programming and Specification with Concurrency and Mobility: [Ph. D. Thesis]. Universidade Nove de Lisboa, 1999
7 Buneman P, Davidson S B, Hillebrand G G, Suciu D. A Query Language and Optimization Techniques for Unstructured Data. In: SIGMOD Conf. 1996. 505～516
8 Caires L, Cardelli L. A Spatial Logic for Concurrency(Part Ⅰ). Information and Computation, 2003, 186(2): 194～235
9 Caires L, Cardelli L. A Spatial Logic for Concurrency(Part Ⅱ). In CONCUR 2002, number 2421 in Lecture Notes in Computer Science. Springer-Verlag, 2002
10 Cardelli L, Gordon A D. Anytime, Anywhere: Model Logics for Mobile Ambients. In: Proc. of POPL. ACM Press, 2000
11 Cardelli L, Gordon A D. Logical Properties of Name Restriction. In: Proc. of TCLA'01, volume 2044 of LNCS, Spriger, 2001
12 Cardelli, Ghelli G. A Query Language Based on the Ambient Logic. In ESOP 2001, volume 2028 of LNCS, Springer-Verlag, 2001
13 Cardelli L, Gardner P, helli G. Manipulating Trees with Hidden Labels. In FoSSaCS'03, LNCS. Springer-Verlag, 2003
14 Berry G, Boudol G. The chemical abstract machine. Theoretical Computer Science, 1992, 96(1): 217～248
15 Calcagno C, Cardelli L, Gordon A D. Deciding Validity in a Spatial Logic for Trees. In: ACM Workshop on Types in Language Design and Implementation, New Orleans, USA, ACM Press, 2003. 62～73
16 Charatonik W, Talbot J-M. The Decidability of Model Checking Mobile Ambients. In: Proc. of FoSSaCS'04, LNCS. Springer-Verlag, 2004
17 Charatonik W, Gordon A D, Talbot J-M. Finite-control Mobile Ambients. In ESOP'02, number 2305 in LNCS. Springer-Verlag, 2002
18 Caires L, Lozes E. Elimination of Quantifiers and Undecidability in Spatial Logics for Concurrency. In CONCUR 2004, LNCS. Springer-Verlag, 2004
19 Conforti G, Ghelli G. Decidability of Freshness, Undecidability of Revelation. In: Proc. of FoSSaCS'04, number 2987 in LNCS. Springer-Verlag, 2004
20 Lozes E. Adjunct Elimination in the Static Ambient Logic. In: Proc. of EXPRESS'2003, to appear in ENTCS, Elsevier, 2003
21 Hirschkoff D, Lozes E, Sangiorgi D. Minimality results for the spatial logics. In: Proc. of FSTTCS'2003, number 2914 in LNCS. Springer-Verlag, 2003
22 Caires L. Behavioral and Spatial Properties in a Logic for the Pi-Calculus. In: Proc. of FoSSaCS'2004, number 2987 in LNCS. Springer Verlag, 2004