适用于空间推理的 DLSR(DP)描述逻辑*)

刘亚彬　陈　岗
(上海财经大学信息系　上海200433)

摘　要　本文扩展了描述逻辑 ALC(D)并提出了空间推理的描述逻辑 DLSR(DP)，在利用描述逻辑进行空间推理时，通过术语推理把定性信息和定量信息相结合。

关键词　具体域，描述逻辑，ALC(D)，DLSR(DP)，空间推理

Descriptin Logic DLSR(DP) for Spatial Reasoning

LIU Ya-Bin　CHEN Gang
(Department of Information, Shanghai University fo Finance&Economics, Shanghai 200433)

Abstract　This paper extends the description logic ALC(D) and developes a description logic DLSR(DP) for spatial reasoning. In spatial reasoning by this description logic, we can integrate quantitative and qualitative information with terminological reasoning.

Keywords　Concrete domain, Description logic, ALC(D), DLSR(DP), Spatial reasoning

1　引言

描述逻辑(Description Logic)是为了表示概念和概念层次知识而设计的知识表示语言，可以合并且给出众所周知的基于框架(Frames)、语义网络(Semantic Networks)、面向对象的表示(Object-oriented Representations)、语义数据模型(Semantic Data Models)的系统的逻辑基础。描述逻辑的主要推理工作是包含和可满足性检验[3,4]。

传统描述逻辑系统的概念描述形式方法限制了描述逻辑的应用，必须进行扩展。扩展描述逻辑系统的方法有两种[5,6]：通过具体域扩展和通过传递闭包扩展。

定性空间推理被广泛应用于地理信息系统、机器人导航、高级视觉、自然语言理解、工程设计和物理位置的常识推理等领域[7]。已经发表了有很多关于空间结构的形式推理理论，例如众所周知的 RCC 理论[8]。尽管这些定性空间推理理论获得了广泛的应用，但是在定性空间推理的许多应用中，还必须考虑到如何把概念知识和定量数据结合起来，因此其它方法也是必需的。

为了充分地支持空间区域间定性关系的可判定推理和定量数据性质的可判定推理，我们在定义了适用于由点、线和区域共同组成的具体域 DP 的基础上，扩展了 ALC(D)描述逻辑，提出了适用于以点、线或区域为空间实体基元的空间推理的描述逻辑 DLSR(DP)描述逻辑。其主要思想是利用具体域对象上的谓词来表示空间对象及空间对象之间的关系，利用众所周知的描述逻辑理论来表示抽象域的知识和对空间对象之间的关系进行推理。在利用该描述逻辑进行定性空间推理时，可以通过术语推理把定性信息和定量信息相结合。DLSR(DP)提供与具体域谓词相关联的任务，通过这些结构，DLSR(DP)扩展了 ALC(D)的表达能力。为了确保可满足性算法的终止，我们利用了术语公理集合的语法格式约束，尽管模型非常严格，但是，术语约束确保了语言的可判定性。

描述逻辑 ALC(D)通过与具体域的结合来支持具体知识的表示，具体域方法可以把概念知识与具体空间知识相结合[1]。可是，为了定义有意义的空间概念，必须对定性空间知识进行表示，并且必须利用它们的各种性质进行推理；由于还必须以描述逻辑的形式方法对这些关系进行量化，因此这些关系将被表示为任务；此外，这种形式方法必须保证定性关系知识与定量空间知识一致。在这部分中，我们引进了描述逻辑 DLSR(DP)，它通过基于具体域谓词的任务形成运算符扩展了 ALC(D)，这种新的运算符允许定义具有非常复杂性质的任务。

2　DLSR(DP)的语法

定义1　令 DR 和 DF 分别是互不相交的任务名和特征名集合，我们把 $DR \cup DF$ 的任意一个元素称为一个原子任务。特征合成(记为 $f_1 f_2 \cdots$)被称为特征链。如果 $P \in \Phi_{DP}$ 是一个带有数量 $n+m$ 的谓词名，$u_1, \cdots, u_n, v_1, \cdots, v_m$ 是特征链，那么表达式 $\exists (u_1, \cdots, u_n) \exists (v_1, \cdots, v_m).P$、$\exists (u_1, \cdots, u_n) \forall (v_1, \cdots v_m), P$、$\forall (u_1, \cdots, u_n) \exists (v_1, \cdots, v_m)$、$P$ 和 $\forall (u_1, \cdots, u_n) \forall (v_1, \cdots, v_m).P$ 也是任务，也称之为复合任务或任务形成谓词约束。令 A 是一个任务名，D 是一个任务，$A \doteq D$ 则是一个任务定义，也称之为术语公理。

定义2　令 DC 是一个与 DR 和 DF 都不相交的概念名集合，我们把 DC 的任意一个元素称为一个原子概念。如果 C 和 D 是概念，R 是任务，$P \in \Phi_{DP}$ 是一个带有数量 n 的谓词名，$u_1, \cdots, u_k$ 是特征链，则下面的表达式也是概念：$C \cap D$(与)、$C \cup D$(或)$\neg C$(非)、$\exists R.C$(存在约束)、$\forall R.C$(值约束)和 $\exists u_1, \cdots, u_n.P$(谓词存在约束)。令 A 是一个概念名，D 是一

*)本文研究得到国家自然科学基金(空间推理和空间知识表示研究及应用69883003)、国家836高科技项目基金(农业专家系统开发平台836-306-ZB05-01-2)和吉林大学符号计算与知识工程国家教育部开放实验室基金和资助。**刘亚彬**　博士，副教授，主要研究方向为知识工程和空间推理。**陈　岗**　副教授，主要研究方向为知识工程和空间推理。

个概念，则 $A \doteq D$ 是一个概念定义，也称之为术语公理。

定义3 令 T 是一个由有限的任务定义和概念定义组成的集合，如果 T 中的概念名和任务名出现在定义的左侧的次数都不多于一次，并且没有循环定义，则 T 是一个术语或 $TBox$。

3 DLSR(DP)的语义

定义4 解释 $I=(\Delta_I, I)$ 是由表示抽象域的集合 Δ_I 和解释函数 I 组成的，集合 Δ_{DP} 和 Δ_I 一定不相交，解释函数进行如下映射：每个概念名 C 到 Δ_I 的子集 C^I，每个任务名 R 到 $\Delta_I \times \Delta_I$ 的子集 R^I，每个特征名 f 到由 Δ_I 至 $\Delta_{DP} \cup \Delta_I$ 的部分映射 f^I，把 $f^I(a)=x$ 记为 $(a,x) \in f^I$。如果 $u=f_1 \cdots f_n$ 是一个特征路径，则把 u^I 定义为 $f_1^I \circ \cdots \circ f_n^I$，"$\circ$"表示函数的合成。令符号 $C, D, R, P, u_1, \cdots, u_n, v_1, \cdots, v_m$ 分别与定义1和定义2中的相同，则可以把解释函数扩展到任意的概念和任务：

$(C \cap D)^I := C^I \cap D^I \quad (C \cup D)^I := C^I \cup D^I$

$(\neg C)^I := \Delta_I - C^I$

$(\exists\ R.C)^I := \{a \in \Delta_I \mid \exists\ b \in \Delta_I : (a,b) \in R^I, b \in C^I\}$

$(\forall\ R.C)^I := \{a \in \Delta_I \mid \forall\ b \in \Delta_I ; (a,b) \in R^I, b \in C^I\}$

$(\exists\ u_1, \cdots, u_n.\ P)^I := \{a \in \Delta_I \mid \exists\ x_1, \cdots, x_n \in \Delta_{DP} : (a, x_1) \in u_1^I, \cdots, (a, x_n) \in u_n^I, (x_1, \cdots, x_n) \in P^{DP}\}$

$(\exists\ (u_1, \cdots, u_n) \exists\ (v_1, \cdots v_m).\ P)^I := \{(a,b) \in \Delta_I \times \Delta_I \mid \exists\ x_1, \cdots, x_n \in \Delta_{DP}, \exists\ y_1, \cdots, y_m \in \Delta_{DP} : (a, x_1) \in u_1^I, \cdots, (a, x_n) \in u_n^I, (b, y_1) \in v_1^I, \cdots, (b, y_m) \in v_m^I, (x_1, \cdots, x_n, v_1, \cdots v_m \in P^{DP}\}$

$(\exists\ (u_1, \cdots, u_n) \forall\ (v_1, \cdots v_m).\ P)^I := \{(a,b) \in \Delta_I \times \Delta_I \mid \exists\ x_1, \cdots, x_n \in \Delta_{DP}, \forall\ y_1, \cdots, y_m \in \Delta_{DP} : (a, x_1) \in u_1^I, \cdots, (a, x_n) \in u_n^I, (b, y_1) \in v_1^I, \cdots, (b, y_m) \in v_m^I, (x_1, \cdots, x_n, v_1, \cdots v_m \in P^{DP}\}$

$(\forall\ (u_1, \cdots, u_n) \exists\ (v_1, \cdots, v_m).\ P)^I := \{(a,b) \in \Delta_I \times \Delta_I \mid \forall\ x_1, \cdots, x_n \in \Delta_{DP}, \exists\ y_1, \cdots, y_m \in \Delta_{DP} : (a, x_1) \in u_1^I, \cdots, (a, x_n) \in u_n^I, (b, y_1) \in v_1^I, \cdots, (b, y_m) \in v_m^I, (x_1, \cdots, x_n, v_1, \cdots v_m \in P^{DP}\}.$

$(\forall\ (u_1, \cdots, u_n) \forall\ (v_1, \cdots, v_m).\ P)^I := \{(a,b) \in \Delta_I \times \Delta_I \mid \forall\ x_1, \cdots, x_n \in \Delta_{DP}, \forall\ y_1, \cdots, y_m \in \Delta_{DP} : (a, x_1) \in u_1^I, \cdots, (a, x_n) \in u_n^I, (b, y_1) \in v_1^I, \cdots, (b, y_m) \in v_m^I, (x_1, \cdots, x_n, v_1, \cdots v_m \in P^{PD}\}$

当且仅当解释 I 对于 T 中的所有的任务定义和概念定义 $A \doteq D$ 来说，都满足 $A^I = D^I$，则称解释 I 是 $TBoxT$ 的一个模型。当且仅当对 T 的所有模型 I，都满足 $D^I \subseteq C^I$，则称概念 C 关于 $TBoxT$ 包含概念 D，记为 $D \leq_T C$。当且仅当存在 T 的模型 I，使 $D^I \neq \Phi$，则称概念 D 关于 $TBoxT$ 是可满足的。

由于当且仅当 $C \cup \neg D$ 是不可满足的才有 $C \leq_T D$，因此包含和可满足性可以被相互简化。下面的定义引进了可以用于表示个体世界知识的 DLSR(DP)断言语言。

4 DLSR(DP)的断言语言和 ABox

定义5 令 O_D 和 O_A 分别是两个不相交的对象名集合，如果 C 是一个概念，R 是一个任务，f 是一个特征名，P 是一个带有数量 n 的谓词名，a 和 b 是 O_A 的元素，x 及 $x_1, \cdots, x_n$ 是 O_D 的元素，则 $a:C$、$(a,b):R$、$(a,x):f$ 和 $(x_1, \cdots, x_n):P$ 是断言。

一个有限的断言集合被称为 $ABox$。概念语言的解释可以通过如下的附加映射扩展到断言语言：把 O_A 的每个对象名映射为 Δ_I 的单一元素，把 O_D 的每个对象名映射为 Δ_{DP} 的单一元素：

$a : C \leftrightarrow a^I \in C^I$

$(a,b) : R \leftrightarrow (a^I, b^I) \in R^I$

$(a,x) : f \leftrightarrow f^I(a^I) \in x^I$

$(x_1, \cdots, x_n) : P \leftrightarrow (x_1^I, \cdots, x_n^I) \in P^{DP}$

当且仅当一个解释 I 是 T 的模型并且满足 $ABoxA$ 中的所有断言，则称这个解释 I 是 $ABoxA$ 的关于 $TBoxT$ 的模型。当且仅当 $ABox$ 有一个模型，$ABox$ 是关于 $TBoxT$ 一致的。

$ABox$ 的一致性问题是判定一个给定的 $ABoxA$ 是否关于 $TBoxT$ 一致。概念的可满足性可以被简化为如下的 $ABox$ 的一致性：当且仅当 $ABox\{a:C\}$ 是一致的，概念 C 是可满足的。遗憾的是如果没有术语约束，DLSR(DP)推理问题是不可判定的。

定理1 DLSR(DP)概念 C 关于 $TBoxT$ 的可满足性是不可判定的。

Baader 和 Hanschke 在文[9]中给出把"Post 对应问题(Post Correspondence Problem)"简化为 ALC(D)概念的可满足性的方法。以此类似，我们可以通过把"post 对应问题"简化为 DLSR(DP)概念 C 关于 $TBoxT$ 的可满足性来证明 DLSR(DP)概念 C 关于 $TBoxT$ 的可满足性是不可判定的。

5 DLSR 的约束术语

定义6 当且仅当概念 C 使用的概念名和任务名都不出现在 T 中的任意任务定义和概念定义的左侧，概念 C 是关于 $TBoxT$ 展开的。当且仅当关于 $TBoxT$ 展开的概念 C 的"非"仅出现在概念名的前面，关于 $TBoxT$ 展开的概念 C 是 NFF(Negation Normal Form)形式。

任意一个概念可以通过概念和任务定义来重复替换概念名和任务名而被变换为展开式，这个算法可以终止，因为 T 中的术语公理是有限的，并且术语中没有循环的定义。还可以把这个展开式进一步转化为 NFF 形式，转化为 NFF 形式是通过重复应用"向下"传播"非"到原子概念的变换规则来实现的：把 $\neg(C \cap D)$ 变换为 $\neg C \cup \neg D$，把 $\neg(C \cup D)$ 变换为 $\neg C \cap \neg D$，$\neg(\exists\ R.C)$ 把变换为 $\forall\ R.\neg C$，把 $\neg(\forall\ R.C)$ 可以变换为 $\exists\ R.\neg C$，把 $\neg(\exists\ u_1, \cdots, u_n.\ P)$ 变换为 $\forall\ u_1, \cdots, u_n.\ \neg P$。

定义7 当且仅当概念 X 存在一个与之等价的关于 $TBoxT$ 展开的概念 X'，并且 X' 的 NFF 形式满足下列条件，概念 X 被称为关于 $TBoxT$ 约束的。

(1) X' 的任意(子)概念 C 的形式为 $\forall\ R_1.D$，D 不包含任意 $\exists\ R_2.E$ 形式的项，其中，R_1 和 R_2 是复合任务。

(2) X' 的任意(子)概念 C 的形式为 $\exists\ R_1.D$，D 不包含任意 $\forall\ R_2.E$ 形式的项，其中，R_1 和 R_2 是复合任务。

(3) X' 的任意(子)概念 C 项的形式为 $\forall\ R.D$ 或 $\exists\ R.D$，其中，R 是复合任务，D 仅包含对长度为1的特征链进行量化的谓词存在约束和不被 C 中的任意存在约束和值约束包含的谓词存在约束。

当且仅当出现在 T 中的任务定义和概念定义的右侧的所有概念都是关于 T 约束的，术语 T 是有约束的。当且仅当 T 有约束并且所有用于 A 的概念都是关于术语 T 约束的，$ABoxA$ 被称为关于 $TBoxT$ 约束的。

下面我们分析具有约束的 DLSR(DP)描述逻辑的基本推理问题的可判定性。

定理2 有约束的 DLSR(DP)的 $ABox$ 的一致性问题是可以判定的。

我们通过给出的可靠的且完备的表算法来证明定理2。

6 DLSR(DP)的表算法

这个算法是一个标准的基于表格的算法，为了判定概念 C 的可满足性，这个算法从初始的 $ABox A_0:=\{a:C\}$ 开始，重复利用 DLSR(DP)的子代生成规则来建立一个或多个子代 $ABox$。因此，子代生成规则的应用构造了一棵 $ABox$ 树 Υ。最后，所有 $ABox$ 即 Υ 的叶节点要么是矛盾的，要么是非矛盾的；矛盾意味着 C 是不可满足的，非矛盾的意味着没有更多的可应用的完成规则。在后面的情况下，初始的 $ABox$ 是完备的，并且定义了一个 C 的模型。可以把子代生成规则的目的理解为把隐含的事实明确化。子代生成规则集是对 $ALC(D)$ 中的用于判定 $ABox$ 一致性的规则集的扩展。在给出子代生成规则之前，我们首先对相应的技术进行定义和说明。

令 A 是 $ABox$，R 是任务，a 和 b 是来自于 O_A 的对象名，γ 是符号但不是 O_D 的元素，u 是特征链 $f_1\circ\cdots\circ f_k, u_1,\cdots,u_n$，$v_1,\cdots,v_m$ 是任意特征链，我们定义如下三个函数：

$$filler1_A(a,u):=\begin{cases} x & \text{其中 } x\in O_D \text{ 满足}\exists\ b_1,\cdots,b_{k-1}\in O_A:((a,b_1):f_1\in A,\cdots,(b_{k-1},x):f_k\in A)\\ \gamma & \text{不存在这样的 } x\end{cases}$$

$$filler2_A(a,b,R):=\begin{cases} \text{true} & \text{如果}(a,b):R\in A\\ \text{true} & \text{如果 } R \text{ 的格式为}\exists\ (u_1,\cdots,u_n)\exists\ (v_1,\cdots,v_m).P,\exists\ (u_1,\cdots,u_n)\forall\ (v_1,\cdots,v_m).P,\\ & \forall\ (u_1,\cdots,u_n)\exists\ (v_1,\cdots,v_m).P \text{ 或}\exists\ (u_1,\cdots,u_n)\exists\ (v_1,\cdots,v_m).P \text{ 并且}\\ & \exists\ x_1,\cdots,x_n,y_1,\cdots,y_m\in O_D \text{ 使 } \mathrm{filler}_A(a,u_1)=x_1\wedge\cdots\wedge\mathrm{filler}_A(a,u_n)=x_n\\ & \wedge\mathrm{filler}_A(b,v_1)=y_1\wedge\cdots\wedge\mathrm{filler}_A(b,v_m)=y_m\wedge(x_1,\cdots,x_n,y_1,\cdots,y_m):p\in A\\ \text{false} & \text{其它情况}\end{cases}$$

$chain_A(a,x,u):=\{(a,c_1):f_1,\cdots,(c_{k-1},x):f_k\}$ 其中 $c_1,\cdots,c_{k-1}\in O_A$ 不用于 A。

定义8 如果 $ABox A$ 包含两个断言 $(a,b):f$ 和 $(a,c):f$ 其中 a 和 b 来自于 O_A 或 O_D，则称 $ABox A$ 包含一个特征 f 的分支。

分支可以通过以 b 置换 A 中 c 的所有具体值来排除。我们假设在任何规则用于初始 $ABox A_0$ 之前，分支就被排除了。

定义9 如下的子代生成规则将通过作为 $ABox A$ 的后代的一个 $ABox A'$ 或两个 $ABox A'$ 和 $ABox A''$ 来置换 $ABox A$。在下面，令 C 和 D 表示概念，R 表示任务，P 表示来自于 Φ_{DP} 的谓词名，$f_1,\cdots,f_n$ 表示特征名，$u_1,\cdots,u_n,v_1,\cdots,v_m$ 是特征链，a 和 b 表示来自于 O_A 的对象名。

$\cap-R$ 合取规则

前提：$a:C\cap D\in A, a:C\notin A\vee a:D\notin A$

结论：$A'=A\cup\{a:C,a:D\}$

$\cup$-R 析取规则

前提：$a:C\cup D\in A, a:C\notin A\wedge a:D\notin A$

结论：$A'=A\cup\{a:C\}, A''=A\cup\{a:D\}$

$\exists$ -R 存在约束规则

前提：$a:\exists\ R.C\in A,\neg\exists\ b\in O_A:(filler2_A(a,b,R)\wedge b:C\in A)$

结论：$A'=A\cup\{(a,b):R, b:C\}$。其中 $b\in O_A$，不用于 A

$\forall$ -R 值约束规则

前提：$a:\forall\ R.C\in A,\exists\ b\in O_A:(filler2_A(a,b,R)\wedge b:C\notin A)$

结论：$A'=A\cup\{b:C\}$

$\exists$ P-R 谓词存在约束规则

前提：$a:\exists\ u_1,\cdots,u_n.P\in A,\neg\exists\ x_1,\cdots,x_n\in O_D:(filler1_A(a,u_1)=x_1\wedge\cdots\wedge filler1_A(a,u_n)=x_n\wedge(x_1,\cdots,x_n):P\in A)$

结论：$A'=A\cup\{(x_1,\cdots,x_n):P\}\cup chain_A(a,x_1,u_1)\cup\cdots\cup chain_A(a,x_n,u_n)$其中，对象 $x_i\in O_D$ 不用于 A

$\exists$ $Pr-R1$复合任务规则1

前提：$(a,b):\exists\ (u_1,\cdots,u_n)\exists\ (v_1,\cdots,v_m).P\in A,\neg\exists\ x_1,\cdots,x_n,y_1,\cdots,y_m\in O_D:(filler1_A(a,u_1)=x_1\wedge\cdots\wedge filler1_A(a,u_n)=x_n\wedge filler1_A(b,v_1)=y_1\wedge\cdots\wedge filler1_A(b,v_m)=y_m\wedge(x_1,\cdots x_n,y_1,\cdots,x_m):P\in A)$

结论：$A'=A\cup\{(x_1,\cdots,x_n,y_1,\cdots,y_m):p\}\cup chain_A(a,x_1,u_1)\cup\cdots\cup china_A(a,x_n,u_n)\cup chain_A(b,y_1,v_1)\cup\cdots\cup chain_A(b,y_m,v_m)$其中，对象 $x_i\in O_D$ 和 $y_i\in O_D$ 不用于 A

$\exists$ $Pr-R2$复合任务规则2

前提：$(a,b):\exists\ (u_1,\cdots,u_n)\forall\ (v_1,\cdots,v_m).P\in A,\neg\exists\ x_1,\cdots,x_n,y_1,\cdots,y_m\in O_D:(filler1_A(a,u_1)=x_1\wedge\cdots\wedge filler1_A(a,u_n)=x_n\wedge filler1_A(b,v_1)=y_1\wedge\cdots\wedge filler1_A(b,v_m)=y_m\wedge(x_1,\cdots,x_n,y_1,\cdots,x_m):P\in A)$

结论：$A'=A\cup\{(x_1,\cdots,x_n,y_1,\cdots,y_m):P\}\cup chain_A(a,x_1,u_1)\cup\cdots\cup chain_A(a,x_n,u_n)\cup chain_A(b,y_1,v_1)\cup\cdots\cup chain_A(b,y_m,v_m)$其中，对象 $x_i\in O_D$ 和 $y_i\in O_D$ 不用于 A

$\exists$ $Pr-R3$复合任务规则3

前提：$(a,b):\forall\ (u_1,\cdots,u_n)\exists\ (v_1,\cdots,v_m).P\in A,\neg\exists\ x_1,\cdots,x_n,y_1,\cdots,y_m\in O_D:(filler1_A(a,u_1)=x_1\wedge\cdots\wedge filler1_A(a,u_n)=x_n\wedge filler1_A(b,v_1)=y_1\wedge\cdots\wedge filler1_A(b,v_m)=y_m\wedge(x_1,\cdots,x_n,y_1,\cdots,x_m):P\in A)$

结论：$A'=A\cup\{(x_1,\cdots,x_n,y_1,\cdots,y_m):P\}\cup chain_A(a,x_1,u_1)\cup\cdots\cup chain_A(a,x_n,u_n)\cup chain_A(b,y_1,v_1)\cup\cdots\cup chain_A(b,y_m,v_m)$其中，对象 $x_i\in O_D$ 和 $y_i\in O_D$ 不用于 A

$\exists$ $Pr-R4$复合任务规则4

前提：$(a,b):\forall\ (u_1,\cdots,u_n)\forall\ (v_1,\cdots,v_m).P\in A,\neg\exists\ x_1,\cdots,x_n,y_1,\cdots,y_m\in O_D:(filler1_A(a,u_1)=x_1\wedge\cdots\wedge filler1_A(a,u_n)=x_n\wedge filler1_A(b,v_1)=y_1\wedge\cdots\wedge filler1_A(b,v_m)=y_m\wedge(x_1,\cdots,x_n,y_1,\cdots,x_m):P\in A)$

结论：$A'=A\cup\{(x_1,\cdots,x_n,y_1,\cdots,y_m):P\}\cup chain_A(a,x_1,u_1)\cup\cdots\cup chain_A(a,x_n,u_n)\cup chain_A(b,y_1,v_1)\cup\cdots\cup chain_A(b,y_m,v_m)$其中，对象 $x_i\in O_D$ 和 $y_i\in O_D$ 不用于 A

*Choose-R1*选择规则1

前提：$a:\forall\ (\exists\ (u_1,\cdots,u_n)\exists\ (v_1,\cdots,v_m).P).C\in A,\exists\ b\in O_A\exists\ x_1\cdots,x_n,y_1,\cdots,y_m\in O_D:(filler1_A(a,u_1)=x_1\wedge\cdots\wedge filler1_A(a,u_n)=x_n\wedge filler1_A(b,v_1)=y_1\wedge\cdots\wedge filler1_A(b,v_m)=y_m\wedge(x_1,\cdots,x_n,y_1,\cdots,y_m):P\notin A\wedge(x_1,\cdots,x_n,y_1,\cdots,y_m):\overline{P}\notin A)$

结论：$A'=A\cup\{(x_1,\cdots,x_n,y_1,\cdots,y_m):P\}, A''=A\cup\{(x_1,\cdots,x_n,y_1,\cdots,y_m):\overline{P}\}$

*Choose-R2*选择规则2

前提：$a:\forall\ (\exists\ (u_1,\cdots,u_n)\forall\ (v_1,\cdots,v_m).P).C\in A,\exists\ b\in O_A\exists\ x_1,\cdots,x_n,y_1,\cdots,y_m\in O_D:(filler1_A(a,u_1)=x_1\wedge\cdots\wedge filler1_A(a,u_n)=x_n\wedge filler1_A(b,v_1)=y_1\wedge\cdots\wedge filler1_A(b,V_m)=y_m\wedge(x_1,\cdots,x_n,y_1,\cdots,y_m):P\notin A\wedge(x_1,\cdots,x_n,y_1,\cdots,x_m):\overline{P}\notin A)$

结论：$A'=A\cup\{(x_1,\cdots,x_n,y_1,\cdots,y_m):P\}, A''=A\cup\{x_1,\cdots,x_n,,y_1,\cdots,y_m\}:\overline{P}\}$

*choose-R3*选择规则3

前提：$a:\forall\ (\forall\ (u_1,\cdots,u_n)\exists\ (v_1,\cdots,v_m).P).C\in A,\exists\ b\in O_A\exists\ x_1,\cdots,x_n,y_1,\cdots,y_m\in O_D:(fillerl_A(a,u_1)=x_1\wedge\cdots\wedge fillerl_A(a,u_n)=x_n\wedge fillerl_A(b,v_1)=y_1\wedge\cdots\wedge fillerl_A(b,v_m)=y_m\wedge(x_1,\cdots,x_n,y_1,\cdots,x_m):P\notin A\wedge(x_1,\cdots,x_n,y_1,\cdots,x_m):\overline{P}\notin A)$

结论：$A'=A\cup\{(x_1,\cdots,x_n,y_1,\cdots,y_m):P\}$，$A''=A\cup\{(x_1,\cdots,x_n,y_1,\cdots,y_m):\overline{P}\}$

*choose-R4*选择规则4

前提：$a:\forall\ (\forall\ (u_1,\cdots,u_n)\forall\ (v_1,\cdots,v_m).P).C\in A,\exists\ b\in O_A,\exists\ x_1,\cdots,x_n,y_1,\cdots,y_m\in O_D:(fillerl_A(a,u_1)=x_1\wedge\cdots\wedge filllerl_A(a,u_n)=x_n\wedge fillerl_A(b,v_1)=y_1\wedge\cdots\wedge fillerl_A(b,v_m)=y_m\wedge(x_1,\cdots,x_n,y_1,\cdots,x_m):P\notin A\wedge(x_1,\cdots,x_n,y_1,\cdots,x_m):\overline{P}\notin A)$

结论：$A'=A\cup\{(x_1,\cdots,x_n,y_1,\cdots,y_m):P\}$，$A''=A\cup\{(x_1,\cdots,x_n,y_1,\cdots,y_m):\overline{P}\}$

定义10 令 C 表示概念，R 表示任务，$P_i(i=1,\cdots,k)$表示来自于 Φ_{DP}的谓词名，a 和 b 表示来自于 O_A 的对象名，令 f 是一个特征，如果如下的冲突触发器中任意一个被触发，则 $ABox A$ 被称为矛盾的：①基元冲突：$a:C\in A,a:\neg C\in A$；②特征域冲突：$(a,x):f\in A,(a,b):f\in A$；③所有域冲突：$(a,x):f\in A,a:\forall\ f.C\in A$；④具体域冲突：$(x_1^{(1)},\cdots,x_{n1}^{(1)}):P_1\in A,\cdots,(x_1^{(k)},\cdots,\cdots x_{nK}^{(K)}):P_k\in A$ 并且相应的合取 $\wedge_{i=1}^{k}P_i(x^{(i)})$ 在 D 中是不可满足的。这是可以判定的，因为 D 被要求是容许的。

DLSR(DP)的冲突触发器与 ALC(D)的冲突触发器是相同的。与 ALC(D)的相应规则相比，∃ Pr-R1-∃ Pr-R4 和 Choose-R1-Choose-R4是新增加的规则。

规则∃ Pr-R1—∃ Pr-R4的意义显而易见。如果已知两个对象通过复合任务相联系，那么我们可以立即建立具体对象作为用于定义复合任务的特征链的填充符，用于定义在新建立的具体对象上成立的谓词也就知道了，并且可以进一步添加适当的约束。

规则 Choose-R1—Choose-R4的意义可以理解如下：如果一个特征链集合用于定义复合任务 R，并且对于这个特征链集合中的所有特征链来说，存在两个对象 a 和 b 的适当的具体填充符，则 b 可能是 a 的任务 R 填充符。如果真是这种情况，必须存在一个显式的约束来规定定义任务 R 的谓词 P 成立(或不成立)。

我们可以利用与文[9]类似的方法来证明算法的有效性和完备性。算法的有效性是从子代生成规则的局部有效性中得出的。算法的完备性是通过证明任意完备的 $ABox$ 是根据定义初始 $ABox A_0$ 模型的算法经计算得到的来证明的。

结束语 我们展示了如何利用描述逻辑 DLSR(DP)把空间和术语推理结合在 TBox 中，关于描述逻辑的研究已经扩展到解决定性空间关系和定量空间数据，主要思想之一是引进基于具体对象性质定义的任务概念。目前的检测有限个合取式的可满足性的工作是在目前的定性空间推理理论的基础上进行的。

参 考 文 献

1 Baader F, Hanschke P. A scheme for integrating concrete domain into concept languages. In: Twelfth Intl. Joint Conf. Artificial Intelligence, Aug. 1991. 452～457

2 Lutz C. Reasoning with concrete domain. In: Proc. of the Sixteenth Intl. Joint Conf. on Artificial Intelligence IJCAI-99, Stockholm, Sweden, 1999

3 Lutz C. Reasoning with Complexity of terminological reasoning revisited. In: Proc. of the 6th Intl. Joint Conf. on Logic for Programming and Automated Reasoning (LPA'99), number 1705 in Lecture Notes in Artificial Intelligence, Springer-Verlag, Step. 1999. 181～200

4 Horrocks I, Sattler U, Tessaris S, Tobies S. Query Containment using a DLR ABox: [LTCS-Report 00-15]. LuFG Theoretical Computer Science, RWTH Aachen, Germany

5 Borgida A , Franconi E, Horrocks I, McGuinness D. Explaining ALC subsumption. In: Proc. of DL'99, 1999. 37～40

6 Artele A, Franconi E. A terporal description logic for reasoning about action and plans. In: J. Doyle, E. Sandewall, P. Torasso, eds. Fourth Intl. Conf. on Principles of Knowledge Representation, May 1994

7 Faltings B. Qualitative spatial reasoning using algebraic topology. In A U Frank and W Kuhn, editors, Spatial Information Theory: a theoretical basis for GIS, volume 988 of Lecture Notes in Computer Science, Springer-Verlag, 1995. 17～30

8 Randell D A, Cui Z, Cohn A G. A spatial logic based on regions and connection. In: B. Nebel, W. Swartout, C. Rich, eds. Principles of Knowledge Representation and Reasoning of the 3rd Intl. Conf. Cambridge MA, Morgan Kaufmann, 1992. 165～176

9 Baader F, Hanschke P. A scheme for integrating concrete domains into concept languages. In: Twelfth Intl. Joint Conf. Artificial Intelligence, Aug. 1991. 452～457

10 Nebel B. Computational properties of qualitative spatial reasoning: First results. In: I. Wachsmuth, C. -R. Rollinger, W. Brauer, eds. Proc. KI-95: Advances in Artificial Intelligence, 19th Annual German Conf. on Artificial Intelligence, Volume 981 of LNAI, Springer-Verlag, Sep. 1995. 233～244

11 Renz J, Nebel B. On the complexity of qualitative spatial reasoning: A maximal tractable fragment of the region connection calculus. In: Proc. of 15th Intl. Joint Conf. on Artificial Intelligence (IJCAI97), Aug. 1997. 522～527

(上接第158页)

图5中的测试结果表明，在不同网络负载下，包的平均传输延迟和丢包率，都处于很低的水平。上述性能指标，完全可以满足高性能路由器在效率、可靠性方面的需求。

结论 本文研究了分布式路由器软件体系结构通用化设计问题，提出了高性能路由器系统中的硬件抽象层概念及其功能，并就其功能及其关键模块在高性能 IPv6路由器基础平台及实验系统中予以实现。通过所构建的测试环境，对 HAL 进行了较系统的测试，测试结果表明：HAL 达到较高的性能指标，其运行效率与可靠性都能满足高性能路由器的需求。对 HAL 的功能分析结果表明，HAL 的提出较好地解决了路由器分布式体系结构所带来的软件系统设计的难题，在上层软件实现中，基本上可以不考虑底层硬件细节，增强了路由器的开放性及可扩展性，具有较强的普适性。

参 考 文 献

1 Keshav S, Sharma R. Issues and trends in router design. IEEE Communications Magazine, 1998, 36(5): 144～151

2 Basic Scalability and Performance Considerations for Evaluating Large-Scale Router Designs. http://www.cisco.com/warp/public/cc/pd/rt/12000/tech/ruar_wp.htm

3 Karlin S, Peterson L. VERA: An Extensible Router Architecture. In: Proc. of the 4th Intl. Conf. on Open Architectures and Network Programming, April, 2001

4 Intel Corporation. IXP1200 Network Processor Datasheet. Sep. 2000

5 Vitesse Semiconductor Corporation, Longmont, Colorado, IQ2000 Network Processor Product Brief, 2000

6 Andersen A T, Nielsen B F. A markovian approach for modeling packet traffic with long-range dependence. IEEE Journal on Selected Areas in Communications (JSAC), 1998, 16 (5): 719～732