

文章编号:1001-9081(2006)08-1912-04

线性时态逻辑中的特性模式

黎升洪^{1,2}, 缪淮扣², 张新林²

(1. 江西财经大学 信息管理学院, 江西 南昌 330013; 2. 上海大学 计算机工程与科学学院, 上海 200072)

(lish@jxufe.edu.cn)

摘要:在模型检查应用中, 需要使用线性时态逻辑对软件具备的特性进行描述。虽然, 不同应用背景涉及不同方面的特性描述, 但是线性时态逻辑描述软件特性方式上具有共性。本文从两个方面抽取这种共性, 首先, 按照线性时态逻辑所描述性质划分, 常见性质包括活性、安全性等; 其次, 按照线性时态逻辑公式的作用范围划分。通过对共同问题, 找到共同的描述方法得到线性时态逻辑的特性模式。最后介绍了线性时态逻辑特性模式在 SPIN 中的应用。

关键词:线性时态逻辑; 特性模式; 模型检查; SPIN

中图分类号: TP311.52 **文献标识码:** A

Property patterns of linear temporal logics

LI Sheng-hong^{1,2}, MIAO Huai-kou², ZHANG Xin-lin²

(1. School of Information and Technology, Jiangxi University of Finance and Economics, Nanchang Jiangxi 330013, China;

2. School of Computer Engineering and Science, Shanghai University, Shanghai 200072, China)

Abstract: In the applications of model checking, the software properties are usually depicted by Linear Temporal Logic (LTL) formulas. The descriptions of software properties based on LTL formulas have common patterns although different applications background have to depict different software properties. The common property patterns of LTL formulas was abstracted in two dimensions, which one was according to the properties depicted by the developers, for example, liveness and safeness, and the other was according to the scope of formulas. Property pattern is a manner that finds the common solution to the common problems in software specification. At last, the application of Property patterns of LTL formulas was given in SPIN tool.

Key words: Linear Temporal Logic (LTL); property patterns; model checking; SPIN

模型检查^[1,2]在软件开发中扮演非常重要的作用, 模型检查的基本思想是使用 Kripke 结构^[3] $M = (S, R, L)$ 描述软件规格模型, 使用时态逻辑^[4~6] 描述软件应具备的特性 ϕ , 然后使用一个算法检查 $M, s \models \phi$ 。这里 Kripke 结构 M 是面向机器, 特性 ϕ 是面向人。要确认 $M, s \models \phi$, 就是 $L(M) \cap L(\phi)$ 为空。可以看出如何刻画特性 ϕ 非常重要。这里的特性是指期望系统具备的特性, 特性模式就是将这些常用的特性用 LTL 公式加以刻画。

1 线性时态逻辑

1.1 语义域定义

一个逻辑系统通常由语法、语义域和语法到语义域映射三个部分构成。这里先给出语义域的定义。

定义 1 标记模态模型: 对原子谓词 $q \in Q$, 一个模态模型是形如 $M = (S, \xrightarrow{a}_{a \in \Sigma}, V)$ 的元组。其中 S 为状态集; $\xrightarrow{a}_{a \in \Sigma}: S \times S$ 为两个状态经过标记 a 的转换关系; $V: q \mapsto 2^S$ 是赋值映射, 它求得状态集 $V(q) \subseteq S$, 即使 q 为真的所有状态集合。

通常, 称 $T = (S, \xrightarrow{a}_{a \in \Sigma})$ 是标记转换系统。原子谓词又被称为原子命题, 通常假设命题集 Q 为可数、无限集合。通常要求模态模型是完备的, 即 $\forall s \in S, \exists a, \exists s'$ 使得 $s \xrightarrow{a} s'$, 这

样可以处理无限的运行。本文假设所有的模型和转换系统是完备的。

增强模态逻辑描述能力的方法之一是将模态模型中的状态改为运行或迹。

定义 2 运行: 一个在标记模态模型 $M = (S, \xrightarrow{a}_{a \in \Sigma}, V)$ 的运行是一个 ω 序列 $r = s_0 \xrightarrow{a_1} s_1 \xrightarrow{a_2} \dots$ 。

这里引入记号 $r(i) = s_i$ 表示第 i 个状态; $r^i = s_i \xrightarrow{a_{i+1}} s_{i+1} \xrightarrow{a_{i+2}} \dots$ 表示运行 r 的第 i 个以后部分; $\lambda(r, i) = a_i$ 是第 i 个标记。

1.2 线性模态逻辑 LTL

定义 3 LTL 语法

$\phi ::= q \mid \neg q \mid P \wedge R \mid P \vee R \mid P \rightarrow R \mid PUR \mid PWR \mid \diamond P \mid \bigcirc P \mid \square P$

LTL 语法的前部分是人们熟悉的命题逻辑操作符, 时态逻辑操作符 \square 表示总是 (Global); 操作符 \diamond 表示最终 (Eventual 或 Future); 操作符 \bigcirc 表示下个状态 (Next time); 操作符 U 表示直到 (Until); 操作符 W 表示除非 (Unless)。这些操作符间, 有如下关系:

$P \vee R = \neg(\neg R \vee \neg R)$

收稿日期: 2006-02-27; 修订日期: 2006-05-08

基金项目: 国家自然科学基金资助项目 (60373072); 江西省教育厅科技项目 (赣教科技便函字 [2002]01 号)

作者简介: 黎升洪 (1962-), 男, 江西南昌人, 副教授, 博士研究生, 主要研究方向: 形式化方法、软件工程; 缪淮扣 (1953-), 男, 江苏淮阴人, 教授, 博士生导师, 主要研究方向: 形式化方法、软件工程; 张新林 (1963-), 男, 湖南永兴人, 副教授, 博士研究生, 主要研究方向: 形式化方法、软件工程。

$$\begin{aligned}
 P \rightarrow R &= \neg P \vee R = \neg (P \wedge \neg R) \\
 \diamond P &= UP \\
 \square P &= \neg \diamond \neg P = \neg (U\neg P) \\
 PWR &= (\square P) \mid (PUR) = \diamond(\neg P) \rightarrow (PUR) \\
 &= PU(R \mid \square P)
 \end{aligned}$$

定义 4 LTL 语义

$s \models q$ iff $s \in V(q)$, 这里由于标记模态模型 M 为默认前提, 故省略。

$$s \models \neg q \text{ iff } s \not\models q$$

$$s \models P \wedge R \text{ iff } s \models P \text{ 且 } s \models R$$

$s \models \bigcirc P$ iff $\forall s'$ if $s \rightarrow s'$ then $s' \models P$, 其含义为状态 s 经过转换后的所有状态 s' 总是满足谓词 P 。

$r \models PUR$ iff $\exists i \in \omega, r^i \models P$ and $\forall j; 0 \leq j \leq i, r^j \models R$, 其含义为: 对所有 $0 \leq j \leq i$, 如果有 $r^j \models P$ 存在, 则必然 $\exists i \in \omega$ 使得 $r^i \models R$ 成立。

2 模型检查中的特性描述

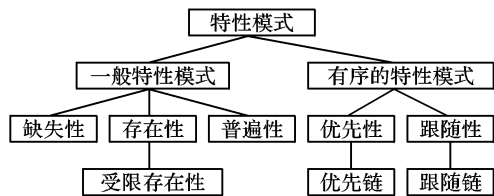


图 1 特性模式的分类

为简化起见, 使用术语“一个给定的状态(或事件)发生。”表示这样一个状态, 其使得给定的状态公式为真; 或在给定的析取事件集的一个事件发生。

(1) 缺失性 (Absence): 在指定的范围内, 给定的状态(或事件)不会发生, 也称为 Never。在模型检查中, 如果指定的范围是全部范围, 称为安全性。

(2) 存在性 (Existence): 在指定的范围内, 给定的状态(或事件)必然发生, 也被称为必然性 Eventually。

(3) 受限存在性 (Bounded Existence): 在指定的范围内, 给定的状态(或事件)必然发生 k 次。这种模式的变体是给定的状态(或事件)在指定的范围内必然最少(最多)发生 k 次。

(4) 普遍性 (University): 在指定的范围内, 给定的状态(或事件)必然处处发生, 即其只包含满足特性的状态(或事

表 1 缺失性模式, 设 P 为假

| 范围 | 特性模式 |
|---------------------|--|
| Globally | $\square(\neg P)$ |
| Before R | $\diamond R \rightarrow (\neg P \cup R)$ |
| After Q | $\square(Q \rightarrow \square(\neg P))$ |
| Between Q and R | $\square((Q \wedge \neg R \wedge \diamond R) \rightarrow (\neg P \cup R))$ |
| After Q until R | $\square(Q \wedge \neg R \rightarrow (\neg PWR))$ |

表 2 存在性模式, 设 P 为真

| 范围 | 特性模式 |
|---------------------|---|
| Globally | $\diamond(P)$ |
| Before R | $\neg RW(P \wedge \neg R)$ |
| After Q | $\square(\neg Q) \mid \diamond(Q \wedge \diamond P)$ |
| Between Q and R | $\square(Q \wedge \neg R \rightarrow (\neg RW(P \wedge \neg R)))$ |
| After Q until R | $\square(Q \wedge \neg R \rightarrow (\neg RU(P \wedge \neg R)))$ |

表 3 受限存在性模式, 设转换到状态 P 最少二次

| 范围 | 特性模式 |
|---------------------|---|
| Globally | $(\neg PW(PW(\neg PW(PW\square\neg P))))$ |
| Before R | $\diamond R \rightarrow ((\neg P \wedge \neg R)UR \mid ((P \wedge \neg R)UR \mid ((\neg P \wedge \neg R)UR \mid ((P \wedge \neg R)UR \mid (\neg PUR))))))$ |
| After Q | $\diamond Q \rightarrow (\neg QU(Q \wedge (\neg PW(PW(\neg PW(PW\square\neg P))))))$ |
| Between Q and R | $\square((Q \wedge \diamond R) \rightarrow ((\neg P \wedge \neg R)UR \mid ((P \wedge \neg R)UR \mid ((\neg P \wedge \neg R)UR \mid ((P \wedge \neg R)UR \mid (\neg PUR))))))$ |
| After Q until R | $\square(Q \rightarrow ((\neg P \wedge \neg R)UR \mid ((P \wedge \neg R)UR \mid ((\neg P \wedge \neg R)UR \mid ((P \wedge \neg R)UR \mid (R \mid (\neg PWR) \mid \square P))))))$ |

件), 也称为 Henceforth and Always。在模型检查中, 如果指定的范围是全部范围, 则称为活性。

(5) 状态(或事件)的优先性 (Precedence): 在指定的范围内, 状态(或事件) P 总是优先于状态(或事件) Q 。优先性被用来描述状态(或事件)间的关系, 状态(或事件) P 是状态(或事件) Q 发生的前置条件, 也可以说状态(或事件) Q 由状态(或事件) P 使能(enable)。

(6) 状态(或事件)的跟随性 (Response): 在指定的范围内, 状态(或事件) P 总是尾随于状态(或事件) Q 。跟随性被用来描述状态(或事件)间的因果关系, 状态(或事件) Q 称为因, 状态(或事件) P 称为果。也称为 Follows and Leads-to。

(7) 状态序列(或事件序列)的优先性 (Precedence Chain): 在指定的范围内, 状态序列(或事件序列) $P_1 \cdots P_n$ 总是先于状态序列(或事件序列) $Q_1 \cdots Q_n$ 。

(8) 状态序列(或事件序列)的跟随性 (Response Chain): 在指定的范围内, 状态序列(或事件序列) $P_1 \cdots P_n$ 总是先于状态序列(或事件序列) $Q_1 \cdots Q_n$ 。

3 特性模式作用范围

每个模式有一个作用范围, 即程序的作用范围, 在该范围内模式必须保持为真。常见的五种范围是: global(全局)、before(在……之前)、after(在……之后)、between(在……之间)和 after-until(在之后……直到)。global 表示整个程序执行范围模式为真; before 表示在给定的事件/状态发生之前模式为真; after 表示在给定的事件/状态发生之后模式为真; between 表示在一个给定的事件/状态发生之后和另一个给定的事件/状态之前模式为真; after-until 和 between 类似, 不同之处在于 between 强调事件之间的发生情况, 而 after-until 即使第二个事件/状态不发生, 第一个事件将继续执行。

需要注意, 特性模式的作用范围由用来说明该特性模式的起始和结束事件/状态决定, 即特性模式的作用范围始于包括起始事件/状态的所有事件/状态, 直到但不包括结束事件/状态。

4 特性模式的 LTL 描述

根据特性模式的划分方法和模式作用范围可以得到常见的 LTL 特性模式表。

表 4 普遍性模式, 设 P 为真

| 范围 | 特性模式 |
|---------------------|---|
| Globally | $\Box(P)$ |
| Before R | $\Diamond R \rightarrow (PUR)$ |
| After Q | $\Box(Q \rightarrow \Box(P))$ |
| Between Q and R | $\Box((Q \wedge \neg R \wedge \Diamond R \rightarrow (PUR)))$ |
| After Q until R | $\Box(Q \wedge \neg R \rightarrow (PWR))$ |

表 5 优先性模式, 设 S 优先于 P

| 范围 | 特性模式 |
|---------------------|--|
| Globally | $\neg PWS$ |
| Before R | $\Diamond R \rightarrow (\neg PU(S R))$ |
| After Q | $\Box \neg Q \Diamond(Q \wedge (\neg PWS))$ |
| Between Q and R | $\Box((Q \wedge \neg R \wedge \Diamond R) \rightarrow (\neg PU(S R)))$ |
| After Q until R | $\Box(Q \wedge \neg R \rightarrow (\neg PW(S R)))$ |

表 6 跟随性模式, 设 S 跟随于 P

| 范围 | 特性模式 |
|---------------------|--|
| Globally | $\Box(P \rightarrow \Diamond S)$ |
| Before R | $\Diamond R \rightarrow (P \rightarrow (\neg RU(S \wedge \neg R)))UR$ |
| After Q | $\Box(Q \rightarrow \Box(P \rightarrow \Diamond S))$ |
| Between Q and R | $\Box((Q \wedge \neg R \wedge \Diamond R) \rightarrow (P \rightarrow (\neg RU(S \wedge \neg R)))UR)$ |
| After Q until R | $\Box(Q \wedge \neg R) \rightarrow ((P \rightarrow (\neg RU(S \wedge \neg R)))WR)$ |

表 7 优先链模式一, 设为二个原因, 一个结果的优先链模式, 即 (S, T) 优先于 P

| 范围 | 特性模式 |
|---------------------|--|
| Globally | $\Diamond P \rightarrow (\neg PU(S \wedge \neg P \wedge \Diamond(\neg PUT)))$ |
| Before R | $\Diamond R \rightarrow (\neg PU(R (S \wedge \neg P \wedge \Diamond(\neg PUT))))$ |
| After Q | $(\Box \neg Q) (\neg QU(Q \wedge \Diamond P \rightarrow (\neg PU(S \wedge \neg P \wedge \Diamond(\neg PUT))))))$ |
| Between Q and R | $\Box((Q \wedge \Diamond R) \rightarrow (\neg PU(R (S \wedge \neg P \wedge \Diamond(\neg PUT))))))$ |
| After Q until R | $\Box(Q \rightarrow (\Diamond P \rightarrow (\neg PU(R (S \wedge \neg P \wedge \Diamond(\neg PUT))))))$ |

表 8 优先链模式二, 设为一个原因, 二个结果的优先链模式, 即 P 优先于 (S, T)

| 范围 | 特性模式 |
|---------------------|---|
| Globally | $(\Diamond(S \wedge \Diamond T) \rightarrow ((\neg S)UP))$ |
| Before R | $\Diamond R \rightarrow ((\neg(S \wedge (\neg R) \wedge \Diamond(\neg RU(T \wedge \neg R))))U(R P))$ |
| After Q | $(\Box \neg Q) ((\neg Q)U(Q \wedge ((\Diamond(S \wedge \Diamond T) \rightarrow ((\neg S)UP))))))$ |
| Between Q and R | $\Box((Q \wedge \Diamond R) \rightarrow ((\neg(S \wedge (\neg R) \wedge \Diamond(\neg RU(T \wedge \neg R))))U(R P)))$ |
| After Q until R | $\Box(Q \rightarrow (\neg(S \wedge (\neg R) \wedge \Diamond(\neg RU(T \wedge \neg R))))U(R P) \Box(\neg(S \wedge \Diamond T)))$ |

表 9 跟随链模式一, 设二个刺激, 一个响应的跟随链模式, 即 P 跟随于 (S, T)

| 范围 | 特性模式 |
|---------------------|--|
| Globally | $\Box(S \wedge \Diamond T \rightarrow \Diamond(\Diamond(T \wedge \Diamond P)))$ |
| Before R | $\Diamond R \rightarrow (S \wedge \Diamond(\neg RUT) \rightarrow \Diamond(\neg RU(T \wedge \Diamond P)))UR$ |
| After Q | $\Box(Q \rightarrow \Box(S \wedge \Diamond T \rightarrow \Diamond(\neg TU(T \wedge \Diamond P))))$ |
| Between Q and R | $\Box((Q \wedge \Diamond R) \rightarrow (S \wedge \Diamond(\neg RUT) \rightarrow \Diamond(\neg RU(T \wedge \Diamond P)))UR)$ |
| After Q until R | $\Box(Q \rightarrow (S \wedge \Diamond(\neg RUT) \rightarrow \Diamond(\neg RU(T \wedge \Diamond P)))U(R \Box(S \wedge \Diamond(\neg RUT) \rightarrow \Diamond(\neg RU(T \wedge \Diamond P))))$ |

表 10 跟随链模式二, 设一个原因, 二个结果的优先链模式, 即 (S, T) 跟随于 P

| 范围 | 特性模式 |
|---------------------|--|
| Globally | $\Box(P \rightarrow \Diamond(S \wedge \Diamond T))$ |
| Before R | $\Diamond R \rightarrow (P \rightarrow (\neg RU(S \wedge \neg R \wedge \Diamond(\neg RUT))))UR$ |
| After Q | $\Box(Q \rightarrow \Box(P \rightarrow (S \wedge \Diamond T)))$ |
| Between Q and R | $\Box((Q \wedge \Diamond R) \rightarrow (P \rightarrow (\neg RU(S \wedge \neg R \wedge \Diamond(\neg RUT))))UR)$ |
| After Q until R | $\Box(Q \rightarrow (P \rightarrow (\neg RU(S \wedge \neg R \wedge \Diamond(\neg RUT))))U(R \Box(P \rightarrow (S \wedge \Diamond T))))$ |

表 11 受限链模式, 设即无 Z 的 (S, T) 跟随于 P

| 范围 | 特性模式 |
|---------------------|---|
| Globally | $\Box(P \rightarrow \Diamond(S \wedge \neg Z \wedge \Diamond(\neg ZUT)))$ |
| Before R | $\Diamond R \rightarrow (P \rightarrow (\neg RU(S \wedge \neg R \wedge \neg Z) \wedge \Diamond((\neg R \wedge \neg Z)UT)))UR$ |
| After Q | $\Box(Q \rightarrow \Box(P \rightarrow (S \wedge \neg Z \wedge \Diamond(\neg ZUT))))$ |
| Between Q and R | $\Box((Q \wedge \Diamond R) \rightarrow (P \rightarrow (\neg RU(S \wedge \neg R \wedge \neg Z \wedge \Diamond((\neg R \wedge \neg Z)UT))))UR)$ |
| After Q until R | $\Box(Q \rightarrow (P \rightarrow (\neg RU(S \wedge \neg R \wedge \neg Z \wedge \Diamond((\neg R \wedge \neg Z)UT))))U(R \Box(P \rightarrow (S \wedge \neg Z \wedge \Diamond(\neg ZUT))))$ |

5 实例

特性模式已广泛用于 SPIN^[7-9]。SPIN 的工作原理如下:

- 1) 将每个进程翻译为有限状态自动机。
- 2) 构造所有进程有限状态自动机的异步交叉积, 形成一个新的全局有限状态自动机(通常称为状态空间)。

- 3) 将特性 LTL 公式翻译为一个自动机。
- 4) 计算自动机与全局有限状态自动机的同步积, 形成一个自动机。
- 5) 检查最后这个自动机的语言是否为空。如果为非空, 则所描述的进程满足指定的特性模式。
- 6) SPIN 中, LTL 公式用来描述坏的特性模式, 如果自动机的语言为非空, 则知道描述的系统不能满足特性模式要求, 此时可以发现反例, 这对调试非常有用。

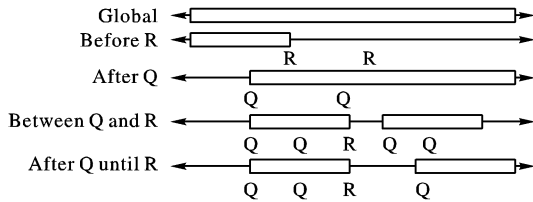


图 2 特性作用范围

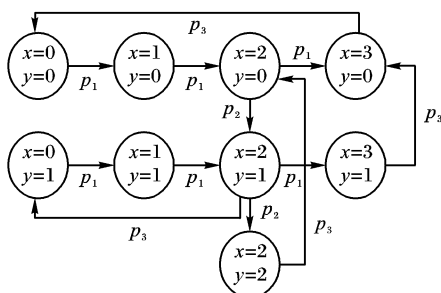


图 3 特性模式的分类

这里, 给出一个如图 3 所示的状态图, 其对应的 SPIN 代码如下所示:

```

int x = 0, y = 0;
proctype p1() {
  do
    :: atomic{ (y < 2) && (x < 3) -> x = x + 1 }
  od
}
proctype p2() {
  do
    :: atomic{ (x + y) == 3 -> x = 0 }
    :: atomic{ (x + y) == 4 -> y = 0 }
  od
}
proctype p3() {
  do
    :: atomic{ (y < 2) && (x < 3) -> x = x + 1 }
  od
}
init {
  run p1();

```

```

run p2();
run p3();

```

在 SPIN 4.2.5 下, 分别验证其下列特性:

- 1) 安全性示例: 验证系统是否满足 LTL 公式 $\square \diamond (x + y = 1)$, 通过 SPIN 验证和仿真功能, 发现 (x, y) 的冲突序列为: $(0, 0), (1, 0), (2, 0), (2, 1), [(2, 2), (0, 2), (1, 2)]^*$ 。
- 2) 活性示例: 验证系统是否满足 $\square (2x > y)$, 通过 SPIN 验证和仿真功能, 发现冲突序列为: $(0, 0)$ 。
- 3) 响应性示例: 验证系统是否满足 $\square ((x + y = 1) \Rightarrow \diamond (x + y = 3))$, 通过 SPIN 验证功能, 特性可满足。

6 结语

高层次的定义和抽象是自动编写形式化规格说明的重要方法, 线性时态逻辑特性模式特别适用于有限状态验证工具 SPIN 中的规格说明。本文从特性模式的分类和作用范围描述 LTL 公式, 对自动编写系统规格说明特性有很大帮助, 如文献[7]给出了 SPIN 中常用的特性模式是其应用的一个方面。

参考文献:

- [1] CLARKE EM, SCHLINGLOFF BH. Model Checking[A]. Handbook of Automated Reasoning[C]. Band II, S. Elsevier, 2001. 1637 - 1790.
- [2] BÉRARD B, BIDOIT M, FINKEL A. Systems and Software Verification: Model-Checking Techniques and Tools[M]. Berlin: Springer, 1999.
- [3] KRIPKE SA. Semantical considerations on modal logic[J]. Acta Philosophica Fennica, 1963, 16: 83 - 94.
- [4] PNUELI A. The Temporal Semantics of Concurrent Programs[A]. Proceedings of the International Symposium on Semantics of Concurrent Computation, Lecture Notes In Computer Science[C]. Springer-Verlag, 1979, Vol 70.
- [5] BEN-ARI M, PNUELI A, MANNA Z. The Temporal Logic of Branching Time[J]. Acta Informatica, 1983, 20(3): 207 - 226.
- [6] EMERSON EA, HALPERN JY. "Sometimes" and "Not Never" revisited: on branching versus linear time temporal logic[J]. Journal of the ACM, 1986, 33(1): 151 - 178.
- [7] HOLZMANN GJ. Spin Model Checker: The Primer and Reference Manual[M]. New York: Addison Wesley, 2003. 608.
- [8] HOLZMANN GJ. Design and validation of computer protocols[M]. London: PRENTICE-HALL, 1991. 22 - 78.
- [9] HOLZMANN GJ. The Model Checker SPIN[J]. IEEE transactions on software engineering, 1997, 23(5).

(上接第 1889 页)

参考文献:

- [1] HOPPER F. Fuzzy cluster analysis[M]. Chichester: John Wiley, 1999.
- [2] HAN JW, KAMBER M. Data mining: Concept and Techniques[M]. San Mateo: Morgan Kaufmann, 2001.
- [3] BEZDEK JC. Pattern recognition with fuzzy objective function algorithms[M]. New York: Plenum Press, 1981.
- [4] 沈红斌, 王士同. 离群模糊核聚类算法[J]. 软件学报, 2004, 15(7): 1021 - 1029.
- [5] YEUNG DS, WANG XZ. Improving Performance of Similarity-Based Clustering by Feature Weight Learning[J]. IEEE Transactions On PAMI, 2002, 24(4): 556 - 561.
- [6] CHANA EY, CHINGA WK. An optimization algorithm for clustering using weighted dissimilarity measures[J]. Pattern Recognition, 2004, 37(5): 943 - 952.
- [7] FRIGUI H, NASRAOUI O. Unsupervised learning of prototypes and attribute weights[J]. Pattern Recognition, 2004, 37: 567 - 581.
- [8] 张莉, 周伟达, 焦李成. 核聚类算法[J]. 计算机学报, 2002, 25(6): 587 - 590.
- [9] GIROLAMI M. Mercer Kernel - Based Clustering in Feature Space[J]. IEEE Transactions on Neural Networks, 2002, 13(3): 780 - 784.
- [10] 沈红斌, 杨杰, 王士同. 基于信息理论的合作模糊聚类算法研究[J]. 计算机学报, 2005, 8: 1287 - 1294.
- [11] 王焜, 陆文聪. 宋代汝窑古瓷的微量元素——支持向量机算法研究[J]. 计算机与应用化学, 2004, 2: 191 - 194.