归纳法推理中的各种推理策略*

李卫华　张　黔　龙　泉

(武汉大学计算机科学系　武汉　430072)

摘要　分元符删除;项推广;无关式删除;交融是归纳法推理系统中一些重要的推理策略. 文中逐一介绍了这些推理策略,给出了使用这些推理策略的方法,列出了不同推理策略的编译 LISP 语言实现.

关键词　分元符删除,项推广,无关式删除,交融,归纳法推理.

归纳法推理是以数学归纳法、结构归纳法为指导思想,用递归函数为工具,综合多种证明策略的一种证明方法. 它完全取消了谓词,代之以函数,使用一套有效的启发式方法指导归纳法推理.[1~5]

本文介绍的分元符删除策略旨在将公式化为不含劣项的子句集;项推广策略旨在将待证公式转换成较强的公式以达到证明更一般公式的目的;无关式删除策略旨在删除待证公式中的无关假设,以达到简化或反驳原公式的目的;交融策略旨在合理使用归纳法假设.

1　分元符删除策略

定义 1. 称公式(Implies test (equal left var))为一删除引理,称(d v_1 … v_n)为分元符项的条件是:var 为一变量;left 中至少存在一个形如(d v_1 … v_n)的真子项,其中 d 为一函数符,v_i 是公式中互不相同的变量;var 仅出现在 left 的(d v_1 … v_n)项中.

例如,下面是 2 条删除引理,其中分元符项分别为(sub1 x),(car x)和(cdr x):

```
(rewrite-rule sub1-elim ;对数据类型{add1}
  ((number? x) (not (equal x '0))) (equal (add1 (sub1 x)) x) ())
(rewrite-rule cdr-car-elim ;对数据类型{cons}
  ((list? x)) (equal (cons (car x) (cdr x)) x) ())
```

一个公式若满足下述条件,则可对该公式作用分元符删除策略:

·此公式中的函数 f 在引入其定义时有对应的删除引理特性;

·函数对应的分元符项中存在某个参量 v_i,或等于删除引理中的 var,或自身也是一个可

---

*本文研究得到国家863高科技项目和国家教委跨世纪优秀人才基金资助. 作者李卫华,1952年生,教授,博士导师,主要研究领域为人工智能,知识工程,多媒体软件. 张黔,女,1973年生,硕士生,主要研究领域为归纳法推理,多媒体软件. 龙泉,1970年生,博士生,主要研究领域为归纳法推理,多媒体软件.

本文通讯联系人:李卫华,武汉430072,武汉大学计算机科学系

本文1995－08－31收到修改稿

作用删除策略的公式；

·若其参量等于引理中的 var，则此参量不能再出现在公式的其它地方.

(elim-destructors-cand? term)判断可否用删除引理对 term 作用分元符删除策略.

设公式 P 中含有项(d x_1 … x_i … x_n)，该项是某删除引理 (Implies test (equal left v_i))中分元符项(d v_1 … v_i … v_n)的例示. 将删除引理作用于待证公式 P 后得到 P_1 与 P_2 的合取式：P_1：(Implies (not $test_1$) P)，弱化情况；

P_2：(Implies (and $test_1$(equal $left_1$ x_i)) P)，其中 $test_1$，$left_1$ 是用 x_i 替代 v_i 后的结果.

我们知道，若(Implies u v)成立，则(Implies (and t u) v)亦成立，故可将关于分元符项的定理 gen 加入假设之中，得：P_3：(Implies (and gen $test_1$(equal $left_1$ x_i)) P).

再用一个未出现在 P_3 中的变量替代 $left_1$ 中的分元符项得：

P_4：(Implies (and gen_1 $test_2$(equal $left_2$ x_i)) P′)，其中 gen_1，$test_2$，$left_2$，P′是用变量替代后的结果. 最后在 P_4 中用 $left_2$ 替代 x_i，删去假设(equal $left_2$ x_i)得：

P_5：(Implies (and gen_2 $test_3$) P″)，其中 gen_2，$test_3$，P″均为用 $left_2$ 替代 x_i 后的结果，这是推广情况. 经上述变换后，对 P 的推理就转换成对 P_1 和 P_5 的推理. (elim-destructors-cl cl hist)实现分元符删除策略，其中，(elim-destructors-cands cl)找出子句 cl 中的候选分元符项. (sort-destructor-cands lst)对候选分元符项排序. 因为在对待证公式作用分元符删除策略时，公式中有时存在多个分元符，例如 sub1，difference 会出现在同一待证公式中，这就存在一个先处理哪个分元符的问题. 选择时，应先删除最简单的分元符项. 因此，(sort-destructor-cands lst)还要判断分元符谁复杂，谁简单(若函数 A 在函数 B 之前引入定义，且函数 B 的定义中需依赖函数 A，则称函数 A 比函数 B 简单). 系统用 get-level-no 取出函数的复杂度. 这个复杂度是系统在引入某函数的定义时计算出来的，并将此复杂度通过(put-level-no fn)加入到 level-no 特性中. 一个函数将比其定义体中任何函数具有相等或更高的复杂度，递归定义的函数比非递归函数具有更高的复杂度.

2 推广策略

定义 2. 设公式 A，B 中 A 是 B 的例示，即 B 比 A 更一般，则称 B 比 A 强.

在归纳法推理过程中，常常需证明比待证公式更强的公式. 例如，假定要证公式 P：(p (f x_1 … x_n))，其中 f 是满足定义条件的递归函数；同时假定函数 f 的定义体为：(defn (f x_1 … x_n) (h (f y_1 … y_n)))，其中(y_1 … y_n)与(x_1 … x_n)按良基关系下降. 对 P 式作用归纳法模式自动生成策略[3]后得归纳步骤：(Implies (p (f y_1 … y_n)) (p (f x_1 … x_n)))
展开 f 的定义，并对上式作用简化策略[5]后得：

P_1：(Implies (p (f y_1 … y_n)) (p (h (f y_1 … y_n))))

由上可见，(f y_1 … y_n)同时出现在归纳假设和归纳结论两边. 这时，可引入一新变量 z，并用 z 替代(f y_1 … y_n)得 P_2：(Implies (p z) (p (h z)))，若能证明 P_2，则把 P_2 中的 z 例示成(f y_1 … y_n)即可证明 P_1；同时找到一条证明 P_1 的可行方法，即先证明 P_2. 这就是引入推广策略的思想. 推广策略由 2 部分组成，选推广项和作用推广策略：

```
(define (gen-cl cl hist)
  (call/cc (lambda (return)
```

```
(let ((commonsubterms ()))
  (when (not (assq 'being-proved stack)) (return ())) ;推广策略应在归纳法推理过程中运用
  (set! commonsubterms (genrlterms cl)) ;寻找最小公共子项
  (if (null? commonsubterms)
      (return ()) ;没有公共子项,将无法作用推广策略
      (begin ;生成推广后的子句
        (set! process-clauses (list (generalize1 cl commonsubterms gen-var-names1)))
        (set! process-hist ;记录作用推广策略的历史
              (list gen-skos commonsubterms obvious-restricts gen-lemma-names))
        (set! all-lemmas-used ;将推广引理加入已用引理集
              (union gen-lemma-names all-lemmas-used))
        (return t)))))
```

定义3. 称项可推广是指,该项不是变量,不是(quote …)表达式,不是显式值模板,且该项的函数符不提示一个分元符删除引理.

定义如下的gen?来判定一个项可否推广:

```
(define (gen? x)
  (cond ((almost-value? x) ()) ;x 是否为变量、quote 表达式或显式值模板?
        ((and (set! tmp (getprop (car x) 'elim-destructors-seq))
              (not (disabled? (access rewrite-rule name tmp))))
         ()) ;函数是否提示分元符删除引理?
        (t t)))
```

从推广策略的引入可以看出,系统需用一个新变量去推广待证公式中的公共子项,于是系统将找出满足以下条件之一的子项:

·子项同时出现在2个或多个子句中;

·某子句中含(equal left right),子项同时出现在等式两边.

找到的子项x应具有属性P:(p x),表明x不是变量、quote表达式、显式值,且不存在任何x的子项也具有属性P.

为找出2个子句A,B的公共子项,(comsubterms t_1 t_2)采用以下步骤:

(1)将2子句中元素较多的子句放入B中;

(2)若A为变量或quote表达式,则判断A是否出现在B中;

(3)对于A的每一个参量均递归地判断是否出现在B中;

(4)若某个A出现在B中,且A不为显式值、底对象或外壳名,A的函数符也不提示一个分元符删除引理,则A为公共子项的一部分;

(5)重复(2)~(4)的操作,直到找出最小的公共子项.

系统在作用推广策略时,引入一个新变量来替代公式中的公共子项,由于变量的类型集为type-set-unknown,这个替代将使推广后的公式过于一般,若已知公共子项具有一定的类型限制,那么可将这个限制作用到新加变量上.

仍以上面的P式为例,假定存在关于f的已证推广引理P_4:(gen (f x_1 … x_n)),再假定(f y_1 … y_n)是(f x_1 … x_n)的例示.考虑上文中的P_2式,(f y_1 … y_n)是一个公共子项,因为P_4为已证引理,则P_4的一个例示P_5:(gen (f y_1 … y_n))也为真.将P_5作为一个假设加入P_2中得:

P_6:(Implies (and (gen (f y_1 … y_n)) (p (f y_1 … y_n))) (p (h (f y_1 … y_n))))

因P_5为真,若P_6为真,则P_2必为真.用一个新变量z替代公共子项(f y_1 … y_n)后得:

P_7:(Implies (and (r z) (p z)) (p (h z))).

观察 P_7,它比 P_3多了一个假设限制(r z).若已知公共子项返回唯一的外壳对象,则把新变量限制在那个外壳类中,例如,若(f y_1 … y_n)总返回一张表,则可增加(list? z)作为新公式的额外假设.若公共子项的类型集中含有多个类型,则相应的限制就是一个析取式.

为了对要推广的项作用类型限制,(get-types term cl)采用如下步骤:

(1)计算公共子项的类型集;

(2)若公共子项返回某外壳类型的一个对象,则将此外壳识别符作用到公共子项上作为推广的类型限制;

(3)例示推广引理,并生成带类型限制的项.

用以上步骤可找到最小公共子项和作为假设的推广引理,同时也得出了公共子项的类型限制,下面就可用未在公式中出现的新变量替代要推广的公共子项.用(define (generalize1 cl subterml var-names) …)对公式作用推广策略之后,将再次作用简化策略.

3 无关式删除策略

在证明待证公式时,作用无关式删除策略的目的是要在用归纳法模式自动生成策略之前删去该公式中的无关假设,以简化后续推理工作.在反驳一个待证公式时,通过作用该策略删除公式中的无关式,若公式删除后得 F,则达到了反驳目的.

定义4. 称一个项为原语是指,项是一个变量;项是 quote 表达式或项满足以下2个条件:

·项的函数未引入定义或项为 not 表达式;

·项的每个参量均为原语.

定义5. 称某集合可删除是指,集合中所有的项均为原语或集合中仅含一个项;

·若项形为(f x_1 … x_n),则 x_1,…,x_n 均为不同的变量;

·若项形为(not (f x_1 … x_n)),则 x_1, …,x_n 均为不同的变量.

无关式删除策略主要是在公式中找出与剩余部分完全没有联系的假设.系统按以下步骤完成对无关式的删除:

·确定系统正处在归纳法推理的过程中;

·对待证公式分区,对公式中的任意2个子句,若它们存在一些公共变量,则将这2个公式置于同一分区中.对公式中的子句集按上述分区方法进行传递闭包计算,最终得到待证公式的分区表;

·用(eliminable? set)判断每个分区可否删除;

·若某子句在可删除分区中,则从待证公式的子句集中去掉该子句.

```
(define (elim-irrelevance-cl cl hist)
  (call/cc (lambda (return)
    (let ((partit ()) (eliminables ()))
      (when (not (assq 'being-proved stack)) (return ())) ;是否正在归纳证明
      (set! partit ;利用闭包计算对子句进行分区
        ;(transitive-closure sets pred)计算集合 sets 在关系 pred 上的传递闭包
        (transitive-closure
          (for lit in cl save (cons (all-vars lit) (list lit)))
          (lambda (x y) ;集合中每一元素形为(子句变量 子句)
```

```
        (if (intersect? (car x) (car y)) ;两元素是否有公共变量?
            (cons (union-equal (car x) (car y)) ;对两个元素求集合并
                  (union-equal (cdr x) (cdr y)))
            ())))))
(setq eliminables (do ((pair partit (cdr pair)) (lans () lans))
                      ((null? pair) lans)
                      (when (eliminable? (cdar pair)) ;判断分区可否删除?
                        (setq lans (nconc (cdar pair) lans)))))
(if (null? eliminables)
    (return ())
    (begin (setq process-clauses ;保存不在可删除分区中的子句
                 (list (for lit in cl when (not (memq lit eliminables)) save lit)))
           (setq process-hist ()) (return t)))))))
```

4　交融策略

定义6. 在待证公式中,若存在形如(equal left right)的等式假设,在待证公式中可用left替代right(或用right替代left),并删去此等式假设.这种推理策略称为交融.

归纳法推理的主要思想是:为证明公式(P x),构造出公式的归纳法模式(Implies (P x′) (P x)),尽可能地将归纳结论(P x)简化为含x′的形式,以便利用归纳假设.

若待证公式形为(equal left right),对此公式作用归纳法,并简化归纳步骤,其结果为P:(Implies (equal $left_1$ $right_1$) (equal left (f $right_1$))).对于P式的处理有2种方案:

(1)对于上式直接作用归纳法有一定困难,因为形如(Implies a b)的公式,其归纳步骤为(Implies (Implies a_1 b_1) (Implies a b)),要证归纳步骤为真,实为证明当a_1为假时,(Implies a b)为真.而a_1此时是一个equal表达式,equal表达式的否定较难证明,故对P式直接作用归纳法的方法不可取.

(2)一个归纳法无法证明公式(equal left right)的原因在于:当把right简化为$right_1$时,left无法被简化为相应的$left_1$,left与right要求作用不同的归纳法模式.若用$left_1$替代$right_1$,将公式(Implies (equal $left_1$ $right_1$) (equal left (f $right_1$)))变换成(equal left (f $left_1$)),经常可以提示一个适合等式2边的归纳法.

于是,系统在对一个公式作用了简化策略之后,尽可能的删去合适的分元符,再试图利用所有的等式假设,并抛弃这些假设.

对于待证公式P,若其子句集中有形为(not (equal left right))的子句(即存在等式假设),就有可能对公式作用交融策略.若left(或right)满足以下条件,则可用left(或right)对right(或left)进行交融:

·left(或right)的函数不是已引入的外壳;

·left(或right)或者为变量,或者其函数不提示一个分元符删除引理;

·待证公式子句集中存在一个不同于等式假设的子句,使得left(或right)出现在这个子句中;

·推理过程的历史记录表明以前没有使用同一个等式假设进行交融.

系统用(fertilize-feasible lit cl term hist)来判定是否可用term对公式作用交融.

```
(define (fertilize-feasible lit cl term hist)
  (and (not (almost-value? term)) ;term 的函数是否为引入的外壳?
       (or (variable? term) ;term 是否为变量?
           (not (sko-dest-nest? term ()))) ;term 的函数是否提示一个分元符删除引理?
       (do ((lit2 cl (cdr lit2)))
           ((or (null? lit2) (and (not (eq? (car lit2) lit)) (occur term (car lit2))))
            (if (null? lit2) () t))) ;term 是否出现在其它子句中?
       (not (do ((entry hist (cdr entry)) (lhs () lhs) (rhs () rhs))
                ((or (null? entry)
                     (and (set| tmp (car entry)) ;hist 中是否有同一记录?
                          (match tmp (fertilize-clause & & & & lhs rhs &)) ;匹配模式
                          (equal? (fargn (fargn lit 1) 1) lhs)
                          (equal? (fargn (fargn lit 1) 2) rhs)))
                 (if (null? entry) () t))))))
```

一旦确定交融,系统更倾向于将复杂度低的表达式向复杂度高的表达式交融.

定义7. 一个表达式的复杂度定义为:

·变量的复杂度为0;

·(quote …)表达式的复杂度为0;

·其它表达式的复杂度为其函数的复杂度与其参量的复杂度的最大值之和.

系统用(complexity term)来确定 term 的复杂度.

当待证公式中存在等式假设(equal left right)时,系统需要判断可否作用交融策略?是用 left 交融 right,还是用 right 交融 left?

·若既可用 left 对公式进行交融,又可用 right 对公式进行交融,则判断 left 和 right 的复杂度,若 left 的复杂度较低,则用 left 交融此公式(left-for-right),否则用 right 交融此公式(right-for-left);

·若仅可以用 left 对公式进行交融,则交融方向为(right-for-left);

·若仅可以用 right 对公式进行交融,则交融方向为(left-for-right);

·否则不能利用此等式假设进行交融.

系统用(fertilize1 lit cl left right hist)来确定交融方向.

定义8. 把 T,F,某外壳的底对象、将外壳构元符作用到满足类型限制的显式值上的结果均称为显式值.

定义9. 将完全由外壳构元符、底对象和变量组成的非变量项称为显式值模板.

下面给出使用等式假设的正确方法.

(1)给定一个待证公式,在其子句集中寻找形为(not (equal left right))的子句(这就是等式假设).判断这个等式假设是否满足交融条件,在满足条件的基础上确定等式交融方向.

(2)在对待证公式子句集进行具体操作之前,先确定3个控制交融的标志:

·若 left 或 right 为显式值模板,则设置海量替代标志;

·若 left 不是显式值模板、right 不是显式值模板、正在推理归纳法模式中的归纳步骤,则设置删除子句标志;

·若满足以下条件,则设置交叉交融标志:

(a)正在推理过程中;

(b)待证公式的子句集中存在形为(equal $left_1$ $right_1$)的等式,确定的交融方向为用左部

交融公式(left-for-right),且 $right_1$ 出现在 right 中,或者确定的交融方向为用右部交融公式(right-for-left),且 $left_1$ 出现在 left 中;

(c)待证公式的子句集中存在形为(equal $left_1$ $right_1$)的等式,确定的交融方向为用左部交融公式(left-for-right),且 $right_1$ 出现在 left 中,或者确定的交融方向为用右部交融公式(right-for-left),且 $left_1$ 出现在 right 中.

(3)下面具体对待证公式的每一个子句进行处理:

·若未设置删除子句标志且处理的子句不同于等式假设,则保持子句不变;

·若满足下面条件之一,则当交融方向为用左部交融公式(left-for-right)时,将处理子句中所有的 right 替换成 left,否则将处理子句中所有的 left 替换成 right:

(a)处理的子句是不同于原等式假设的另一等式假设(not (equal $left_2$ $right_2$));

(b)设置了海量替代标志;

(c)未设置交叉交融标志.

·若处理子句为一等式结论(equal $left_3$ $right_3$),则当交融方向为用左部交融公式 (left-for-right)时,将处理子句右部($right_3$)中所有的 right 替换成 left,否则将处理子句左部($left_3$)中所有的 left 替换成 right.

系统运用(fertilize-cl cl hist)对待证公式作用交融策略.在对待证公式作用了交融策略之后,将重新对得到的结果简化,若在子句集中有多个等式假设可以使用,每次交融时仅对所找到的第一个等式假设进行处理.其它的等式假设则在简化了新公式并作用了分元符删除策略之后,再次遇到时方予处理.

参考文献

1 Boyer R S, Moore J S 著,李卫华译.计算逻辑.计算机工程与应用,1982,**154(4)**,**155(5)**,1983,**169(7)**.

2 李卫华,张黔,刘娟,石自力.归纳法推理系统.计算机学报,1996,**19(3)**:230～236.

3 李卫华,张黔,张亮,刘娟.归纳法模式的自动生成.软件学报,1996,**7(3)**:168～174.

4 李卫华,张黔,承雪琦.归纳法推理中的子句简化策略,软件学报,1996,**7(增刊)**:558～564.

5 李卫华,张黔,韩波.归纳法推理中的项重写策略,软件学报,1996,**7(增刊)**:565～571.

6 李卫华,陈兆乾,潘金贵.人工智能程序设计.北京:科学出版社,1989.

STRATEGIES IN INDUCTION INFERENCE

Li Weihua　Zhang Qian　Long Quan

(*Department of Computer Science　Wuhan University　Wuhan*　430072)

Abstract　Destructor Eliminating, term generalizing and irrelevance eliminating, fertilizing are several important inference strategies in induction inference system. Starting from the introduction of the above strategies, this paper discusses the basic idea of the strategies and the method of using the strategies. The system has been implemented by using compiler LISP on micro computers.

Key words　Destructor eliminating, term generalizing, irrelevance eliminating, fertilizing, induction inference.