# 一个上下文无关文法获取过程的设计和实现*

张瑞岭

(中国科学院软件研究所计算机科学开放研究实验室 北京 100080)

**摘要** 文章介绍一个基于复用的上下文无关文法获取过程的设计和实现,该过程用于获取以上下文无关文法表示的概念.它从待获取概念的有限实例和句型以及可能复用的已知概念出发,通过一个交互式文法推断过程,最终得到概念的文法定义.

**关键词** 文法推断,示例学习,上下文无关语言,复用.

**中图法分类号** TP18

以有限的实例作为样本,推断出一个文法,使其产生的语言包含这些实例.这个过程属于语言的归纳学习或文法推断.Gold和Feldman分别在文献[1,2]中论述了文法推断的基本理论.在以后的20多年中,许多作者从不同的出发点对文法推断理论或特定的推断算法作了研究.

尽管文法推断问题被讨论了很长时间,但由于计算复杂性等因素的限制,使得该问题的研究进展缓慢.概括地说,以往的研究工作有如下特点:(1) 主要侧重于上下文无关CF(context-free)文法或其子类的研究,大量有效的推断算法都是针对CF子类文法的推断问题;(2) 推断的出发点除了实例样本外,往往附加其他信息,如实例的轮廓信息[3],而这些附加信息通常难以提供;(3) 对推断出的文法一般没有质量要求,因此,推断出的文法通常比较繁琐,并且结构不自然.

本文描述一个基于复用的CF文法的获取过程,其基本原理首先在文献[4~6]中提出.该过程是形式规约获取系统SAQ[7,8]的重要组成部分,SAQ是在概念库的支持下,进行形式规约的获取与检验.我们的CF文法获取过程是人(文法刻画者)与机器间高度交互的一个过程,其指导思想是:充分利用刻画者关于所要定义的语言(不是文法本身)的片断知识,由人与机器交互,推断出符合要求且结构自然的文法.所谓符合要求是指,获取的文法正确地刻画所要定义的语言.所谓结构自然是指,推断出的文法能自然地反映语言的内部结构,从而能自然地刻画其定义的语言的含义(语义).对刻画者的要求是,对所要定义的语言有认识(能够列举出语言的一些句子),能够判断一个串是否为所要定义的语言的合法句子,同时,他对语言的更进一步的片断知识(如能提供语言的一些简单句型)越多,则文法获取过程越快,获取的文法也更能符合要求.

在SAQ系统中,我们把一个CF文法定义的语言称为一个概念,文法定义中的开始非终极符即为概念的名称,其他非终极符称为子概念,文法定义的语言的句子称为概念的实例.其中包含非终极符的句子称为概念的句型.我们的获取模型是:在一次获取过程中同时获取一组相互关联的概念(概念的联立定义).刻画者首先给出待获取概念的实例和句型(后者不是必须的)以及可能复用的已知子概念及其文法定义,系统从这些信息出发进行猜测,猜测结果用文法定义中的产生式表示;对每个猜测,系统根据一系列规则进行取舍,将系统不能决定取舍的猜测再提交给刻画者确认;猜测确认过程结束,系统对获取猜测再进行化简(求精),从而得到更精炼的猜测,最终得到待获取概念的文法定义.

下面我们详细描述获取过程,然后对其作些讨论.

## 1 记号约定

$\lambda$表示空串,$\varnothing$表示空集.

字母表$V=V_N \cup V_T$,其中$V_N$表示非终极符集合,其元素用大写字母$A,B,C,\ldots$表示;$V_T$表示终极符集合,其元素用小写字母$a,b,c,\ldots$表示;$V_N \cap V_T=\varnothing$.

---

* 本文研究得到国家自然科学基金、国家"九五"科技攻关项目基金和国家863高科技项目基金资助.作者张瑞岭,1969年生,博士生,主要研究领域为机器学习.

本文通讯联系人:张瑞岭,北京 100080,中国科学院软件研究所计算机科学开放研究实验室

本文1997-04-29收到原稿,1997-09-05收到修改稿

$V_T^*$ 表示终极符串集合(包括空串).$V^*$ 表示终极符和非终极符串集合,其元素用希腊字母 $\alpha,\beta,\gamma,\ldots$ 表示.

语言 $L$ 是 $V_T$ 上的一个串集,即 $L\subseteq V_T^*$.CF 语言的定义用一个 CF 文法表示.一个 CF 文法 $G$ 是一个四元组,$G=(V_N,V_T,P,X)$,其中 $P$ 为产生式集合;$X$ 为开始非终极符.产生式集合 $P$ 中的产生式形如 $A\to\alpha$,其中 $A\in V_N,\alpha\in V^*$.

在下面的描述中非终极符与概念名等价.

## 2 获取过程

概念的获取过程是一个人机交互过程,其间可以利用已知概念(概念复用)进行归纳学习.亦即从特殊到一般,对刻画者提供的实例进行有根据地推广,推广的结果要得到认可.

获取过程从联立定义的概念的有限实例和句型以及可能复用的已知概念出发,最终得到概念的 CF 文法定义.未知概念的句型是刻画者对概念的片断知识的一种,它与一般产生式右部的区别是其中出现的非终极符较少.对本获取过程来说,未知概念的句型不是必须的.

获取过程包括相继进行的 5 个子过程:系统猜测、猜测确认、猜测求精、空字处理和冗余去除.下面分别加以描述.

### 2.1 系统猜测

获取过程一开始是由刻画者给出未知概念名及其相应的有限数目的实例和句型,如果需要的话,同时给出复用的已知概念名及其相应的文法定义(在 SAQ 系统中,已知概念的定义直接从概念库中提取).系统猜测的实质是,寻找字符串与字符串、字符串与语法定义之间的结构包含或从属关系.

我们把刻画者提供的未知概念 $X$ 的所有实例组成的有穷集合记为 $Sample(X)(\subset V_T^*)$,所有的句型组成的集合记为 $SentForm(X)(\subset V^*)$;将已知概念 $X$ 的语法定义记为 $G(X)$.

**定义 1.** 未知概念 $X$ 的实例 $s$ 的一个项猜测是形如 $X\to\alpha(\alpha\in V^*)$ 的产生式,它满足:对于 $\alpha$ 中出现的每个非终极符(即概念名)$Y$,有 $Sample(Y)$ 中的一个实例(若 $Y$ 是未知概念)或 $G(Y)$ 的一个合法句子(若 $Y$ 是已知概念),当把这些概念名用对应的实例或句子串替换后,即可还原成实例 $s$.若 $\alpha$ 中不含非终极符,则 $X\to\alpha$ 称为实例 $s$ 的一个平凡项猜测.概念 $X$ 的一个实例 $s$ 产生的所有项猜测的右部组成的集合记为 $Term(X,s)$.即

$$Term(X,s)=\{\alpha\mid X\to\alpha \text{ 为 } s \text{ 的一个项猜测}\}$$

**定义 2.** 实例 $s$ 的一个许可分解 $abc(a,b,c\in V_T^*)$ 是指,$s$ 分解为 3 个子串,其中 $b$ 为非空串,$a$ 和 $c$ 不同时为空,且满足:存在一个关知概念 $Y$,有 $b\in Sample(Y)$;或存在一个已知概念 $Y$,有 $b$ 为 $G(Y)$ 的合法句子.

**定义 3.** 实例 $s$ 的全体许可分解组成的集合记为 $ADecomp(s)$.$s$ 的一个许可分解 $abc$ 已在 $ADecomp(s)$ 中出现是指如下两种情形之一.

(1) $ADecomp(s)$ 中存在一元素 $a_1b_1c_1$,满足 $a=a_1,b=b_1,c=c_1$.

(2) $ADecomp(s)$ 中存在一元素 $a_1b_1c_1$,满足 $a_1$ 为 $a$ 的端,$c_1$ 为 $c$ 的尾,$b$ 为 $b_1$ 的真子串.

**定义 4.** 集合 $T_1=\{\alpha\mid\alpha\in V^*\}$ 和 $T_2=\{\beta\mid\beta\in V^*\}$ 的笛卡儿积

$$T_1\times T_2=\{\alpha\beta\mid\alpha\in T_1,\beta\in T_2\}$$

其中 $\alpha\beta$ 表示由 $\alpha$ 和 $\beta$ 连接得到的串,显然,$\alpha\beta\in V^*$.

系统猜测阶段的任务是,由系统对每个未知概念 $X$ 的所有 $s(\in Sample(X))$,依次构造 $Term(X,s)$.构造 $Term(X,s)$ 的过程如下.

(1) 置 $Term(X,s)$ 和 $ADecomp(s)$ 为空;

(2) 求 $s$ 的一个许可分解 $abc$,若不存在新的许可分解,或 $s$ 根本不存在许可分解,则转步骤(4).这里假设 $b$ 对应的概念为 $X_b$,即:若 $X_b$ 为未知概念,则 $b\in Sample(X_b)$;若 $X_b$ 为已知概念,则 $b$ 为 $G(X_b)$ 的合法句子.若该分解在 $ADecomp(s)$ 中出现,则继续求 $s$ 的下一个许可分解;否则,将该分解加入 $ADecomp(s)$,并将猜测 $aX_bc$ 加入 $Term(X,s)$ 中,继续步骤(3);

(3) 对许可分解 $abc$ 中的 $a$,假定一个临时的非终极符 $X_a$,并在本步骤完成后丢弃.若 $a$ 为非空字,则令 $Sample(X_a)=\{a\}$,并利用本算法求 $Term(X_a,a)$(递归调用).若 $a$ 为空字,则令 $Term(X_a,a)=\{\lambda\}$.类似地,对 $c$ 做同样处理,求得 $Term(X_c,c)$,将集合 $Term(X_a,a)\times\{X_b\}\times Term(X_c,c)$ 并入 $Term(X,s)$ 中,继续求 $s$ 的下一个许可分解;

(4) 求 $s$ 的平凡项猜测,即

$$Term(X,s)=Term(X,s)\cup\{s\}$$

系统猜测阶段还对每个未知概念 $X$ 的各个句型 $f(\in SentForm(X))$ 构造猜测集合,其方法如下:设 $f=a_1A_1a_2A_2\ldots a_nA_n$,其中 $a_i\in V_T^*$,$A_i\in V_N(i=1,\ldots,n)$,记由 $f$ 产生的所有猜测构成的集合为 $Term(X,f)$,则

$$Term(X,f)=Term(X,a_1)\times\{A_1\}\times\ldots\times Term(X,a_n)\times\{A_n\}$$

其中若 $a_i=\lambda$，则 $Term(X,a_i)=\{\lambda\}$，否则，用与前面对实例构造猜测集合相同的过程求 $Term(X,a_i)$.

例 1：[4]设待获取概念为 $X=\{a^nb^nc^m\mid n,m=1,2,\ldots\}$，给定样本集合为 $\{abc,aabbc,aaabbbcc\}$，复用已知概念 $X_1$，其定义为 $G(X_1)=(\{X_1\},\{a,b\},P,X_1)$，其中 $P=\{X_1\to ab,X_1\to aX_1b\}$. 系统猜测结果如表 1 所示.

表 1

| 概念名 | 实例 | 系统猜测结果 |
|---|---|---|
| $X$ | $aaabbbcc$ | $X_1cc$ |
| | $aabbc$ | $X_1c$ |
| | $abc$ | $X_1c$ |

## 2.2 猜测确认

系统产生出所有猜测后，下一步将依次决定每个猜测的取舍. 这一步需要刻画者参与决策，即由刻画者回答系统不能决定而提交给刻画者回答的问题.

**定义** 5. 当前接受的产生式集合 *AcceptSet* 是指，目前为止，所有确认接受的猜测和所有未处理(即未决定是否接受)的平凡猜测以及所有已知概念的语法定义.

**定义** 6. 产生式 $X\to\alpha$ 可由产生式集合 $P$ 推出是指，由 $P$ 中产生式可识别出 $\alpha$ 为 $X$ 的句型或实例.

系统依次对每个未知概念的各个样本实例和句型对应的猜测进行取舍处理，同一实例或句型对应的猜测的处理顺序是，先处理通用性相对强的猜测，若这种猜测不被接受，则继续处理通用性相对弱的猜测，最后处理平凡猜测. 我们称猜测 $X\to\alpha$ 比 $X\to\beta$ 的通用性强是指其依次满足以下 1 个或多个条件：① $\alpha$ 中出现的终极符数比 $\beta$ 少；② $\alpha$ 比 $\beta$ 短，即 $\alpha$ 中终极符和非终极符的总数较少；③ $X\to\alpha$ 产生递归，即 $\alpha$ 中出现非终极符 $X$.

系统对每个猜测 $X\to\alpha$ 的取舍处理过程如下：

(1) 计算当前 *AcceptSet*；

(2) 判断 $X\to\alpha$ 是否由 *AcceptSet* 推出，若是，自动予以拒绝，否则，继续下一步；

(3) 产生出有限的、由 $X\to\alpha$ 生成的新实例. 产生新实例的方法是：将 $\alpha$ 中出现的未知概念名用其对应的实例替换，已知概念名用其对应的语法定义生成的句子(生成句子时，限定每个产生式的最大使用次数[6])替换. 将猜测 $X\to\alpha$ 连同其产生的新实例交付刻画者确认是否接受. 很显然，若刻画者不能接受产生的新实例中的任何一个，则猜测 $X\to\alpha$ 就不能被接受.

## 2.3 猜测求精

求精过程是一个再推广过程，相当于在系统猜测过程中，将 $Sample(X)$ 用 $Term(X)$ 替换后再求 $Term(X)$. 它将生成一些更精炼的产生式. 求精过程如下：

(1) 将 *AcceptSet* 中的产生式按右部的长度降序排列；

(2) 依次对 *AcceptSet* 中每个产生式 $X\to\alpha$ 判断是否包含其他某产生式的右部，若不存在这样的 $X\to\alpha$，则求精过程结束；否则，即存在产生式 $X\to\alpha$ 和 $Y\to\beta$(这里 $X$ 和 $Y$ 可以是同一非终极符)，$\beta$ 为 $\alpha$ 的子串，则将 $\alpha$ 中 $\beta$ 子串替换成概念名 $Y$，得一新终极符和非终极符串 $\alpha'$，从而得到一个新的产生式 $X\to\alpha'$.

(3) 判断产生式 $X\to\alpha'$ 是否可以由 *AcceptSet* 推出，若是，则自动拒绝 $X\to\alpha'$，并对 *AcceptSet* 中下一产生式按步骤(2)处理. 否则，将 $X\to\alpha'$ 交付刻画者确认是否接受.

(4) 若刻画者接受 $X\to\alpha'$，则将 *AcceptSet* 中产生式 $X\to\alpha$ 替换成产生式 $X\to\alpha'$，并转步骤(1). 否则，放弃 $X\to\alpha'$，并对 *AcceptSet* 中下一产生式进行步骤(2)中的处理.

例 2：假设例 1 中的系统猜测结果被确认接受，则系统求精结果如表 2 所示.

表 2

| 概念名 | 系统猜测结果 | 猜测求精结果 | 获取产生式 |
|---|---|---|---|
| $X$ | $X_1cc$ | $Xc$ | $X\to Xc$ |
| | $X_1c$ | $X_1c$ | $X\to X_1c$ |

## 2.4 空字处理

未知概念所拥有的实例可以是一个空字(即空串)，一个已知概念的定义中同样可能包含空字产生式，即形如

$X\rightarrow\lambda$的产生式.在系统猜测和猜测求精阶段对空字不予考虑,等到已经得到一个精化的产生式集合后,再进行空字处理.处理过程如下:

(1) 对所有包含空字实例的未知概念 $X$,向 *AcceptSet* 中加入产生式:$X\rightarrow\lambda$;

(2) 对 *AcceptSet* 中任一产生式 $X\rightarrow\alpha$,将 $\alpha$ 中出现的具有空字的概念名部分或全部替换成空串.比如,在产生式 $X\rightarrow aYbZc$ 中,概念 $Y$ 和 $Z$ 有空字,则将 $Y$ 和 $Z$ 部分或全部替换成空串,于是,得到如下产生式:$X\rightarrow abZc$,$X\rightarrow aYbc$,$X\rightarrow abc$.若得到的某个产生式已在 *AcceptSet* 中出现,如 $X\rightarrow abZc$ 在 *AcceptSet* 中出现,则从 *AcceptSet* 中将其去除.

### 2.5 冗余排除

在得到最终结果之前,还需进行一遍冗余排除处理,即对当前 *AcceptSet* 集合中任一产生式,若能由 *AcceptSet* 中其他产生式推出,则询问刻画者是否将其删除.

例如,在 *AcceptSet* 中出现产生式

(1) $X\rightarrow Y$

(2) $Z\rightarrow Xb$

(3) $Z\rightarrow Yb$

其中产生式(3)可以从式(1)(2)推出,所以,系统会询问是否去除产生式(3).

### 2.6 定义编辑

一轮获取过程结束,最后的 *AcceptSet* 就是我们得到的概念的文法定义.如果用户对获取的文法定义不需作任何修改,则可继续入库操作(即将获取的新概念存入概念库),获取过程结束;否则,用户可以根据前一轮获取过程得到的经验,对待获取概念或其实例作适当修改,重新进行获取;或者,用户直接编辑获取的定义,如修改、删除和增加有关产生式.编辑后的文法定义中若出现未知概念,这时系统提示进入新一轮概念的获取,并将新一轮获取的定义合并于上次获取的结果之后,直至最终结果中不含未定义的子概念.

## 3 讨 论

以上学习过程在 SunOs5.3 的 OpenWin 下实现,并完成了表达式、初等函数、Lisp 语言等概念的学习.结果表明,只要给予的未知概念的样本集合合理(样本具有典型性)和丰富(包罗所有结构情形),该学习过程就能够高效率地获取高质量的概念定义.这里的高效率是指交互过程较短;高质量是指获取的文法能自然地反映概念的内部结构和子概念,从而自然地刻画概念的含义.同时,根据我们对当前系统的使用经验,可以对现有获取过程作如下几点改进.

如果刻画者给出的样本随意性比较强,可能得不到简洁的猜测,为此,可以在求精阶段加入如下处理过程,以试图得到通用性强的猜测.

设当前 *AcceptSet* 中包含如下猜测

(1) $X\rightarrow\alpha\gamma_1\beta$

(2) $X\rightarrow\alpha\gamma_2\beta$

其中 $\alpha,\beta\in V^*$,且 $\alpha,\beta$ 不同时为空,显然 $\gamma_1\neq\gamma_2$.若 $\gamma_1$ 和 $\gamma_2$ 皆为非终极符,且当前 *AcceptSet* 中不存在猜测

(3) $\gamma_1\rightarrow\gamma_2$

(4) $\gamma_2\rightarrow\gamma_1$

则试图引入猜测(3)或(4).若引入(3),则删除猜测(2);若引入(4),则删除猜测(1).

若 $\gamma_1$ 和 $\gamma_2$ 中,一个为非终极符,另一个是由终极符和非终极符组成的非空串,不妨设 $\gamma_1$ 为非终极符,且当前 *AcceptSet* 中不包含猜测(3),则试图引入猜测(3),若被接受,则用(3)替换 *AcceptSet* 中的(2).

还有一种对付随意性样本,以得到通用性强的猜测方法,那就是充分利用已知概念,以进行最大限度的推广.例如,我们在获取简单的四则表达式的文法定义时,给出的实例如下(实例之间以逗号分开).

*expr*: 2, 3+4

*term*: 1, 1×2

*factor*: 3, (2)

并复用已知概念 *int*,这里,*expr*、*term*、*factor* 和 *int* 分别表示表达式、项、因子和整数.在系统猜测阶段,得不到我们希望得到的一些猜测,如 $expr\rightarrow expr+term$ 和 $term\rightarrow term\times factor$.对此,我们可以采用如下方法:在系统猜测阶段进行两遍扫描.第1遍扫描可以产生诸如 $expr\rightarrow int$,$term\rightarrow int$ 以及 $term\rightarrow term\times int$ 等猜测.原则上说,这些猜测产生的实例皆能符合要求,但提交给刻画者确认时,会感到很别扭,原因是其通用性不够;但当我们进行第2遍扫描时,将*expr*,

*term* 和 *factor* 看作已知概念，其定义由第 1 遍扫描得到的猜测进行刻画. 比如，对 *expr* 的实例 2 重新猜测，将 *term* 看作已知概念，其定义中包含第 1 遍扫描产生的猜测：*term*→*int*，显然 2 可以被这时的 *term* 接受，于是，我们可以从 *expr* 的实例 2 得到猜测：*expr*→*term*，用类似的方法可以从 *term* 的实例 1×2 得到猜测：*term*→*term*×*factor* 等等. 这些猜测正是我们希望得到的，而在第 1 遍扫描中是得不到的. 虽然它们有可能在后面的猜测求精阶段得到，但没有在系统猜测阶段来得直接. 当然，这种两遍扫描方法可能会给系统带来很大负担.

**致谢**　本文的工作是在董韫美院士的指导下完成的，在此表示感谢. 万战勇和陈自明同志对系统进行了试用，并提出诸多有益建议，一并表示谢意.

## 参考文献

1 Gold E M. Language identification in the limit. Information and Control, 1967,10:447～474

2 Feldman J. Some decidability results on grammatical inference and complexity. Information and Control, 1972,20:244～262

3 Sakakibara Y. Learning context-free grammars from structural data in polynomial time. Theoretical Computer Science, 1990, 76:223～242

4 董韫美. 获取上下文无关文法的一种交互式算法. 计算机学报，1996,19(3):168～173
(Dong Yun-mei. An interactive algorithm for acquisition of context-free grammars. Chinese Journal of Computers, 1996,19 (3):168～173)

5 董韫美. 基于复用的上下文无关文法推断. 软件学报，1996,7(863 专刊):178～181
(Dong Yun-mei. Context-free grammatical inference based on reusing. Journal of Software, 1996,7(863 special issue):178～181)

6 Dong Yun-mei *et al*. Collection of SAQ Report no. 8～16. 中国科学院软件研究所计算机科学实验室报告，ISCAS-LCS-96-1. Mar. 1996
(Dong Yun-mei *et al*. Collection of SAQ Report no. 8～16. Technical Report no. ISCAS-LCS-96-1, Laboratory of Computer Science, Institute of Software, The Chinese Academy of Sciences, Beijing, March 1996)

7 张瑞岭. 以上下文无关语言表示的概念的一个交互式学习过程的实现[硕士论文]. 中国科学院软件研究所，1996
(Zhang Rui-ling. The implementation of an interactive learning procedure for acquisition of concepts represented as CFL[M.S. Thesis]. Institute of Software, The Chinese Academy of Sciences, 1996)

8 Dong Yun-mei. Collection of SAQ Report no. 1～7. 中国科学院软件研究所计算机科学实验室报告，ISCAS-LCS-95-09. August 1995
(Dong Yun-mei. Collection of SAQ Report no. 1～7. Technical Report no. ISCAS-LCS-95-09, Laboratory of Computer Science, Institute of Software, The Chinese Academy of Sciences, Beijing, August 1995)

## Design and Implementation of a Procedure for Acquisition of Context-Free Grammars

ZHANG Rui-ling

(*Laboratory of Computer Science　Institute of Software　The Chinese Academy of Sciences　Beijing*　100080)

**Abstract**　In this paper, the author introduces the design and implementation of an interactive procedure for grammatical inference. It serves as a tool to help the users acquire concepts represented as context-free grammars from finite examples and sentential forms, which some known concepts can be reused as sub-concepts.

**Key words**　Grammatical inference, learning by examples, context-free language, reuse.