目　录

第 01-02 学时

章节：

第 1 章绪论

1.1 集合的基础知识

1.2 关系

1.3 图

1.4 语言

主题：课程总体介绍及基础知识回顾

教学目的：使学生了解教学目的与教学基本内容；了解本门课程的理论重要性，以及学习本课程应注意的问题。了解集合和关系的相关概念；了解图的相关概念；掌握归纳法、递归定义和反证法。

重点：教学目的，基本内容，学习本课程应注意的问题。

难点：如何让学生能较好地认识到学习这门课的重要性，以及本课程的教学在计算机高级人才培养中的地位。

教学过程：

2、课程介绍 10 分钟

理论来源

1. Chomsky 对自然语言的研究
2. ALGOL 60 语言的语法描述方式
3. Turing、Kleene、Neumann、Huffman 等 对自动机的研究

作用

1. 相关理论是计算机学科基础。
2. 应用范围已被扩展到生物工程、自动控制系统、图象处理与模式识别等许多领域。

根本问题

1. 理论方面的知识是计算机科学的真正灵魂。
2. 计算机科学的根本问题是：什么能被(有效地)自动化？

怎样进行自动化是一个实践的过程

课程目的和基本要求

课程性质

技术基础

基础知识要求

数学分析（或者高等数学），离散数学

主要特点

抽象和形式化

理论证明和构造性

基本模型的建立与性质

本专业人员 4 种基本的专业能力

计算思维能力

算法的设计与分析能力

程序设计和实现能力

计算机软硬件系统的认知、分析、设计与应用能力

计算思维能力

逻辑思维能力和抽象思维能力

构造模型对问题进行形式化描述

理解和处理形式模型

知识

掌握正则语言、下文无关语言的文法、识别模型及其基本性质、图灵机的基本知识。

能力

培养学生的形式化描述和抽象思维能力。

使学生了解和初步掌握“问题、形式化描述、自动化（计算机化）”这一最典型的计算机问题求解思路。

3、教材及参考资料介绍 2 分钟，强调主要使用中文资料，英文资料可以辅助以后阅读科技论文。

教材及参考资料

教材：形式语言与自动机理论，蒋宗礼，姜守恒，清华大学出版社，2013 年，第 3 版

参考资料：

[1] 有限自动机理论，陈文宇，电子工业出版社，2013 年，第 2 版
[2] 自动机理论、语言和计算导论，霍普克罗夫特 (John E.Hopcroft) (作者), Rajeev Motwani (作者), Jeffrey D.Ullman (作者), 孙家骕 (译者)，机械工业出版社，2008 年，第 3 版
[3] 形式语言、自动机理论与计算导论，卡马拉(Kamala Krithivasan)、拉玛(Rama R)、冯晓宁、 孟宇龙(译者)，电子工业出版社，2012 年
[4] John E Hopcroft, Rajeev Motwani, Jeffrey D Ullman. Introduction to Automata Theory, Languages, and Computation (2nd Edition). Addison-Wesley Publishing Company, 2001

4、内容安排介绍 3 分钟，要求学生一周内定好答疑时间。强调作业的重要性。

考核方式及课程构成比例

1. 平时作业及课堂表现 20%；
2. 中期考核 10%，开卷，随堂考；
3. 期末考核 70%，闭卷，考试。

讲课学时数：40

其中：课题讲授 37 学时；习题课 3 学时

（一）复习旧课

在学习本门课程之前，需要对已有知识进行复习，它们主要是离散数学的内容，希望能够温顾而知新。（快速讲解）

第 1 章 绪论

1.1 集合的基础知识

1.2 关系

1.3 图

1.4 语言

1.1 集合的基础知识

1.1.1 集合及其表示

一些没有重复的对象的全体称为**集合**(set)，而这些被包含的对象称为该集合的元素(element)。集合中元素可以按任意的顺序进行排列。一般，使用大写英文字母

表示一个集合。

列举法

对于元素个数较少的集合，可以采用列举法，即将集合的所有元素全部列出，并放在一对花括号中。例如集合 $A=\{0,1,2,3,4,5,6,7,8,9\}$，表示集合 A 由 0，1，2，3，4，5，6，7，8，9 共 10 个元素组成。

命题法

对于集合元素较多的或者是由无穷多个元素组成的集合，可以使用集合形成模式$\{x \mid P(x)\}$进行描述，其中，x 表示集合中的任一元素, $P(x)$是一个谓词，对 x 进行限定，$\{x \mid P(x)\}$表示由满足 $P(x)$的一切 x 构成的集合。可以使用自然语言，或者数学表示法来描述谓词 $P(x)$。

例如，$\{n \mid n$ 是偶数$\}$，或者 $\{n \mid n \bmod 2 = 0\}$，都表明了一个由所有偶数组成的集合。

集合的基数

如果集合 A 包含元素 x（也称元素 x 在集合 A 中），记为 $x \in A$。

对于任意的有穷集合 A，使用$|A|$表示该集合包含的元素的个数，也称基数或势。显然，$|A| = 0 \Leftrightarrow A = \varnothing$ 。

如果一个集合中的元素个数是有限的，称该集合为有穷集合。如果一个集合包含的元素是无限的，称该集合为无穷集合。无穷集合又分为可数集(如自然数集，有理数集)和不可数集(如实数集)。

1.1.2 集合之间的关系

定义 1-2 设 A, B 是两个集合，如果集合 A 中的每个元素都是集合 B 的元素，则称集合 A 是集合 B 的子集，集合 B 是集合 A 的包集。记作 $A \subseteq B$，或 $B \supseteq A$。

定义 1-3 设 A, B 是两个集合，如果 $A \subseteq B$ ，且$\exists x \in B$，但 $x \notin A$，则称 A 是 B 的真子集，记作 $A \subset B$。

有几个关于集合的结论比较有用。详细描述集合基数相等的问题。

几个结论

1. $A = B$ iff $A \subseteq B$ 且 $B \subseteq A$。

2. 如果 A 是有穷集，且 $A \subset B$ ，则$|A| < |B|$。

3. 对于无穷集，这个结论并不适用。如奇数集是自然数集的真子集，自然数集与奇数集之间存在一一对应关系，即它们的基数相等。该映射可表示为：

$f(x) = 2x$-1

1.1.3 集合的运算

1. 并，交，注意多个集合并、交的写法
2. 差，注意它不要求两个集合存在子集关系
3. 补集，论域的概念
4. 笛卡儿积，生成的集合中元素为原集合元素的有序对

笛卡尔集的概念允许我们利用多维空间来思考问题，它将是语言的重要基础。

集合 A 和 B 的笛卡儿乘积使用 $A \times B$ 表示（也简记为 AB）

$A \times B = \{(a, b) \mid a \in A$ 且 $b \in B\}$

当集合 A、B 为有穷集时

$$|A \times B| = |A| * |B|$$

下面这个例子说明了笛卡尔集的生成。

设 $A = \{a, b, c\}$, $B = \{0, 1\}$,

则 $A \times B = \{(a, 0), (a, 1), (b, 0), (b, 1), (c, 0), (c, 1)\}$

$B \times A = \{(0, a), (0, b), (0, c), (1, a), (1, b), (1, c)\}$

关于笛卡尔集有一个思考题。

什么情况下： $A \times B = B \times A$?

幂集是我们本书中将要用到的一个重要概念，也是许多理论性课程的重要基础。我们常说："在一个层次不能解决的问题，在其更高层次通常能够解决"，而幂集常常为我们提供了更高的层次。

幂集

设 A 为一个集合，那么 A 的幂集为 A 的所有子集组成的集合，记为 $2A$，即 $2A=\{B \mid B \subseteq A\}$。

例如，集合 $A=\{1，2，3\}$，则 A 的幂集为:

$2A=\{Ø，\{1\}，\{2\}，\{3\}，\{1，2\}，\{1，3\}，\{2，3\}，\{1，2，3\}\}$。

当集合 A 为有穷集时，如果集合 A 包含的元素个数为 n，那么集合 $2A$ 的元素个数(集合 A 的所有子集的个数)为 $2n$，这就是称 $2A$ 为集合 A 的幂集的原因。

1.2 关系

1.2.1 二元关系

设 A 和 B 为两个集合，则 A 到 B 的关系是 $A\times B$ 的任何子集。(重要概念)

若 $A=B$，则称为 A 上的关系。

若 R 为 A 到 B 的关系，当 $(a, b) \in R$ 时，可记为 aRb。

例如，设 A 为整数集合，则 A 上的关系"<"是集合 $\{(a, b) \mid a, b \in A$,且 $a < b\}$。

关系数据库是当前运用得最广的数据库，下面我们来回顾关系本身的概念。

最简单的关系是二元关系。强调"任意子集"。

下面描述关系的三歧性。

设 R 是 A 上的二元关系，有

(1) 如果对 $\forall a \in A$，都有 $(a, a) \in R$，则称 R 是自反的。

(2) 如果对 $\forall a \in A$，都有 $(a, a) \notin R$，则称 R 是反自反的。

(3) 如果对 $\forall a, b \in A$，$(a, b) \in R \Rightarrow (b, a) \in R$，则称 R 是对称的。

(4) 如果对 $\forall a, b \in A$，$(a, b) \in R$ 且 $(b, a) \in R \Rightarrow a=b$，则称 R 是反对称的。

(5) 如果对 $\forall a, b, c \in A$，$(a, b) \in R$ 且 $(b, c) \in R \Rightarrow (a, c) \in R$，则称 R 为传递的。

1.2.2 等价关系与等价类

根据关系的三歧性，有一种最特殊的关系，就是等价关系。

定义 1-14　如果集合 A 上的二元关系 R 是自反的、对称的和传递的，则称 R 为等价关系。

等价关系的一个重要性质为：集合 A 上的一个等价关系 R 可以将集合 A 划分为若干个互不相交的子集，称为等价类。对 A 中的每个元素 a，使用 $[a]$ 表示 a 的等价类，即 $[a]=\{b \mid aRb\}$。等价关系 R 将集合 A 划分成的等价类的数目，称为该等价关系的指数。

见下面的例子。

非负整数集合 N 上的模 3 同余关系 R,

$R=\{(a, b) \mid a, b \in N,\ 且\ a \bmod 3 = b \bmod 3\}$。

$\{0,3,6,\ldots,3n,\ldots\}$形成一个等价类；

$\{1,4,7,\ldots,3n+1,\ldots\}$形成一个等价类；

$\{2,5,8,\ldots,3n+2,\ldots\}$形成一个等价类；

分别记为[0]，[1]和[2]。

1.2.3 关系的合成

关系的合成有点像函数的合成。

设 $R1 \subseteq A \times B$ 是 A 到 B 的关系， $R2 \subseteq B \times C$ 是 B 到 C 的关系，则 $R1$ 与 $R2$ 的合成是 A 到 C 的关系

$$R1\ R2 = \{(a, c) \mid \exists (a, b) \in R1\ 且 (b, c) \in R2\}$$

有了关系的合成，就相当于定义了关系的乘法，在其基础上可以进一步定义关系的 n 次幂。

1.2.4 递归定义与归纳证明

递归定义

(1) 基础

(2) 归纳

(3) 极小性限定(有限性)

归纳证明

(1) 基础

(2) 归纳

(3) 由归纳法原理，结论对…成立

关系的 n 次幂

定义 1-17 设 R 是 S 上的二元关系，则 R^n 如下递归定义：

(1) $R^0 = \{(a, a) \mid a \in S\}$

(2) $R^i = R^{i-1} R\ (i = 1, 2, 3, \ldots)$

1.2.5 关系的闭包

而下面要介绍的正闭包和克林闭包则是本书中将不断用到的概念。

定义 1-19 设 R 是 S 上的二元关系， R 的正闭包 R^+ 定义为

(1) $R \subseteq R^+$ ；

(2) 如果$(a, b), (b, c) \in R^+$, 则$(a, c) \in R^+$ ；

(3) 除(1)，(2)外， R^+不再含有其他任何元素。

$R^+ = R \cup R^2 \cup R^3 \cup \ldots$

且当 S 为有穷集时，有

$R^+ = R \cup R^2 \cup R^3 \cup \ldots R^{|S|}$

关系的克林闭包

$R^* = R^0 \cup R^+$

R0 表示该关系满足自反性

本课程由于是理论性质课程，将遇到非常多的证明。我们现在回顾两种证明方法和一种定义方法。

首先是反证法。反证法可以说成是“没有办法的办法”。

反证法也称为归谬法。利用反证法证明一个命题时，一般的步骤为：

(1) 假设该命题不成立；

(2) 进行一系列的推理；

(3) 在推理的过程中如果出现了下列情况之一：

1. 与已知条件矛盾；
2. 与公理矛盾；
3. 与已证过的定理矛盾；
4. 与临时的假定矛盾；
5. 自相矛盾；

(4) 由于上述矛盾的出现，可以断言“假设该命题不成立”的假设不正确；

(5) 肯定原命题正确。

然后是非常简单的归纳法。我们后面会看到，其实归纳法也不一定简单，在某

些地方它还非常有趣。

递归定义是本书中使用最多的定义，它与归纳证明有着密切的联系。注意这里的归纳证明更倾向于运算的封闭性。

后面在讲到自动机时我们将使用到图的概念，在讲到句子的推导时我们将使用到树的概念，因此在这里首先对这两个基本概念进行回顾。

图

现实世界中，许多现象可以抽象成图来表示。

直观地，图是由一些点和连接两点的边组成的。

无向图

定义 1-19 设 V 是一个非空的有穷集合，$E\subseteq V\times V$，称 $G=(V,E)$ 为一个无向图。

其中，V 称为顶点集，V 中的元素称为顶点；E 称为无向边集，E 中的元素称为无向边。

无向图中的边都没有方向，(vi, vj) 和 (vj, vi) 表示的是同一条边

无向图顶点的度

例 1-21

$V=\{v1, v2, v3, v4, v5\}$

$E=\{(v1, v2), (v1, v3), (v1, v4), (v2, v3), (v2, v5), (v3, v4), (v3, v5), (v4, v5)\}$

$\deg(v1) = \deg(v2) = \deg(v4) = \deg(v5) = 3$

$\deg(v3) = 4$

有向图

(vi, vj) 表示的是从顶点 vi（前导）出发，到达顶点 vj（后继）的一条边。

其中：vi 称为 vj 的前导，vj 称为 vi 的后继。显然，(vi, vj) 和 (vj, vi) 表示的是不同边

有向图顶点的度：入度与出度

思考：有向图和无向图，哪种更特殊？

$G2$

$V=\{v1, v2, v3, v4, v5\}$

$E=\{(v1, v2), (v1, v3), (v2, v3), (v3, v4), (v3, v5), (v4, v1),(v4, v5), (v5, v2),(v5, v4)\}$

ideg($v1$)= 1,odeg ($v1$)= 2

ideg($v2$) =2, odeg($v2$) =1

有向回路

设 $G=(V,E)$为一个有向图。如果对于 $0 \le i \le k-1$，均有$(vi, vi+1) \in E$，则称 $v0$，$v1$，…，vk是 G的一条有向路。当 $v0=vk$时，称 $v0$，$v1$，…，vk是一条有向回路。

思考：从 $v1$ 到 $v4$ 有多少条路？

特别评讲两个思考。

1.3.3 树

设 $G=(V,E)$为一个有向图。当 G满足如下条件时，称 G为一棵(有向)树：

(1) $\exists v\in V$，v没有前导，且 v到树中的其他顶点都有一条有向路，该顶点称为树 G的根；

(2) 每一个非根顶点有且仅有一个前导；

(3) 每个顶点的后继按其拓扑关系从左到右排序。

（二）小结

形式语言与自动机的重要性总结。

集合的基础知识；关系的相关概念；图的相关概念；归纳法、递归定义和反证法这些内容需要大家非常的熟悉。

（三）布置作业

1(9)(10)　2(1)(3)　3(5)　4(1)　7(2)

10(1)(3)(5)　20(5)

第 03-04 学时

章节：

第 2 章文法

2.1 启示

2.2 形式定义

主题：形式语言及其相关的基本概念

教学目的：使学生理解字母表、句子等概念；了解字母及其特性、出现等概念；了解句子的长度、空语句、句子的前后缀、语言及其运算等概念及相关的计算；掌握语言的表示，即根据语言的描述获得其集合方式的形式描述。

重点：形式语言的基本概念

难点：语言的形式化描述

教学过程：

（一）复习旧课

上节课复习了离散数学中与本书相关的一些内容，包括集合、关系、证明和证明的方法，以及图与树等知识。对关系的闭包进行复习，尤其是正闭包和克林闭包，将在以后的学习中经常用到，需要熟练掌握。对于树，尤其是二元树，在以后的语法树中也会用到，在此做一个讲解。

（二）讲授新课

今天学习本门课程的核心概念：语言。如前面所讲，我们研究的语言是形式语言，它是对任意语言，包括自然语言、受限语言、程序设计语言的抽象。

关于字符串的概念大家比较熟悉。

这里给出语言的概念，后面将给出形式定义。

字/字母是语言的基本要素，它是比较确定的，也比较简单。

语法也是语言的基本要素，它有时候会非常复杂。关于“下雨天留客，天留我不留”。

语言是“为相当大的团体的人所懂得并使用的字和组合这些字的方法的统一”，但这个定义对于计算机界的人们来说毫无意义。

语言的最基本元素是字母表。

字母表(alphabet) 是一个非空有穷集合，字母表中的元素称为该集合中的字母(letter)，也可叫作符号 (symbol)或字符(character)

字母表中的字母具有整体性和可辨认性(特殊情况下才需要注意，如“≥”表示成“>=”)

设Σ^1 , Σ^2 是两个字母表， Σ^1 与Σ^2 的乘积：

$\Sigma^1 \Sigma^2 =\{ab|a\in\Sigma^1, b\in\Sigma^2\}$

例：{0, 1}{0, 1}={00, 01, 10, 11}

回顾：集合的笛卡尔集

语言的研究包括如下三个方面 见课件第 67 页)，主要是语言规律的研究。

下面介绍常用术语，前三个术语已经介绍过（见课件第 68 页）。

字符串的连接比较有特点。

设∑是一个字母表，$x, y \in \Sigma^*$, x, y 的并置(连接)是由串 x 直接接串 y 组成的，记作 xy。

连接的性质：(P31)

1. 结合律
2. 消去律
3. 唯一分解性
4. 单位元素：ε

集合的连接与前面回顾的集合的乘积稍有差别，但这个差别仅仅是表示方式，而不影响其本质。如 ab 如果用集合的乘积方式应写为(a, b)（使用黑板）。

需要注意，在考察这一运算时，我们使用了字母表而不是一般的集合 A,B 来表示。

现在讨论什么情况下两个字母表的乘积可以交换顺序，而最后一种情况比较特殊，它对于一般的集合运算而言并不成立。

字母表的 n 次连接（幂）使用了递归定义。使用黑板演示。

设∑是一个字母表，∑的 n 次幂定义为：

(1) $\Sigma^0 = \{\varepsilon\}$

(2) $\Sigma^n = \Sigma^{n-1}\Sigma \qquad n \geq 1$

其中 ε 是由∑中的 0 个字符组成的。

思考：‘’和‘ ’是相等的吗？

与平常的集合运算一样，定义了 n 次幂后就可以定义克林闭包了。这里的克林闭包并没有尽头。

设∑是一个字母表，∑的正闭包：

$\Sigma^+ = \Sigma \cup \Sigma^2 \cup \Sigma^3 \cup \ \ldots$

∑的克林闭包：

$\Sigma^* = \Sigma^0 \cup \Sigma^+$

下面的例子给出了直观的描述

字母的闭包

$\{0, 1\}^+=\{0, 1, 00, 01, 10, 11, \ \ldots\}$

$\{0, 1\}^*=\{\varepsilon, 0, 1, 00, 01, 10, 11, \ \ldots\}$

字母表上的串。

字母表的克林闭包是该字母表上所能描述最“大”的集合，而正闭包只比克林闭包少一个空串。用 2 分钟与前面关系的闭包进行对比。

由于正闭包只比克林闭包少一个空串，空串本身非常有趣，它所构成的集合与有着如下区别。特别注意分析字母表的零次方。$\Sigma^0 = \{\varepsilon\}$

另外与之相关有两个值得思考的问题。几个思考注意与学生的交互。

$\{\varepsilon\}$与$\emptyset$

$|\{\varepsilon\}|=1 \qquad\qquad |\emptyset|=0$

$\{\varepsilon\}\times A=A \qquad\qquad \emptyset \times A= \emptyset$

对于加法来说， ε 是什么？对于乘法呢？对于关系 R 呢？P16 定义 1-17

$R^0 =\{(a, a) \mid a \in S\}$

前面给出了语言的描述，这里给出形式的定义。

设∑是一个字母表，$\forall L \subseteq \Sigma^*$, L 称为字母表∑上的一个语言(language), $\forall x \in L$,

x 叫做 L 的一个句子。

可按语言的基数分为有穷语言与无穷语言。

例子：

字母表 $\{0, 1\}$ 上的一些语言：

$\{00, 11\}$

$\{0, 1, 00, 11\}$

$\{0\}\{0, 1\}^*\{1\}$

$\{0, 1\}^*\{111\}\{0, 1\}^*$

设 Σ_1, Σ_2 是一个字母表，$L_1 \subseteq \Sigma_1^*, L_2 \subseteq \Sigma_2^*$, 语言 L_1 与 L_2 的乘积是一个语言：

$L_1L_2=\{xy \mid x \in L_1, y \in L_2\}$

该语言是字母表 $\Sigma_1 \cup \Sigma_2$ 上的语言。

设 Σ 是一个字母表，$\forall L \subseteq \Sigma^*$, L 的 n 次幂是一个语言

(1) 当 $n = 0$ 时， $L^n = \{\varepsilon\}$

(2) 当 $n \geq 1$ 时， $L^n = L^{n-1} L$

L 的正闭包 $L+$ 是一个语言：

$L^+ = L \cup L^2 \cup L^3 \cup \ldots$

L 的克林闭包 L^* 是一个语言：

$L^+ = L^0 \cup L^+$

句子的长度及前、后缀快速讲解。注意不要使学生陷入迷茫状态。

设 Σ 是一个字母表，$\forall x \in \Sigma^*$, x 叫做 Σ 的一个句子。

两个句子称为相等的，如果它们对应位置上的字符都相等。（有序）

设 Σ 是一个字母表，$x, y \in \Sigma^*$, $a \in \Sigma$, 句子 xay 中的 a 叫作 a 在该句子中的一个出现。

设 Σ 是一个字母表，$\forall x \in \Sigma^*$,句子 x 中字符出现的总个数叫做该句子的长度，记做 $|x|$。

长度为 0 的字符串叫空句子，记作 ε 。

设Σ是一个字母表，$x, y, z \in \Sigma^*$,且 $x = yz$,则称 y 是 x 的前缀；如果 $z \neq \varepsilon$ ，则称 y 是 x 的真前缀。 z 是 x 的后缀；如果 $y \neq \varepsilon$，则称 z 是 x 的真后缀。

设Σ是一个字母表，$w, x, y, z \in \Sigma^*$,且 $w = xyz$，则称 y 是 w 的子串。

ε 是任何字符串的子串

公共子串，最大公共子串都不唯一。如 *acbac* 与 *accbb* 有最大公共子串 *ac* 与 *cb*。

以上这次概念将在后面使用到，它们不需要死记，以后自然会理解。

（三） 小结

用形式化的方法描述什么是语言。

形式语言与自动机理论的产生与作用。

形式语言的基本概念。

（四） 布置作业

21(1)(3)(5)　　22(需要知道个数，不需要写完)　　28(1)(2)(6)

29　　2(3)(6)(8)

第 05-06 学时

章节：

第 2 章文法

2.2 形式定义

主题： 文法的直观意义与形式定义

教学目的： 使学生了解文法的直观意义；理解文法的形式定义；掌握推导的方法、以及根据文法产生的语言句子和句型的方法。

重点： 文法的直观意义与形式定义，推导，文法产生的语言、句子、句型。

难点： 形式化的概念。

教学过程：

（一）复习旧课

回顾语言的集合定义。

字母表是一个非空有穷集合。

如{a，a′，b，b′}， {∞，∧，∨，≥，≤}

字母表的乘积

$$\sum_1\sum_2=\{ab|a\in\sum_1，b\in\sum_2\}$$

如 {0,1}{0,1}={00，01，10，11}

字母表∑的 n 次幂

$\sum^0=\{\varepsilon\}$ ε是由∑中的 0 个字符组成的。

$\sum^n=\sum^{n-1}\sum$

∑的正闭包

$\sum^+=\sum\cup\sum^2\cup\sum^3\cup\sum^4\cup\ldots$

∑的克林闭包

$\sum^*=\sum^0\cup\sum^+ \quad =\sum^0\cup\sum\cup\sum^2\cup\sum^3\cup\ldots$

句子

∑是一个字母表，$\forall x\in\sum^*$，x 叫做∑上的一个句子。

出现、句子的长度、并置（连接）、前缀与后缀 、公共前缀与后缀、子串、公共子串、最大公共子串

语言(language)

$\forall L\subseteq\sum^*$，L 称为字母表∑上的一个语言，$\forall x\in L$，x 叫做 L 的一个句子。

语言的乘积

$L_1 \subseteq \sum_1^*$，$L_2 \subseteq \sum_2^*$，语言 L_1 与 L_2 的乘积是一个语言，该语言定义为：

$L_1L_2=\{xy|\ x\in L_1,\ y\in L_2\}$ 是字母表$\sum_1 \cup \sum_2$上的语言。

幂

$\forall L\in \sum^*$，L 的 n 次幂是一个语言，该语言定义为

⑴ 当 n=0 是，$L^n=\{\varepsilon\}$。

⑵ 当 $n\geq 1$ 时，$L^n= L^{n-1}L$ 。

正闭包

$L^+=L\cup L^2\cup L^3\cup L^4\cup \ldots$

克林闭包

$L^*= L^0\cup L\cup L^2\cup L^3\cup L^4\cup \ldots$

（二） 讲授新课

上一章从集合的角度讨论了语言，本节课开始从语言的规律对其进行研究：一是从语言产生的角度；另一个是从接收(或识别)语言的角度。

文法：从语言产生的角度研究语言

有穷语言(如英语词汇表)

无穷语言，有穷描述，如：$(\{0, 1\}\ \{0, 1\})^*$

通过分析句子的“结构”研究语言(即平常所说的语法)

首先来看一些例子。

哈尔滨是美丽的城市。

集合是数学的基础。

中国进入 WTO。

句子结构：

<名词短语><动词短语><句号>

尖括号表示必须出现。

根据句子结构，可以构造如下合法句子：

集合进入 WTO。

且合法句子总数为：

|<名词短语>||<动词短语>||<句号>|

其中，符号“| |”表示集合的基数

关于语义问题

“集合进入 WTO。”符合语法，但它没有适当含义，这是语义问题，不在本书研究范畴之内。

形成规则：

“a可以是 b ”表示为“$a \to b$” 也可读作 “a定义为 b ”

<句子>$\to$ <名词短语><动词短语><句号>

<动词短语> $\to$ <动词> <形容词短语>

<动词短语> $\to$ <动词> <名词短语>

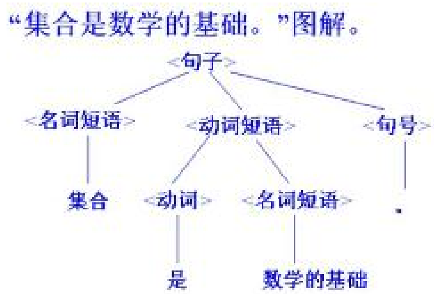
<动词> $\to$是

<动词> $\to$走在

<名词短语> $\to$北京

…其余部分见书 P45

下面来看句子图解



练习：“形式语言是很抽象的。”图解。

讨论：自然语言中，句子的分解到此结束了吗？

总结：表示语言四要素

1. 一系列“符号”，即语法变量，如<名词短语>
2. 最终定义的结构，如<句子>

3. 终极符号，如“集合”，“数学的基础”

4. 规则，即产生式，如<动词短语> $\to$ <动词> <名词短语>

以上我们使用若干个例子语言进行了讲解，现在我们对文法进行形式定义。

文法(grammar)G是一个四元组：

$G=(V, T, P, S)$

✓ V—变量(variable)的非空有穷集。一个语法变量表示了一个语法范畴。

✓ T—终极符(terminal)的非空有穷集。$V\cap T=\emptyset$ 。

✓ P—产生式(production)的非空有穷集。P中元素具有形式 $\alpha\to\beta$, 其中 $\alpha\in(V\cup T)+$,且 α 中至少有 V中的一个元素出现。$\beta\in(V\cup T)^*$ 。α, β 依次称为产生式 $\alpha\to\beta$ 的左部和右部。

✓ S—$S\in V$，文法 G的开始符号(start symbol) 。

文法的判定：

(1) $(\{A\}, \{0, 1\}, \{A\to 01, A\to 0A1, A\to 1A0\}, A)$

(2) $(\{A\}, \{0, 1\}, \{A\to 0, A\to 0A\}, A)$

(3) $(\{A, B\}, \{0, 1\}, \{A\to 01, A\to 0A1, A\to 1A0, B\to BA, B\to 0\}, A)$

讨论

$(\{S\}, \{a, b\}, \{S\to 00S, S\to 11S, S\to 00, S\to 11\}, S)$ 是文法吗？为什么？

不是。产生式右部 0，1 等符号既不为语法变量也不为终极符。即不满足定义中 $\beta\in(V\cup T)^*$ 。更改：可将 a, b 依次改为 0, 1 。

一些产生式经常有相同的左部，这时可以将它们合在一起简记。

对一组有相同左部的产生式

$\alpha\to\beta_1, \alpha\to\beta_2, \ldots, \alpha\to\beta_m,$

可以简记为

$\alpha\to\beta_1 \mid \beta_2 \mid\ldots\mid \beta_n$

简要说明一些常用的约定

1. $A, B, C,$ …表示语法变量
2. $a, b, c,$ …表示终极符号
3. $X, Y, Z,$ …表示语法变量或终极符号
4. $x, y, z,$ …表示终极符号组成的行
5. $\alpha, \beta, \gamma,$ …表示语法变量或终极符号组成的行

推导在前面实际上已经用过，但未进行形式化的定义。

设 $G=(V, T, P, S)$是一个文法，如果 $\alpha\to\beta\in P, \gamma, \delta\in(V\cup T)^*$，则称 $\gamma\alpha\delta$在G中直接推导出 $\gamma\beta\delta$ ，记作 $\gamma\alpha\delta \underset{G}{\Rightarrow} \gamma\beta\delta$ 。读作 $\gamma\alpha\delta$在文法 G中直接推导出 $\gamma\beta\delta$ 。直接推导可简称为推导(derivation)或派生。

解释推导的实质、意义及与归约的关系。与父/子关系进行类比。

推导符号实质上是对产生式符号“→”(从某种意义上)的扩展，以逐步获得最终的字符串。

与之相对应，也可称$\gamma\beta\delta$直接归约成$\gamma\alpha\delta$。直接归约可简称为归约(reduction)。

对推导符号有两个重要的扩充，它们的记法与正闭包、克林闭包的记法一致，实际上，它们都表达了“一次与多次”，“零次与多次”的涵义。

直接推导

n 步推导

多步推导

任意步推导

注意再次强调加号与星号的联系与区别。

我们知道，两个集合的乘积的任意子集均可称为一个关系，常见的关系如>, <, =等，推导实际上也是一个关系，由于我们前面进行了扩展，因此有三类的推导关系。

对$\forall x, y\in\sum+$，

为获得$\{x^n y^n \mid n\geq 1\}$，可使用产生式组 $\{D\to xy \mid xDy\}$实现；

为获得$\{x^n \mid n\geq 0\}$，可使用产生式组 $\{D\to\varepsilon \mid xD\}$实现；

为获得$\{ x^n y^n \mid n \geq 0\}$，可使用产生式组 $\{D \to \varepsilon \mid xDy\}$实现。

有两类比较常用的推导。当然，其它的推导方式也是可以的，但这两类更加有规律，便于研究。

我们上次课讲了括号匹配的例子，这里从形式化的角度再次提出来。注意之所以只需要写产生式，是因为可以从它获得其它三个部分的完整信息。根据该例进行详细说明。但我们时刻要牢记：文法是四元式。

文法(grammar)G是一个四元组： $G = (V, T, P, S)$

V—变量(variable)的非空有穷集。一个语法变量表示了一个语法范畴。

T—终极符(terminal)的非空有穷集。$V \cap T = \varnothing$ 。

P—产生式(production)的非空有穷集。P中元素具有形式 $\alpha \to \beta$， 其中 $\alpha \in (V \cup T)^+$,且 α 中至少有 V中的一个元素出现。 $\beta \in (V \cup T)^*$ 。α , β 依次称为产生式$\alpha \to \beta$ 的左部和右部。

S—$S \in V$，文法 G的开始符号(start symbol)。

讲解例 2-6，详细说明各文法的特点。

设 $G = (\{A\}, \{0, 1\}, \{A \to 01, A \to 0A1\}, A)$

$L(A) = \{0^n 1^n \mid n \geq 1\}$

设文法 $G = (V, T, P, S)$，则称 $L(G) = \{w \mid w \in T^*$且 S $w\}$ 为文法 G产生的语言。

对于$\forall$ $w \in L(G)$，w称为 G 产生的句子(sentence)。参见 P33 定义 1-43

设文法 $G = (V, T, P, S)$，对于$\forall \alpha \in (V \cup T)^*$ ，如果S α ，则称 α是 G产生的一个句型(sentential form)。

句子 w是从S开始，在 G中可以推导出来的终极符号行，它不含语法变量。

句型 α是从S开始，在 G中可以推导出来的符号行，它可能含有语法变量。

联系与区别：句子一定是句型，但句型不一定是句子。

由句子的定义，容易得到文法产生语言的定义。注意这里使用了扩展的推导符

号，因为它使得描述变得非常简便。一个文法确实产生语言 $L(G)$，必须确定两个命题：①该文法推导出的串都在语言中；②所有语言中的串都可以由该文法产生。

例 2-7 给定文法 $G=(\{S, A, B, C, D\}, \{a, b, c, d, \#\}, \{S\rightarrow ABCD | abc\#, A\rightarrow aaA, AB\rightarrow aabbB, BC\rightarrow bbccC, cC\rightarrow cccC, CD\rightarrow ccd\#, CD\rightarrow d\#, CD\rightarrow \#d\}, S)$，求句型 $aaaaaabbbbcccc\#d$ 和 $aaaaaaaaAbbccccd\#$ 的推导。

$\underline{S}\Rightarrow \underline{A}BCD$　　需要增加 4 个连续的 a

$\Rightarrow aa\underline{A}BCD$　　需要增加 2 个连续的 a

$\Rightarrow aaaa\underline{AB}CD$　　需要 $aabb$

$\Rightarrow aaaaaabb\underline{BC}D$　　需要 $bbcc$

$\Rightarrow aaaaaabbbbc\underline{cC}D$　　需要增加 cc

$\Rightarrow aaaaaabbbbcccc\underline{CD}$　　需要 $\#d$

$\Rightarrow aaaaaabbbbcccc\#d$

讨论：

由 $C\rightarrow cC$ 可得 $cC\Rightarrow ccC$

由 $cC\Rightarrow ccC$ 可得 $C\rightarrow cC$ 吗？

不可以。因为 $cC \Rightarrow ccC$ 可能是由 $cC \rightarrow ccC$ 得到的，而 $cC \rightarrow ccC$ 成立并不意味着 $C \rightarrow cC$ 成立。

另外，$C \Rightarrow c$ 也不意味着产生式 $C \rightarrow c$ 的存在，因为它可以由产生式组 $C \rightarrow cC, C \rightarrow \varepsilon$ 得到。

总之，规则一般作为条件，而不是结论存在

$\underline{S}\Rightarrow \underline{A}BCD$　　*需要增加 8 个连续的 a*

$\Rightarrow aa\underline{A}BCD$　　*需要增加 6 个连续的 a*

$\Rightarrow aaaa\underline{A}BCD$　　*需要增加 4 个连续的 a*

$\Rightarrow aaaaaa\underline{A}BCD$　　*需要增加 2 个连续的 a*

$\Rightarrow aaaaaaaaA\underline{BC}D$　　*需要 bbcc*

$\Rightarrow aaaaaaaaAbbcc\underline{CD}$　　*需要 ccd#*

$\Rightarrow aaaaaaaaAbbccccd\#$　　*需要 d#*

讨论：推导过程是唯一的吗？

不一定。如：例 2-7 中第一个句型推导过程唯一，但第二个推导过程不唯一(变量 A 及其左边前面的推导与其右边的推导互不影响)。

（三）小结

文法的形式定义

推导与归约

推导的分类

文法产生的语言、句子、句型。

（四）布置作业

4，5

第 07-08 学时

章节：

第 2 章文法

2.3 文法的构造

主题：文法的构造

教学目的：使学生掌握基本的文法构造方法；了解文法等价的概念，了解根据文法产生句子或句型比较容易，但根据语言构造文法比较困难，甚至不可行。

重点：文法的构造。

难点：文法构造方法。

教学过程：

（一）复习旧课

由于文法的概念非常重要，而且与其它地方的概念有较大不同，我们需要对其加强。复习文法的定义、推导，以及句子（约 7 分钟）。

定义 2-1 文法(grammar)G是一个四元组：

$G=(V, T, P, S)$

V—变量(variable)的非空有穷集。一个语法变量表示了一个语法范畴。

T—终极符(terminal)的非空有穷集。$V\cap T=\emptyset$ 。

P—产生式(production)的非空有穷集。P中元素具有形式 $\alpha \to \beta$。 其中 $\alpha \in(V\cup T)^+$,且 α 中至少有 V中的一个元素出现。 $\beta \in(V\cup T)^*$ 。α , β 依次称为产生式 $\alpha\to\beta$的左部和右部。

S—$S\in V$，文法 G的开始符号(start symbol) 。

定义 2-2 设 $G=(V, T, P, S)$是一个文法，如果 $\alpha \to\beta\in P$, $\gamma, \delta\in(V\cup T)^*$，则称 $\gamma\alpha\delta$在 G中直接推导出 $\gamma\beta\delta$。

与之相对应，也可称 $\gamma\beta\delta$直接归约成 $\gamma\alpha\delta$。直接归约可简称为归约(reduction)。

对语言、句子、句型等重要概念进行复习。

（二）讲授新课

2.3 文法的构造

构造文法 G ，使得 L(G) = {0, 1, 00, 11}

方法 1(直接构造，可用于构造任意有穷语言)：$G1=(\{S\}, \{0, 1\}, \{S\rightarrow 0|1|00|11\}, S)$

方法 2(引入两个变量)：$G2=(\{S, A, B\}, \{0, 1\}, \{S\rightarrow A|B|AA|BB, A\rightarrow 0, B\rightarrow 1\}, S)$

方法 3(将产生式的右部第一个符号改为终极符号)：$G3=(\{S, A, B\}, \{0, 1\}, \{S\rightarrow 0|1|0A|1B, A\rightarrow 0, B\rightarrow 1\}, S)$

上例说明：一个语言可以由不同的文法产生。因此，我们有如下的定义。

定义 2-5 设有两个文法 G1 和 G2，如果 L(G1) = L(G2) ，则称 G1 与 G2 等价(equivalence)。

如果变量/终极符号/产生式对最终生成的句子没影响，则对语言也没有影响。

约定：对一个文法，只列出该文法的所有产生式，且所列的第一个产生式的左部是该文法的开始符号。

这是我们关于等价的定义，注意本书中的等价绝大多数是以对应的语言（它是一个集合）相等来定义的。两个文法一般不会说它们相等，因为要相等意味着四元组要相等。而这在上例中是不成立的。

强调：即使这样简单的文法，要清楚说明其语言也不容易。

在进一步学习文法的构造之前，我们要认清这项任务的困难。因为困难，所以我们只能完成一些比较容易，或者说规律性比较强的任务。

构造文法 G_{10}， 使 $L(G_{10})=\{ww^T|w\in\{0，1，2，3\}^+\}$。

文法

$S\rightarrow HE$

$H\rightarrow 0|1|2|3|0H|1H|2H|3H$

$E\rightarrow 0|1|2|3|E0|E1|E2|E3$

能否生成 $L(G_{10})$ ？？？

分析$\{ww^T|w\in\{0，1，2，3\}^+\}$的句子的特点

设 $w=a_1a_2…a_n$，从而有 $w^T= a_n…a_2 a_1$，故 $ww^T= a_1a_2…a_na_n…a_2 a_1$

满足 $f(ww^T，i)=f(ww^T，|ww^T|-i+1)$。

递归地定义 L

(1) 对$\forall a\in\{0，1，2，3\}$，$aa\in L$；

⑵如果 x∈L，则对∀a∈{0，1，2，3}，axa∈L；

⑶ L 中不含不满足(1)、(2)任何其他的串。

根据递归定义中的第一条，有如下产生式组：

$S\rightarrow 00 \mid 11 \mid 22 \mid 33$

再根据递归定义第二条，又可得到如下产生式组：

$S\rightarrow 0S0 \mid 1S1 \mid 2S2 \mid 3S3$

从而，

G_{10}：$S\rightarrow 00 \mid 11 \mid 22 \mid 33 \mid 0S0 \mid 1S1 \mid 2S2 \mid 3S3$

构造文法 G_{12}，使 $L(G_{12}) = \{w \mid w$ 是十进制有理数$\}$。

解：易知有理数分为负有理数和非负有理数，

以 S 为开始符号，有 $S \rightarrow R \mid +R \mid -R$

其中 R 表示非负有理数

R 又可以划分成无符号整数、无符号带小数和无符号纯小数 $R \rightarrow N \mid B \mid P$

$B \rightarrow N.D$　　　　　　$P \rightarrow 0.D$

由于 B 表示的为无符号带小数，N 不应以 0 开始：

$N \rightarrow AM$　　　　$A \rightarrow 1|2|3|4|5|6|7|8|9$

M 为任意的十进制数串，包括为空

$M \rightarrow \varepsilon \mid 0M \mid 1M \mid 2M \mid 3M \mid 4M \mid 5M \mid 6M \mid 7M \mid 8M \mid 9M$

与之相对应， D 不应以 0 结束，而其余部分可以是任意的十进制数串，包括为空

$D \rightarrow MA$

$S \rightarrow R \mid +R \mid -R \mid 0$

$R \rightarrow N \mid B \mid P$

$B \rightarrow N.D$　　　　　　$P \rightarrow 0.D$

$N \rightarrow AM$　　　　　　$D \rightarrow MA$

$A \rightarrow 1|2|3|4|5|6|7|8|9$

$M \rightarrow \varepsilon \mid 0M \mid 1M \mid 2M \mid 3M \mid 4M \mid 5M \mid 6M \mid 7M \mid 8M \mid 9M$

思考：

与书上的结果有什么不同？

不允许 D 为 0 (会对结果产生影响，特别注意是否允许 0.0 的存在) ，且多一个 P(不会对结果产生影响)

由于 N 不允许以 0 开始，这使得 0 $\notin$ N，即按前面分类方式没有包括 0，因此在最终结果时将其加上： $S \rightarrow R| +R \mid -R| 0$。

构造产生语言$\{a^nb^nc^n|n\geq1\}$的文法。

文法 $G=(\{S_1\}, \{a, b\}, \{S_1\rightarrow ab \mid aS_1b\}, S_1)$

产生的语言$\{a^nb^n \mid n\geq1\}$形式上看起来与语言$\{a^nb^nc^n \mid n\geq1\}$比较接近。

$G=(\{S_2\}, \{c\}, \{S_2\rightarrow c \mid cS_2\}, S_2)$产生的语言是$\{c^n|n\geq1\}$。

$\{a^nb^nc^n \mid n\geq1\}$?= $\{a^nb^n \mid n\geq1\}\{c^n \mid n\geq1\}$

文法

$S\rightarrow S_1S_2$

$S_1\rightarrow ab \mid aS_1b$

$S_2\rightarrow c \mid cS_2$

不能产生语言

$\{a^nb^nc^n \mid n\geq1\}$

而是产生语言

$\{a^nb^n \mid n\geq1\}\{c^n \mid n\geq1\}$

文法

G：$S\rightarrow abc \mid aSbc$

产生的语言为：

$\{a^n(bc)^n \mid n\geq1\}$

焦点：交换 b 和 c 的位置。因此考虑引入变量

G_{14}：$S\rightarrow abc|aSBc$，

$bB\rightarrow bb$

$cB\rightarrow Bc$

（三） 小结

根据文法产生句子或句型比较容易，但根据语言构造文法比较困难（没有太直接的方法可用），甚至不可行。

（四）布置作业

8.(1)(3)

第 09-10 学时

章节：

第 2 章文法

2.4 文法的乔姆斯基体系

主题：文法的乔姆斯基体系

教学目的：使学生理解乔姆斯基体系，掌握文法类别的判断方法。

重点：乔姆斯基体系。

难点：定理 2-1 的证明。

教学过程：

（一） 复习旧课

复习文法的构造 （5 分钟）

一些特殊语言的文法构造；

一些经验性的文法构造方法：直接法，引入变量法，分析特征构造递归定义法等。

（二） 讲授新课

今天我们要介绍一系列重要的定义，它们描述了 Chomsky 对文法的分类。

设文法 $G=(V, T, P, S)$，则

(1) G 叫作 0 型文法，或短语结构文法。

(2) 如果对于 $\forall\ \alpha \to \beta \in P$，均有 $|\beta| \geq |\alpha|$ 成立，则称 G 为 1 型文法，或上下文有关文法(CSG)。对应地，$L(G)$叫作 1 型语言或者上下文有关语言。

(3) 如果对于 $\forall\ \alpha \to \beta \in P$，均有 $|\beta| \geq |\alpha|$，并且 $\alpha \in V$ 成立，则称 G 为 2 型文法，或上下文无关文法(CFG)。对应地，$L(G)$叫作 2 型语言或者上下文无关语言。

(4) 如果对于 $\forall\ \alpha \to \beta \in P$，$\alpha \to \beta$ 均具有形式

$A \to w$

$A \to wB$

其中 $A, B \in V$，$w \in T^*$，则称 G 为 3 型文法，或正则文法(RG)。对应地，$L(G)$叫作 3 型语言或者正则语言。

该体系指明了形式语言研究的方式，0 型文法最为一般，3 型文法最为特殊。

1 型文法规定推导过程中句型长度递增；

2 型文法规定产生式的左边只能为一个语法变量，这样某个语法变量的推导不会依赖于其上下文，这也是“上下文无关语言”，以及与之相对的“上下文有关语言”名称的来历；

3 型文法则允许从一个语法变量最多产生出一个语法变量，且该语法变量只能在句型尾。

短语结构文法是我们所关心的最一般的文法，而右线性文法则是最特殊的文法，本课程将重点对其进行研究。

例 2-16 文法的分类

(1) $G1: S \rightarrow 0|\ 1|\ 00|\ 11$

$G3: S \rightarrow 0|\ 1|\ 0A|\ 1B, A \rightarrow 0, B \rightarrow 0$

(2) $G5: S \rightarrow A|\ B|\ BB, A \rightarrow 0, B \rightarrow 0$

(3) $G14: S \rightarrow aBC|\ aSBC$

$aB \rightarrow ab, bB \rightarrow bb, bC \rightarrow bc, cC \rightarrow cc$

(4) $G7: S \rightarrow \varepsilon|\ 0S$

定理 2-1

L 是 RL 的充要条件是存在一个文法，该文法产生语言 L，并且它的产生式要么是形如 $A \rightarrow a$ 的产生式，要么是形如 $A \rightarrow aB$ 的产生式，其中 A, B 为语法变量，a 为终极符号。

定理的证明过程。

该定理实际上是说明：产生式组 $A \rightarrow w\ |wB$ 与产生式组 $A \rightarrow a_1A_1$, $A_1 \rightarrow a_2A_2$, …, $A_{n-1} \rightarrow a_n|a_nB$ 的表达能力相同，因此它们产生的语言也等价。

形式化的证明。

几点注意：

(1) 这里是对两个模型的等价性证明。

(2) 为证明一个特殊的结论，可以通过证明一个更一般的结论来完成。

(3) 施归纳于推导的步数，这里实际上为字符串的个数。有的情况则为字符串的长度。

(4) 构造与证明。这种思路非常重要

（三）小结

文法的乔姆斯基体系，

文法类别的判断方法。

（四）布置作业

2

第 11-12 学时

章节：

第 2 章文法

2.4 文法的乔姆斯基体系

2.5 空语句

主题：正则文法与空语句

教学目的：使学生理解正则文法（即左线性文法）的概念，了解右线性文法，并理解左/右线性文法与推导/归约的对应关系。掌握空语句的概念，了解空语句的意义，理解空语句的引入不改变文法的类型。

重点：正则文法

难点：空语句的引入不改变文法的类型

教学过程：

（一）复习旧课

复习几种文法的定义（5 分钟）。

（二）讲授新课

下面介绍正则语言

设 $G=(V,T,P,S)$，如果对于 $\forall\ \alpha\to\beta\in P$， $\alpha\to\beta$ 均具有如下形式：

$A\to w$

$A\to wBx$

其中 $A,B\in V$， $w,x\in T^*$，则称 G 为线性文法，对应地，$L(G)$叫作线性语言。

设 $G=(V,T,P,S)$，如果对于 $\forall\ \alpha\to\beta\in P$, $\alpha\to\beta$ 均具有如下形式：

$A\to w$

$A\to wB$

其中 $A,B\in V$, $w\in T^+$，则称 G 为右线性文法，对应地，$L(G)$叫作右线性语言。

左线性文法与右线性文法刚好形成对应。

显然，左/右线性文法都是线性文法，而线性文法可以既不是左线性文法，又不是右线性文法。

定理 2-2

L 是一个左线性语言的充要条件是存在文法 G，G 中的产生式要么是形如 $A \to a$ 的产生式，要么是 $A \to Ba$ 的产生式，且 $L(G) = L$。其中 A, B 为语法变量，a 为终极符号。

但是，我们有一个非常有意思的定理，它说明左、右线性文法是等价的。

定理 2-3 左线性文法与右线性文法等价。

右线性文法自然地对应于句子的推导过程；

左线性文法自然地对应于句子的归约过程。

对于产生语言而言，左、右线性的关系。下面使用一个例子进行说明。细致讲解。

例子

语言{0123456}的左线性文法与右线性文法的构造。

解：限定产生式的右部只有一个终极符号。

右线性文法 Gr:

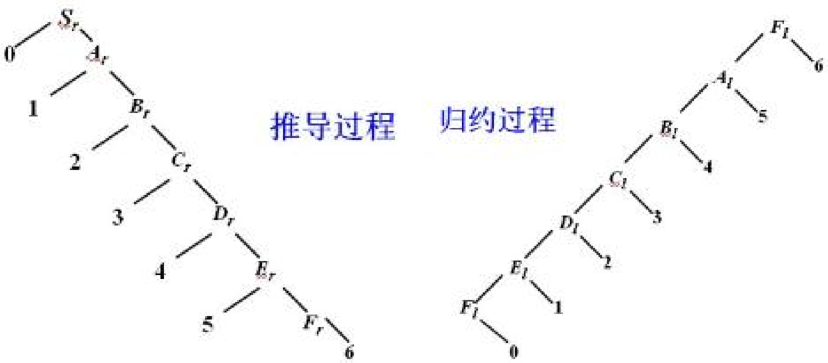
$Sr \to 0Ar$　　$Ar \to 1Br$　　$Br \to 2Cr$　　$Cr \to 3Dr$

$Dr \to 4Er$　　$Er \to 5Fr$　　$Fr \to 6$

左线性文法 Gl:

$Sl \to Al6$　　$Al \to Bl5$　　$Bl \to Cl4$　　$Cl \to Dl3$

$Dl \to El2$　　$El \to Fl1$　　$Fl \to 0$



讨论：左线性文法是否也可以用推导过程？

可以表示，但不符合从左至右的习惯。

定理 2-4:

左线性文法的产生式与右线性文法的产生式混用所得到的文法不是 RG。

证明：只需要举一个反例。

如 $G16$: $S \rightarrow 0A$ $A \rightarrow S1 \mid 1$

只需要举一个反例就可以说明一个断言不成立。

2.5 空语句 （1 学时）

形如 $A \rightarrow \varepsilon$ 的产生式叫作空产生式，也可叫作 ε 产生式。

ε 产生式提出的目的：简化问题处理

ε 产生式存在的原因：除了为生成空语句 ε 外，可以不被用于语言中其他任何句子的推导中。

定理 2-5

设 $G = (V, T, P, S)$为一文法，则存在与 G 同类型的文法 $G' = (V', T, P', S')$，使得 $L(G) = L(G')$，但 G 的开始符号 S' 不出现在的任何产生式的右部。

当文法 $G = (V, T, P, S)$的开始符号 S 不出现在 P 中任何产生式的右部时， G 就是所求。否则

取 $G' = (V', T, P', S')$，其中

$V' = V \cup \{S'\}$

$P' = P \cup \{S' \rightarrow \alpha \mid S \rightarrow \alpha \in P\}$

显然， G' 与 G 有相同的类型。

为证明 $L(G) = L(G')$，需要证明

$L(G') \subseteq L(G)$ 与 $L(G) \subseteq L(G')$

先证 $L(G') \subseteq L(G)$

对 $\forall x \in L(G')$，由文法产生的语言的定义知，在 G' 中存在如下推导：

$S' \Rightarrow \alpha$ 使用产生式 $S' \rightarrow \alpha$

x 使用 P 中除 $S' \rightarrow \alpha$ 以外的产生式

由于对 $\forall$ 产生式 $S' \rightarrow \alpha \in P'$，都 $\exists$ $S \rightarrow \alpha \in P$ 与之对应，有

$S \Rightarrow \alpha$ x

即 $x \in L(G)$

因此有 $L(G')\subseteq L(G)$

再证 $L(G)\subseteq L(G')$

对 $\forall x\in L(G)$，由文法产生的语言的定义知，在 G 中存在如下推导：

$S\Rightarrow\alpha$ 使用产生式 $S\to\alpha$

x 使用 P 中的产生式

而 $P'=P\cup\{S'\to\alpha\mid S\to\alpha\in P\}$，有

$S'\Rightarrow\alpha$ x

即 $x\in L(G')$

因此有 $L(G)\subseteq L(G')$

综上所述，有 $L(G)=L(G')$，定理得证。

思考：如果取 $P'=P\cup\{S'\to S\}$ 易证有 $L(G)=L(G')$，为什么不这样做呢？

因为定理要求同类型的文法，当原文法是正则文法（产生式具有 $A\to w/\ wB$ 的形式）时，产生式 $S'\to S$ 会导致其不是正则文法。

设 $G=(V,T,P,S)$ 是一个文法，如果 S 不出现在 G 的任何产生式的右部，则

(1) 如果 G 是 CSG，则仍然称 $G=(V,T,P\cup\{S\to\varepsilon\},S)$ 为 CSG； G 产生的语言仍然称为 CSL。

(2) 如果 G 是 CFG，则仍然称 $G=(V,T,P\cup\{S\to\varepsilon\},S)$ 为 CFG； G 产生的语言仍然称为 CFL。

(3) 如果 G 是 RG，则仍然称 $G=(V,T,P\cup\{S\to\varepsilon\},S)$ 为 RG； G 产生的语言仍然称为 RL。

定理 2-6

下列命题成立。

(1) 如果 L 是 CSL，则 $L\cup\{\varepsilon\}$ 仍然是 CSL

(2) 如果 L 是 CFL，则 $L\cup\{\varepsilon\}$ 仍然是 CFL

(3) 如果 L 是 RL，则 $L\cup\{\varepsilon\}$ 仍然是 RL

定理 2-7

下列命题成立。

(1) 如果 L 是 CSL，则 $L-\{\varepsilon\}$ 仍然是 CSL

(2) 如果 L 是 CFL，则 $L-\{\varepsilon\}$ 仍然是 CFL

(3) 如果 L 是 RL，则 $L-\{\varepsilon\}$ 仍然是 RL

空产生式/空语句的加入不会对文法和语言产生实质性影响。

进一步说明：

如果 $\varepsilon\notin L$，则 $L-\{\varepsilon\}=L$

更大的命题：如果 $a\notin B$，则 $B-\{a\}=?$

对于 $\forall A\in V$，产生式 $A\rightarrow\varepsilon$ 不会改变文法产生语言的类型。因此约定：在需要的时候，可以出现形如 $A\rightarrow\varepsilon$ 的产生式。

（三）小结

正则文法（即左线性文法）

右线性文法

左/右线性文法与推导/归约的对应关系

空语句

空语句的引入不改变文法的类型

（四）布置作业

8.(1)(3)，9 (1)(2)，10 (1) (2)，11 (1) (2)

第 13-14 学时

章节：

第 3 章有穷状态自动机

3.1 语言的识别

3.2 有穷状态自动机

主题：确定的有穷状态自动机 DFA

教学目的：了解 DFA 是对实际问题的抽象，了解其直观物理模型，理解并掌握 DFA 的形式定义；理解接受一个字符到一个字符串的扩充以及 DFA 接受的句子。

重点：DFA 直观物理模型，DFA 的形式定义，DFA 接受的句子。

难点：DFA 的形式定义，DFA 接受的句子。

教学过程：

（一） **讲授新课**

确定的有穷状态自动机 DFA （1 学时）

定义语言

可以从两个方面进行：

1 ）从产生语言的角度；

2 ）从接收(或识别)语言的角度。

形式语言研究内容

产生一个语言:

1)定义语言中的基本句子;

2)根据其余句子的形成规则，产生出该语言所包含的所有句子。

有限自动机研究内容

使用某种自动机模型来接收字符串

接收的所有字符串形成的集合，也是一个语言

统一的理论

形式语言与自动机作为统一的理论,实际上包括 3 个方面的内容:

1) 形式语言理论(产生语言)

2) 自动机理论(接收语言)

3) 形式语言与自动机的等价性理论

3.1 语言的识别√

识别一个句子是否合法的困难：可以选择的直接推导/归约可能不止一个，走入“死胡同”后需要“回溯”，增加了问题的难度。

例：$G: S\rightarrow aA|aB, A\rightarrow aA|c, B\rightarrow aB|d$

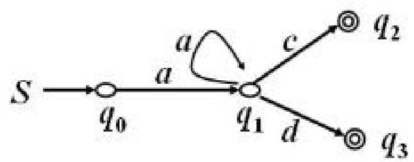
$aaad$ 是该方法所定义的语言的句子吗？

使用推导，总共有 $2^4=16$ 种不同的选择；

使用归约比较规范，但复杂。

但问题规模增大后，可能导致两者都不容易

自动机模型



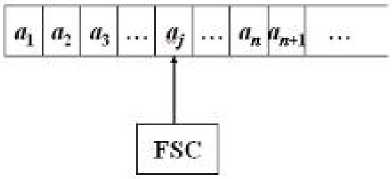
与例中文法作用相同的一个识别系统，如果一个句子使用得状态由 $q0$ 转移到任意一个最终状态($q2$ 或 $q3$)，那么该句子就是合法的。

它把关键性的决策(使用 $S\rightarrow aA$ 还是 aB)留到了能真正作决策的地方(出现 c 或 d)，避免了回溯。

模型的几个方面

1. 系统具有有穷个状态，不同的状态代表不同的意义。
2. 系统所有可能处理的字符构成一个字母表。
3. 系统根据当前状态和当前读入的字符转到新状态。
4. 系统有一个初始状态。
5. 系统有一些终止状态，如果一个字符串引导系统从初始状态到达某个终止状态，则该字符串是系统所识别语言的句子。

有穷状态自动机的物理模型



系统的每一个动作由三个节拍构成：

1. 读入读头所指向的字符；
2. 根据当前状态和读入的字符改变有穷控制器的状态；
3. 将读头向右移动一格。

3.2 有穷状态自动机

有穷状态自动机(finite automaton, FA) M 是一个五元组：

$M = (Q, \Sigma, \delta, q0, F)$

其中，

Q—状态的非空有穷集合。$\forall q \in Q$, q 称为 M 的一个状态(state)。

Σ—输入字母表。输入字符串都是 Σ 上的字符串。

δ—状态转移函数(transition function)。$\delta : Q \times \Sigma \to Q$。对$\forall(q, a) \in Q \times \Sigma$，$\delta(q, a) = p$ 表示 M 在状态 q 读入 a，将状态变成 p，并将读头向右移动一个带方格而指向输入字符串的下一个字符。

$q0$—M 的开始状态(initial state), $q0 \in Q$。

F—M 的终止状态(final state)集合, $F \subseteq Q$。$\forall q \in F$, q 称为 M 的终止状态。也称为接受状态(accept state)。

例 3-1

下面是有穷状态自动机

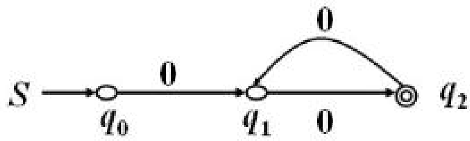
$M_1 = (\{q_0, q_1, q_2\}, \{0\}, \delta_1, q_0, \{q_2\})$

δ_1 如下表所示

状态说明	状态	输入字符
		0
开始状态	$q0$	$q1$

	$q1$	$q2$
终止状态	$q2$	$q1$

思考：哪些字符串能使其达到终止状态？



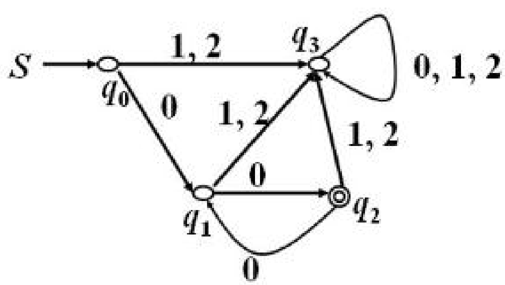
偶数(至少为 2)个 0。

M2 = ({q0, q1, q2 , q3}, {0, 1, 2}, δ2 , q0 , {q2})

δ_2 如下表所示

状态说明	状态	输入字符		
		0	1	2
开始状态	$q0$	$q1$	$q3$	$q3$
	$q1$	$q2$	$q3$	$q3$
终止状态	$q2$	$q1$	$q3$	$q3$
	$q3$	$q3$	$q3$	$q3$

思考：哪些字符串能使其达到终止状态？



偶数(至少为 2)个 0。

扩充 δ

将 δ 扩充为~δ : Q× Σ* → Q。对∀q∈ Q, w ∈ Σ*：

(1) ~δ (q, ε) = q

(2) $\tilde{\delta}(q, wa) = \delta(\tilde{\delta}(q, w), a)$

扩充的作用：

(1) 加入单位元素 ε ；

(2) 从概念上允许一次接收字母表的任意一个字符串，而不仅是一个字符。

$\tilde{\delta}$ 与 δ “兼容”

δ 的定义域 Q× Σ 是 $\tilde{\delta}$ 的定义域的 Q× Σ*真子集，

需要考虑在上 Q× Σ ，$\tilde{\delta}$ 是否与 δ 有相同的函数值：

$\tilde{\delta}(q, a) = \tilde{\delta}(q, \varepsilon a)$

$= \delta(\tilde{\delta}(q, \varepsilon), a)$ 根据(2)

$= \delta(q, a)$ 根据(1)

因此 与 δ “兼容”。约定：以后直接用 δ 代替 。

运算符/函数“兼容”的另一个例子：加号 “+”。它在整数集、有理数集、实数集中都有定义，而在后者中的定义均与前者保持“兼容”。

确定的有穷状态自动机

对∀q ∈ Q, a ∈ Σ , δ(q, a)均有确定的值。将这种 FA 称为确定的有穷状态自动机(deterministic finite automaton, DFA)

设 $M = (Q, \Sigma, \delta, q0, F)$ 为一个 FA，对于 $\forall x \in \Sigma^*$，如果 $\delta(q0, x) \in F$，则称 x 被 M 所接受；如果 $\delta(q0, x) \notin F$，则称 M 不接受 x。

$L(M) = \{x \mid x \in \Sigma^* 且\ \delta(q0, x) \in F\}$ 称为由 M 接受(识别)的语言。

例 3-1 中两个 DFA 所识别的语言是什么？

$L(M_1) = L(M_2) = \{0^{2n} \mid n \geq 1\}$

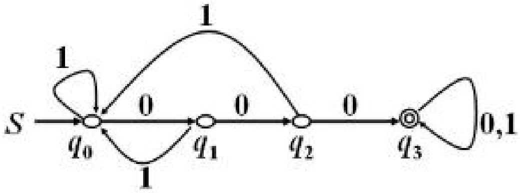
设 M_1, M_2 为 FA，如果 $L(M_1) = L(M_2)$，则称 M_1 与 M_2 等价。

- ✓ 再次注意：“等价”与“相同/相等”是不同概念。
- ✓ 本定义及定义 3-2 针对 FA，而不仅仅是 DFA。

例 3-2

构造一个 DFA，它接受的语言为 $\{x000y \mid x, y \in \{0, 1\}^*\}$

- ✓ 语言特点为：有 3 个连续的 0
- ✓ 状态：初始状态：q_0，已接收 1 个 0 ：q_1，已接收连续的 2 个 0 ：q_2，已接收连续的 3 个 0 ： q_3。如果接收 1 个或 2 个 0 后接收到 1，就应返回初始状态重新计数；如果到了 q_3，就表示条件已经满足，因此 q_3 应为一个终止状态。另外，在状态 q_3 无论接收什么字符都不会离开。



接受语言$\{x000y \mid x, y \in \{0, 1\}^*\}$的ＤＦＡ M的图示

五元组表示：

$M = (\{q_0, q_1, q_2, q_3\}, \{0, 1\},$

$\{\delta(q_0, 0) = q_1, \delta(q_1, 0) = q_2, \delta(q_2, 0) = q_3,$

$\delta(q_0, 1) = q_0, \delta(q_1, 1) = q_0, \delta(q_2, 1) = q_0,$

$\delta(q_3, 0) = q_3, \delta(q_3, 1) = q_3\}, q0, \{q_3\})$

其状态转移函数（实际上为一个集合）的有几个元素？为什么？

状态转移图中是否有相应数目的箭头？

4*2 = 8

本例中少一个，因为有两个 $q3$ 向自己转移被合并。

设 $M = (Q, \Sigma, \delta, q_0, F)$ 为一个ＦＡ，满足如下条件的有向图称为 M 的状态转移图(t r a s i d i a g r a m)

(1) $q \in Q \Leftrightarrow q$是该有向图中的一个顶点。

(2) $\delta(q, a) = p \Leftrightarrow$ 图中有一条从顶点 q 到顶点 p 的标记为 a 的弧。

(3) $q \in F \Leftrightarrow$ 标记为 q 的顶点用双层圈标出。

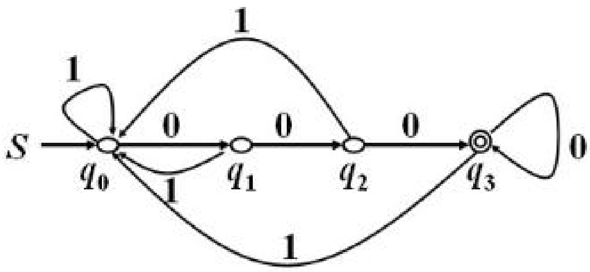
(4)用标有 S 的箭头指出 M 的开始状态。

状态转移图已清楚地给出了 FA 五元组的各元组（不包括无用的字符）；

对于 DFA 来说，各顶点的出度恰好等于输入字母表中所含字符数；

字符串 x 被 F A M 所接受 iff 在 M 的状态转移图中存在一条从开始状态到某一终止状态的有向路，该路的标记依次并置构成 x；

一个 FA 可以有多个终止状态。



接受语言 $\{x000 \mid x \in \{0,1\}^*\}$ 的 D F A

注意：应以 000 结尾，因此在 q_3 状态接收 1 需要回到 q_0 。

（二）小结

语言的识别

有穷状态机的形式定义

有穷状态机 M 接受(识别)的语言

（三）布置作业

1. 左图　 2. (1)(3)(9)

第 15-16 学时

章节：

第 3 章有穷状态自动机

3.2 有穷状态自动机

主题：确定的有穷状态自动机 DFA

教学目的：使学生掌握 DFA 识别的语言；

重点：DFA 识别的语言

难点：DFA 识别的语言

教学过程：

（一）复习旧课

对上次课所学语言的识别、有穷状态机的形式定义等内容进行复习（5 分钟）

（二）讲授新课

设 $M=(Q,\Sigma,\delta,q_0,F)$ 为一个 FA，$x,y\in\Sigma^*$，$\delta(q_0,x)=q$，xqy 称为 M 的一个即时描述（instantaneous description，ID），它表示 xy 是 M 正在处理的一个字符串，x 引导 M 从 q_0 启动并到达状态 q，当前正指向的 y 首字符。

如果 $xqay$ 是 M 的一个即时描述（x 是一个字符串，从 q_0 开始处理它后到达当前状态；q 为当前状态，ay 可看作字符 a 与字符串 y 连接而成的字符串，当前指针正指向字符 a），且 $\delta(q,a)=p$，则经过字符 a 的处理后，即时描述变为 $xapy$ 这一过程记作：$xqay \vdash_M xapy$

对例 3-3 定义的 M 处理输入串 1010010001，有

$$
\begin{aligned}
& q_0 1010010001 \vdash_M 1q_0 010010001 \vdash_M 10q_1 10010001 \\
& \vdash_M 101q_0 0010001 \vdash_M 1010q_1 010001 \vdash_M 10100q_2 10001 \\
& \vdash_M 101001q_0 0001 \vdash_M 1010010q_1 001 \vdash_M 10100100q_2 01 \\
& \vdash_M 101001000q_3 1 \vdash_M 1010010001q_0
\end{aligned}
$$

即时描述扩展符号

可将 $\vdash_M$ 看成是M的所有即时描述集合上的二元关系。

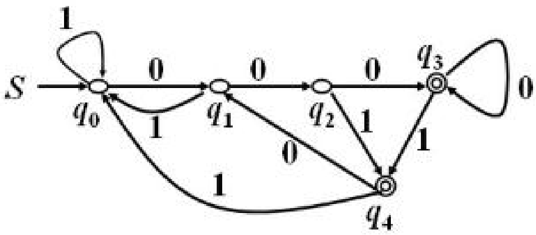
$\vdash_M^n$，或 $(\vdash_M)^n$ 表示经过n步移动到达(n为一个确定的正整数)；

$\vdash_M^*$，或 $(\vdash_M)^*$ 表示经过若干步(可以为0)移动到达；

$\vdash_M^+$，或 $(\vdash_M)^+$ 表示经过至少一步移动到达。

设 $M = (Q, \Sigma, \delta, q_0, F)$ 为一个 FA 对 $\forall\ q \in Q$，能引导FA从开始状态到达 q 的字符串的集合为

$set(q) = \{x \mid x \in \Sigma^*, \delta(q_0, a) = q\}$



接受语言

$\{x000 \mid x \in \{0, 1\}^*\} \cup \{x001 \mid x \in \{0, 1\}^*\}$ 的 DFA

根据状态进行划分

根据图 3-5 所给的 DFA

⊙ $set(q_0) = \{x \mid x \in \{0, 1\}^*$，$x = \varepsilon$ 或 x 以 1 但不包括 001 结尾$\}$

⊙ $set(q_1) = \{x \mid x \in \{0, 1\}^*$，$x = 0$ 或 x 以 10 结尾$\}$

⊙ $set(q_2) = \{x \mid x \in \{0, 1\}^*$，$x = 00$ 或 x 以 100 结尾$\}$

⊙ $set(q_3) = \{x \mid x \in \{0, 1\}^*$，$x$ 以 000 结尾$\}$

⊙ $set(q_4) = \{x \mid x \in \{0, 1\}^*$，$x$ 以 001 结尾$\}$

⊙ 它们是对 $\{0, 1\}^*$ 的划分

例 3-4 √

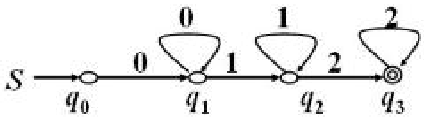
接受语言$\{0^n1^m2^k \mid n, m, k \geq 1\}$的 D F A

q_0 ： 启动状态，未读入任何字符

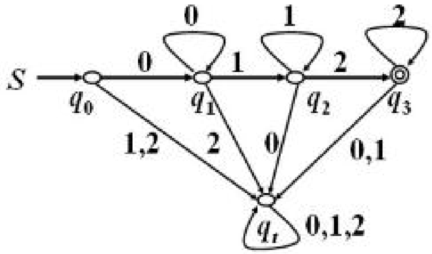
q_1 ： 已读入一个或多个 0

q_2 ： 到达 q_1 状态后读入一个或多个 1

q_3 ： 到达 q_2 状态后读入一个或多个 2



出现该语言不能接受的序列时进入陷井状态 q_t（t 表示 t r a p ）



陷井状态

当 FA 进入陷井状态后，就无法离开

如果陷井状态为终止状态，就表示后面无论是什么字符串都可以被该 DFA 识别，如：接受语言 $\{x00y \mid x, y \in \{0,1\}^*\}$的 F A；

如果陷井状态不为终止状态，就表示后面无论是什么字符串都不可以被该 DFA 识别，如：接受语言 $\{x \mid x \in \{0,1\}^+$且 x 中不含形如 00 的子串$\}$的 F A

特别是后一种情况，用文法表示比较困难，用语言表示也往往需要用集合的减法。

FA 的构造 （1 学时）

用画图的方式构造 FA 比较方便、直观。可以先根据语言的主要特征画出其“主体框架”，再考虑一些细节要求。

FA 的记忆功能

FA 的状态具有一定的记忆功能，不同的状态对应于不同的情况，如：构造语言$\{x000y \mid x, y \in \{0, 1\}^*\}$的 FA 时，需要记忆已经读入 0 个连续的 0，已经读入 1 个连续的 0，已经读入 2 个连续的 0，已经读入至少 3 个连续的 0 等至少 4 个状

态。当需要记忆无穷个状态时，就无法构造相应的 F A 如语言 $\{0^n1^n / n \geq 1\}$ 。当然，语言 $\{0^m1^n / m, n \geq 1\}$ 只需要 4 个状态。

例 3-5

构造一个接受语言 $\{x \mid x \in \{0, 1\}^*$, 且当把 x 看成二进制数时，x 模 3 与 0 同余$\}$的 D F A

分析：模 3 同余自然就构成了二进制数的一个划分。该划分导致 3 个集合：

$\{3n \mid n$ 为非负整数$\}$，其元素包括$\{0, 11, 110, 1001, \cdots\}$;

$\{3n+1 \mid n$ 为非负整数$\}$，其元素包括$\{1, 100, 111, 1010, \cdots\}$;

$\{3n+2 \mid n$ 为非负整数$\}$ ，其元素包括$\{10, 101, 1000, 1011, \cdots\}$。

先画出如下三个状态。

q_0 : 希望能代表$\{3n \mid n$ 为非负整数$\}$

q_1 : 希望能代表$\{3n+1 \mid n$ 为非负整数$\}$

q_2 : 希望能代表$\{3n+2 \mid n$ 为非负整数$\}$

当 $x = 3n$ 时，$x0 = 2*3n + 0$，仍然应由 $q0$ 代表

$x1 = 2*3n + 1$，应由 $q1$ 代表

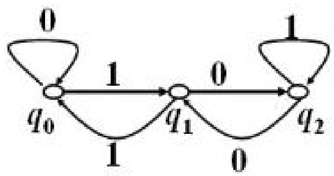
当 $x = 3n + 1$ 时，$x0 = 2*(3n + 1) + 0 = 6n + 2$，应由 $q2$ 代表;

$x1 = 2*(3n + 1) + 1 = 6n + 3$，应由 $q0$ 代表

当 $x = 3n + 2$ 时，$x0 = 2*(3n + 2) + 0 = 6n + 3 + 1$,应由 $q1$ 代表;

$x1 = 2*(3n + 2) + 1 = 6n + 3 + 2$，应由 $q2$ 代表

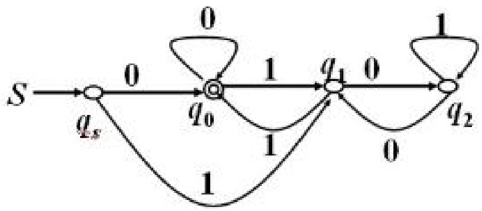
其状态转移图(部分)如下所示：



应将 q_0 设置为终止状态。

空句子不可能代表二进制数，所以应设置一个新的状态 q_s 作为初始状态，在 q_s 下读入 0 应转到状态 q_0 ，读入 1 则应转到状态 q_1。

其完整状态转移图如下所示：



思考：

为表示模 3 与 1 同余，应怎样画状态图？

模 4 与 0 同余呢？

只需要改变其终止状态。

同样的方法可得，其状态要多一个，转移也不同。

（三）小结

即时描述

DFA 的构造

（四）布置作业

1.（1）使用即时描述

第 17-18 学时

章节：

第 3 章有穷状态自动机

3.3 不确定的有穷状态自动机

主题：不确定的有穷状态自动机 NFA

教学目的：使学生理解 NFA 的定义，掌握 NFA 向 DFA 的转换。

重点：NFA 的定义，NFA 向 DFA 的转换。

难点： NFA 向 DFA 的转换。

教学过程：

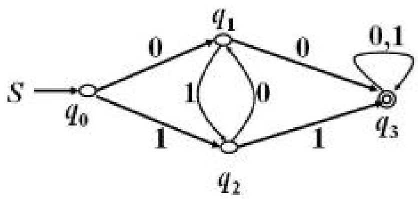
（一）复习旧课

对上次课所学 DFA 的形式定义，DFA 接受的句子，DFA 识别的语言等内容进行复习（约 7 分钟）

（二）讲授新课

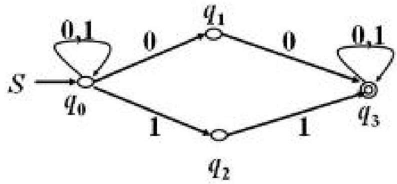
3.3 不确定的有穷状态自动机 （1 学时）

作为对 DFA 的修改



接受语言$\{x|\ x\in\{0,1\}^*$,且 x含有子串 00 或 11$\}$的 D F A

“直接”的 F A



希望是接受语言$\{x|\ x\in\{0,1\}^*$,且 x含有子串 00 或 11$\}$的 F A

与 DFA 的区别

并不是对于所有的$(q, a) \in Q \times \Sigma$，$\delta(q, a)$都有一个状态与之对应 (相当于 $f(x)$对某些 x 没有函数值)；

并不是对于所有的 $(q, a) \in Q \times \Sigma$，$\delta(q, a)$都只对应一个状态(相当于 $f(x)$对某些 x 有多个函数值)。

这就是“不确定”的由来

理解N F A

$\delta(q, a)$对应的是状态的一个子集，当这个子集为空时，表示没有状态与之对应；当这个子集的元素个数大于 1 时，表示有多个状态与之对应。

从这个意义上，$\delta(q, a)$仍是通常意义下的一个函数，只是其值域发生了改变。

当 $\delta(q, a)$对应的所有子集元素个数为 1 时，NFA 退化为 D F A。

NFA 的特点

具有“智能”

只要在 FA 中存在一条从开始状态出发，最终到达某一个终止状态的标记为 x 的路径，就认为它接受了串 x，否则认为它不接受串 x。从这个意义上来说，这类 FA 与 DFA 的作用是一致的(识别句子是否合法)。

不确定的有穷状态自动机的形式定义(1 学时)

不确定的有穷状态自动机(non-deterministic finite automaton，NFA)M是一个五元组：

$M=(Q, \Sigma, \delta, q_0, F)$

其中，Q, Σ, q_0, F 状态的意义同D F A(定义 3-1)

δ —状态转移函数。$\delta: Q \times \Sigma \rightarrow 2^Q$。对$\forall (q, a) \in Q \times \Sigma$，$\delta(q, a)=\{p_1, p_2, \cdots, p_m\}$ 表示 M在状态 q 读入字符 a，可以选择地将状态变成 p_1，p_2，…，或者 p_m，并将读头向右移动一个带方格而指向输入字符串的下一个字符。

扩充 δ

将 δ扩充为$\tilde{\delta}: Q \times \Sigma^* \rightarrow 2^Q$。对$\forall q \in Q$，$w \in \Sigma^*$，$a \in \Sigma$：

(1) $\tilde{\delta}(q, \varepsilon)=\{q\}$

(2) $\tilde{\delta}(q, wa) = \{p \mid \exists r \in \tilde{\delta}(q, w), \text{ s t } p \in \delta(r, a)\}$

扩充的作用：

(1) 加入单位元素 ε

(2) 从概念上允许一次接收字母表的任意一个字符串，而不仅是一个字符

设 $M = (Q, \Sigma, \delta, q_0, F)$ 为一个 N F。A对于 $\forall x \in \Sigma^*$：

如果 $\delta(q_0, x) \cap F \neq \emptyset$,则称 x 被 M所接受;如果 $\delta(q_0, x) \cap F = \emptyset$,则称 M不接受 x。

$L(M) = \{x \mid x \in \Sigma^*$且 $\delta(q_0, x) \cap F \neq \emptyset\}$

称为由 M接受(识别)的语言。

分析

NFA 适于构造“包含子串…”,“以串…开始,串…结束”,“(倒数)第…个字母是…”，“满足条件…”的语言;

NFA 不适于构造“不包含子串…”，“不满足条件…”的语言,在这些情况下它可能退化为D F。A

定理 3-1 √

NFA 与 DFA 等价。

NFA 与 DFA 等价是指 NFA 和 DFA 能识别相同的语言类。实际上，它们都是正则语言的识别模型。

回顾等价的概念：识别相同的语言(定义 3-3)。

证明思路

(1) 显然，DFA 是 NFA 的特殊形式；即所有的 DFA 已经用 NFA 的形式表示；

(2) 需要证明对于任意给定的N F,A存在与之等价的D F。A

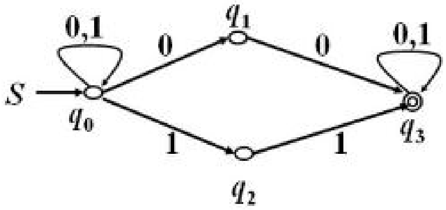
根据 NFA 构造 D F,A 将状态集合从 Q变换到 $2Q$,这样 DFA 中的任意一个状态实际上是 NFA 中某些状态的整体(这里避免使用术语“集合”)。

使用归纳法证明 DFA 与 NFA 的状态转移存在一一对应关系。

证明由 DFA 和 NFA 能识别相同的语言。

例 3-7

求与下图所示 NFA 等价的 D F。A



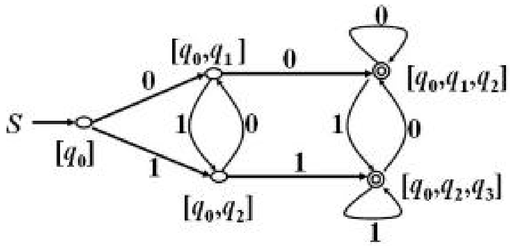
分析：DFA 的一个状态对应对 NFA 的一个状态集合，因此，需要 $2^4=16$ 个状态。下表列出其状态转移。

从[q_0]开始可以到达的状态为可达状态，对于 DFA 来说，只有可达状态才是有用的。

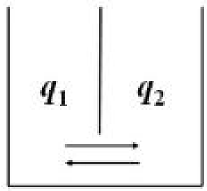
状态说明		状态	输入字符	
			0	1
开始状态	√	[$q0$]	[$q0$, $q1$]	[$q0$, $q2$]
		[$q1$]	[$q3$]	[Ø]
		[$q2$]	[Ø]	[$q3$]
终止状态		[$q3$]	[$q3$]	[$q3$]
	√	[$q0$, $q1$]	[$q0$, $q1$, $q3$]	[$q0$, $q2$]
	√	[$q0$, $q2$]	[$q0$, $q1$]	[$q0$, $q2$, $q3$]
终止状态		[$q0$, $q3$]	[$q0$, $q1$, $q3$]	[$q0$, $q2$, $q3$]
		[q 1 ,q 2]	[q 3]	[q 3]
终止状态		[q 1 ,q 3]	[q 3]	[q 3]
终止状态		[q 2 ,q 3]	[q 3]	[q 3]
		[q 0 ,q 1 , q 2]	[q 0 ,q 1 ,q 3]	[q 0 ,q 2 ,q 3]
终止状态	√	[q 0 , q 1 q 3]	[q 0 ,q 1 ,q 3]	[q 0 ,q 2 ,q 3]
终止状态	√	[q 0 , q 2	[q 0 ,q 1 ,q 3]	[q 0 ,q 2 ,q 3]

		q 3]		
终止状态		[q 1 , q 2 q 3]	[q 3]	[q 3]
终止状态		[q 0 , q 1 q 2 q 3]	[q 0 q 1 q 3]	[q 0 q 2 q 3]
		[Ø]	[Ø]	[Ø]

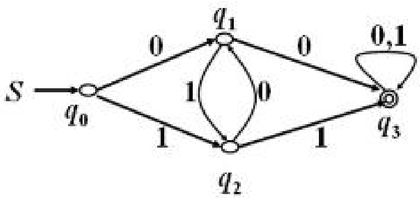
获得的D F。A



“准”陷井状态。



简化后的D F。A



减少工作量的方法:

只列出可达状态的转移，即填表时不需要填完

1. 填表时可用[0, 1]代替[q_0, q_1]
2. [Ø]总是不可达的，可将该状态对应的转移去掉

说明:

[Ø]这种记法主要是为了保持上下文一致,实际上,它与[q_0, q_1]应有所区别。

（三）小结

NFA 的定义,

NFA 向 DFA 的转换。

（四）布置作业

10. (2) (1) (5)

第 19-20 学时

章节：

第 3 章有穷状态自动机

3.4 带空移动的有穷状态自动机

主题： 带空移动的有穷状态自动机 ε-NFA

教学目的： 使学生理解 ε-NFA 的定义，掌握 ε-NFA 向 NFA 的转换。

重点： ε-NFA 的定义，ε-NFA 向 NFA 的转换

难点： ε-NFA 向 NFA 的转换

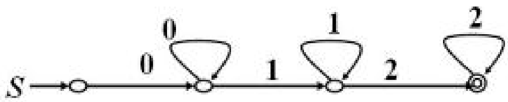
教学过程：

（一）复习旧课

对上次课所学重点内容 NFA 的定义，NFA 向 DFA 的转换进行复习（约 5 分钟）

（二）讲授新课

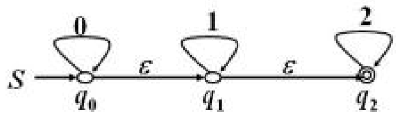
3.4 带空移动的有穷状态自动机（1 学时）



接受语言$\{0^n1^m2^k \mid n,\ m,\ k \ge 1\}$的 N F A

接受语言$\{0^n1^m2^k \mid n,\ m,\ k \ge 0\}$的 NFA 见图 3-13

对 NFA 进行修改



接受语言$\{0^n1^m2^k \mid n,\ m,\ k \ge 0\}$的 ε -N F A

定义 3-9

带空移动的不确定的有穷状态自动机(non-deterministic finite automaton with ε-moves，ε-NFA) M 是一个五元组：

$M=(Q,\ \Sigma,\ \delta,\ q_0,\ F)$

其中，Q, Σ, q_0, F 状态的意义同DFA定义3-1)

δ —状态转移函数。 δ: $Q\times(\cup\Sigma\{\varepsilon\})\rightarrow 2^Q$。

对$\forall(q,\ a)\in Q\times\Sigma$， $\delta(q,\ a)=\{p_1,\ p_2,\ \cdots,\ p_m\}$ 表示M在状态q读入字符a,可以选择地将状态变成将状态变成p_1, p_2, $\cdots$,或者p_m,并将读头向右移动一个带方格而指向输入字符串的下一个字符。

对$\forall q\in Q$，$\delta(q,\ \varepsilon)=\{p_1,\ p_2,\ \cdots,\ p_m\}$表示$M$在状态$q$不读入任何字符，可以选择地将状态变成$p_1$, p_2, $\cdots$,或者p_m。

扩充δ

将δ扩充为$\tilde{\delta}$: $Q\times\Sigma^*\rightarrow 2^Q$。

δ与$\tilde{\delta}$的重要区别:由于空移动ε的存在,即便对于单个的字符,$\tilde{\delta}$也可能产生多次移动。由于该区别的存在，不可以再用δ代替$\tilde{\delta}$。

ε-$CLOSURE(q)=\{p|$ 从q到p有一条标记为ε的路$\}$

与NFA及DFA的区别：NFA及DFA中，$\tilde{\delta}(q,\ \varepsilon)=\{q\}$。

约定

对$\forall\ q\in Q$， $q\in\delta(q,\ \varepsilon)$

1. 若不包含，则根据ε-$CLOSURE$的定义， $(q,\ \varepsilon)$也不应包含它，这导致书中表3-8相应表项不易解释(0个ε);
2. 将NFA看成ε-NFA时，可以认为每个状态都有到自身标记为ε的弧。

但对表的改变不会对ε-NFA到NFA的转换产生影响,因此可将此作为一种约定。

定理3-2

ε-NFA与NFA等价。

证明思路：

(1) 显然， NFA是ε-NFA的特殊形式；即所有的NFA已经用ε-NFA的形式表示；

(2) 需要证明对于任意给定的ε-NFA存在与之等价的NFA

构造方法

设有 ε-N F A

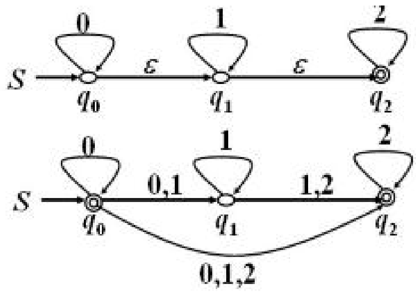
$M1 = (Q, \Sigma, \delta1, q0, F)$

构造 N F $M2 = (Q, \Sigma, \delta2, q0, F2)$，其中

$F2 =$	$F \cup \{q0\}$ 如果 $F\cap$ ε-C L O S(q0)R $\neq\emptyset$
	$F \cup \{q0\}$ 如果 $F\cap$ ε-C L O S(q0)R $\neq\emptyset$

对$\forall(q, a) \in Q\times\Sigma$，使 $\delta2\ (q, a) = \quad (q, a)$

例 3-8 √



接受语言$\{0^n1^m2^k \mid n, m, k \geq 0\}$的 ε-NFA 与 N F A

（三）小结

ε-NFA 的定义

ε-NFA 向 NFA 的转换

（四）布置作业

15. (1)

要求:画出 ε-NFA 状态转移图;用表列出 NFA 状态转移函数,画出 NFA 状态转移图。

第 21-22 学时

章节：

第 3 章有穷状态自动机

3.5FA 是正则语言的识别器

3.6FA 的一些变形

主题：FA 是正则语言的识别器及其变形

教学目的：使学生掌握 RG 与 FA 相互转换方法。

重点：RG 与 FA 相互转换方法

难点：RG 与 FA 相互转换方法

教学过程：

（一）复习旧课

对上次课的重点内容 ε-NFA 的定义，ε-NFA 向 NFA 的转换等进行复习（约 5分钟）

（二）讲授新课

3.5 FA 是正则语言的识别器

正则文法的推导过程与 DFA 的处理过程相对应。

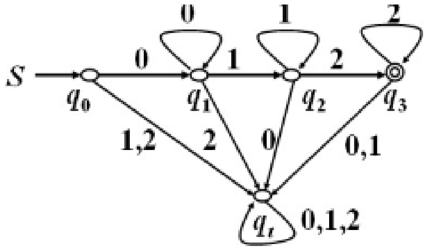
- $A_0 \Rightarrow a_1 A_1 \Rightarrow a_1 a_2 A_2 \Rightarrow \ldots \Rightarrow a_1 a_2 \ldots a_n$
- $q_0 a_1 a_2 \ldots a_n \vdash_M a_1 q_1 a_2 \ldots a_n \vdash_M a_1 a_2 q_1 \ldots a_n \vdash_M a_1 a_2 \ldots a_n q_n$

定理 3-3

FA 接受的语言是正则语言。

分析：由于正则语言是能被正则文法产生出来的语言（见定义 2-6　文法的 Chomsky 体系)，因此只需要证明：对于任意 F A(识别语言)，都存在相应的正则文法与之对应（产生语言）　。

例 3-9



文法

$$q_0 \to 0q_1 \mid 1q_t \mid 2q_t$$
$$q_1 \to 0q_1 \mid 1q_2 \mid 2q_t$$
$$q_2 \to 0q_t \mid 1q_2 \mid 2q_3 \mid 2$$
$$q_3 \to 0q_t \mid 1q_t \mid 2q_3 \mid 2$$
$$q_t \to 0q_t \mid 1q_t \mid 2q_t$$

由于这里的陷井状态不是终止状态,它可以在文法中被删除。化简后文法:

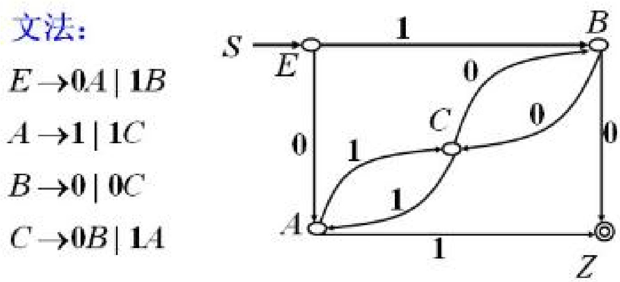
$$q_0 \to 0q_1$$
$$q_1 \to 0q_1 \mid 1q_2$$
$$q_2 \to 1q_2 \mid 2q_3 \mid 2$$
$$q_3 \to 2q_3 \mid 2$$

定理 3-4

正则语言可以由 FA 接受。

分析：只需要证明，对于任意正则文法，都存在相应的 FA 与之对应 。

例 3-10



练习(见习题)：22(1)

推论 3-1

FA 与正则文法等价。

3.6 FA 的一些变形

双向有穷状态自动机

带输出的ＦＡ

Moore 机：根据状态决定输出

Mealy 机：根据移动决定输出

（三）小结

RG 与 FA 相互转换方法

（四）布置作业

20， 22（1）

第 23-24 学时

对第一、二、三章的内容给出一个系统的复习。（约 25 分钟）

进行第一、二章习题的评讲。学生评讲约 35 分钟。

进行第一、二章习题的评讲。老师评讲约 30 分钟。

第 25-26 学时

章节：

第 4 章正则表达式

4.1 启示

4.2 正则表达式的形式定义

主题：正则表达式的形式定义

教学目的：使学生理解正则表达式的形式定义。

重点：正则表达式的形式定义。

教学过程：

（一）随堂中期考试（1 学时）

第二章 8(8) 9(7)

第三章 10(7)(8)

（二）讲授新课

4.1 启示

正则文法

$G: A \to aA \mid aB \mid cE$

$B \to bB \mid bC$

$C \to cC \mid c$

$E \to cE \mid bF$

$F \to dF \mid eF \mid aH$

$H \to aH \mid a$

相应的 NFA

各状态对应的集合

1. $set(A) = \{a^n \mid n \geq 0\} = \{a\}^*$
2. $set(B) = set(A)\{a\}\{b^m \mid m \geq 0\} = \{a^n a\, b^m \mid m, n \geq 0\} = \{a\}^*\{a\}\{b\}^* = \{a\}^+\{b\}^*$
3. $set(C) = set(B)\{b\}\{c\}^* = \{a\}^+\{b\}^+\{c\}^*$
4. $set(D) = set(C)\{c\} = \{a\}^+\{b\}^+\{c\}^+$

5. ...

相应的表达式

$L(M) = \{a\}^+\{b\}^+\{c\}^+$

$\cup \{a\}^*\{c\}^+\{b\}(\{d\} \cup\{e\})^*\{a\}^+\{a\}$

简洁形式

$a^+b^+c^+ + a^*c^+b(d+e)^*a^+a$

$= aa^*bb^*cc^* + a^*cc^*b(d+e)^*aaa^*$

4.2 正则表达式的形式定义

定义 4-1 设 Σ 是一个字母表，

(1) Ø是 Σ 上的正则表达式(regular expression, RE)，它表示语言Ø 。

(2) ε是 Σ 上的正则表达式，它表示语言$\{\varepsilon\}$

(3) 对于$\forall a\in \Sigma$，a 是 Σ 上的正则表达式，它表示语言$\{a\}$

(4) 如果 r 和 s 分别是 Σ 上的表示语言 R 和 S 的正则表达式，则

r 与 s 的“和”$(r+s)$是 Σ 上的正则表达式，

$(r+s)$表达的语言为 $R\cup S$

r 与 s 的“乘积”(rs)是 Σ 上的正则表达式，

(rs)表达的语言为 RS

r 的克林闭包 (r^*)是 Σ 上的正则表达式，

(r^*)表达的语言为 R^*

(5) 只有满足(1)(2)(3)(4)的表达式才是 Σ 上的正则表达式。

例 4-1

设 Σ ={0, 1}，下面是 Σ 上的一些正则表达式及其对应的语言：

(1) 正则表达式 0，表示语言{0}

(3) 正则表达式(0+1)，表示语言{0, 1}

(4) 正则表达式(01)，表示语言{01}

(5) 正则表达式$((0+1)^*)$，表示语言$\{0, 1\}^*$

(6) 正则表达式$((00)((00)^*))$，表示语言$\{00\}\{00\}^*$

(8) 正则表达式$((((0+1)^*)000((0+1)^*))$，表示字母表{0, 1}上至少含有 3 个连续 0 的串组成的语言

约定

众多的括号增加表达式的复杂性，约定：

(1) $r^+ = (r(r^*)) = ((r^*)r)$

(2) 闭包优先级最高，乘运算次之，加运算最低

(3) 在意义明确时，正则表达式 r 表示的语言记为 $L(r)$，也可以直接记为 r

(4) 各运算均执行左结合规则

定义 4-2

设 r, s 是字母表上的两个正则表达式，如果 $L(r) = L(s)$，则 r 称为与 s 相等(equivalence, 也称作等价)，记作 $r = s$

基本规律

(1) 结合律: $(rs)t = r(st)$

$$(r + s) + t = r + (s + t)$$

(2) 分配律: $r(s + t) = rs + rt$

$$(s + t)r = sr + tr$$

(3) 交换律: $r + s = s + r$ (注意 $rs \neq sr$)

(4) 幂等律: $r + r = r$

(5) 加法运算零元素: $r + \emptyset = r$

(6) 乘法运算单位元素: $r\varepsilon = r$ (注意 $r + \varepsilon \neq r$)

(7) 乘法运算零元素: $r\emptyset = \emptyset$

基本规律(续)

(8) $L(\emptyset) = \emptyset$

(9) $L(\varepsilon) = \{\varepsilon\}$

(9) $L(\emptyset^*) = \{\varepsilon\}$(参见 $P_{41}30(10)$)

定义 4-3

设 r 是字母表上的一个正则表达式，r 的 n 次幂定义为：

(1) $r^0 = \varepsilon$

(2) $r^n = r^{n-1}r$

基本规律：

$r^n r^m = r^{n+m}$ (注意 $(rs)^n \neq r^n s^n$)

例 4-2

设 Σ ={0, 1}，

00 表示语言{00}

$(0+1)^*00(0+1)^*$表示所有至少含两个连续 0 的串组成的语言

$(0+1)^*1(0+1)^9$ 表示所有倒数第 10 个字符为 1 的串组成的语言

（三）小结

正则表达式的形式定义

（四）布置作业

1. (1)(5)(10)

第 27-28 学时

章节：

第 4 章正则表达式

4.3 正则表达式与 FA 等价

4.4 正则语言等价模型的总结

主题：正则表达式与 FA 等价及正则语言等级模型

教学目的：使学生掌握 RE 向 ε-NFA 的转换，以及 DFA 向 RE 的转换，理解正则语言等价模型。

重点：RE 向 ε-NFA 的转换，DFA 向 RE 的转换，正则语言等价模型

难点：RE 向 ε-NFA 的转换，DFA 向 RE 的转换

教学过程：

（一）复习旧课

对上次课所学重点内容正则表达式的形式定义进行复习（约 3 分钟）

（二）讲授新课

4.3 正则表达式与 FA 等价（1 学时）

定义 4-4 正则表达式 r 称为与 FA M 等价，如果 $L(r) = L(M)$。

定理 4-1

正则表达式表示的语言是正则语言。

定理的证明过程就是根据正则表达式构造相应与之等价的 FA 的过程。

例 4-3

构造与正则表达式$(0+1)^*0+(00)^*$等价的 FA。

步骤 1, 2

构造不含运算符的基本 ε-NFA。包括：与 0 等价的 M_1，与 1 等价的 M_2。

步骤 3

构造与 0+1 等价的 M_3 (基本方法见图 4-7) 。

步骤 4

构造与$(0+1)^*$等价的 M_4 (基本方法见图 4-9)。

步骤 5

构造与$(0+1)^*0$ 等价的 M_5 (基本方法见图 4-8)。

步骤 6

构造与 00 等价的 M_6 (基本方法见图 4-8) 。

步骤 7

构造$(00)^*$等价的 M_7 (基本方法见图 4-9) 。

步骤 8

构造与$(0+1)^*0+ (00)^*$等价的 M_5 (基本方法见图 4-7)。

定理 4-2

正则语言可以用正则表达式表示。

定理的证明过程就是根据正则表达式构造相应与之等价的 FA 的过程。

例 4-4

求与下图所示的 DFA 等价的正则表达式。

预处理

增加两个状态 X，Y及相应空移动边(不存在不可到达状态)

去掉状态 q_3

使用“去状态 2”，$q = q_2$，$p= q_3$，$t=Y| q_4$

去掉状态 q_4

使用“去状态 1” ，$q = q_2$，$p= q_4$，$t=Y| q_1 | q_0$

并弧

去掉状态 q_0

使用“去状态 2”，$q = q_2 | q_1 | X$，$p= q_0$，$t=q_1$

并弧

去掉状态 q_1

使用“去状态 2”，$q = q_2 | X$，$p= q_0$，$t=q_2$

去掉状态 q_2

使用“去状态 2”

说明

如果不存在从状态 X 经过某一状态 q 到状态 Y 的路，则状态 q 应被删除。

补充练习：

接受语言 $\{x00 \mid x \in (0, 1)^*\}$ 的 DFA，求其正则表达式

讨论

直接根据其 DFA 可以知道它的正则表达式吗？

可以，为(0+1)*00

4.4 正则语言等价模型的总结 (1 学时)

正则语言的 5 种等价描述模型

1. 正则文法(RG)
2. 确定的有穷状态自动机(DFA)
3. 不确定的有穷状态自动机(NFA)
4. 带空移动的不确定的有穷状态自动机(ε-NFA)

正则表达式(RE)

转换关系

（三） 小结

字母表 Σ 上的正则表达式用来表示 Σ 上的正则语言。

正则表达式的运算。

正则表达式是正则语言的一种描述。

正则语言的 5 种等价描述模型及其转换关系。

（四） 布置作业

5. (1)，6(1)

第 29-30 学时

章节：

第 5 章正则语言的性质

5.1 正则语言的泵引理

主题：正则语言的泵引理

教学目的：使学生了解泵引理只可用于证明一个语言不是RL,但要证明一个语言是 RL 需要构造 FA 或 RE 或 RG；灵活运用泵引理，对哪些语言是正则语言有一个基本、快速的判断。掌握利用泵引理证明一个语言不是 RG。

重点：利用泵引理证明一个语言不是 RG

难点：利用泵引理证明一个语言不是 RG

教学过程：

（一）复习旧课

对第四章给出一个总复习。（约 10 分钟）

并进行第三章习题的评讲。学生评讲约 25 分钟。

老师进行第三章习题的评讲总结，约（10 分钟）

（二）讲授新课

研究动机

$\{0^n1^n \mid n\geq 1\}$不存在 RG，FA，RE 等形式的描述。

RL 有什么性质

对给定的 RL，找到最小的 DFA

5.1 正则语言的泵引理（1 学时）

一个 DFA 只有有穷个状态，当该 DFA 识别的语言 L 是无穷语言时，L 中必定存在一个足够长(大于 DFA 的可达状态数)的句子，使得 DFA 在识别该句子的过程中，肯定要重复地经过某一状态。

如句子 $z=a_1a_2 \ldots a_m$,假设 DFA M 在识别它的过程中经过的状态依次为 $q_0q_1 \ldots q_m$,则当 m 大于 M 的可达状态数时，这些状态至少有一对是重复的，不妨令为 q_k 和 q_j，显然 $k\neq j$。

泵引理推导示意图

泵引理证明

设 L 是一个 RL，DFA $M=(Q, \Sigma, \delta, q_0, F)$满足 $L(M)=L$，$|Q|=N$，并假设 M 中不含任何不可达状态。取 L 的句子

$z=a_1a_2\ldots a_m\ (m\geq N)$

对$\forall h\in[1..m]$，令

$\delta(q_0, a_1a_2\ldots a_h)=q_h$

由于 $m\geq N$，所以$\exists k, j\in[0..N], k<j$ 且 $q_k=q_j$

泵引理证明(续)

此时有

$\delta(q_0, a_1a_2\ldots a_k)=q_k$

$\delta(q_k, a_{k+1}\ldots a_j)=q_j=q_k$

$\delta(q_j, a_{j+1}\ldots a_m)=q_m$

所以对于$\forall$非负整数 i

$\delta(q_k, (a_{k+1}\ldots a_j)^i)=\delta(q_k, (a_{k+1}\ldots a_j)^{i-1})$

$=\ldots=\delta(q_k, a_{k+1}\ldots a_j)=q_k$

泵引理证明(续)

因为 $z\in L(M)$，所以 $q_m\in F$

而 $\delta(q_0, a_1a_2\ldots a_k(a_{k+1}\ldots a_j)^i a_{j+1}\ldots a_m)=q_m\in F$

所以 $a_1a_2\ldots a_k(a_{k+1}\ldots a_j)^i a_{j+1}\ldots a_m\in L(M)$

取 $u=a_1a_2\ldots a_k$

$v=a_{k+1}\ldots a_j$

$w=a_{j+1}\ldots a_m$

有：对$\forall$非负整数 i, $uv^iw\in L$

注意到 $j\leq N$, $k<j$, 有：$|uv|\leq N$, $|v|\geq 1$

引理 5-1

设 L 为一个 RL，则存在仅依赖于 L 的正整数 N，对于 $\forall z \in L$，如果 $|z| \geq N$，则 $\exists u$，v，w，st.

(1) $z = uvw$

(2) $|uv| \leq N$

(3) $|v| \geq 1$

(4) 对于 $\forall$ 整数 $i \geq 0$，$uv^i w \in L$

(5) N 不大于接受 L 的最小 DFA M 的状态数

例 5-1

证明 $\{0^n1^n | n \geq 1\}$ 不是 RL。

证明：令 $L = \{0^n1^n | n \geq 1\}$。假设 L 是 RL，则它满足泵引理。不妨设 N 是泵引理中仅依赖于 L 的正整数，取

$$z = 0^N1^N，显然\ z \in L$$

此时必然 $\exists u, v, w$ st. $z = uvw$, $|uv| \leq N$, $|v| \geq 1$，因此 v 只可能是由 0 组成的非空串

设 $v = 0^k$，$k \geq 1$; $w = 0^j1^N$

则 $u = 0^{N-k-j}$

从而有

$$uv^i w = 0^{N-k-j}(0^k)^i 0^j 1^N$$

当 $i = 2$ 时，$uv^2 w = 0^{N-k-j}0^{2k}0^j1^N = 0^{N+k}1^N$

由于 $k \geq 1$，$N+k > N$

也就是说，$uv^2 w \notin L$，这与泵引理矛盾。

所以，L 不是 RL

例 5-2

证明 $\{0^n | n$ 为素数$\}$ 不是 RL。

简证：假设它为 RL，且有一个长度为 $N+p$(取这样的长度是为了保证其 $\geq N$)的串，则可以有长度为 k 的 0 串可以被随意地泵进，$N+p+ik$ 均保持为一个素数，显然，当

$$i = N+p \quad 时$$

$$N+p+ik = (N+p)(k+1)$$

为一个合数。矛盾。因此它不是 RL。

例 5-3

证明$\{0^n1^m2^{n+m} \mid m, n\geq 1\}$不是 RL。

证明：令 $L=\{0^n1^m2^{n+m} \mid m, n \geq 1\}$ 。假设 L 是 RL，则它满足泵引理。不妨设 N 是泵引理中仅依赖于 L 的正整数，取

$z=0^N1^N2^{2N}$，显然 $z\in L$

此时必然$\exists u, v, w$ st. $z=uvw$; $|uv|\leq N$, $|v|\geq 1$，因此 v 只可能是由 0 组成的非空串

设 $v=0^k$， $k\geq 1$; $w=0^j1^N2^{2N}$

则 $u=0^{N-k-j}$

从而有

$$uv^iw=0^{N-k-j}(0^k)^i0^j1^N2^{2N}$$

当 $i=0$ 时， $uv^0w=0^{N-k}1^N2^{2N}$

由于 $k\geq 1$ ，$N-k+N<2N$

也就是说， $uv^0w\notin L$，这与泵引理矛盾。

所以， L 不是 RL。

例 5-3(续)

注意：

本例与例 5-1 非常相似。实际上，证明一个语言不是 RL 的题都需要相同的步骤。虽然取 $z=0^N1^N2^{2N}$，但当 0 串被泵进/泵出时，应说明它不满足 $0^n1^m2^{n+m}$ ，而不是 $0^n1^n2^{2n}$ 。

（三）小结

泵引理

泵引理的应用

（四）布置作业

1. (1)(3)(8)

第 31-32 学时

章节：

第 5 章正则语言的性质

5.2 正则语言的封闭性

主题：正则语言的封闭性

教学目的：使学生理解正则语言的封闭性。

重点：正则语言的封闭性

难点：正则语言的封闭性

教学过程：

（一）复习旧课

对上次课所学重点内容正则语言的泵引理进行复习（约 5 分钟）

（二）讲授新课

5.2 正则语言的封闭性（1 学时）

定义 5-1 如果任意的、属于某一语言类的语言在某一特定运算下所得的结果仍然是该类语言，则称该语言类对此运算是封闭的，并称该语言类对此运算具有封闭性(closure property)。

我们所熟悉的封闭性：整数对于加法是封闭的，有理数对于乘法/除法是封闭的

有效封闭性

给定一个语言类的若干个语言的描述。如果存在一个算法，它可以构造出这些语言在给定运算下所获得的运算结果的相应形式的语言描述，则称此语言类对相应的运算是有效封闭的，并称此语言类对相应的运算具有有效封闭性(valid closure property)。

定理 5-1

RL 在并、乘积、闭包运算下是封闭的。

使用正则文法，见 P_{85} 习题 2-11

使用 FA

1. 两个 RL 的并：合并两个 NFA，使其仅具有相同的开始状态；
2. 两个 RL 的乘积：以第一个 NFA 的终止状态为第二个的开始状态；
3. 闭包：在 FA 开始和终止状态间添加空移动边。

定理 5-2

RL 在补运算下是封闭的。

补：两个 DFA 具有互补的终止状态。

如果 DFA $M = (Q, \Sigma, \delta, q_0, F)$识别的语言为 L，那么 DFA $M' = (Q, \Sigma, \delta, q_0, Q - F)$ 识别的语言应为 $\Sigma^* - L$。

定理 5-3

RL 在交运算下是封闭的。

交：同时满足两个 RL。

定义 5-2

设 Σ, Δ 是两个字母表，映射 $f: \Sigma \to 2^{\Delta^*}$

称为是从 Σ 到 Δ 的代换(substitution)。如果对于$\forall\ a \in \Sigma$，$f(a)$是上 Δ 的 RL，则称 f 为正则代换(regular substitution)。

复习： Δ^*是指字母表 Δ 上的克林闭包，即 Δ 中若干个字符连接而成字符串的集合(P_{30})；而 2^{Δ^*}是指 Δ^*的幂集(P_{10})

扩展

先将的 f 定义域扩展到 Σ^*上：$f: \Sigma^* \to 2^{\Delta^*}$

(1) $f(\varepsilon) = \{\varepsilon\}$ (允许空字符)

(2) $f(xa) = f(x) f(a)$ (允许字符串)

再将 f 的定义域扩展到 2^{Σ^*}上： $f: 2^{\Sigma^*} \to 2^{\Delta^*}$

例 5-4

设 $\Sigma = \{0, 1\}$, $\Delta = \{a, b\}$, $f(0) = a$, $f(1) = b^*$，(注意没有严格区分 a 与$\{a\}$, b^*本身是一个正则表达式，它表示了一个集合)则

1. $f(010) = f(0)\,f(1)\,f(0) = ab^*a$
2. $f(\{11, 00\}) = f(11) \cup f(00) = f(1)f(1) \cup f(0)f(0) = b^*b^* + aa = b^* + aa$
3. $f(L(0^*(0+1)1^*) = L(a^*\ (a+b^*)\ b^*)$
   $= L(a^*ab^* + a^*b^*\ b^*) = L(a^*b^*)$

定义 5-3

设 Σ, Δ 是两个字母表，映射 $f: \Sigma \to 2^{\Delta^*}$ 为正则代换，则

(1) $f(\emptyset) = \emptyset$

(2) $f(\varepsilon) = \varepsilon$

(3) 对于 $\forall a \in \Sigma$， $f(a)$ 是 Δ 上的正则表达式

(4) 如果 r, s 是 Σ 上的正则表达式，则

$f(r+s) = f(r) + f(s)$ $\quad f(rs) = f(s)f(s)$ $\quad f(r^*) = f(r)^*$

是 Δ 上的正则表达式

定理 5-4

设 L 是 Σ 上的一个 RL, $f: \Sigma \to 2^{\Delta^*}$ 为正则代换，则 $f(L)$ 也是 RL。

证明过程：根据定义 5-2，定义 5-3 及定义 4-1，利用数学归纳法。

例 5-5

设 $\Sigma = \{0, 1, 2\}$, $\Delta = \{a, b\}$, $f(0) = ab$, $f(1) = b^*a^*$, $f(2) = a^*(a+b)$ ，则

1. $f(00) = abab$
2. $f(010) = abb^*a^*ab = ab^+a^+b$
3. $f((0+1+2)^*) = (ab + b^*a^* + a^*(a+b))^* = (ab + b^*a^* + a^+ + a^*b)^* \quad = (a+b)^*$
4. $f(0(0+1+2)^*) = ab\,(a+b)^*$
5. $f(012) = abb^*a^*a^*(a+b) = ab^+a^*(a+b)$
6. $f((0+1)^*) = (ab + b^*a^*)^* = (a+b)^*$
7. $f(1^*) = (b^*a^*)^* = (a+b)^*$

定义 5-4

设 Σ, Δ 是两个字母表，映射 $f: \Sigma \to \Delta^*$ 为映射。如果对于 $\forall x, y \in \Sigma^*$，

$f(xy)=f(x)f(y)$

则称 f 为从 Σ 到 Δ^*的同态映射(homomorhism)。

定义 5-4(续)

对于$\forall w\in\Delta^*$， w 的同态原像是一个集合 $f^{-1}(w)=\{x| f(x) = w$ 且 $x\in\Sigma^*\}$

对于$\forall L\subseteq\Delta^*$， L 的同态原像是一个集合 $f^{-1}(L)=\{x| f(x) \in L\}$

例 5-6

设 $\Sigma=\{0, 1\}$, $\Delta=\{a, b\}$, $f(0) = aa$, $f(1) = aba$,则

1. $f(01) = aaaba$
2. $f((01)^*) = (aaaba)^*$
3. $f^{-1}(aab) = \emptyset$
4. $f^{-1}(\{aaa, aba, abaaaaa, abaaaaaa\}) = \{0, 100\}$
5. $f^{-1}((ab+ ba)^*a) = \{1\}$
6. $f(f^{-1}((ab+ ba)^*a)) = \{aba\}$

注意

同态映射是正则代换的特例

原因：比较其值域可知

1. 正则代换的值域为 2^{Δ^*}
2. 同态映射的值域为 Δ^*

推论 5-1

RL 的同态像是 RL。

定理 5-5

RL 的同态原像是 RL。

定义 5-5

设 $L_1, L_2\subseteq\Sigma^*$， L_1 除以 L_2 的商（quotient）定义为

$L_2 / L_1 = \{x \mid \exists\, y \in L_2 \text{ st. } xy \in L_1\}$

对商的计算：考虑句子的后缀

例 5-7

考虑 {0, 1} 上的如下语言

1. 设 $L_1 = (0 + 1)^*, L_2 = 0(0 + 1)^*$, 则 $L_1 / L_2 = L_1$
2. 设 $L_1 = 01, L_2 = 01$, 则 $L_1 / L_2 = \{\varepsilon\}$
3. 设 $L_1 = (01)^*, L_2 = (01)^*$,则 $L_1 / L_2 = L_1$
4. 设 $L_1 = 0^*011^*, L_2 = 00$, 则 $L_1 / L_2 = \emptyset$
5. 设 $L_1 = 00^*1^*1, L_2 = 00^*1^*1$, 则 $L_1 / L_2 = 0^*$
6. 设 $L_1 = 00^*1^*1, L_2 = 0^*1^*1$, 则 $L_1 / L_2 = 0^* + 00^*1^*$

定理 5-6

设 $L_1, L_2 \subseteq \Sigma^*$，如果 L_1 是 RL，则 L_1 / L_2 也是 RL。

注意 L_2 可以是任意语言，所以这种封闭性不是有效封闭性。

（三）小结

正则语言的封闭性

（四）布置作业

1. (1)(2)(3)(8)

对第五章给出一个总复习。（约 5 分钟）

进行第四、五章习题的评讲。(约 20 分钟)

课程总复习 （约 20 分钟）