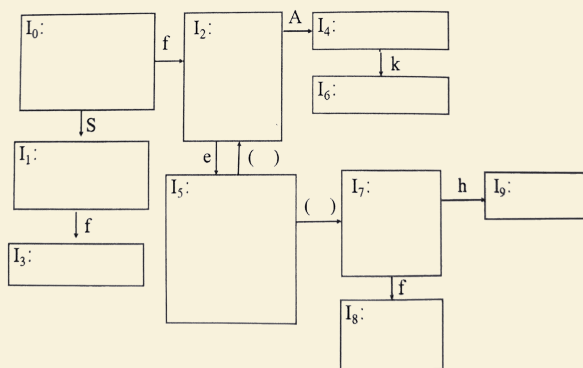


已知文法 $G[S]$ ：

$$S \rightarrow Sf \mid fAk$$

$$A \rightarrow eSf \mid eSh \mid \varepsilon$$

- 写出拓广文法，并将下图识别活前缀的 DFA 补充完整。
- 求出 S, A 的 FIRST 集和 FOLLOW 集。
- 该文法是 LR(0) 文法吗？是 SLR(1) 文法吗？说明理由（指出 (1) 中的 LR(0) 项目集规范族是否有冲突？如果有冲突，哪个项目集存在何种冲突？）。判断后构造出相应的 SLR(1) 分析表。



第 (1) 问 几个概念：

拓广文法 对原有的以 S 为开始符的文法 $G[S]$ ，增加产生式 $S' \rightarrow S\$$ （教材和资料中多采用 $S' \rightarrow S$ 的产生式），得到的新文法 $G'[S']$ 为「拓广文法」。易知拓广文法与原文法等价。

DFA 指确定性的有穷自动机。

LR(0) 项目 在每个产生式的右部添加圆点以构成项目。比如产生式

$S \rightarrow ab$ 对应三个项目： $S \rightarrow \bullet ab$ 、 $S \rightarrow a \bullet b$ 、 $S \rightarrow ab \bullet$ 。特别地，空产生式 $S \rightarrow \varepsilon$ 只对应一个项目： $S \rightarrow \bullet$ 。

原文法的拓广文法 $G'[S']$ 为：

$$S' \rightarrow S\$$$

$$S \rightarrow Sf \mid fAk$$

$$A \rightarrow eSf \mid eSh \mid \varepsilon$$

在补充 DFA 时，若遇到了形如 $S \rightarrow \bullet A \dots$ 的产生式（圆点紧接着非终结符），则需要把所有以该非终结符 A 开头的产生式也加入到 DFA 的同一状态中。得到的 DFA 图如下：

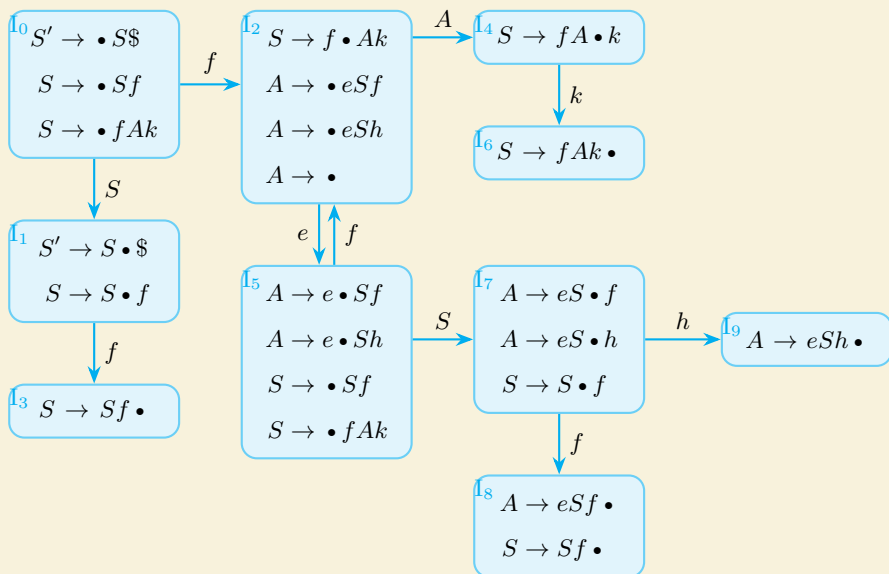


图 1: DFA 状态图

需要记得将括号里的转移字符补全。

第 (2) 问 求 FIRST 集和 FOLLOW 集，参考「知识小料」·其三十七，其中有详细步骤。这里对拓广文法 G' 进行处理，为了在 FOLLOW 集得到 $\$$ ，以备后续操作。比较简单。

易知， $\text{NULLABLE} = \{A\}$ 。

- $\text{FIRST}(S) = \{f\}$;
- $\text{FIRST}(A) = \{e\}$ 。
- $\text{FOLLOW}(S) = \{f, h, \$\}$;
- $\text{FOLLOW}(A) = \{k\}$ 。

第 (3) 问 询问是否为 LR(0)、SLR(1) 文法，需要先明确概念。

移进—归约冲突 对于某一状态，例如题中的 I_2 ，若其既可对应移进操作，也可对应归约操作，则存在冲突。

归约—归约冲突 对于某一状态，例如题中的 I_8 ，若存在多个归约项目，且对应的产生式不同，则存在冲突。

LR(0) 文法 当 DFA 状态图中不存在移进—归约冲突和归约—归约冲突，则文法是 LR(0) 文法。

SLR(1) 文法 当 DFA 状态图中不存在上述两种冲突，或者虽存在冲突但可以利用 FOLLOW 集来消解，则文法是 SLR(1) 文法。此时需要考虑前看符号（即 FOLLOW 集的符号）。

两种文法的关系 一个文法若为 LR(0) 文法，则它一定是 SLR(1) 文法的候选者，但反之不然。

项目集规范族 指同一状态的 LR(0) 项目集合，如 I_0 的三条项目的集合。

查看 DFA 状态图，发现状态 I_2 的 4 个项目中，前 3 个项目 $S \rightarrow f \cdot Ak$ 、 $A \rightarrow \cdot eSf$ 、 $A \rightarrow \cdot eSh$ 对应移进操作，而后 1 个项目 $A \rightarrow \cdot$ 对应归约操作，因此在这一状态下不能确定是移进 e 还是归约为 A 。据此得知，项目集 I_2 存在**移进—归约冲突**。

查看 DFA 状态图，还会发现状态 I_8 的 2 个项目中，项目 $A \rightarrow eSf \cdot$ 、 $S \rightarrow Sf \cdot$ 分别对应于产生式 $A \rightarrow eSf$ 、 $S \rightarrow Sf$ 的归约操作，因此在这一状态下不能确定是归约为 A 还是归约为 S 。据此得知，项目集 I_8 存在**归约—归约冲突**。

因此，原文法不是 LR(0) 文法。但这种情况可以用 FOLLOW 集处理，可以试着使用 SLR(1) 分析方法填表，再判断是否是 SLR(1) 文法。

在 SLR(1) 文法中，只有归约操作与 LR(0) 文法不同：对于状态 I_i 的项目 $X \rightarrow \cdots \cdot$ ，仅对 $y \in \text{FOLLOW}(X)$ 添加 $\text{ACTION}[i, y]$ 为需要归约的产生式编

号。(也就是说,只有当面临 X 的前看符号时,才能做出归约操作,否则报错。)

为了便于归约,这里把拓广文法 $G'[S']$ 产生式展开写,并编号:

$$0 : S' \rightarrow S\$$$

$$1 : S \rightarrow Sf$$

$$2 : S \rightarrow fAk$$

$$3 : A \rightarrow eSf$$

$$4 : A \rightarrow eSh$$

$$5 : A \rightarrow \varepsilon$$

状态	ACTION 表					GOTO 表	
	e	f	h	k	$\$$	S	A
I_0		s_2				g_1	
I_1		s_3			Acc.		
I_2	s_5			r_5			g_4
I_3		r_1	r_1		r_1		
I_4				s_6			
I_5		s_2				g_7	
I_6		r_2	r_2		r_2		
I_7		s_8	s_9				
I_8		r_1	r_1	r_3	r_1		
I_9				r_4			

表 1: SLR(1) 分析表

详细说明分析表的几处:

- 移进项 s_i, g_i 的下标 i 表示状态号 I_i , 归约项 r_j 的下标 j 表示产生式号。
- 举例说明: 对于 r_5 这一格, 因为状态 I_2 的归约项目 $A \rightarrow \bullet$ 的产生式左部为 A , $\text{FOLLOW}(A) = \{k\}$, 所以在第 I_2 行仅对 k 列添加 $\text{ACTION}[2, k]$, 对应的产生式是 $[5 : A \rightarrow \varepsilon]$, 所以填写 r_5 。(如果按照 LR(0) 分析法, 那么 ACTION 表部分的每一列都要填写 r_5 。)

- 举例说明：状态 I_8 这一行，存在归约—归约冲突。

- i) 对项目 $A \rightarrow eSf \bullet$ ， $\text{FOLLOW}(A) = \{k\}$ ，仅对 k 列添加 $\text{ACTION}[8, k] = r_3$ （用 3 号产生式归约）；
- ii) 对项目 $S \rightarrow Sf \bullet$ ， $\text{FOLLOW}(S) = \{f, h, \$\}$ ，对 $f, h, \$$ 列均需要添加 $\text{ACTION}[8, f] = \text{ACTION}[8, h] = \text{ACTION}[8, \$] = r_1$ （用 1 号产生式归约）。

（事实上，本例满足 $\text{FOLLOW}(S) \cap \text{FOLLOW}(A) = \emptyset$ 。如果不满足这一条件，那么需要在同一个单元格内填写多个归约，此时冲突无法解决，就不是 SLR(1) 文法了。）

- Acc. 是「接受」(Accept) 的意思。
- 在分析 FOLLOW 集时需要用拓广文法 G' ，这是为了使原开始符 S 的 FOLLOW 集包含终结符 $\$$ ，以将 ACTION 表的 $\$$ 列恰当补全。

填写分析表后，发现分析表内无冲突（ $\text{FOLLOW}(S) \cap \text{FOLLOW}(A) = \emptyset$ 这一条件立大功）。所以原文法是 SLR(1) 文法。

【结论】

- (1) 拓广文法见解析，DFA 状态图如图 1 所示。
- (2) S, A 的 FIRST 集分别为 $\{f\}$ 、 $\{e\}$ ；FOLLOW 集分别为 $\{f, h, \$\}$ 、 $\{k\}$ 。
- (3) 该文法不是 LR(0) 文法，是 SLR(1) 文法，题 (1) 中的项目集规范族 I_2 存在移进—归约冲突， I_8 存在归约—归约冲突。SLR(1) 分析表如表 1 所示。

【点评】这是一道编译原理的大题，考察了 LR(0) 和 SLR(1) 文法的分析方法。此题需要同学们会补充 DFA 状态图，会填写 ACTION 表和 GOTO 表，以及进行一些分析。