有穷自动机的最小化

Yajun Yang

yjyang@tju.edu.cn

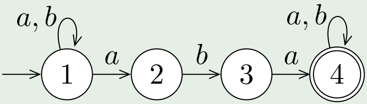
School of Computer Science and Technology
Tianjin University

2015

有穷自动机的等价性

问题：对同一个正则语言L,是否存在多个不同的DFA接受它?

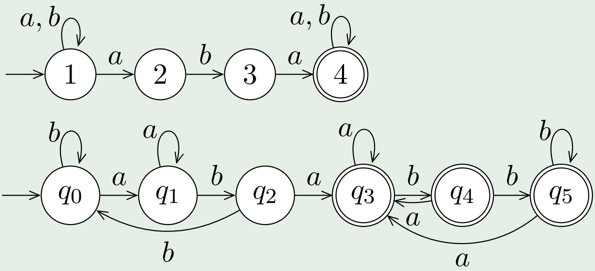
Example (Converting an NFA to DFA)



有穷自动机的等价性

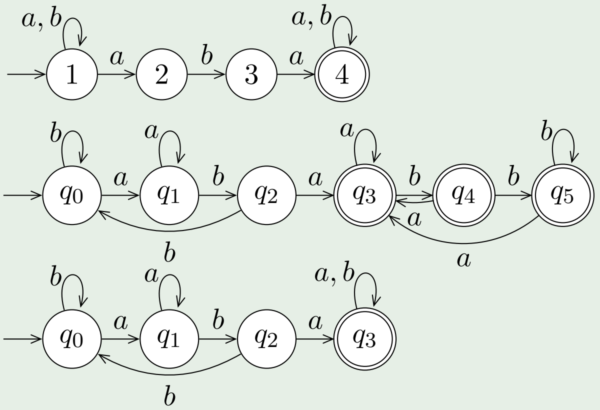
问题：对同一个正则语言L,是否存在多个不同的DFA接受它?

Example (Converting an NFA to DFA)



问题：对同一个正则语言L,是否存在多个不同的DFA接受它?

Example (Converting an NFA to DFA)



有穷自动机的等价性

Definition (状态的等价)

对于给定的DFA $M = (Q, \Sigma, \delta, q_0, F)$，定义$Q$上的等价关系如下：
对于$p, q \in Q$,若对于$\forall w \in \Sigma^*$，$\widehat{\delta}(p, w) \in F$当且仅当$\widehat{\delta}(q, w) \in F$，则称$p$和$q$等价，或者$p$和$q$是不可区分的。

有穷自动机的等价性

Definition (状态的等价)

对于给定的DFA $M = (Q, \Sigma, \delta, q_0, F)$，定义$Q$上的等价关系如下：
对于$p, q \in Q$,若对于$\forall w \in \Sigma^*$，$\widehat{\delta}(p, w) \in F$当且仅当$\widehat{\delta}(q, w) \in F$，则称$p$和$q$等价，或者$p$和$q$是不可区分的。

不要求$\widehat{\delta}(p, w)$和$\widehat{\delta}(q, w)$是相同的状态，只要求要么是都接受，要么是都不接受即可。

Definition (状态的等价)

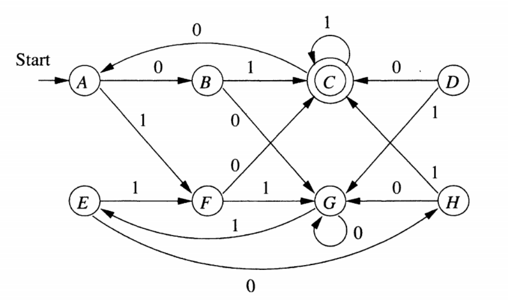
对于给定的DFA $M=(Q,\Sigma,\delta,q_0,F)$，定义$Q$上的等价关系如下：
对于$p,q\in Q$,若对于$\forall w\in\Sigma^*$，$\widehat{\delta}(p,w)\in F$当且仅当$\widehat{\delta}(q,w)\in F$，则称$p$和$q$等价，或者$p$和$q$是不可区分的。

不要求$\widehat{\delta}(p,w)$和$\widehat{\delta}(q,w)$是相同的状态，只要求要么是都接受，要么是都不接受即可。

该关系是自反的、对称的、传递的。

状态的等价

Example (An example)



- $\{C, G\}$是否可区分?
- $\{A, G\}$是否可区分? 串ϵ? 串0? 串1? 串01?
- $\{A, E\}$是否可区分? 串ϵ? 串1? 串0?

计算等价状态：填表算法

Basis: 如果$p \in F$且$q \notin F$，则$\{p, q\}$是可区分的状态对；

计算等价状态：填表算法

Basis: 如果$p \in F$且$q \notin F$，则$\{p, q\}$是可区分的状态对；

Indution: 设p, q是满足下列条件的状态：对于某个输入符号a，$r = \delta(p, a)$和$s = \delta(q, a)$是可区分的状态，则$\{p, q\}$为可区分的状态对。

计算等价状态：填表算法

Basis: 如果$p \in F$且$q \notin F$，则$\{p, q\}$是可区分的状态对；

Indution: 设p, q是满足下列条件的状态：对于某个输入符号a，$r = \delta(p, a)$和$s = \delta(q, a)$是可区分的状态，则$\{p, q\}$为可区分的状态对。

为所有状态对画一张表，开始时表中每个格子均为空白（不作任何标记）；

计算等价状态：填表算法

Basis: 如果$p \in F$且$q \notin F$，则$\{p, q\}$是可区分的状态对；

Indution: 设p, q是满足下列条件的状态：对于某个输入符号a，$r = \delta(p, a)$和$s = \delta(q, a)$是可区分的状态，则$\{p, q\}$为可区分的状态对。

为所有状态对画一张表，开始时表中每个格子均为空白（不作任何标记）；

1. 对$p \in F$，$q \notin F$的一切状态对，在相应的格子中做标记$\times$；

计算等价状态：填表算法

Basis：如果$p \in F$且$q \notin F$，则$\{p, q\}$是可区分的状态对；

Indution：设p, q是满足下列条件的状态：对于某个输入符号a，$r = \delta(p, a)$和$s = \delta(q, a)$是可区分的状态，则$\{p, q\}$为可区分的状态对。

为所有状态对画一张表，开始时表中每个格子均为空白（不作任何标记）；

1. 对$p \in F$，$q \notin F$的一切状态对，在相应的格子中做标记$\times$；
2. 重复以下过程，直到表中内容不再改变：若存在一个未标记状态$\{p, q\}$，且对于某个$a \in \Sigma$，$\{\delta(p, a), \delta(q, a)\}$已做标记，则在$\{p, q\}$对应格子内做标记；

计算等价状态：填表算法

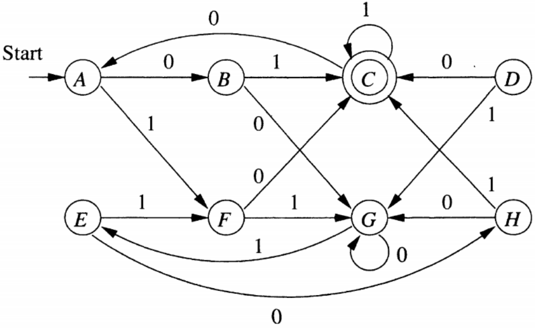
Basis：如果$p \in F$且$q \notin F$，则$\{p, q\}$是可区分的状态对；

Indution：设p, q是满足下列条件的状态：对于某个输入符号a，$r = \delta(p, a)$和$s = \delta(q, a)$是可区分的状态，则$\{p, q\}$为可区分的状态对。

为所有状态对画一张表，开始时表中每个格子均为空白（不作任何标记）；

1. 对$p \in F$，$q \notin F$的一切状态对，在相应的格子中做标记$\times$；
2. 重复以下过程，直到表中内容不再改变：若存在一个未标记状态$\{p, q\}$，且对于某个$a \in \Sigma$，$\{\delta(p, a), \delta(q, a)\}$已做标记，则在$\{p, q\}$对应格子内做标记；
3. 完成（1）和（2）之后，所有未标记的$\{p, q\}$都是等价的。

An Example



	A	B	C	D	E	F	G
B	x						
C	x	x					
D	x	x	x				
E		x	x	x			
F	x	x	x		x		
G	x	x	x	x	x	x	
H	x		x	x	x	x	x

正确性证明

定理6.3.1

在填表算法中，$\{p, q\}$被标记，当且仅当p, q是可区分的。

正确性证明

定理6.3.1

在填表算法中，$\{p, q\}$被标记，当且仅当p, q是可区分的。

This theorem has two directions.

用归纳法证明。

DFA的最小化算法

将所有状态按照等价关系分块，需证明等到的块集合是状态集的划分。

DFA的最小化算法

将所有状态按照等价关系分块，需证明等到的块集合是状态集的划分。

定理6.3.2

对于DFA中的状态集Q，按照等价关系分块，即每个状态p和与其等价的所有状态组成一个块，则不同的状态块形成集合的划分。

将所有状态按照等价关系分块，需证明等到的块集合是状态集的划分。

定理6.3.2

对于DFA中的状态集Q，按照等价关系分块，即每个状态p和与其等价的所有状态组成一个块，则不同的状态块形成集合的划分。

同一块中的所有成员均等价，不同块中选择的状态一定不等价。

DFA的最小化算法

给定一台DFA $M = (Q, \Sigma, \delta, q_0, F)$;

DFA的最小化算法

给定一台DFA $M = (Q, \Sigma, \delta, q_0, F)$;

1. 用填表算法找到所有的等价状态对;

DFA的最小化算法

给定一台DFA $M = (Q, \Sigma, \delta, q_0, F)$；

1. 用填表算法找到所有的等价状态对;
2. 根据等价状态对，将状态集合Q划分成状态块；

DFA的最小化算法

给定一台DFA $M = (Q, \Sigma, \delta, q_0, F)$；

1. 用填表算法找到所有的等价状态对;
2. 根据等价状态对，将状态集合Q划分成状态块；
3. 将状态块作为状态，构造最小化DFA $M' = (Q', \Sigma, \delta', q_0', F')$，其中：

DFA的最小化算法

给定一台DFA $M = (Q, \Sigma, \delta, q_0, F)$；

1. 用填表算法找到所有的等价状态对;
2. 根据等价状态对，将状态集合Q划分成状态块；
3. 将状态块作为状态，构造最小化DFA $M' = (Q', \Sigma, \delta', q_0', F')$，其中：

   - $Q' = \{[p] | p \in Q\}$;

DFA的最小化算法

给定一台DFA $M = (Q, \Sigma, \delta, q_0, F)$；

1. 用填表算法找到所有的等价状态对;
2. 根据等价状态对，将状态集合Q划分成状态块；
3. 将状态块作为状态，构造最小化DFA $M' = (Q', \Sigma, \delta', q_0', F')$，其中：

    - $Q' = \{[p] | p \in Q\}$;
    - $\delta'([p], a) = [\delta(p, a)]$，$[p] \in Q'$，$a \in \Sigma$;

DFA的最小化算法

给定一台DFA $M = (Q, \Sigma, \delta, q_0, F)$；

1. 用填表算法找到所有的等价状态对;
2. 根据等价状态对，将状态集合Q划分成状态块；
3. 将状态块作为状态，构造最小化DFA $M' = (Q', \Sigma, \delta', q_0', F')$，其中：
    - $Q' = \{[p] | p \in Q\}$;
    - $\delta'([p], a) = [\delta(p, a)]$，$[p] \in Q'$，$a \in \Sigma$;
    - $q_0' = [q_0]$;

DFA的最小化算法

给定一台DFA $M = (Q, \Sigma, \delta, q_0, F)$；

1. 用填表算法找到所有的等价状态对;
2. 根据等价状态对，将状态集合Q划分成状态块；
3. 将状态块作为状态，构造最小化DFA $M' = (Q', \Sigma, \delta', q_0', F')$，其中：

    - $Q' = \{[p] | p \in Q\}$;
    - $\delta'([p], a) = [\delta(p, a)]$，$[p] \in Q'$，$a \in \Sigma$;
    - $q_0' = [q_0]$;
    - $F' = \{[p] | p \in F\}$.

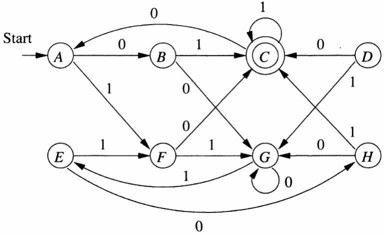
DFA的最小化算法

给定一台DFA $M = (Q, \Sigma, \delta, q_0, F)$；

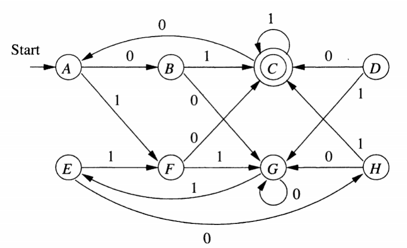
1. 用填表算法找到所有的等价状态对;
2. 根据等价状态对，将状态集合Q划分成状态块；
3. 将状态块作为状态，构造最小化DFA $M' = (Q', \Sigma, \delta', q_0', F')$，其中：
    - $Q' = \{[p] | p \in Q\}$;
    - $\delta'([p], a) = [\delta(p, a)]$，$[p] \in Q'$，$a \in \Sigma$;
    - $q_0' = [q_0]$;
    - $F' = \{[p] | p \in F\}$.

通过最小化算法得到的DFA M' 也被成为DFA M 的商自动机

An Example

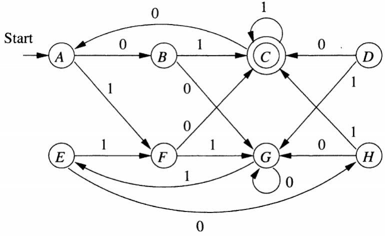


An Example

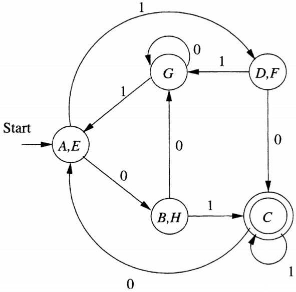


	A	B	C	D	E	F	G
B	x						
C	x	x					
D	x	x	x				
E		x	x	x			
F	x	x	x		x		
G	x	x	x	x	x	x	
H	x		x	x	x	x	x

An Example



	A	B	C	D	E	F	G
B	x						
C	x	x					
D	x	x	x				
E		x	x	x			
F	x	x	x		x		
G	x	x	x	x	x	x	
H	x		x	x	x	x	x



正确性证明

定理6.3.3

任意的DFA M 与它的商自动机 M' 是等价的。

正确性证明

定理6.3.3

任意的DFA M 与它的商自动机 M' 是等价的。

M'是否能以同样的算法得到状态数更少的DFA?

正确性证明

定理6.3.3

任意的DFA M 与它的商自动机 M' 是等价的。

M'是否能以同样的算法得到状态数更少的DFA?

定理6.3.4

若M是任意一台DFA，M'是通过最小化算法得到的DFA，则M'不比与M等价的DFA具有更多的状态。