**计算理论课程报告**

**2022113557-宫名扬**

（库克定理）是完全的。

证明：首先，是问题：给定一个实例，可以设计一个非确定算法，它首先猜测一个满足的赋值，然后检查这个赋值是否满足中所有子句，这显然可以在非确定多项式时间内完成。

下面要证任意，。由于中的语言是无限的，所以我们不可能给的每一个语言都单独设计一个不同的到的多项式时间归约。但是，中的每一个语言都可以用一个通用的方式来描述，也就是，一个识别它的多项式时间。这样一来，我们可以设计这样一个通用的归约，给定一个通用的多项式时间，这个归约把这个所识别的语言归约到。这个通用的归约，当给定一个具体的多项式时间（识别的语言为），就可以给出一个从到的多项式时间规约。这样一来，我们就实际上给出了“任意，”的证明。

下面设计这个通用的归约，给定一个通用的多项式时间，把这个所识别的语言多项式时间归约到。

首先，令为任意一个多项式时间，它所识别的语言为。令为的多项式时间限制（不失一般性，可以假设对于任意，）。记这个通用的归约为，其中，是从上的字符串到中实例的映射，且对于任意，。

如果输入能被接受，那么就存在一个在上的接受的计算，这个计算持续不超过步，且带上的有效符号（带上最左边与最右边无限的空格无效）不超过个（上面假设了），其中。考虑这个计算对应的ID转移序列，则这个序列中最多包含个ID（加上起始ID），且每个ID的长度最多为（加上代表带头位置以及状态的一位）。这使得我们能够用有限个布尔变量和一个满足赋值来描述这个计算。

下面引入“画面”的概念，这个概念有助于描述该归约。

在上运行对应的画面是一张的表格，其中行代表在输入上的一个计算分支的ID，如图1所示。

如图1所示，为了方便，假定每一个ID都以符号#开始和结束，画面的第一列和最后一列都是#号。画面的第一行是在上的起始ID，每一行都根据的转移函数从上一行得到。如果画面的某一行是接受ID，则称该画面为接受的。

在上的每一个接受画面都对应在上的一个计算分支。所以判定是否接受的问题等价于判定是否存在在上的接受画面的问题。

现在开始描述从到的多项式时间规约。在输入上，该归约产生一个公式。

首先描述中的变量。令，对于每个介于1到的i和j以及中的每个符号，存在一个变量。

个画面格子中的每一格称为一个单元。第i行第j列的单元称为cell[i,j]，并且包含中的一个符号。用的变量表示单元中的内容，若取值1，则意味着cell[i,j]包含s。

现在设计，使得变量的一个满足赋值确实对应在上的一个接受画面。公式是四部分的AND运算：（分别对应英文单词unique、start、next、accept）。下面依次描述每一部分。

首先，为了获得在赋值与画面之间的对应，必须保证的第一件事是赋值为每个单元恰好放置一个符号。公式确保了这一要求，它用布尔运算的语言来表示这一点：



该公式在方括号中的第一部分



是说，cell[i,j]中至少包含一个符号s；第二部分



是说，cell[i,j]中不能同时包含两个不同的符号s和t。这两部分合起来便说，每个单元包含且仅包含一个符号。所以，满足（从而也满足）的任何赋值都给表中的每个单元恰好指定了一个符号。，和部分则保证该表格确实是一个接受画面，如下。

公式保证表的第一行是在上的起始ID，它的定义如下：



显然，满足（从而也满足）的任何赋值都使得表中的第一行是在上的起始ID。

公式保证接受ID出现在画面中，它确保的某一个接受状态出现在画面的最后一行对应的最终ID中：



最后，公式保证画面的每一行都对应于从上一行的ID按照的规则合法转移得到的ID。它通过确保每一个2×3的窗口单元都是合法的来保证这一点（图1中给出了窗口的直观表现形式）。如果一个2×3的窗口不违反由的转移函数指定的动作，则称该窗口是合法的。换言之，如果它可以出现在从一个ID正确地转移到另一个ID的过程中，该窗口就称为合法的。

例如，设a，b，c是带符号，和是的状态。假设在状态，带头读取a时，写一个b，保持状态，然后右移。在状态，带头读取b时，非确定地：

1. 写一个c，进入状态然后左移，或者
2. 写一个a，进入状态然后右移。

形式化地表示为，。该机器的合法窗口的例子如图2所示。

在图2中，窗口(a)、(b)是合法的，分别对应以、的方式移动。窗口(c)是合法的，因为出现在顶行的右边，不确定带头在什么符号上，若这个符号是a，则可以用规则得到该窗口。窗口(d)显然是合法的，因为顶行与底行是相同的，当带头与窗口的边缘位置相隔一定距离时就一定会出现这种情况。注意，在合法窗口中，#可以出现在顶行和底行的左边或者右边。窗口(e)是合法的，因为紧靠顶行的右边可能是，那么可以用规则得到底行结果。最后，窗口(f)是合法的，因为状态可能紧挨着顶行的左边，则可以用规则得到底行结果。

特别地，我们规定一种特殊的合法窗口，如图3所示（只画出了在中间单元的情况，位于左边单元或者右边单元同样合法），其中代表的任意一个接受状态。定义这种合法窗口的目的是为了解决该归约过程可能存在的一个问题：我们规定的画面的行数为，这的确保证了计算过程中产生的ID序列可以全部包含在该画面中，但是可能会出现这样的情况，在输入上运行，很快就接受了并得到ID序列中的最后一个接受ID，即计算过程中产生的ID序列中的ID数量小于该画面能够容纳的最大行数，那么该如何处理画面中剩余的行呢？引入这样的特殊的合法窗口正是为了解决这一问题，有了这种合法窗口，就能够把ID序列中的最后一个接受ID复制到画面中所有剩余的行中，以便填满整个画面。

图4所示的窗口对于机器是不合法的。

在窗口(a)中，顶行中间的符号不可能改变，因为没有状态与它相邻。窗口(b)不是合法的，因为转移函数指明b应变为c而不是a。窗口(c)不是合法的，因为在底行出现了两个状态。

可以断言：如果画面的顶行是起始ID，画面中的每一个窗口都是合法的，那么画面的每一行都是从上一行合法转移得到的ID。

为证明该断言，考虑画面中任意两个相邻ID，称为上ID与下ID。在上ID中，对于每一个包含带符号且不与状态符号相邻的单元，这个单元必定是某个窗口顶行的中间单元，且该窗口顶行不含状态，那么该符号必定保持不变，出现在窗口的底行中间，即出现在底行ID的同一位置；对于包含状态符号的单元以及与状态符号相邻的带符号单元，考虑这样的窗口，该窗口顶行的中间单元包含着状态符号，这就使相应的三个位置按照转移函数的要求一致更新。因此，在上ID中，包含带符号且不与状态符号相邻的单元、包含状态符号的单元、包含带符号且与状态符号相邻的单元，这三种单元都在下ID中得到正确更新，所以，如果上ID是合法ID，那么下ID也是合法ID，并且下ID是根据的规则从上ID转移得到。

现在给出公式的构造，它规定画面中的所有窗口都是合法的。每个窗口包含6个单元，它们可以以固定数目种方式设置为合法窗口。公式指出，这6个单元的设置必须是这些方式之一，即



其中-窗口代表顶行中间单元为cell[i,j]的窗口。把这个公式中的文字“-窗口是合法的”替换为下面的公式，把窗口中6个单元的内容记为：



关于公式的构造，可能有一点疑问，即上面提到的“它们（）可以以固定数目种方式设置为合法窗口”是真的吗？因为假如将设置为合法窗口的方式不是固定数目种，那么就有可能破坏该归约的多项式时间性。“它们（）可以以固定数目种方式设置为合法窗口”确实为真，因为中的每一个代表一个中一个符号，设是中符号的数目，那么只依赖于图灵机，而不依赖于输入的长度，所以对于为一常数，那么的所有可能取值有种，而对于依然为一固定数目，而根据上面的合法窗口与非法窗口的示例与分析过程，给定的一个具体取值，一定可以根据的所有转移规则立刻判断出该取值是否构成一个合法窗口（其中“立刻”指的是，的所有转移规则的数量只依赖于，不依赖于输入长度，对于也为一常数，因此也不破坏该归约的多项式时间性），所以该命题为真。

下面分析该归约的复杂性，证明它在多项式时间内完成。为此考察的大小。首先，估计一下它的变量的数目。画面是一个表格，所以它包含个单元。每个单元有与它相关联的个变量，其中是中符号的数目。因为只依赖于图灵机，而不依赖于输入的长度，所以变量总数是

估计一下的每个部分的大小。对于画面的每个单元，公式包含固定长度的公式片段，所以长度为。公式对画面第一行的每个单元包含一个片段，所以长度为。公式对画面最后一行的每个单元包含一个固定长度的公式片段，所以长度为。公式对画面的每个单元包含固定长度的公式片段，所以长度为。于是的总长为。该结果完全符合目标，因为它说明的长度是的多项式。

为看出能在多项式时间内生成该公式，注意它的高度重复性。公式的每一部分由许多几乎相同的片段组成，只是在下标上有简单的变化。因此可以容易地构造一个归约，在多项式时间内从输入生成。

下面给出一个该归约的具体例子：让我们来利用这个通用归约的模板将一个几乎最简单的语言多项式归约到。

找到一个多项式时间内接受的如图5所示，其中只包含与。注意，实际上是一个，但是任意一个都可以看作一个，因此我们就将看作一个。对于，其多项式时间限制，因为显然在任何输入上运行所经过的最大步数为2，但是为了不失一般性，要假设，所以我们令（这样保证了时）。。

接下来我们给出两个输入与的归约结果与，其中而，期望得到的结果是与。

如下：

首先给出公式，由于，所以，，所以有



公式如下，其中起始ID为。



公式如下：



公式如下：



其中，



所以，。

同理，，其中，、、、



因为起始ID为。

至于公式中的“构成一个合法窗口”部分，需要考虑种的取值，其中是中符号的数目，然后再判断哪些取值构成合法窗口，由于只依赖于图灵机，而不依赖于输入的长度，所以对于为一常数，所以这个过程没有破坏归约的多项式时间性。但是要在该报告中给出所有满足“构成一个合法窗口”的取值也不太现实，毕竟依然是一个很大的数，故报告中只好忽略这一部分。

下面给出一个满足的赋值，以此来证明：

通过画面的方式给出一个满足的赋值，如图6所示，其中，任意cell[i,j]=s表示：

且

而则没有满足的赋值，因为不存在一个正确的在运行导致接受状态的ID序列，从而也不存在一个在运行对应的接受画面。

所以确实有与。

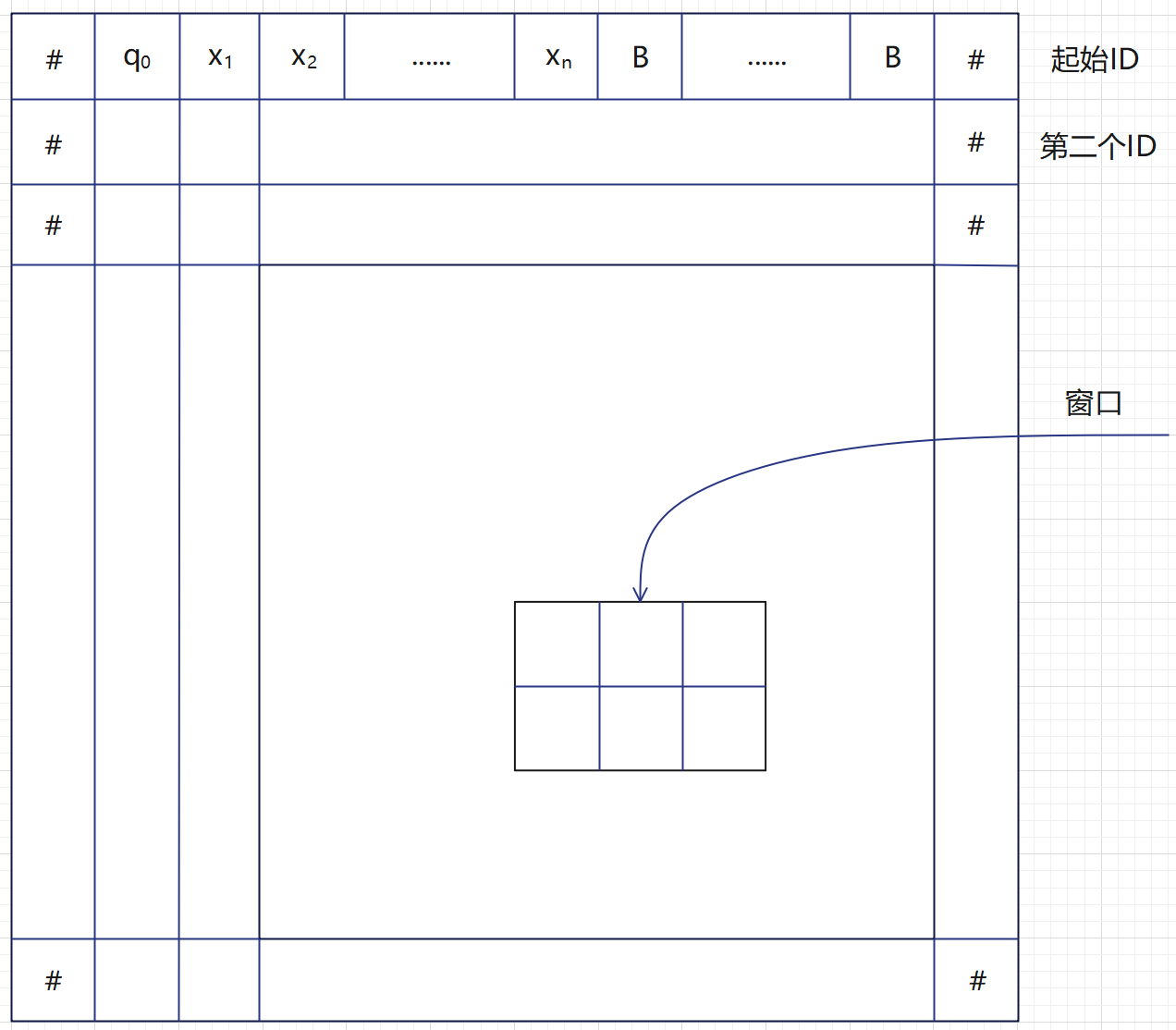


图1 在上运行对应的画面

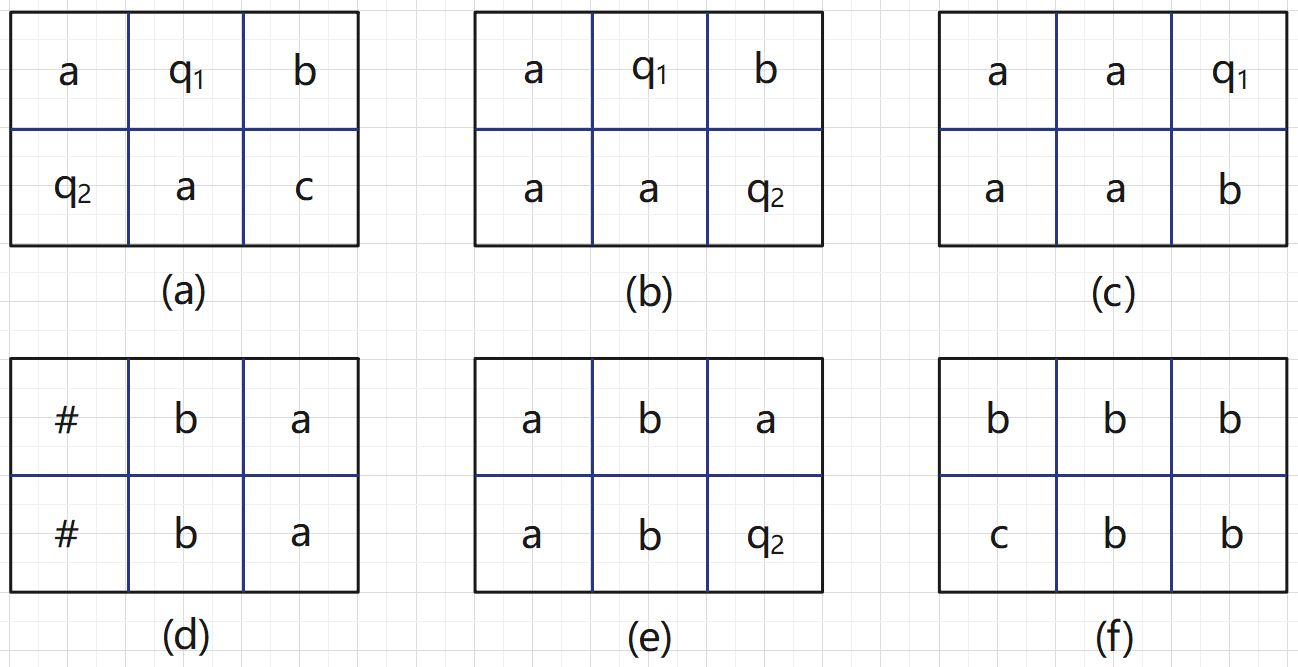


图2 合法窗口示例

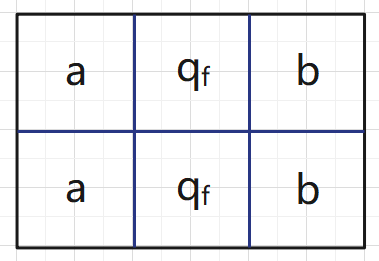


图3 一种特殊的合法窗口

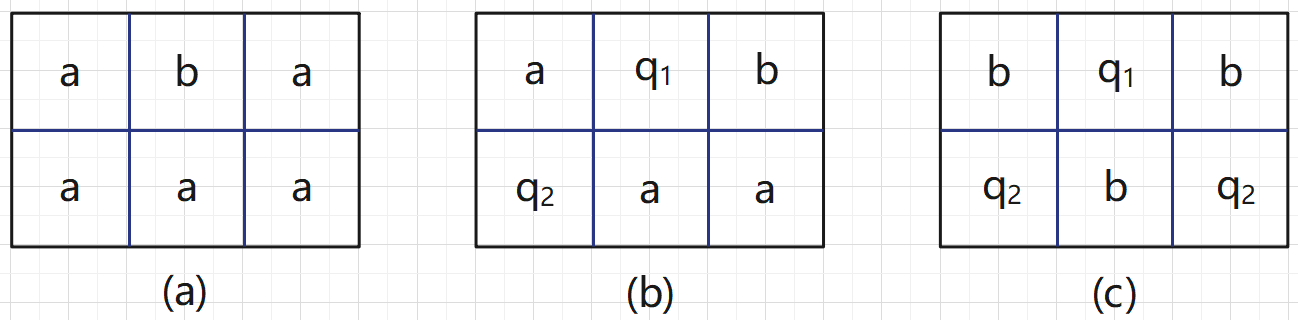


图4 非法窗口示例

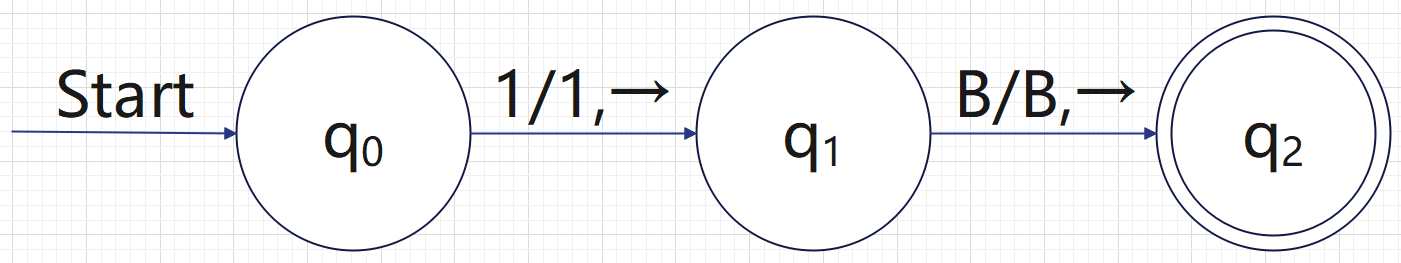


图5 接受的

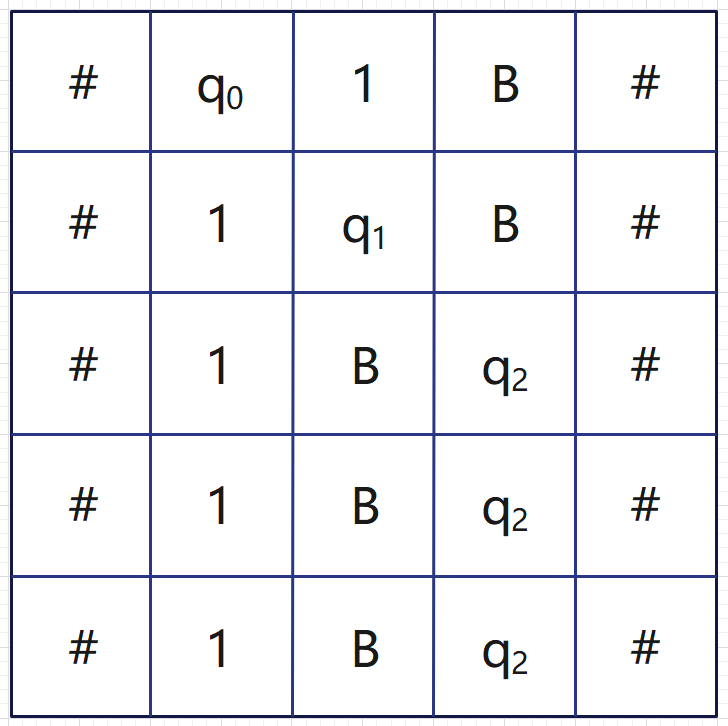


图6 满足的赋值