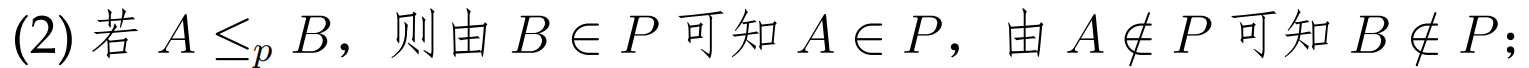


证明：假定，，那么有多项式时间可计算函数和使得，。考虑复合函数，构造一个多项式时间复杂性图灵机按以下方式计算：首先，用一个多项式时间复杂性图灵机模拟，输入为，输出为（由于已经假设是多项式时间可计算的，所以这样的图灵机是存在的）；然后，用一个多项式时间复杂性图灵机模拟，输入为，输出为，因此是多项式时间可计算函数，且。因此，可通过规约函数得到。



证明：若，则有判定的多项式时间算法，因为则有是从到的多项式时间规约。那么判定的多项式时间算法的描述如下：

=“对输入：

1. 计算。
2. 在输入上运行，输出的输出。”

因为是从到的规约，所以。于是，只要，就接受，从而接受；若，则，不接受，从而不接受。且因为的两个步骤都在多项式时间内运行，所以在多项式时间内运行。其中，步骤2在多项式时间内运行是因为两个多项式的合成还是多项式。

所以，由可知，而其逆否命题“由可知”同样成立。



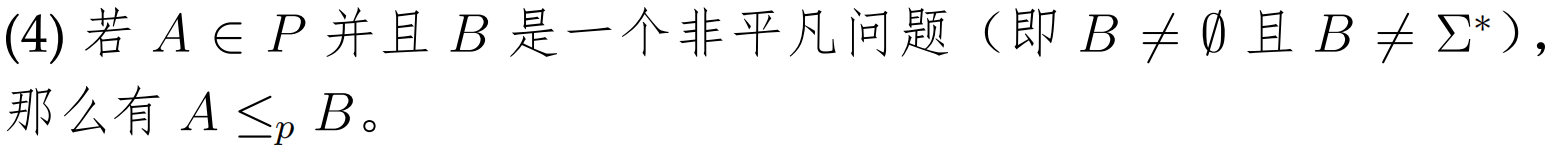
证明：若，则有判定的非确定型多项式时间图灵机，因为则有是从到的多项式时间规约。那么构造判定的非确定型多项式时间图灵机如下：

=“对输入：

1. 计算。
2. 在输入上运行，输出的输出。”

因为是从到的规约，所以。于是，只要，中就存在一条计算分支接受，从而中存在一条计算分支接受；若，则，不接受，从而不接受。且因为的两个步骤都在多项式时间内运行，所以在多项式时间内运行。其中，步骤2在多项式时间内运行是因为两个多项式的合成还是多项式。

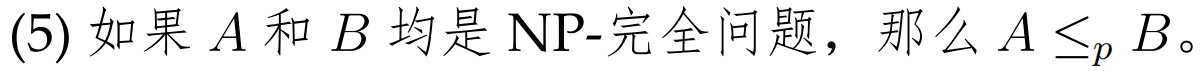
所以，由可知，而其逆否命题“由可知”同样成立。



证明：由于为非平凡问题，所以存在输入与，与都有固定长度。因为，所以有判定的多项式时间算法。构造使得当且仅当如下：

在输入上运行，若，则；若，则。

显然。因为为多项式时间算法，且与长度都固定，所以可以在多项式时间内计算，所以。



证明：因为是-完全问题，所以是问题。由于是-完全问题，任意问题可以多项式时间归约到-完全问题，所以。



证明：因为是-完全问题，所以任意问题可以多项式时间归约到，又因为，所以任意问题都为问题，；又，所以。



证明：首先，容易看出：属于。给定无向图和整数，猜测中个顶点并验证这些顶点构成一个-团（分别验证每个顶点与其余顶点都相连，可以在多项式时间内完成）。

若，由是-完全问题可得是-完全问题。

下证：

给定一个的实例，归约为字符串，其中为如下定义的无向图：

中的顶点分成组，每组三个顶点，各组称为三元组。每个三元组对应中的一个子句，三元组中的每个顶点对应于相应子句的一个文字。的每个顶点用它对应的中的文字做标记。除以下两种情形以外，中的其他顶点对都相连：同一个三元组内的顶点无边相连、相反标记的两个顶点（如和）无边相连。

下证是可满足的当且仅当有-团：



若是可满足的，在满足赋值下的每个子句中至少有一个文字为真。在的每个三元组中，选择在该满足赋值下为真的文字对应的顶点（如果在某一子句中不止一个文字为真，则任意选择其中一个文字）。选择出来的个顶点恰好构成一个-团，因为其中任意两个顶点不来自同一个三元组，且其中也不存在相反标记的两个顶点（否则其对应的两个文字不能在该满足赋值下同时为真），所以其中任意两个顶点间有边相连。



若有-团，因为在同一个三元组中的顶点都无边相连，所以该团中的任何两个顶点不在同一个三元组中，因此该团由每一个三元组中的一个顶点组成。因为团中任意两个顶点对应的文字都不是相反标记的（否则这两个顶点间无边），所以可以给的变量赋值，使得团中每个顶点对应的文字都为真。在该赋值下，的每个子句都包含一个赋值为真的文字，所以此时为真。所以是可满足的。



证明：是问题：给定图和上界，猜测一个个顶点的集合，验证的每条边都至少有一个端点属于这个集合，可以在多项式时间内完成。

下证：

给定的一个实例：图和下界，归约为图和上界作为的实例，其中为的顶点个数。显然能在多项式时间内完成这个变换。

下证有规模为的独立集，当且仅当有规模为的顶点覆盖：



设是的顶点集合，设是规模为的顶点覆盖。断言是独立集。假设不然，即在中存在顶点对和，在和之间有边。和都不属于，但和之间有边，这与为顶点覆盖相矛盾，所以假设不成立，是独立集。而显然有个顶点，得证。



设是规模为的独立集。断言：规模为的顶点覆盖。假设不然，那么存在某条边没有被覆盖，所以和都不属于而属于，但和之间有边，这与为独立集相矛盾。



证明：是问题：给定无向带权图和限度，猜测一个的所有顶点的排序，验证沿着这个顶点序列可以走出一个总权低于的哈密顿回路，这可以在多项式时间内完成。

下证、，从而：

先证：

给定图，归约构造带权图，其顶点和边与的相同，每条边权为1，限度等于的顶点数。显然，有长度为的哈密顿回路当且仅当有哈密顿回路。

再证：

给定有向图，归约构造的无向图称为。对于中每个顶点，在中存在三个顶点，的边如下：

1. 对于中所有顶点，在中有边和。
2. 如果在中有有向边，那么在有边。

下证中有哈密顿回路当且仅当中有有向哈密顿回路：



假设是中的哈密顿回路，于是有是中的无向哈密顿回路。



因为中的每个顶点只有两条边，因此若在哈密顿回路中出现，则一定把和其中一个作为直接前驱，另一个作为直接后继。因此的哈密顿回路的顶点的上标一定在模式或相反的模式之间变化，两种模式对应着遍历无向哈密顿回路的不同方向。不妨设中存在的哈密顿回路的顶点上标模式为。考虑回路中从上标为2的顶点到上标为0的顶点之间的边，则这些边是中的有向边，并且中存在哈密顿回路沿着这些有向边所指的方向经过每一条边。因此，中的无向哈密顿回路就产生中的有向哈密顿回路。

因为是-完全的，，所以是-完全的。



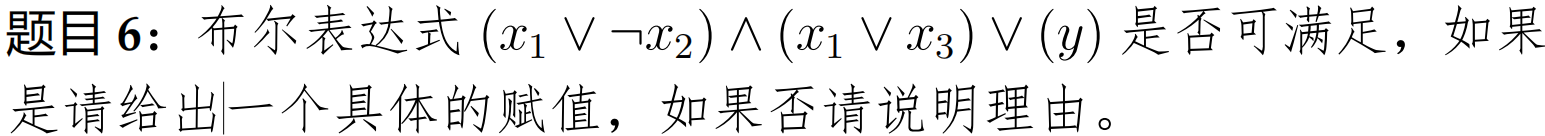
证明：包括所有存在至少一个不可满足赋值的布尔表达式，以及不是有效布尔表达式的串，这个语言是的：首先判断给定的串是否是有效的布尔表达式，不是则属于这个语言；如果是，接着猜测一个这个布尔表达式的赋值，验证在该赋值下该布尔表达式为假。因为是的，则其补是问题。

要证是-完全问题，即证任意，，即，即任意，，又因为是的，即证是-完全的。

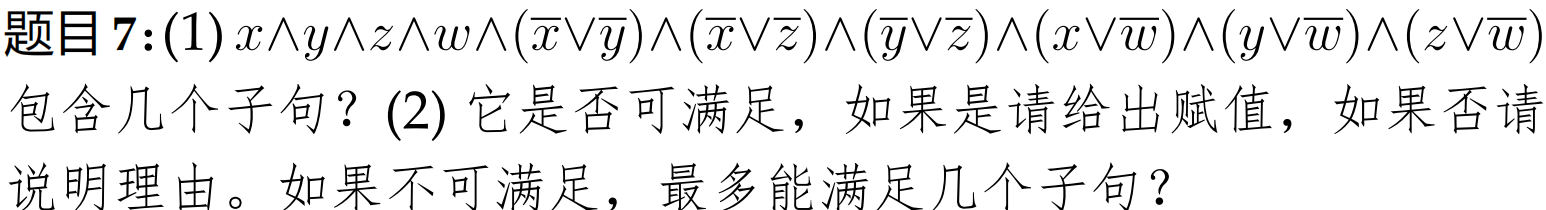
下证：

给定的一个实例，归约得到作为的实例。

若可满足，则的一个可满足赋值即为的一个不可满足赋值，则；若，即有一个不可满足赋值，则这个赋值即为的一个可满足赋值，则可满足。



证明：可满足。如，取值任意、即满足；更具体地，即为一个满足该式的赋值。



证明：（1）10。

（2）不可满足。假设可满足，则有x=1，y=1，那么，矛盾，所以假设不成立，该式不可满足。



证明：是完全的：给定子句集合和整数，猜测一个对中所有子句中所有变元的赋值，验证此赋值至少满足中的个子句，这可以在多项式时间内完成。

下证：

给定一个实例，其中有个子句，对于其中每个子句（其中是文字），引入一个不同于中所有变元的新变元，产生10个子句：。这样总共产生的个子句组成一个子句集，这个子句集和整数作为归约得到的的实例。

下证：对于上述的10个子句，如果一个赋值满足，那么存在一个在这个赋值基础上对也赋值的赋值，这个赋值满足这10个子句中的7个，且不存在其他赋值满足更多子句；如果一个赋值不满足，那么任何在这个赋值基础上对也赋值的赋值最多满足这10个子句中的6个。

1.若，设置，则满足上述10个子句中的第一行和第三行的所有子句，共7个。若设置，则只满足第一行的前3个子句和第三行的所有子句，共6个，不多于7个。

2.若，设置，则满足第一行的前3个子句、第二行的后2个子句、第三行的前2个子句，共7个。若设置，则满足第一行的前2个子句、第二行的后2个子句、第三行的所有子句，也为7个。中任意两个为1时同理。

3.若，设置，则满足第一行的1个子句、第二行和第三行的所有子句，共7个。若设置，则只满足第一行的2个子句和、第二行的所有子句、第三行的1个子句，共6个，不多于7个。中任意一个为1时同理。

4.若，即该赋值不满足时，假如设置，则只满足第一行的和第二行的所有子句，共4个；假如设置，则只满足第二行和第三行的所有子句，共6个。

因此，若可满足，则中个子句可同时满足，则可以恰好满足中个子句；若不可满足，则中个子句不可同时满足，则中至少有一个子句不能满足，则其对应的中的10个子句中最多有6个满足，则中被满足的子句个数严格小于。证毕。



证明：：给定一个实例，归约直接将作为的实例。由于是格式，而格式一定也是格式，所以。由于是完全的，所以任意， 。又因为，所以任意， ，所以是难的。



证明：（1）1.

2.

3.

得到：



（2）1.；2.；3.；

4.；5.；

6.；7.。

得到：



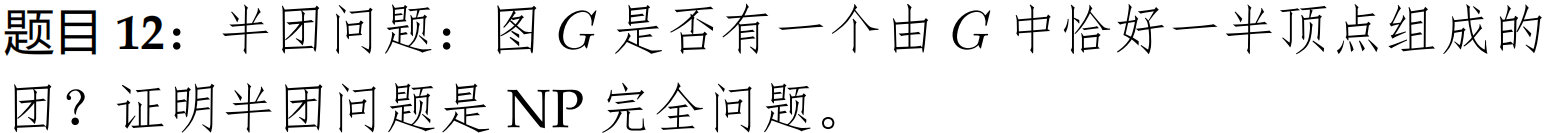


证明：是NP问题：给定一个布尔表达式，猜测4个不同的的赋值，验证这4个赋值满足，可以在多项式时间内完成。

下证：

给定一个的实例，引入2个不出现在中的变元，归约得到作为的实例。

若可满足，则存在4个可满足赋值（在的可满足赋值的基础上加上或或或）；若不可满足，则显然也不可满足。



证明：是问题：给定无向图，猜测中一半顶点，验证这些顶点中每个顶点与其余顶点都相连，即这些顶点构成一个团，可以在多项式时间内完成。

下证：

给定一个的实例：无向图和整数，假定有个顶点，归约得到作为的实例，的构造如下：

1.若，。显然，有半团当且仅当有-团；

2.若，在的基础上添加个独立顶点（没有边与这些顶点相连）得到，此时有个顶点，显然，有半团当且仅当有-团；

3.若，在的基础上添加个顶点，这些顶点与中原有的每一个顶点都相连，并且这些顶点互相之间也都相连，得到，此时有个顶点，有半团当且仅当有-团（若有个顶点组成的半团，则除去新添加的个顶点之外，其中至少包含个中的顶点，意味着有-团；若有-团，则这个顶点加上新添加的个顶点构成一个的半团）。