7.3

a.

代码见附录

b.

对于任意的模型，恒有下列几个语句可以判断真假

$A \wedge \neg A = False \qquad A \vee \neg A = True \quad True \vee \neg A = True$

$False \wedge \neg A = False$

c.

在实际中，可能会有一些模型比较复杂，比如说是多项∧、∨、⇒、¬运算的复合运算，当函数复合程度越高、逻辑运算次数越多。即使真值存在，在有限时间内也难以判断真值。

d.

可以将原公式化成主析取范式或者主合取范式。这样通过判断每个子句的结果，然后进行析取或者合取得到公式的真值。例如，在部分模型中，A 为真，B 未知，$A \vee B = True$，$\neg A \wedge B = False$。但是存在着在部分模型中无法检测真值的情况，例如 A 为真，B 未知时，$A \wedge B$、$\neg A \vee B$ 、$True \wedge B$均未知。

e.

改进的算法是将公式化为主析取范式或者主合取范式，通过对子句析取或者合取获得公式的真值。对于子句，通常可以快速判断真值，如$A \wedge B \wedge C \ldots . \wedge N$，如果有任何一个变量为假，则该子句为假。这样的例子还有很多。由此可以看到，改进的算法可以加速判断一个语句是否由 KB 包含，即 TT-ENTAILS?更有效。

7.4

题号	a	b	c	d
True/False	true	false	true	false
题号	e	f	g	h
True/False	true	true	true	true
题号	i	j	k	l
True/False	false	true	true	false

7.7

a.

总模型个数为$2^4 = 16$，当$B \vee C = False$时，当且仅当$B = C = False$，因此该语句对应的模型数为$16 - 1 * 2^2 = 12$。

b.

当$\neg A \vee \neg B \vee \neg C \vee \neg D = False$时，当且仅当$A = B = C = D = False$，因此该语句对应的模型数为$2^4 - 1 = 15$。

c.

考虑$A \wedge \neg B \wedge C \wedge D = True$时，当且仅当$A = C = D = True, B = False$，此时$A \Rightarrow B = Flase$，因此语句$(A \Rightarrow B) \wedge A \wedge \neg B \wedge C \wedge D \equiv Flase$，该语句对应的模型数为 0。

7.20

首先将 s1~s6 蕴含式化简成合取范式

s_1: $(\neg A \vee B \vee E) \wedge (\neg B \vee A) \wedge (\neg E \vee A)$

s_2: $\neg E \vee D$

s_3: $\neg C \vee \neg F \vee \neg B$

s_4: $\neg E \vee B$

s_5: $\neg B \vee F$

s_6: $\neg B \vee C$

DPLL 算法过程：

组合成子句$\{(\neg A \vee B \vee E), (\neg B \vee A), (\neg E \vee A), \neg E \vee D, \neg C \vee \neg F \vee \neg B, \neg E \vee B, \neg B \vee F, \neg B \vee C\}$
初始化$symbols = A, B, C, D, E, F$
提取纯文字D,从$symbols$消去D，$symbols = A, B, C, E, F$
从子句中消除D相关子句
令$A = True$
　　考察子句$\{B \vee E, \neg C \vee \neg F \vee \neg B, \neg E \vee B, \neg B \vee F, \neg B \vee C\}$
　　令**$B = True$**
　　　　考察子句$\{\neg C \vee \neg F, F, C\}$
　　　　令**$F = C = True$**
　　　　　　此时子句为假
　　　　　　Return False (此时说明 A=B=true 不满足要求)
　　令**$B = False$**

考察子句$\{E, \neg E\}$

显然此时不存在使得子句为真的变量

Return False

上述过程得到$A = True$不能使得子句为真

令$\boldsymbol{A = False}$

考察子句$\{(\neg B), (\neg E), \neg C \vee \neg F \vee \neg B, \neg E \vee B, \neg B \vee F, \neg B \vee C\}$

令$\boldsymbol{B = E = False}$

发现此时子句全为真

Return True

Finsh

附录

```
def pl_true(s, m):
    if (s == True):
        return True
    elif (s == False):
        return False
    op, args = s.op, s.args
    elif (is_symbol(op)):
    return lookup(s, m)
    elif (op == '~'):
    return not pl_true(args[0], m)
    elif (op == 'or'):
    return pl_true(args[0], m) | pl_true(args[1], m)
    elif (op=='and'):
    return pl_true(args[0], m) & pl_true(args[1], m)
    elif (op=='->'):
    return (not pl_true(args[0], m))|pl_true(args[1], m)
    elif (op='<->'):
    return ((not pl_true(args[0], m)) | pl_true(args[1], m))=((not pl_true(args[1], m)) |
pl_true(args[0], m))
    else:
    print('关键词错误')
```