puzzles

April 28, 2024

```
[1]: from z3 import *
```

0.0.1 Who took drugs

```
[2]: guys = ['S', 'Mi', 'B', 'R', 'Ma']
     Said = [Bool(f'{i}_said') for i in guys]
     Drug = [Bool(f'{i}_drug') for i in guys]
     solver = Solver()
```

```
[3]: p0 = Xor(Drug[1], Drug[2])
     p1 = Xor(Drug[3], Drug[0])
     p2 = Xor(Drug[1], Drug[4])
     p3 = Xor(Drug[2], Drug[4])
     p4 = Xor(Drug[2], Drug[3])
     P = [p0, p1, p2, p3, p4]
```

```
[4]: Q = [Said[i] == P[i] for i in range(5)]
     said_four_true = PbEq([(said, True) for said in Said], 4)
     solver.add(Q)
     solver.add(said_four_true)
```

```
[5]: solver.add( Implies(Drug[3], Drug[2]) )
     solver
```

```
[5]: [S_said == Xor(Mi_drug, B_drug),
      Mi_said == Xor(R_drug, S_drug),
      B_said == Xor(Mi_drug, Ma_drug),
      R_said == Xor(B_drug, Ma_drug),
      Ma_said == Xor(B_drug, R_drug),
      PbEq(((S_said, 1),
            (Mi_said, 1),
            (B_said, 1),
            (R_said, 1),
            (Ma_said, 1)),
           4),
      Implies(R_drug, B_drug)]
```

[9]:

```
solver.check()
m = solver.model()
for said in Said:
    print(f'{said}: {m[said]}')
for drug in Drug:
    print(f'{drug}: {m[drug]}')
```

```
S_said: True
Mi_said: True
B_said: True
R_said: False
Ma_said: True
S_drug: True
Mi_drug: False
B_drug: True
R_drug: False
Ma_drug: True
```

0.0.2 Which one contains salt

[19]:

```
# True: still contrains salt; Flase: was replaced with sugar
Salt = [Bool(f'salt_{i}') for i in range(4)]
Note0 = Salt[0]
Note1 = Salt[1]
Note2 = Not(Salt[2])
Note3 = Not(Salt[1])
Note = [Note0, Note1, Note2, Note3]
One_true = PbEq([(Note[i], True) for i in range(4)], 1)
One_salt = PbEq([(Salt[i], True) for i in range(4)], 1)
```

[20]:

```
solver = Solver()
solver.add(One_true, One_salt)
solver.check()
```

[20]:

```
sat
```

[21]:

```
m = solver.model()
for salt in Salt:
    print(f'{salt}: {m[salt]}')
```

```
salt_0: False
salt_1: False
salt_2: True
salt_3: False
```

0.0.3 When is the birthday

[ ]:

```
G = Int('George')
W = Int('William')
J = Int('John')
A = Int('Abe')
M = Int('Millard')
guys = [G, W, J, A, M]
```

[41]:

```
solver = Solver()
for guy in guys:
    solver.add(1 <= guy)
    solver.add(guy <= 5)

solver.add(Distinct(guys))

# # Not necessarily in a week
# solver.add((M - G)%7 == (W - A)%7)
# solver.add((A - J)%7 == 2)
# solver.add(M == 4)

# Within a week
solver.add((M - G) == (W - A), M >= G, W >= A)
solver.add((A - J) == 2)
solver.add(M == 4)

solver.check()
m = solver.model()
for guy in guys:
    print(f'{guy}: {m[guy]}')
```

```
George: 2
William: 5
John: 1
Abe: 3
Millard: 4
```

[ ]:

```

```