

(M1) SAT - SOLUTION



in z3 python framework. Init operations take a lot of time.
If you do benchmark between different encodings into the same file. Result could be not correct.

TEAM HOUSE OR AWAY PERIOD WEEK

1 DECISION VARIABLE: $X_{1011}, X_{1111}, \dots, X_{m-1 \frac{m}{2} m-1}$

2 DOMAIN: $X_{TPHW} \in \{0,1\}$

4 ADDED CONSTRAINT

3 CONSTRAINT

- 4 Each game has exactly 2 team
- 5 Each game cannot be played by 2 home or 2 away team
- 6 Each team in a day play at home or away (no both!)

- 1 • every team plays with every other team only once;
- 2 • every team plays once a week;
- 3 • every team plays at most twice in the same period over the tournament.

EXAMPLE:

$M=4$

	W1	W2	W3
P1	1-2	1-3	1-4
P2	3-4	2-4	2-3

1 PLAY 3 TIMES X (NOT OK!)

- $M-1$ WEEK
- $M/2$ PERIODS
- M TEAMS

⚠ $M=4$ HAS NO FEASIBLE SOLUTION! ⚠

• MODELLING

	$T=1$	$T=2$	$T=3$	$T=4$
P	$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$
	HOUSE AWAY			

CONSTRAINTS (IS USED INSTANCES ABOVE TO HELP VISUALIZATION)

1 every team plays with every other team only once;

	$T=1$	$T=2$	$T=3$	$T=4$
P	$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$
	H=0			

AT MOST ONE $\rightarrow \forall t_1, t_2 (t_1 \neq t_2) \rightarrow$ AT MOST ONE $\left(\left((X_{t_1 0 1 1} \vee X_{t_1 1 1 1}) \wedge (X_{t_2 0 1 1} \vee X_{t_2 1 1 1}) \right), \left((X_{t_1 0 1 2} \vee X_{t_1 1 1 2}) \wedge (X_{t_2 0 1 1} \vee X_{t_2 1 1 1}) \right), \dots, \left((X_{t_1 0 P W} \vee X_{t_1 1 P W}) \wedge (X_{t_2 0 P W} \vee X_{t_2 1 P W}) \right) \right)$

\downarrow

$\binom{M}{2}$ DIFFERENTS AT MOST $\rightarrow O(M^2)$

EACH AT MOST HAS $|P \cdot W|$ $\rightarrow (A \vee B) \wedge (C \vee D)$

```

1 clauses = []
2 for t1 in range(0,team):
3     for t2 in range(t1+1,team):
4         for p in range(0,periods):
5             for w in range(0,weeks):
6                 t_i = Or(list(vars[t1, :, p, w].flatten()))
7                 t_j = Or(list(vars[t2, :, p, w].flatten()))
8                 clauses.append(And(t_i, t_j))
9                 s.add(at_least_one(clauses))
10 # print("I : number of clause in the at_least_one", len(clauses)) # Dimension check
11 clauses = []

```

② every team plays once a week;

$$P \begin{bmatrix} \overbrace{1 \ 0 \ 0}^w \\ 0 \ 0 \ 0 \end{bmatrix} \begin{matrix} T=1 \\ (0 \ 1 \ 1) \end{matrix} \quad \begin{matrix} T=2 \\ (1 \ 1 \ 0) \end{matrix} \begin{matrix} (0 \ 0 \ 0) \\ (0 \ 0 \ 0) \end{matrix} \quad \begin{matrix} T=3 \\ (0 \ 1 \ 0) \\ (0 \ 0 \ 1) \end{matrix} \begin{matrix} (0 \ 0 \ 0) \\ (1 \ 0 \ 0) \end{matrix} \quad \begin{matrix} T=4 \\ (0 \ 0 \ 1) \\ (1 \ 1 \ 0) \end{matrix} \begin{matrix} (0 \ 0 \ 0) \\ (0 \ 0 \ 0) \end{matrix}$$

EXACTLY ONE $\rightarrow \forall t \forall w$ EXACTLY ONE $(X_{t01w}, X_{t11w}, \dots, X_{t1pw})$
 $|t \cdot w|$ DIFFERENTS EXACTLY ONE \rightarrow EACH HAS $|2 \cdot p|$ ELEMENTS

```
1 # Constraint2 - Every team plays once at week
2 for t in range(0,team):
3     for w in range(0,weeks):
4         c = exactly_one( list(vars[t,:,w].flatten()) )
5         s.add(c)
```

③ every team plays at most twice in the same period over the tournament.

$$P \begin{bmatrix} \overbrace{0 \ 0 \ 0}^w \\ 0 \ 0 \ 0 \end{bmatrix} \begin{matrix} T=1 \\ (0 \ 0 \ 0) \end{matrix} \quad \begin{matrix} T=2 \\ (1 \ 1 \ 0) \end{matrix} \begin{matrix} (0 \ 0 \ 0) \\ (0 \ 0 \ 0) \end{matrix} \quad \begin{matrix} T=3 \\ (0 \ 1 \ 0) \\ (0 \ 0 \ 1) \end{matrix} \begin{matrix} (0 \ 0 \ 0) \\ (1 \ 0 \ 0) \end{matrix} \quad \begin{matrix} T=4 \\ (0 \ 0 \ 1) \\ (1 \ 1 \ 0) \end{matrix} \begin{matrix} (0 \ 0 \ 0) \\ (0 \ 0 \ 0) \end{matrix}$$

AT MOST $\frac{k}{2}$ $\rightarrow \forall t \forall p$ AT MOST 2 $(X_{t0p1}, \dots, X_{t1pw})$
 $|p| \cdot |t|$ DIFFERENTS AT MOST \rightarrow EACH HAS $|2 \cdot |w||$ ELEMENTS

? ENCODING

$$\text{ATMOST } \frac{k}{2} \rightarrow \neg (X_{1011} \wedge X_{1012} \wedge X_{1111}) \wedge \neg (X_{1011} \wedge X_{1022} \wedge X_{1122}) \wedge \neg (X_{1122} \wedge X_{1122} \wedge X_{1122})$$

ANY COMBINATION OF $k-1$ VAR

$O(M^{2+k})$

```
1 # Constraint3 - Every team plays at most twice in the same period over the tournament.
2 for t in range(0,team):
3     for p in range(0,periods):
4         c = at_most_k( list(vars[t,:,p,:].flatten()) , k = 2 )
5         s.add(c)
```

4 Each game has exactly 2 team

$$P \left[\begin{array}{c} T=1 \\ \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 0 \end{pmatrix} \end{array} \right] \quad \begin{array}{c} T=2 \\ \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \end{array} \quad \begin{array}{c} T=3 \\ \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} \end{array} \quad \begin{array}{c} T=4 \\ \begin{pmatrix} 0 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \end{array}$$

$$\boxed{\text{EXACTLY TWO}} \rightarrow \forall p \forall w \text{ EXACTLY TWO } \left(\overset{\text{ELEM}}{X_{10pW} \vee X_{11pW}}, \dots, (X_{m0pW} \vee X_{m1pW}) \right)$$

\downarrow $|P \cdot W|$ DIFFERENTS EXACTLY TWO \downarrow HAS $|M|$ ELEMENTS

```

1 # Constraint4.2 - Each game has exactly 2 team
2 clauses = []
3 for p in range(0,periods):
4     for w in range(0,weeks):
5         for t in range(team):
6             x = Or(list(vars[t,:,p,w].flatten())) ] ELEM
7             clauses.append(x)
8         s.add(exactly_k(clauses, k = 2))
9         # print("4.2 number of clause in the exactly_k: " , len(clauses)) # Dimension check
10    clauses = []
  
```

5 Each game cannot be played by 2 home or 2 away team

$$P \left[\begin{array}{c} T=1 \\ \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 0 \end{pmatrix} \end{array} \right] \quad \begin{array}{c} T=2 \\ \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \end{array} \quad \begin{array}{c} T=3 \\ \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} \end{array} \quad \begin{array}{c} T=4 \\ \begin{pmatrix} 0 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \end{array}$$

$$\boxed{\text{AT MOST ONE}} \rightarrow \forall f \forall p \forall w \text{ AT MOST ONE } (X_{1fPW}, \dots, X_{mfPW})$$

\downarrow $|F \cdot P \cdot W|$ DIFFERENTS AT MOST ONE \downarrow HAS $|G|$ ELEMENTS

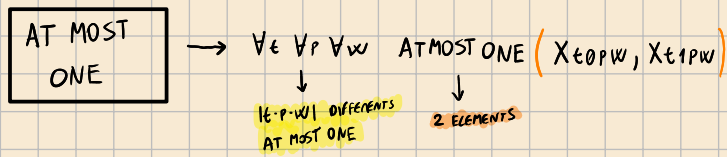
\downarrow Δ DIFFERENT NAME IN HOME, SEQUENTIAL, ... ENCODINGS

```

1 # Constraint5 - Each game cannot be played by 2 home-team or 2 away-team
2 for h in range(home):
3     for p in range(0,periods):
4         for w in range(0,weeks):
5             s.add(at_most_one(list(vars[:,h,p,w].flatten()), name = f"ff({h}-{p}-{w}))) # TODO -> FIND THE BEST NAME FOR EACH CATEGORY (NAME IS IMPORTANT TO DON'T HAVE SAME Y)
6             if(h == 0 and p == 0 and w == 0 ): # DEBUG
7                 print(list(vars[:,h,p,w].flatten())) # DEBUG
8             ttt.add(at_most_one(list(vars[:,h,p,w].flatten()), name = f"T(h-p-w)")) # DEBUG
9             # print("4.3 number of clause in the at_most_one : " , len(list(vars[:,h,p,w].flatten())) # Dimension check
  
```

6 Each team in a day play at home or away (no both!)

$$P \begin{matrix} \overbrace{[}^w \\ \underbrace{]}^w \end{matrix} \begin{matrix} T=1 \\ \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \end{matrix} \begin{matrix} T=2 \\ \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} \end{matrix} \begin{matrix} T=3 \\ \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \end{matrix} \begin{matrix} T=4 \\ \begin{pmatrix} 0 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} \end{matrix} \begin{matrix} \begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 0 \end{pmatrix} \\ \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \end{matrix}$$



```

1 # Constraint6 - Each team in a day play at home or away cannot both
2 for t in range(team):
3     for p in range(periods):
4         for w in range(weeks):
5             s.add(at_most_one(list(vars[t,:,p,w].flatten()) , name = f"fo({t-p-w})"))
6             # print("4.1 number of clause in the at_most_one: " , len(list(vars[t,:,p,w].flatten())) # Dimension check

```

(M1) SAT - SOLUTION - V2

BENEFITS
 + HEURISTIC
 - SEQUENTIAL
 ++ BITWISE
 • PAIRWISE (THE SAME)

4 Each game has exactly 2 team + Each game cannot be played by 2 home-team or 2 away-team

CONSTRAINT

CONSTRAINTS 4,5,6
 OF VERSION1

EXACTLY ONE $\rightarrow \forall F \forall P \forall W$ EXACTLY ONE $(X_{0FPW}, X_{1FPW}, \dots, X_{MFPW})$
 |P-W| DIFFERENTS
 M ELEMENTS

```
1 # Constraint 4 - Each game has exactly 2 team + Each game cannot be played by 2 home-team or 2 away-team
2 for h in range(home):
3     for p in range( periods):
4         for w in range( weeks):
5             # exactly one home and exactly one away + exactly 2 teams, opposite sides
6             heule.add(exactly_one_heule([vars[t,h,p,w] for t in range(team)] , name = f'h(h)p(p)w(w)'))
```

(M1) SAT - SOLUTION - V3

BENEFITS
 FIRST COLUMN (n)
 SECOND
 FIRST LINE (n)
 ++ PAIRWISE
 -- HEURISTIC
 • BITWISE
 • SEQUENTIAL

1 SYMMETRY BREAKING

YOU FIXED
 IN THE FIRST
 COLUMN MATCH
 FROM 1 TO M/2

EX:

M=6

W1
 P1 (1,?)
 P2 (2,?)
 P3 (3,?)

(?,?) IS THE SAME

FIXED COLUMN

IS ALSO
 POSSIBLE TO FIX 2 LINE

W1 W2 ...
 P1 (1,?) (2,?) ...

CUSTOM
 ENCODINGS

$(X_{1011} \vee X_{1111}) \wedge (X_{1012} \vee X_{1121}) \dots (X_{1011} \vee X_{1111})$

Encodings

$O(m^2)$ CLAUSES

• PAIRWISE \rightarrow AT-MOST-ONE $(x_1, x_2, x_3, x_4) \rightarrow (1x_1 \vee 1x_2) \wedge (1x_1 \vee 1x_3) \cdots \binom{m}{2}$

• SEQUENTIAL \rightarrow AT-MOST-ONE (x_1, x_2, x_3, x_4)

m CLAUSES + $m-1$ AUXILIAR VAR

$$\begin{array}{ccc} \downarrow & & \\ \begin{array}{l} (1x_1 \vee s_1) \wedge \\ (1x_2 \vee s_2) \wedge \\ (1x_3 \vee s_3) \wedge \end{array} & \begin{array}{l} (1s_1 \vee s_2) \wedge \\ (1s_2 \vee s_3) \wedge \end{array} & \begin{array}{l} (1x_2 \vee 1s_1) \wedge \\ (1x_3 \vee 1s_2) \wedge \\ (1x_4 \vee 1s_3) \end{array} \end{array}$$

$$(1x_1 \vee s_1) \wedge x_1 \wedge (1x_2 \vee 1s_1)$$

- (a) If x_i is true, then s_i must be true
- $\neg x_1 \vee s_1$
 - $\neg x_2 \vee s_2$
 - $\neg x_3 \vee s_3$
- (b) Monotonicity: once s_i is true, all following must be true
- $\neg s_1 \vee s_2$
 - $\neg s_2 \vee s_3$
- (c) At most one: if x_{i+1} is true, then s_i must be false
- $\neg s_2 \vee \neg s_1$
 - $\neg s_3 \vee \neg s_2$
 - $\neg s_4 \vee \neg s_3$

(M2) SAT - SOLUTION - V3

1 DECISION VARIABLE:

$\binom{m}{2}$ ALL POSSIBLE TEAM COMBINATION

is t1,t2 play a game on week w ?

$M[0][1][0], \dots, M[m][m-1][w]$ } $\binom{m}{2} \cdot |w|$ VARIABLES

$H[0][0], \dots, H[t][w]$ } $|t| \cdot |w|$ VARIABLES is team t home in week w ?

$P[0][0][0], \dots, P[t][p][w]$ } $|t| \cdot |p| \cdot |w|$ VARIABLES

is team is playing in the period and week p,w ?

2 DOMAIN: $M, H, P \in \{0,1\}$

4 ADDED CONSTRAINT

3 CONSTRAINT

- every team plays with every other team only once;
- every team plays once a week;
- every team plays at most twice in the same period over the tournament.

- in a game (1 home team - 1 away team)
- each game is played by 2 team
- TODO

5 PRECOMPUTING? YES (ROUND ROBIN PAIRS)

■ = NUMBER OF CARDINALITY CONSTRAINTS FOR 2 CONSTRAINTS (WHEN YOU USE ENCODINGS LIKE HEURLE, SEQUENTIAL, ... YOU SHOULD HAVE m UNIQUE NAMES (LABEL))
■ = NUMBER OF VARIABLES FOR 2 CARDINALITY CONSTRAINT

- ✓ 1 • every team plays with every other team only once;

EXACTLY-ONE → $\forall t_1, t_2$ E-ONE ($M[t_1][t_2][0], \dots, M[t_1][t_2][w]$)

MIP → $\forall t_1, t_2$ $M[t_1][t_2][0] + \dots + M[t_1][t_2][w] = 1$

```
1 # Constraint 1 : every team plays with every other team only once;
2 for t1 in range(team):
3     for t2 in range(t1 + 1, team):
4         S.add(PbEq([M[t1][t2][w], 1] for w in range(weeks)), 1))
```

- ✓ 2 • every team plays once a week;

EXACTLY-ONE → $\forall w, t$ E-ONE ($P[t][0][w], \dots, P[t][p][w]$)

MIP → $\forall w, t$ $P[t][0][w] + \dots + P[t][p][w] = 1$

```
1 # Constraint2 : each team play exactly one per week, sum over p of P[t][p][w] == 1
2 for t in range(team):
3     for w in range(weeks):
4         S.add(PbEq([P[t][p][w], 1] for p in range(weeks)), 1))
5 # SAT translation of exactly one 1 (PbEq also enabled weight bit = Add + ...)
```

- ✓ 3 • every team plays at most twice in the same period over the tournament.

EXACTLY-TWO → $\forall t, p$ E-TWO ($P[t][p][0], \dots, P[t][p][w]$)

MIP → $\forall t, p$ $P[t][p][0] + \dots + P[t][p][w] = 2$

$t=1$

	w_1	w_2	\dots	w_m
p_1				
p_2				
\vdots				
p_m				

→ = 1

$t=1$

	w_1	w_2	\dots	w_m
p_1				
p_2				
\vdots				
p_m				

→ = 2

```
1 # Constraint3 : each team plays two per period : sum over weeks of P[t][p][w] <= 2
2 for t in range(team):
3     for p in range(weeks):
4         S.add(PbLe([P[t][p][w], 1] for w in range(weeks)), 2))
```

4 • in a game (1 home team - 1 away team)

A	B	XOR
0	0	0
0	1	1
1	0	1
1	1	0

CUSTOM $\rightarrow \forall w, t_1, t_2 \quad M[t_1][t_2][w] \rightarrow \text{XOR}(H[t_1][w], H[t_2][w])$

MIP $\rightarrow ?$

```
1 # Constraint4 : Home/away consistency: when (t1,t2) plays in w, HOME differs
2 for w in range(weeks):
3     for t1 in range(team):
4         for t2 in range(t1 + 1, team):
5             S.add(Implies(M[t1][t2][w], Xor(HOME[t1][w], HOME[t2][w])))
```

5 each game is played by 2 team

EXACTLY-TWO $\rightarrow \forall p, \forall w \quad \text{E-TWO}(P[0][p][w], \dots, P[t_1][p][w])$

MIP $\rightarrow \forall p, \forall w \quad P[0][p][w] + \dots + P[t_1][p][w] = 2 \rightarrow \text{GAME}(p, w) \downarrow t_1 + t_2 + \dots + t_m = 2$

```
1 # Constraint5 : Each game is played by 2 team : sum over t of P[t][p][w] == 2
2 for w in range(weeks):
3     for p in range(teams):
4         S.add(PbEq([P[t][p][w] for t in range(team)], 2)) # SAT translation of x1+x2+x3 =2 (PbEq also enabled weight 3x1 + 4x2 + ...)
5         S.add(exactly_k([P[t][p][w] for t in range(team)], 2)) # TODO : use exactly k of sequential
```

LINK - VARIABLES

CUSTOM $\rightarrow \forall w, t_1, t_2 \quad M[t_1][t_2][w] \rightarrow (P[t_1][0][w] \wedge P[t_2][0][w]) \vee \dots \vee (P[t_1][p][w] \wedge P[t_2][p][w])$

MIP $\rightarrow ???$

```
1 # B) Link periods to pair variables M
2 for w in range(weeks):
3     for t1 in range(team):
4         for t2 in range(t1 + 1, team):
5             # If the pair plays in week w, they share exactly one period
6             S.add(Implies(M[t1][t2][w], Or([And(P[t1][p][w], P[t2][p][w]) for p in range(weeks)])))
7
8             # If two teams share (period p, week w), then that (t1,t2) is the match that week
9             for p in range(weeks):
10                 S.add(Implies(And(P[t1][p][w], P[t2][p][w]), M[t1][t2][w]))
```


ROUND ROBIN PAIRS → PAIR GENERATION

```

1 def round_robin_pairs(n):
2     arr = list(range(n))
3     weeks_list = []
4     for w in range(n - 1):
5         pairs = []
6         for i in range(n // 2):
7             a = arr[i]
8             b = arr[n - 1 - i]
9             if a < b: pairs.append((a, b))
10            else: pairs.append((b, a))
11        weeks_list.append(pairs)
12        # rotate (fix arr[0])
13        arr = [arr[0]] + [arr[-1]] + arr[1:-1]
14    return weeks_list

```

LAST ELEM OF THE LIST FIRST 2 TO THE LAST (NOT INCLD)

EX: $M=4$

$ARR = [0, 1, 2, 3]$

$W-L = []$

↓ ...

$W-L = [[0, 3], [1, 2]]$ $W=1$

$ARR = [0, 3, 1, 2]$

$W-L = [..., [0, 1], [2, 3]]$ $W=2$

↓ ...

$W=1$ $W=2$ $W=3$
 $[[(0, 3), (1, 2)], [(0, 2), (1, 3)], [(0, 1), (2, 3)]]$

USING PRECOMPUTED LIST OF ROUND ROBIN

```

1 if __name__ == '__main__':
2     # Pin ONLY the pairings to a valid round-robin schedule to avoid symmetry blow-up.
3     rr_weeks = round_robin_pairs(team) # weeks -> list of (a,b)
4     M_true = {(min(a,b), max(a,b), w) for w, pairs in enumerate(rr_weeks) for (a,b) in pairs}
5     for t1 in range(team):
6         for t2 in range(t1+1, team):
7             for w in range(weeks):
8                 if (t1, t2, w) in M_true:
9                     s.add(M[t1][t2][w])
10            else:
11                s.add(Nor(M[t1][t2][w]))

```

EXTENDED

```

1 M_true = set() # start with an empty set
2 for w, pairs in enumerate(rr_weeks):
3     print(w, pairs)
4     for a, b in pairs:
5         t1 = min(a, b)
6         t2 = max(a, b)
7         M_true.add((t1, t2, w))

```

EX: $M=4$

$RR_WEEKS = [[(0, 3), (1, 2)], [(0, 2), (1, 3)], [(0, 1), (2, 3)]]$