```
P1 1. \neg(In(A) \land In(B)), \neg(In(A) \land In(F))
```

- In(A) <-> (Order(A, 1) V Order(A, 2) V Order(A, 3) V Order(A, 4)),
 In(B) <-> (Order(B, 1) V Order(B, 2) V Order(B, 3) V Order(B, 4))
- Order(A, 1) V Order(B, 1) V Order(C, 1) V Order(D, 1) V Order(E, 1) V Order(F, 1) V Order(G, 1) V Order(H, 1) V Order(I, 1) V Order(J, 1),
 Order(A, 2) V Order(B, 2) V Order(C, 2) V Order(D, 2) V Order(E, 2) V Order(F, 2) V Order(G, 2) V Order(H, 2) V Order(I, 2) V Order(J, 2)
- 4. ¬(Order(A, 1) ^ Order(B, 1)), ¬(Order(A, 2) ^ Order(B, 2))
- 5. ¬(Order(A, 1) ^ Order(A, 2)), ¬(Order(B, 1) ^ Order(B, 2))
- P2 1. Processor Exclusivity Constraint: For each pair of tasks U, V, on any processor P, and at any time T, \neg (Exec(U, P, T) $^{\wedge}$ Exec(V, P, T))

Eg.
$$\neg$$
(Exec(A, 1, 1) $^{\land}$ Exec(B, 1, 1))

2. Task Precedence Constraint: For each pair of tasks U, V that are connected by an arc: V->U, Exec(U, P, T) -> (Exec(V, 1, 1) V Exec(V, 1, 2) V ... V Exec(V, 1, T-1) V ... V Exec(V, K, T-1)), where K and T are the number of processors and the time U is executed.

Eg. When there's an arc A->D: Exec(D, 1, 2) -> (Exec(A, 1, 1) V Exec(A, 2, 1))

- 3. Temporal Constraint: For each task U, Exec(U, 1, 1) V Exec(U, 1, 2) V ... V Exec(U, 1, M) V ... V Exec(U, K, M), where K and M are the number of processors and maximum time for execution
- Eg. Exec(A, 1, 1) V Exec(A, 1, 2) V Exec(A, 1, 3) V Exec(A, 2, 1) V Exec(A, 2, 2) V Exec(A, 2, 3)
- P3 1. Forall x, y, F(x, y) -> F(y, x) (or Forall x, y, F(x, y) <-> F(y, x))
- 2. Forall x, y, $C(x, y) \rightarrow \neg C(y, x)$
- 3. Exists x, F(A, x) ^ F(B, x)
- 4. Forall x, $C(x, A) \rightarrow \neg F(B, x)$
- 5. Exists x, $C(D, x) ^ (forall y, C(y, B) \rightarrow F(x, y))$
- 6. Forall x, $F(D, x) \rightarrow (exists y, C(y, x))$
- 7. Forall x, $F(A, x) \rightarrow (exists y, C(y, x) \land (forall z, \neg C(z, y)))$
- 8. Exists x, $C(x, B) \wedge (forall y, z, C(y, B) \rightarrow \neg C(y, x))$