A simple version of the game is used: The game graph has a random initial configuration where every player has a random direct trust towards every other player, as well as a random capital. These values may be uniformly distributed in an interval or may follow another distribution such as the exponential, or may have a high probability of being zero. The exact distribution however is not determined at this point, as it is not yet needed.

The players also know that after $R$ rounds (blocks in bitcoin terms) exactly two players are selected at random (these choices follow the uniform distribution). One is dubbed seller and the other buyer. The seller offers a good that costs $C$ (this number is common knowledge from the beginning), which she values at $C - l$ and the buyer values at $C + l$ ($l$ is also common knowledge from the beginning).

The players play simultaneously in each round and can do any of the known actions. If two actions collide (e.g. $A$ reduces $DTr_{A \to B}$ and $B$ steals from $DTr_{A \to B}$ as well), then one of the two actions is chosen with equal probability (50%).

To better model a player's actions and the aforementioned conflict resolution, we demand that each change explicitly mentions the source and the destination of the funds for each of her actions. For example, player $A$ decides on the values of all the following variables. This constitutes a concrete round for $A$.

$$\forall B, C \in \mathcal{V}, move\left(A, (A, B), (A, C)\right) = ?$$
$$\forall B, C \in \mathcal{V}, B \neq A, move\left(A, (B, A), (A, C)\right) = ?$$

The first argument is the player who decides, the second argument is from which direct trust to take the funds and the third is to which direct trust to deposit the funds.

In this simple version, $A$ is not allowed to set $move\left(A, (A, B), (A, B)\right)$ to any value different than 0 for any $B$. This choice is made to facilitate the analysis. According to the implementation modelling we have agreed upon however, $A$ will (probably) be able to do it. This will in practice give $A$ the option to increase the chances that $B$ will not be able to take the funds from this particular direct trust. (Exactly the same goes for $move\left(A, (B, A), (B, A)\right)$. At first sight this means that $A$ can increase her incoming direct trust, but this is not the case. Either way we will avoid this complication for now.)

Now for the constraints for player's $A$ move.

1. It makes no sense to deposit to and withdraw from a specific direct trust in the same round. This constraint applies only to outgoing direct trusts, because incoming direct trusts cannot be increased.

$$\forall B, C, D \in \mathcal{V}, move\left(A, (A, B), (A, C)\right) \times move\left(A, (A, D), (A, B)\right) = 0$$
$$\wedge$$
$$\forall B, C, D \in \mathcal{V}, move\left(A, (A, B), (A, C)\right) \times move\left(A, (D, A), (A, B)\right) = 0$$

2. One cannot use more funds than are available from a single direct trust.

$$\forall B \in \mathcal{V}, \sum_{C \in \mathcal{V}} move\left(A, (A, B), (A, C)\right) \leq DTr_{A \to B}$$
$$\forall B \in \mathcal{V}, \sum_{C \in \mathcal{V}} move\left(A, (B, A), (A, C)\right) \leq DTr_{B \to A}$$

3. If two players try to change the same direct trust, then set the relevant moves of one of the two players (chosen uniformly at random) to 0.

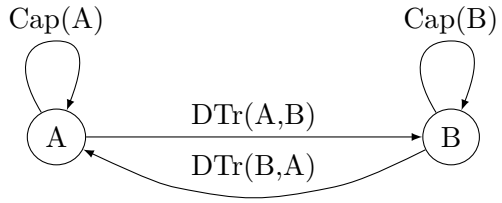
```
resolveConflict(A, B) :
  sum1 = Σ_{C∈V} move(A, (A, B), (A, C))
  sum2 = Σ_{C∈V} move(B, (A, B), (B, C))
  if (sum1*sum2 != 0)
    choice ←$ {A, B}
    if (choice == A)
      ∀C ∈ V, move(A, (A, B), (A, C)) = 0
    else # if (choice == B)
      ∀C ∈ V, move(B, (A, B), (B, C)) = 0

resolveAllConflicts() :
  ∀A, B ∈ V
    resolveConflict(A, B)
    resolveConflict(B, A)
```

`resolveAllConflicts()` is executed after all players choose their moves for a round.

To get an idea of how this game would look like, please take a look here, at p. 3.

Complete 2-player graph:




A's possible actions:

$$Action(A) = \begin{matrix} \\ \\ \\ \end{matrix} \begin{matrix} \multicolumn{3}{c}{from} \\ DTr(B,A) & DTr(A,B) & Cap(A) \end{matrix} \begin{bmatrix} x_1 & x_2 & x_3 \\ y_1 & 0 & y_3 \\ z_1 & z_2 & 0 \end{bmatrix} \begin{matrix} Purchase \\ DTr(A,B)\ to \\ Cap(A) \end{matrix}$$

Results:

$$\begin{aligned} DTr'(A,B) &= DTr(A,B) + y_1 + y_3 - x_2 - z_2 \\ DTr'(B,A) &= DTr(B,A) - x_1 - y_1 - z_1 \\ Cap'(A) &= Cap(A) + z_1 + z_2 - x_3 - y_3 \\ Bought &= \frac{x_1 + x_2 + x_3}{cost(b)} \end{aligned}$$

No funds destroyed/created rule:

$$DTr'(A,B) + DTr'(B,A) + Cap(A) + Bought \times cost(b)$$
$$=$$
$$DTr(A,B) + DTr(B,A) + Cap(A)$$

Individual no funds created rules:

$$\begin{aligned} x_1 + y_1 + z_1 &\leq DTr(B,A) \\ x_2 + z_2 &\leq DTr(A,B) \\ x_3 + y_3 &\leq Cap(A) \end{aligned}$$

No adding and reducing from same place rules:

$$\begin{array}{llll} y_1 x_2 = 0 & z_1 x_3 = 0 & z_2 x_3 = 0 & y_3 x_2 = 0 \\ y_1 z_2 = 0 & z_1 y_3 = 0 & z_2 y_3 = 0 & y_3 z_2 = 0 \end{array}$$