What is Trust

Orfeas Stefanos Thyfronitis Litos

University of Edinburgh
o.thyfronitis@ed.ac.uk

Abstract. We will try to define all the abstract properties that we would like "Trust" to have.

Abstract Trust

1 Definitions

Definition 1 (Agent). *An agent can be thought of as either a programme/Turing machine/protocol (inanimate) or as a pseudonymous identity corresponding to a human. Let $\mathcal{P}$ be the set of all agents.*

Definition 2 (State). *Let agent $P \in \mathcal{P}$. If P is inanimate, then P's state at an instance $t \in \mathbb{N}$, is a function $S : \mathcal{P} \times \mathbb{N} \to \mathcal{S}$ that returns the state of the machine. If P is a human, then $S(P,t)$ is a record of the internal condition of the human, as observed by the human.*

Definition 3 (Global State). *The global state $GS : \mathbb{N} \to \mathcal{S}^{|\mathcal{P}|}$ is the set of the states of all agents $P \in \mathcal{P}$ at a specific instance:*

$$GS(t) = (S(P_1,t), ..., S(P_n,t)) \ ,$$
$$\text{where } \bigcup_{i=1}^{n} \{P_i\} = \mathcal{P} \ .$$

Definition 4 (Trust). *Trust is a function $Tr : \mathcal{P}^2 \times \mathcal{S}^{|\mathcal{P}|} \times \mathbb{P}\left(\mathcal{S}^{|\mathcal{P}|}\right) \times \mathbb{N}^2 \to \mathcal{R}^+ \cup \{\infty\}$.*

Let $in = (P_1, P_2, GS_1, \{GS'_1, ..., GS'_n\}, t_1, t_2) \in \mathcal{P}^2 \times \mathcal{S}^{|\mathcal{P}|} \times \mathbb{P}\left(\mathcal{S}^{|\mathcal{P}|}\right) \times \mathbb{N}^2$. Then $Tr(in)$ is interpreted as the level of commitment P_1 can provide that the actions of P_2 upon a world where $GS(t_1) = GS_1$ will lead to a world where $GS(t_2) \in \{GS'_1, ..., GS'_n\}$.

We use the notation $\mathcal{D}_{Tr} = \mathcal{P}^2 \times \mathcal{S}^{|\mathcal{P}|} \times \mathbb{P}\left(\mathcal{S}^{|\mathcal{P}|}\right) \times \mathbb{N}^2$.

2 Desired Properties

1. Let $t \in \mathbb{N}$. Then $\forall\,(P_1, P_2, GS, States, t, t) \in \mathcal{D}_{Tr}$ it is

$$Tr\,(P_1, P_2, GS, States, t, t) = \begin{cases} \infty, \text{ if } GS \in States \\ 0, \text{ if } GS \notin States \end{cases}.$$

   In other words, all players trust all other players infinitely with respect to the current state of the world.
2. Let $t_1, t_2 \in \mathbb{N} : t_1 > t_2$. Then $\forall\,(P_1, P_2, GS, States, t_1, t_2) \in \mathcal{D}_{Tr}$ it is

$$Tr\,(P_1, P_2, GS, States, t_1, t_2) = \begin{cases} \infty, \text{ if } GS\,(t_2) \in States \\ 0, \text{ if } GS\,(t_2) \notin States \end{cases}.$$

   This means that the past cannot be modified.
3. Let $(P_1, P_2, GS, States, t_1, t_2) \in \mathcal{D}_{Tr}$. If

$$Tr\,(P_1, P_2, GS, States, t_1, t_2) > Tr\,(P_1, P_1, GS, States, t_1, t_2)$$

   and all global states in $States$ are more desirable than $\mathcal{S}^{|\mathcal{P}|} \setminus States$ for P_1 at the moment t_2, then P_1 prefers to hand over whatever she controls to P_2 at the moment t_1 than maintain this control for herself.

Economic Trust

We would like to provide players with an API where they:

1. Entrust coins to another player
2. Appropriate coins previously entrusted by another player
3. Retract coins previously entrusted to another player
4. Query trust towards another player

The following functionality provides such an interface:

$\mathcal{F}_{Trust}$

```
1 Initialize trusts from all players to all players to 0
2 Initialize coins for all players to some values
3
4 Upon receiving entrust(id_2, x) from id_1:
5   If id_1 has at least x coins
6     Increase trust from id_1 to id_2 by x
7     Decrease the coins of id_1 by x
8     Recalculate indirectTrusts
```

```
  Else discard request

Upon receiving steal(id_2, x) from id_1:
  If trust from id_2 to id_1 is equal to or exceeds x
    Decrease trust from id_2 to id_1 by x
    Increase the coins of id_1 by x
    Recalculate indirectTrusts
  Else discard request

Upon receiving distrust(id_2, x) from id_1:
  If trust from id_1 to id_2 is equal to or exceeds x
    Decrease trust from id_1 to id_2 by x
    Increase the coins of id_1 by x
    Recalculate indirectTrusts
  Else discard request

Upon receiving query(id_2) from id_1:
  answer = indirectTrust(id_1, id_2)
  Send answer to id_1
```

References