質と量の組を用いたトラストの定式化

真野 健 †　　櫻田 英樹 †　　塚田 恭章 †

† 日本電信電話株式会社 NTT コミュニケーション科学基礎研究所
243-0198 神奈川県厚木市森の里若宮 3-1
[mano.ken|sakurada.hideki|tsukada.yasuyuki]@lab.ntt.co.jp

あらまし 質と量の組を用いてトラストを定量的に定式化する．ある仮定のもと，トラストを加法性を持つ値として捉え，その総和を超えないことをトラスト計算の健全性と定める．また，個々のトラスト計算の結果が健全であることだけでなく，計算手続きがトラストの付値の変化に対して安定であることの重要性も指摘する．そのような問題設定のもと，さまざまなトラスト合成演算を定義し，それを用いたトラスト計算プロトコルの健全性と安定性を証明する．さらに，プロトコルのプライバシ強化についても考察する．

Formulation of Trust using Pair of Quality and Quantity

Ken Mano†　　Hideki Sakurada†　　Yasuyuki Tsukada†

†NTT Communication Science Laboratories, NTT Corporation.
3-1 Morinosato Wakamiya Atsugi Kanagawa 243-0198 Japan
[mano.ken|sakurada.hideki|tsukada.yasuyuki]@lab.ntt.co.jp

Abstract We present a mathematical formulation of a trust using a pair of quality and quantity. Under a certain assumption, we regard a trust as an additive value and define the soundness of trust computation as not to exceed the summation. Moreover, we point out the importance of, not only soundness of each computed trust, but also the stability of trust computation procedure against the change of trust value assignment. Under this setting we define various trust composition operators, and prove soundness and stability of a trust computation protocol using the operators. We also study privacy enhancement of the protocol.

1　はじめに

本稿ではトラストの数理的な定式化について論じる．トラストの数理的な定式化には，大きく分けて二つのアプローチがある．ひとつは論理的なアプローチ [1, 4, 5, 7] で，知識論理あるいは信念論理と呼ばれる様相論理の一種を用い，トラストという問題の論理的な構造を明らかにすることを目指している．もうひとつは，トラストメトリックに演算や推論を施して必要なトラスト値を計算する計算的アプローチ [2,3,6,8] である．こちらは，確率論や，それにもとづく subjective logic，fuzzy logic などを用い，計算された値の正しさを保証することを目指している．本稿のアプローチは後者に属し，以下の二つの特徴を持つ．

第一の特徴は，トラストメトリックを質と量の組で定式化し，合成演算を定義していることである．トラストには，確率などによって表現される質的側面とともに，量的側面があると考えられる．日常的な意味においても，“あいつはいつも頼りになる” という質的な側面とともに，“あいつとは長い付き合いだ” という量的な側面もトラストに関する重要な判断基準である．本稿ではこれを数理的なメトリックとして定式化し，その基本的な性質を明らかにする．トラスト値に対する合成演算は，従来の計算的アプローチと同様，並列的な合成演算 (複数の情報

源からのトラスト値を統合) と直列的な合成演算 (一方のトラスト値をもう一方のトラスト値に基づいて割り引く) とに分けられる．

質と量の組をメトリックとするアイデア自体は新しいものではなく，例えば [2] で暗に用いられており，並列合成を重み付き加算によって自然に定義できるという特長を持つ．しかし，その性質は十分に調べられてはいない．本稿では，合成演算の代数的性質のいくつかを明らかにする．

第二の特徴は，合成演算を用いた計算の正しさを，演算そのものを用いて定式化していることである．合成演算自体の定義が妥当であっても，その適用は常に妥当とは限らず，トラストの重複勘定によって計算結果が不当に増幅される場合がある．我々は，演算を用いたトラスト計算の正しさ (健全性，安定性) を，情報の多寡を表す半順序 $\leq$ を用いて定式化する．また，そのような議論を可能とする (非形式的な) 前提として，トラストの加法性の概念を導入する．この定式化の特長は，$\leq$ が合成演算のみに依存して定義され，確率的な仮定を必要としないことである．

さらに，トラストを分散的に計算するためのプロトコルを提案し，演算の代数的性質を用いてその正しさを証明する．また，プロトコルのプライバシ強化についても考察する．

本稿の構成は以下のとおりである．2 節では，基本的な問題設定と，トラストの合成演算について述べる．3 節では，トラスト計算の正しさとして健全性と安定性を定義し，正しい計算を表現する構文として線形項を導入する．4 節では，トラストを分散的に計算するためのプロトコルを示し，そのプライバシ強化について述べる．紙面の都合上，ほとんどの証明は割愛する．

2 トラストとは

2.1 トラストの質と量

本稿では，トラスト値を質と量の対として定義する．任意の人 A，B について，

A の B に対するトラスト $t_{AB} = (p_{AB}, q_{AB})$.

q_{AB} は非負実数であり，トラスト t_{AB} の量と呼ぶ．意図された解釈は，A と B のやり取りの量 (やり取りの回数，取引額，など) である．p_{AB} はトラスト t_{AB} の質と呼び，$p_{AB} \in [0,1]$ と仮定する．意図された解釈は，やり取りの中で B が A の期待に応えた割合である．$q_{AB} = 0$ ならば $p_{AB} = 0$ と仮定し，簡単のために $(0,0)$ を 0 と書くことがある．例えば，トラストの量を質問の回数とし，質を正しい返事が得られた割合とする．A が B に過去 100 回質問しその中で 90 回正しい返事が得られたならば，A の B に対するトラストは $(0.9, 100)$ となる．

さらに本稿では，そのようなトラストが分散的に配置された状況を考える．すなわち，各自が自分の，他人に対するトラストを保持しているとし，trusted third party などの存在は仮定しない．各々は，自分が保持するトラスト値について聞かれたら，正しい値を答える場合もあるし，返事をしない，知っていても “知らない” と答える，嘘の値を言うといった場合もあるとする．

2.2 トラストの合成演算

本節では，トラストの合成演算について説明する．合成には，並列と直列がある．

トラストの並列合成 $+$

C に対するトラストを A と B に聞いたら，$t_{AC} = (p_{AC}, q_{AC})$ と $t_{BC} = (p_{BC}, q_{BC})$ だった．そのとき，仮に A，B は 100% 信頼するとして，それらを総合してどのように評価すればいいか．

$$t_{AC} + t_{BC} = \left(\frac{q_{AC} \cdot p_{AC} + q_{BC} \cdot p_{BC}}{q_{AC} + q_{BC}},\ q_{AC} + q_{BC} \right).$$

つまり，量は単なる足し算，質は量で重み付けした平均である．ただし，$(0,0)+(0,0) = (0,0)$ と定める．質の定義を確率の類推で考えると，量を独立な試行の回数とみなし，q_{AC} 回のうち p_{AC} の確率で真であり q_{BC} 回のうち p_{BC} の確率で真である場合に，$q_{AC} + q_{BC}$ 回全体で真となる確率である．例えば，$t_{AC} = (0.9, 100)$，$t_{BC} = (0.8, 1000)$ のとき，$t_{AC} + t_{BC} = (0.81, 1100)$ となる．

$+$ は結合的かつ可換的であり，$0 + t = t$ を満たす．

トラストの直列合成 $*$

B が A に，C に対するトラストが $t_{BC} = (p_{BC}, q_{BC})$ であることを知らせてきたとする．

さらに，A の B に対するトラストが $t_{AB} = (p_{AB}, q_{AB})$ だったとする．そのとき，A の C に対するトラストはいくらだと思えばいいか．

ここでは，そのようなトラストの直列的な合成を，以下のように定義する:

$$t_{AB} * t_{BC} = (p_{AB} \cdot p_{BC}, min(q_{AB}, q_{BC})).$$

質の定義は，確率の類推で考えると，確率 p_{AB} で p_{BC} かつ確率 $1 - p_{AB}$ で 0 であるときの期待値である．別な言葉で言えば，A が B を信頼し，かつ B が C を信頼する割合である．量の定義は，A は B から伝えられたトラスト情報について，B に対するトラストの量以上の信をおかないという考えに基づいている．例えば，$t_{BC} = (0.9, 1000)$，$t_{AB} = (0.8, 100)$ のとき，$t_{AB} * t_{BC} = (0.72, 100)$ となる．

トラストの直列合成は，伝達によるトラスト情報の劣化を表す．伝達によって劣化すると考える理由は，伝達者が嘘をつく可能性があるからである．嘘には，トラストの量や質を過少に言う場合と過大に言う場合が考えられる．しかし本稿の問題設定では，一般にすべてのトラスト情報を集めることはできないため，量的に過少に評価されるのは不可避である．さらにトラストという問題の性質から考えても，嘘としてより深刻なのは過大な評価であり，上の定義もそれを反映している．

大雑把に言って，上記直列合成は，嘘に関して以下の仮定をしていることになる:
A の B に対するトラストが (p_{AB}, q_{AB}) のとき，B は A へのトラスト情報の伝達に際して，

1. 質を過大に偽るとしても，高々もとの値の $1/p_{AB}$ 倍．
2. 量 q_{AB} 以下のトラスト情報については，その量を過大に偽らない．

逆に，このような仮定が成り立つとき，上記直列合成が妥当となるとも言える．3.1 節では，これをより一般的な形で定式化する．

$*$ は結合的かつ可換的であり，$0 * t = 0$ を満たす．

我々は，合成の定義は上記のものが唯一だと主張するつもりはない．上記定義は本稿における working example だが，用途やユーザの好みによってバリエーションが可能である．

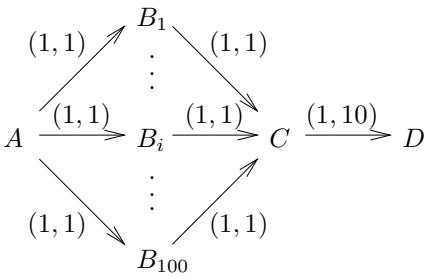

図 1: 重複勘定が起こる状況

2.3 重複勘定の問題

前節において我々は，トラストの並列合成を定義した．しかしこの演算の適用は，常に妥当とは限らない．これは，重複勘定および加法性と深い関係がある．本節では，適用が妥当とならない場合として，3 つのタイプの重複を提示する．

例 2.1 $A, B_1, \cdots, B_{100}, C, D$ の間のトラストを，以下のように仮定する: $t_{AB_i} = (1, 1)$，$t_{B_iC} = (1, 1)$，$t_{CD} = (1, 10)$．そのとき，以下のようなトラスト計算の例を考える:

1. “A の D に対するトラストは $t_{AB_1} * t_{B_1C} * t_{CD} + \cdots + t_{AB_{100}} * t_{B_{100}C} * t_{CD} = (1, 100)$”.
2. “A の C に対するトラストは $t_{AB_1} * t_{B_1C} + \cdots + t_{AB_{100}} * t_{B_{100}C} = (1, 100)$”.

これらの計算は妥当だろうか？ ∎

1 は，量的に 10 しか原資がないのに，重複勘定のため誤って 100 に増幅されてしまっている (ように見える) ことを問題にしている．一方 2 は，表面的には重複勘定はない．しかし，$t_{B_1C}, \cdots, t_{B_{100}C}$ というトラストがどのように定められたかが問題になる．もし，C のたったひとつの行為を $B_1, \cdots, B_{100}$ が同時に目の当たりにした結果 t_{B_iC} を決定したのならば，その量は全体で 1 と勘定されるべきではないか？

例えば，100 人の構成員 $B_1, \cdots, B_{100}$ からなる NGO があったとする．その NGO の活動趣旨に賛同し C が 1 万円の寄付を行い．その事実にもとづきすべての B_i が “C を量的に 1 万円分信頼する” と決心したとする．そのとき，それら信頼を “加算” して，“C は 100 万円分信頼できる” と判断するのは妥当だろうか．

1 と 2 は似ているが違う問題である．後者は，計算の基礎となるトラストの値をいかに決める

かという問題なのに対し，前者は，そのように決めた値を使っていかに計算するかという問題である．

まず 2 について考える．解決として，トラストに基礎的なトラストとそうでないものという区別を導入する．基礎トラストとは，直接的かつ排他的な経験に基づいて各人によって決定されるトラストである．そうでないトラストとは，伝聞情報から計算されたトラストである．

経験が直接的とは，自分の目や耳で見聞きしたことをいう．伝聞や推測は直接的ではない．排他的とは，自分だけで経験したことをいう．もし複数人で 1 つのことを経験した場合は，総和が 1 となるように量を分配する (n 人で一緒に経験したならひとりあたまの量は n 分の 1 にする，など)．分け前が決定できないような場合は，直接的ではないと考える．このような仮定のもとで，並列合成において量を加算することが正当化される．

人の全体集合を $\mathcal{P}$ と書き，有限集合と仮定する．任意の $A, B \in \mathcal{P}$ について，A の B に対する基礎トラストを t_{AB} と表す．単に B に対する基礎トラスト，あるいは基礎トラストと呼ぶこともある．経験が直接的という仮定のもと，本稿では基礎トラストを加法的[1]な値ととらえ，B の総基礎トラスト t_B を以下のように B に対する基礎トラストの総和として定義する:

$$t_B = \sum_{P \in \mathcal{P} - \{B\}} t_{PB}.$$

次に 1 について考える．図 1 において，もし D に対する 0 でない基礎トラストが t_{CD} だけならば，この例は重複勘定のために総基礎トラストを量において上回ってしまっていることになる．加法的な値を正しく計算するためには，このような事態は避ける必要がある．

では，D に対するトラストの計算において，D に対する基礎トラスト以外のトラスト，すなわち伝達経路の基礎トラストは重複勘定してもいいだろうか？

例 2.2 図 2 のような状況において，まず

$$A \to B_i \to C \to D \to E_i \to F$$
$$(i = 1, \cdots, 100)$$

という 100 本の経路各々でトラストを計算し，そのあとそれらの和をとるという計算を考える．これは妥当だろうか，

∎

この例では，計算結果は F の総基礎トラストを超えてはいない．しかし 100 本の経路は C-D 辺を共有しているので，

$$\sum_{i=1}^{100} t_{AB_i} * t_{B_iC} * t_{CD} * t_{DE_i} * t_{E_iF} = (1, 100)$$

という計算では，いわば C-D という小さな量の辺に，大きな量のトラストが 100 回に分けて流れこんでいる．これは合計で考えると，直列合成における量の基本的な考え方である “C は D から伝えられたトラスト情報について，D に対するトラストの量以上の信をおかない” に反している．

実際，もし新たな経験によって t_{CD} が $(0.5, 2)$ に更新されたとすると，計算結果は $(0.5, 100)$ に変化する．つまり，量 100 の計算結果がたった量 1 の変化に大きく左右されることになり，健全とは言えない．よって，このような事態は避けなければならない．

3 トラストのネットワーク

上記の合成演算を使って，ネットワーク上の人からトラストを収集し計算することを考える．

3.1 健全性と安定性

本節では，トラストの計算の正しさを，その具体的な計算方法に依存せずに定式化する．

まず，トラスト上の二項関係 $\leq$ を以下のように定義する:

$$t \leq t' \text{ iff } \exists t_1, t_2 \; t + t_1 = t_2 * t'.$$

この式は，左辺が t' を t_2 分だけ劣化させたものに等しく，かつそれが t に t_1 を加えたものであることを主張する．つまりその直観的な意味は，t が t' にくらべて部分的かつ劣化しているということであり，情報の多寡を表す関係とみなすことができる．

補題 3.1 $\leq$ は以下の性質を満たす．

[1] システム全体の値が，そのサブシステムの値の総和となるということ．

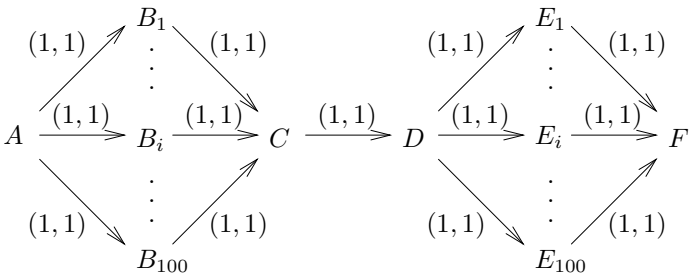

図 2: 伝達経路の基礎トラストの重複勘定

1. 反射性: $t \leq t$
2. 推移性: $t \leq t'$ かつ $t' \leq t''$ ならば $t \leq t''$
3. 反対称性: $t \leq t'$ かつ $t' \leq t$ ならば $t = t'$
4. 減少性: $t * t' \leq t'$
5. 単調性: $t \leq t'$ ならば $t'' + t \leq t'' + t'$
6. 準分配性: $t*(t' + t'') \leq t * t' + t * t''$
7. 強単調性: $t \leq_{\Delta t} t'$ ならば, $t''+t \leq_{\Delta t} t''+t'$ かつ $t'' * t \leq_{\Delta t} t'' * t'$

ただし Δt はトラストであり，$t \leq_{\Delta t} t'$ iff $t' \leq t + \Delta t \ \wedge \ q(t) \leq q(t')$ である．∎

定義 3.2 B に対する計算されたトラスト t は，$t \leq t_B$ のとき健全であるという．∎

本稿の設定では，B に対するトラストを計算する際，一般に $t_B = \sum_{P \in \mathcal{P}} t_{PB}$ をすべて完全に計算はできない．上記健全性は，t は部分的かつ劣化しているかもしれないが，重複計算などを含まず各 t_{PB} を正しく “加法的に” 取り扱うという意味で正しい計算結果のひとつとみなせることをあらわす．

例 3.3 基礎トラストが，図 3 のような状況を考える．A の C に対する計算されたトラスト $s = t_{AB_1} * t_{B_1C} + t_{AB_1} * t_{B_1B_2} * t_{B_2C} = (0.7695, 20)$ を考えると，$(0.7695, 20) \leq (0.9, 20) = t_C$ なので，s は A から C への計算されたトラストとして健全である．また，$s' = t_{AB_2} * t_{B_2C} = t_C$ も同様．しかし，$s + s' = (0.765, 22)$ は，$(0.765, 22) \not\leq t_C$ なので，A の C に対する計算されたトラストとして健全ではない．実際ここでは t_{B_2C} が重複勘定されている．∎

しかし，計算の正しさは個々の計算されたトラストを考えるだけでは不十分である．例 2.2 で示したように，計算結果自体は総基礎トラスト

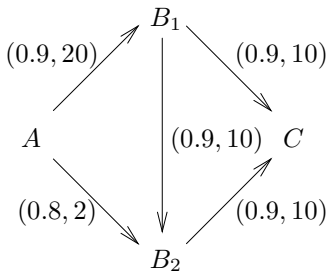

図 3: 重複勘定によって健全でない計算となる例

を超えないが，基礎トラストのわずかな変化に大きく左右されてしまう場合がある．以下ではこのような問題が起こらないことを，計算手順の安定性として定式化する．

ここで問題となるのは，本稿であつかう計算手順は一般に部分的かつ非決定的だということである．次節で我々は，トラストを非決定的プロトコルで分散計算する方法を示すが，それは要求を送る相手の選択や応答の構成に関して非決定性を含む．また問題設定から，要求を受け取った人は，嘘をついたり答えなかったりすることが想定されている．以下では，そのような手順を入力から可能な出力の集合への関数とみなし，その健全性をいわゆる Hoare 前順序を用いて定義する．

定義 3.4 基礎トラスト付値 T は，相異なる人の対からトラストへの関数である．$T(A, B)$ は T における A から B への基礎トラストを表し，t^T_{AB} とも書く．T が文脈から明らかな場合は，従来通り単に t_{AB} と書く．さらに，T における t_{CD} への付値を Δt だけ増やしたものを $T +_{CD} \Delta t$ と書く．∎

定義 3.5 トラスト計算手順 f とは，基礎トラスト付値 T と相異なる人 A，B を与えられて，

停止したら計算されたトラストを出力する手順である．$f(T, A, B)$ は入力 T，A，B に対する可能な出力の全体集合を表す．∎

定義 3.6 トラスト集合上の前順序 $\sqsubseteq$ を以下のように定義する:

$$\mathcal{T} \sqsubseteq \mathcal{T}' \text{ iff } \forall t \in \mathcal{T}\ \exists t' \in \mathcal{T}'\ t \leq t'.$$

さらに，トラスト集合 $\mathcal{T}$ とトラスト t に対して，演算 $\mathcal{T} + t$ を以下のように定義する:

$$\mathcal{T} + t = \{t' + t \mid t' \in \mathcal{T}\}.$$ ∎

定義 3.7 トラスト計算手順 f は，任意の相異なる人 A，B について以下の 2 つの条件を満足するとき，安定であるという．

1. T における B の総基礎トラストが 0 ならば，$f(T, A, B) \sqsubseteq \{0\}$.
2. 任意の相異なる人 C, D とトラスト Δt について，$f(T +_{CD} \Delta t, A, B) \sqsubseteq f(T, A, B) + \Delta t$. ∎

大雑把に言って，トラスト計算手順の安定性は，手順がすべての基礎トラストを加法的に取り扱うことである．

例 3.8 例 2.2 において，t_{CD} のみ $(1, 0.5)$ に変更した基礎トラスト付値を T とし，さらに $T' = T +_{CD} (1, 0.5)$ とする．そのとき，その例にある和の式を用いた決定的な計算手順を考えると，

$$\sum_{i=1}^{100} t^T_{AB_i} * t^T_{B_iC} * t^T_{CD} * t^T_{DE_i} * t^T_{E_iF} = (1, 50),$$

$$\textstyle\sum_{i=1}^{100} t^{T'}_{AB_i} * t^{T'}_{B_iC} * t^{T'}_{CD} * t^{T'}_{DE_i} * t^{T'}_{E_iF} =$$

$$(1, 100) \not\leq (1, 50) + (1, 0.5).$$

よってこの手順は安定ではない．∎

補題 3.9 手順が安定ならば，それが出力する個々の計算されたトラストは健全である．∎

次に，2.2 節で述べた嘘に関する仮定を，$\leq$ を用いてより一般的な形で定式化する．本稿では，各伝達者の嘘をトラスト上の関数として表せると仮定する．参加者 B がトラストを A に伝えるときにつく嘘を表す関数を B の A に対する

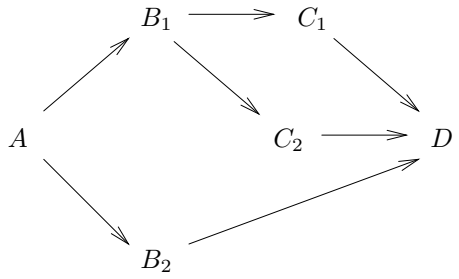

図 4: 線形項が表すグラフ

嘘関数と呼び L_{AB} で表す．B が計算されたトラスト s を A に送る時，実際に A に送られるのは $L_{AB}(s)$ である．そのとき，L_{AB} には以下の不等式が成り立つと仮定する:

$$t_{AB} * L_{AB}(s) \leq s.$$

すなわち，B の A に対する任意の嘘は，“$t_{AB} * \_$” によってキャンセルできるということであり，これを嘘の上限仮定と呼ぶ．

3.2 線形項による計算

人を頂点とする有向グラフ (各辺は，トラストする側からされる側へ向かう) を考え，その上でトラストの計算手順を考える．重複勘定のない手順を表現するために，線形項を定義する．相異なる人の対 A，B に対して，基礎トラスト記号と呼ばれる定数記号 $\tilde{t}_{AB}$ を導入し，それと $+$ および $*$ から構成される項を考える．

定義 3.10 A, B, C を相異なる任意の頂点として，A-B 線形項と，それらが表すグラフ (有向辺の集合) を以下のように定義する:

1. $\tilde{t}_{AB}$ は A-B 線形項 であり， A-B 辺の単一要素集合を表す．
2. A-B 線形項 $S_1, \cdots, S_n$ $(n \geq 1)$ が，それらの表すグラフが互いに有向辺を共有しないならば $S_1 + \cdots + S_n$ は A-B 線形項 であり，$S_1 \cup \cdots \cup S_n$ を表す．
3. S が B-C 線形項 であり，S の表すグラフに A が現れないとき，$\tilde{t}_{AB} * S$ は A-C 線形項であり，S が表すグラフに A-B 辺を加えたものを表す． ∎

例 3.11 図 4 は，線形項 $\tilde{t}_{AB_1} * (\tilde{t}_{B_1C_1} * \tilde{t}_{C_1D} + \tilde{t}_{B_1C_2} * \tilde{t}_{C_2D}) + \tilde{t}_{AB_2} * \tilde{t}_{B_2D}$ が表すグラフを示している．図 1 のグラフを表す線形項は存在しない．このグラフ中で線形項で表せるのは，例えば個々の経路 $\tilde{t}_{AB_i} * \tilde{t}_{B_iC} * \tilde{t}_{CD}$ である． ∎

C, D を任意の人とする．A-B 線形項 S の以下の性質は定義からすぐに導かれる．

- S が表すグラフに C-D 辺が含まれる iff S に $\tilde{t}_{CD}$ が現れる．
- (線形性) $\tilde{t}_{CD}$ は S に高々一度しか現れない．

よって，A-B 線形項 S と基礎トラスト付値 T を与えられた時，S に出現する任意の $\tilde{t}_{CD}$ を t^T_{CD} と解釈して得られる値を，T において S を用いて線形に計算されたトラストと呼び，$[S]^T$ で表す．

補題 3.12 任意の A-B 線形項 S と基礎トラスト付値 T について，$[S]^T \leq t_B$． ∎

4 トラスト計算プロトコル

本節では，トラストを分散的に計算するためのプロトコルを示す．本節で示される結果は，演算の結合性，可換性，ゼロ元，補題 3.1 の性質のみに依存し，演算の具体的定義には依存しない．

4.1 基本プロトコル

以下のメッセージをやり取りする非決定的プロトコルである．

要求 トラストを計算したい対象 C と，要求が辿ってきた人の列 P の組 $\langle C, P\rangle$．

応答 計算されたトラスト s と，それを計算するのに用いた線形項 S の組 $\langle s, S\rangle$．

A が C とやり取りした経験がない場合は，$t_{AC} = 0$ とする．A が D から $\langle C, P\rangle$ という要求を受け取ったら，以下のことを順に行う:

1. $t_{AC} \neq 0$ ならば，応答 $\langle t_{AC}, \tilde{t}_{AC}\rangle$ を 自分自身に送る．
2. 以下の 3 条件を満たす $B_1, \cdots, B_n$ を非決定的に選んで，要求 $\langle C, P \cdot A\rangle$ を送る:
   - B_i は自分自身でも C でもない．
   - t_{AB_i} が 0 でない．
   - B_i が P に含まれない．
3. 応答を待てるだけ待つ．
4. 得られた応答の中から，その第2要素である線形項が互いに同じ基礎トラスト記号を含まないよう $\langle s_{B'_1C}, S_{B'_1C}\rangle, \cdots, \langle s_{B'_kC}, S_{B'_kC}\rangle$ を非決定的に選び (ひとつも選ばなかったら，すぐに終了)，組 $\langle t_{AB'_1} * s_{B'_1C} + \cdots + t_{AB'_k} * s_{B'_kC}, \tilde{t}_{AB'_1} * S_{B'_1C} + \cdots + \tilde{t}_{AB'_k} * S_{B'_kC}\rangle$ を D に返して終了．ただし $B'_i = A$ の場合，$t_{AB'_i} * s_{B'_iC}$ は $s_{B'_iC}$ を，$\tilde{t}_{AB'_i} * S_{B'_iC}$ は $S_{B'_iC}$ を表すものとする．

ただし，個々のメッセージがどの要求の処理に直接関わるかを決定する手段が提供されていると仮定する．自分が C に対するトラスト計算セッションを開始したいなら，自分自身に要求 $\langle C, \lambda\rangle$ を送る (λ は空列)．

定理 4.1 基本プロトコルを用いて定められるトラスト計算手順を f とする．

1. f の実行中，プロトコルの参加者が応答の第一要素を決定するときにおいてのみ，上限仮定の範囲内で嘘をつくとする．そのとき，f によって計算されたトラストは健全．
2. プロトコルの参加者が嘘をつかないならば，f は安定．

参加者が嘘をつく場合，基本プロトコルを用いて定められるトラスト計算手順は安定ではない．嘘を含む場合の上界を別途与えることは可能だが，そのためには $\leq$ や演算子の基本性質だけでは不十分で，$*$ の具体的定義に依存した議論が必要となる．

4.2 プロトコルのプライバシ強化

前節のプロトコルでは，要求の第 2 要素 (要求が辿ってきた人の列) や応答の第 2 要素 (線形項) からメッセージの送受関係が分かるが，これはプライバシ漏洩となる恐れがある．しかし本稿の問題設定では，メッセージを直接送受した相手にある程度情報が漏れることは避けられない．そこで，それ以外の参加者の情報をできるだけ隠蔽することでプロトコルのプライバシ

を強化することを考える．さいわい列や項に対しては，要素や辺が出現するかどうかの簡単なチェックしかしていないので，それらをハッシュ値の集合に置き換えることで，同等な計算を行いつつ送受関係をある程度隠蔽できる．

各人 A には固有の識別子 id_A が定められているとする．さらに，H を衝突のない一方向性ハッシュ関数とし，$\|$ をビット連接とする．

要求 トラストを計算したい対象 C の識別子 id_C と，人 の識別子の ハッシュ値の集合 $\bar{P}$ の組 $\langle id_C, \bar{P}\rangle$.

応答 計算されたトラスト s と，人の識別子の対の ハッシュ値の集合 $\bar{S}$ の組 $\langle s, \bar{S}\rangle$.

A が D から $\langle id_C, \bar{P}\rangle$ という要求を受け取ったら，以下のことを順に行う．

1. $t_{AC} \neq 0$ ならば, 応答 $\langle t_{AC}, \{H(id_A \,\|\, id_C)\}\rangle$ を自分自身に送る．
2. 以下の 3 条件を満たす $B_1, \cdots, B_n$ を非決定的に選んで，要求 $\langle id_C, \bar{P} \cup \{H(id_A)\}\rangle$ を送る:
   - B_i は自分自身でも C でもない．
   - t_{AB_i} が 0 でない．
   - $H(id_{B_i})$ が $\bar{P}$ に含まれない．
3. 応答を待てるだけ待つ．
4. 得られた応答の中から，その第2要素である集合が互いに同じ要素を含まないよう $\langle s_{B'_1C}, \bar{S}_{B'_1C}\rangle, \cdots, \langle s_{B'_kC}, \bar{S}_{B'_kC}\rangle$ を非決定的に選び (ひとつも選ばなかったら，すぐに終了)，組 $\langle t_{AB'_1} * s_{B'_1C} + \cdots + t_{AB'_k} * s_{B'_kC}, \{H(id_A \,\|\, id_{B'_i}) \mid 1 \leq i \leq k \text{ かつ } B'_i \neq A\} \cup S_{B'_1C} \cup \cdots \cup S_{B'_kC}\rangle$ を D に返して終了．ただし $B'_i = A$ の場合，$t_{AB'_i} * s_{B'_iC}$ は $s_{B'_iC}$ を表すものとする．

自分が C に対するトラスト計算セッションを開始したいなら，自分自身に $\langle id_C, \emptyset\rangle$ を送る．基本プロトコルと同様な健全性と安定性が，プライバシ強化版でも成り立つ．

ハッシュを用いても，既知の識別子に関する情報の漏洩は避けられない．この問題に対しては，ハッシュ値の代わりにセッションで一意な乱数を生成する[2]といった改良が考えられる．

[2] $\bar{P}$ のチェックは生成した本人が行う．

5 おわりに

トラストを質と量の組で定式化し，その合成演算の代数的性質を明らかにした．さらに，トラスト計算の正しさを確率的仮定を用いず合成演算のみを用いて定式化し，分散計算のためのプロトコルの正しさを証明した．

合成の定義には，多値化などさまざまなバリエーションが考えられる．また，本稿では安定な計算手順として線形項を用いたものを提案したが，この問題は最大フロー問題とも深い関わりがある．これらの検討は今後の課題である．

参考文献

[1] R. Demolombe. Reasoning about trust: A formal logical framework. In *iTrust 2004*, pages 291–303, 2004.

[2] J. Huang and D. M. Nicol. A calculus of trust and its application to PKI and identity management. In *IDtrust 2009*, pages 23–37, 2009.

[3] A. Jøsang. A logic for uncertain probabilities. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 9(3):279–212, 2001.

[4] C.-J. Liau. Belief, information acquisition, and trust in multi-agent systems – a modal logic formulation. *Artif. Intell.*, 149(1):31–60, 2003.

[5] E. Lorini and R. Demolombe. From trust in information sources to trust in communication systems: An analysis in modal logic. In *KRAMAS 2008*, pages 81–98, 2008.

[6] G. Theodorakopoulos and J. S. Baras. Trust evaluation in ad-hoc networks. In *WiSe 2004*, pages 1–10. ACM, 2004.

[7] P. Venkat Rangan. An axiomatic basis of trust in distributed systems. In *IEEE S&P*, pages 204–211, 1988.

[8] 菊池 浩明. 信頼の数学モデルとセキュリティへの応用について. In **第 *24* 回ファジィシステム シンポジウム**, pages 393–397, 2008.