# A Unifying Theory of Electronic Money and Payment Systems

Ahto Buldas[1] (✉), Märt Saarepera[2], Jamie Steiner[3], and Dirk Draheim[4]

[1] Centre for Digital Forensics and Cyber Security,
Tallinn University of Technology,
Akadeemia tee 15a, 12618 Tallinn, Estonia
https://taltech.ee/en/
ahto.buldas@taltech.ee
[2] mart.saarepera@guardtime.com
[3] Guardtime, A. H. Tammsaare tee 60, 11316 Tallinn, Estonia
https://guardtime.com/
jamie.steiner@guardtime.com
[4] Information Systems Group,
Tallinn University of Technology,
Akadeemia tee 15a, 12618 Tallinn, Estonia
https://taltech.ee/en/
dirk.draheim@taltech.ee

**Abstract.** We present a general theory of payment systems that is capable of describing both traditional and electronic forms of payment. Starting from the three basic functions of money and general non-functional requirements, we derive the necessary and sufficient properties of technical implementations of money and payments. We describe possible scalable implementations of e-money schemes based on a general description of their data structures (money distributions) and payments. We define the notion of *bill scheme*, in which the value units are bills with invariant values, and show that only the bill scheme allows for scalable and practically efficient implementations through decomposition, where the components have to process a considerably smaller amount of data and fewer payment requests, compared to the overall system.

**Keywords:** e-money · e-cash · electronic payment systems · monetary system · digital euro · Fintech · Bitcoin · blockchain technology

## 1 Introduction

Money is a social phenomenon that makes trading between people and organizations more efficient and flexible. Without money, there would only be *barter transactions* – trading one good for another. This is inefficient, as a prerequisite to any trade is *double coincidence of wants* – existence of two parties that can provide goods or services the other party wants. Money appears as the subject of monetary systems and as the object of payment systems. A monetary system regulates the money supply. Governments are steering the money supply via a set of complex measures in a tiered, collateralized system [1,2] – in these endeavors they are supported by resp. team together with independent, legally trusted, accountable institutions [3,4]. Payment systems implement the distribution and exchange of money. They are large-scale systems that consist of organizational and technical measures [5]. Payment systems enable monetary systems; but must not be confused with them [1,2]. In the last decades, electronic payment systems have been crucial for the development of economies and societies. Currently, we see new forms of electronic payment systems emerging, as most obvious instances of *Fintech* [6,7,8], with a proclaimed potential for a next wave of e-commerce [9,10,11,12,13,14], or even with a proclaimed disruptive potential for our societies and monetary systems [15,16,17]. So it is not yet clear, in how far and to what extent such promises are realistic and might take off; it is clear that today's stack of monetary systems and payment systems is not flexible enough to cope with some concrete challenges. During the recent European refugee crisis, a concrete flaw of the existing system became clear: a person can make electronic payments only if he or she has a bank account.

Over the last few years, new, innovative payment systems have gone from being just the dreams of Tech Startups to being seriously discussed by governments and central banks. In her speech at the Bank of England Conference in September 2017, Christine Lagarde said: *"To be clear, this [virtual currencies] is not about digital payments in existing currencies – through Paypal and other 'e-money' providers such as Alipay in China, or M-Pesa in Kenya. Virtual currencies are in a different category, because they provide their own unit of account and payment systems. These systems allow for peer-to-peer transactions without central clearinghouses, without central banks. For now, virtual currencies such as Bitcoin pose little or no challenge to the existing order of fiat currencies and central banks. Why? Because they are too volatile, too risky, too energy intensive, and because the underlying technologies are not yet scalable. Many are too opaque for regulators; and some have been hacked. But many of these are technological challenges that could be addressed over time. Not so long ago, some experts argued that personal computers would never be adopted, and that tablets would only be used as expensive coffee trays. So I think it may not be wise to dismiss virtual currencies."* [18] This 2017 statement of Christine Lagarde addresses the potential disruptive nature of emerging electronic payment systems. For us, it is important that innovations in electronic payment systems do not necessarily have to be disruptive and still can add tremendous value to our economies and societies. In that vein, the European Central Bank (ECB) announced that it "intensifies its work on a digital euro" [19] and in the respective report on the digital euro [20], again from October 2020, it is stated: *"To ensure that consumers continue to have unfettered access to central bank money in a way that meets their needs in the digital age, the ECB's Governing Council decided to advance work on the possible issuance of a digital euro – an electronic form of central bank money accessible to all citizens and firms. A digital euro would be introduced alongside cash, it would not replace it."* [20], p. 2, compare also with [21].

Given these current developments, now is the time to develop a deeper, formal understanding of payment systems, in general, and electronic payment systems, in particular. This deeper understanding is essential for successfully addressing the critical challenge of any future electronic payment systems: scalability! This paper aims to provide exactly this deeper understanding. The work follows in a tradition of many contributions by Guardtime and Tallinn University of Technology to the understanding of ultra-scalable document verification infrastructures [22,23,24,25,26], compare also with [27,28,29]. The theory of payment systems that we build in this paper, relies on the following basic, simple observation. Money has three main functions:

- *Unit of account* – money is a measure of value that can be applied to all items, thus simplifying the accounting process.
- *Store of value* – money helps to preserve values over time. For example, producing goods and saving them in warehouses is not necessarily a good way of storing value, because the goods' value depreciate over time.
- *Medium of exchange* – money acts as intermediary between buyers and sellers. Instead of selling goods for other goods, the seller sells goods for money which can later be used to buy other goods.

An object is more suitable to be used as money if it has properties that allow it to perform these functions. These are not the only properties that should be considered when designing a money system, but they are the most fundamental ones. Various different forms of money are in use, and these different forms work in fundamentally different ways, i.e., there are different *money schemes*. We are not aware of any prior work that attempts to formally describe the requirements and properties of money schemes. This is probably because, in human history, money schemes arose naturally, out of a desire to facilitate certain types of desirable transactions, rather than from a deliberate process of design.

The emergence of electronic money has considerably increased the number of different money schemes in use. To implement electronic money, one must solve a different set of technical problems than for physical money. For example, in ancient times when seashells were used as money, double spending could not be a problem, whereas for digital money it is. The history of banking began a few thousand years ago. Banks introduced accounts as a new type of monetary units, and the banker's job was to keep track of the value of this monetary unit for each of their depositors. Electronic bank money systems that emerged during the era of mainframe computers in the implementation of electronic accounting systems, where the accounts are just numbers stored in a bank's computer database. One of the biggest challenges in electronic account systems has been the settlement of inter-bank transactions, where atomic swap operations are required.

More recently, blockchain [30] money schemes (cryptocurrency) such as Bitcoin [31] introduce new types of monetary units – electronic coins – that offer much more flexible types of payments that may involve several monetary units. A payment may involve creating several new coins while destroying existing coins. Other schemes such as Ethereum [32] offer universal programmable money implemented as smart contracts [33,34] that enable its users to associate a payment with arbitrary verifiable logical preconditions. Blockchain money schemes introduced these new possibilities, but also created new fundamental problems, especially those related to efficient and scalable implementation.

One of the most important requirements for an economy-wide money scheme is that it is capable of supporting a sufficient level of transactions for a long future period. Unfortunately, it is difficult to foresee what volume of transactions may be required for future economies. Therefore, it is important to know if a money scheme can scale as needed in the future. A future-proof electronic money scheme should be derived from a deliberate process of design, starting from fundamental principles that will ensure its scalability.

A new theory is needed to study the essential and most general properties of money schemes in order to understand if some of them can be easily scaled while others cannot. In this paper, we will derive what a money scheme is composed of. Next, we will derive, from those compositional elements, the minimal, yet sufficient properties of any money scheme that are required for it to perform these three basic functions of money and support a volume of transactions that can be easily expanded. By doing this in a systematic way, we will enumerate the full set of possible money schemes. Our aim is to present an abstract mathematical model for describing money schemes that allows one to draw concrete conclusions about their potential for implementation and, in particular, their scalability.

We proceed as follows. In Sect. 2, we investigate fundamental notions of money distribution and redistribution. Based on this, we are able to formalize the dynamics of money and payments in Sects. 3 and Sect. 4. Section 5 takes the theory a significant step further, i.e., from composing payments of a single payment system to the composition of whole payment systems and their interplay. In Sect. 6, we walk through some example money schemes to illustrate the applicability of the contributed theory. In Sect. 7, we exploit the theory to provide an exhaustive classification of all possible money schemes. In Sect. 8, we study a more general notion of decomposability related to full parallel decomposition of state machine, and prove some results concerning the non-decomposability of certain schemes. In Sect. 9, we extend the non-decomposability results to a weaker notion of decomposition. We conclude in Sect. 10.

## 2 Money Distribution and Redistribution

There are several different money schemes in use. Physical cash is represented as physical coins or bills that are marked with values, and can be given in payment. Bank money is represented by an account which has a balance representing the upper limit of value that the account can be exchanged for. Bitcoin and similar money is represented by Unspent Transaction Outputs (UTXOs) in Bitcoin's ledger [35], which can be assigned, in parts, to one or more public keys. All of these schemes share some basic properties. For example, they use some kind of numerical measure that describes the amount of money – its *monetary value*. This is the basic property which allows money in the scheme to function as a unit of account.

An implemented money scheme can be modeled as a system with users. In this system, there is a function, $m(a)$, that describes the amount of money each user $a$ has. Payments in this model are changes to the function $m$. It seems obvious that such a function is necessarily a part of any mathematical model of a money scheme.

However, a single function model is not rich enough to describe how a money scheme can, in practice, be implemented. Since every mathematical model of a money scheme must at least describe such a function $m$, all money schemes would look exactly the same. This means that a function $m$ is necessary, but not sufficient to describe different money schemes.

Our first goal is to find a model that is, on one hand, rich enough to describe implementation aspects. On the other hand, the model has to be simple enough to describe only the most fundamental aspects required to implement the scheme.

A useful observation about existing money schemes is that they all have some kind of *monetary units* that are physical or digital representations of money. Examples are bills, coins, bank accounts, Bitcoin UTXOs,

etc. Every monetary unit has a unique monetary value and a unique bearer – the owner of that monetary unit. Monetary units, while often fungible, are distinguishable. They may have some kind of identifiers, such as the serial number on a bill or a bank account number. They may also be distinguishable because of being separate physical objects, such as coins.

In such a model, the state of money scheme – the so called *money distribution* – is represented by a set $U$ of value units, where every value unit $u \in U$ has a unique monetary value $\nu(u)$ and a unique bearer $\beta(u)$ which represent the user of the system, to whom the money belongs. So, instead of describing a state of the money scheme by a single money function, this new model uses one set and two functions. The two-function model is robust enough to describe how the money distribution can change in the money scheme, i.e., through *payments*.

It turns out that this minor extension of the one-function model is sufficient to study the implementation aspects that affect scalability and show that different money schemes may have dramatically different scalability limitations. These conclusions may be derived in a fundamental manner, and no specific implementation details or techniques may overcome them.

### 2.1 Representation of Money and its Distribution

Following the discussion above, a *money distribution* $M$ involves the following components:

- $U$ is the set of *monetary units*
- $\nu\colon U \to \mathbb{N}$ is the *value function* defining the value $\nu(u)$ of every value unit $u$. The set $\mathbb{N}$ is the set of all natural numbers, but instead, we can use any set of numerals that is totally ordered (e.g. integers, real numbers).
- $\beta\colon U \to \mathfrak{B}$ is the *bearer function* defining the bearer $\beta(u)$ of a unit. The set $\mathfrak{B}$ is the set of possible bearers. The bearer is usually a legal construction defining any type of legal entity, such as a person, a family, a company, a state institution, etc.

Hence, the money distribution $M$ defines monetary units, their values, and their bearers. A schematic view of a money distribution is depicted in Fig. 1.

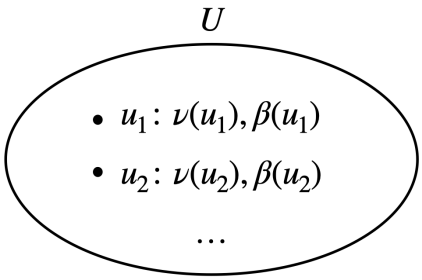


**Fig. 1.** Schematic representation of money distribution $M = (U, \nu, \beta)$.

**Definition 1 (Money Distribution).** A *money distribution* on a bearer set $\mathfrak{B}$ is a triple $M = (U, \nu, \beta)$, where $U$ is a set, $\nu\colon U \to \mathbb{N}$ and $\beta\colon U \to \mathfrak{B}$ are functions, called the *value function* and the *bearer function*, respectively.

We use the indexed representations $M = (U, \nu, \beta) = (U_M, \nu_M, \beta_M)$ to emphasize that these are the components of $M$.

**Definition 2 (Total Value of a Money Distribution).** The *total value* of a money distribution $M = (U, \nu, \beta)$ is the natural number $\sigma(M) = \sum_{u \in U} \nu(u)$.

We only consider money distributions with finite total value.

We define the *money of bearer* $b$ in a money distribution $M$ as the amount of money that $b$ owns in $M$.

**Definition 3 (Money of Bearer).** By the *money of bearer* $b$ in a money distribution $M$ we mean the number $\sigma(M, b) = \sum_{u \in \beta^{-1}(b)} \nu(u)$, where $\beta^{-1}(b) = \{u \colon u \in U,\ \beta(u) = b\}$ is the inverse image of $b$ under $\beta$.

**Definition 4.** By $\mathbb{M}$ we denote the set of all possible money distributions $M$. By $\mathbf{0}_{\mathbb{M}}$, we denote the empty money distribution $\mathbf{0}_{\mathbb{M}} = (\emptyset, \emptyset, \emptyset)$.

### 2.2 Transformation of Money

Money transformations represent changes in the money distribution. If the original money distribution is $M = (U, \nu, \beta)$ and the transformation is $R$, then the changed (transformed) money distribution is $R(M) = M' = (U', \nu', \beta')$. A transformation may change the values and bearers of monetary units. It may also destroy (melt) value units and create (mint) new value units.

**Definition 5 (Money Transformation).** A *money transformation* $T$ is a partial transformation on $\mathbb{M}$. (As usual, we use $\operatorname{dom} T$ and $\operatorname{range} T$ for the domain resp. the range of $T$).

We use $1_{\mathbb{M}}$ to denote the identity mapping, which represents *no change* to the money distribution. Additionally, there is a function which might transform the money distribution, but is defined nowhere; ie. its domain includes no actual bearers, which means its practical effect is nothing. This transformation $\Theta$ with domain $\operatorname{dom}(\Theta) = \emptyset$ is also a partial transformation.

Note that the domain $\operatorname{dom}(T)$ may be a singleton set $\{M\}$, which means that the value of $T(M)$ is only defined for a single money distribution $M$.

The money transformations on $\mathbb{M}$ form a *monoid* under the composition operation:

- Composition $T_1 \circ T_2$ of two money transformations yields a money transformation.
- Composition is associative: $T_1 \circ (T_2 \circ T_3) = (T_1 \circ T_2) \circ T_3$.
- The identity function $1_{\mathbb{M}}$ is a partial transformation.
- $\Theta$ is the zero element of the monoid, i.e., $\Theta \circ T = T \circ \Theta = \Theta$ for every redistribution $T$.

A transformation that preserves the total money, is called a *redistribution*, see Def. 6.

**Definition 6 (Redistribution).** A *redistribution* $R$ is a money transformation so that $\sigma(R(M)) = \sigma(M)$ for every $M \in \operatorname{dom} R$.

**Definition 7 (Initial Emission).** A transformation $E_0$ defined on the empty money distribution (i.e., $\operatorname{dom} E_0 = \{\mathbf{0}_{\mathbb{M}}\}$) that transforms $\mathbf{0}_{\mathbb{M}}$ to a non-empty money distribution $M_0 = E_0(\mathbf{0}_{\mathbb{M}})$, is called *initial emission*.

## 3 Dynamics of Money

At any moment of time, the amount of money and its distribution is defined by the money distribution. Changes in the money distribution are caused by input events, which we call *redistributions*. Eventually, we are interested only in such systems where redistributions are caused by payments, which is to say that we will not consider transformations that change the *overall quantity* of money in the system.

The money distribution and its redistribution over time is what we call *money evolution*, and is represented by a pair $(M(t), R(t))$ of mutually related functions, where $M(t)$ represents the money distribution at time $t$, and $R(t)$ represents the redistribution that transforms some initial money distribution $M_0$ to the current distribution $M(t)$.

Our goal is to study physical implementations of money via $M(t)$ and $R(t)$ as a system. First, we study the relation between $M(t)$ and $R(t)$ and how they are related to the implementation. Intuitively, $M(t)$ represents the state of the system, while $R(t)$ represents the input that causes changes in that state. We observe that in nature similar situation is modeled via differential equations. First, we look at the simplest physical concept – point mass – to look for useful analogies and see if we may apply our existing intuition the task of examining different money schemes.

### 3.1 Descriptions of $R(t)$ and $M(t)$

Money evolutions $(M(t), R(t))$ are not represented by continuous functions but rather *piecewise constant functions* also known as *step functions*, where the change of $M(t)$ happens at a discrete set $T = \{t_1, t_2, t_3, \ldots\}$ of time values $0 = t_0 < t_1 < t_2 < t_3 \ldots$ as depicted in Fig. 2. The function $M(t)$ is represented as a sequence of pairs $(M_0, t_0), (M_1, t_1), (M_2, t_2), (M_3, t_3), \ldots$ and is defined by this sequence as follows:

$$M(t) = M_i,$$

where $i \in \{0, 1, \ldots\}$ is the first index for which $t_{i+1} > t$.

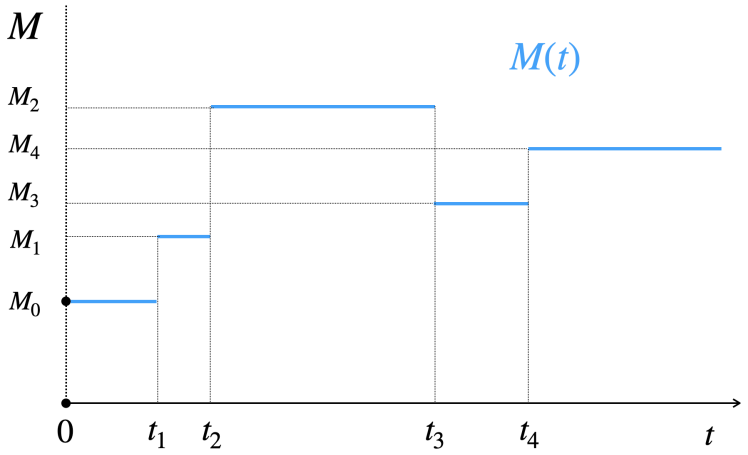


**Fig. 2.** Change of money distribution.

Fig. 3 depicts the corresponding redistribution function $R(t)$, which is $1_{\mathbb{M}}$ everywhere except at the points $t_1, t_2, t_3$, and $t_4$, where redistributions $R_1, R_2, R_3$ happen.

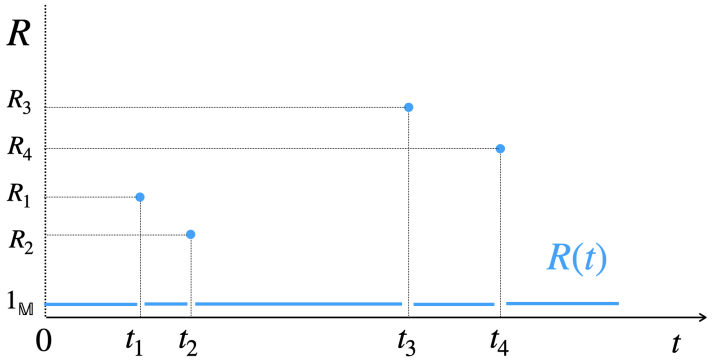


**Fig. 3.** Redistribution function.

Let $(R_1, t_1), (R_2, t_2), (R_3, t_3), \ldots$ be the sequence of non-trivial $(R_i \neq 1_{\mathrm{dom}(R_i)})$ redistributions such that for every $i$, the redistribution is assumed to happen at time $t_i$.

The redistribution function is then defined as follows:

$$R(t) = \begin{cases} R_i \text{ if } t = t_i \in T \\ 1_{\mathbb{M}} \text{ if } t \notin T \end{cases} \tag{1}$$

It is easy to see that the functions $M(t)$ and $R(t)$ are related in the following way:

$$M(t) = (R_1 \circ R_2 \circ \ldots \circ R_n)(M_0) = R_n(R_{n-1}(\ldots R_2(R_1(M_0))\ldots)), \tag{2}$$

where $n$ is the largest natural number such that $t_n \leqslant t$.

From equation (2), it follows that $M(t)$ is uniquely defined by $M_0$ and $R(t)$.

However, it is not yet clear how precisely $M_0$ and $R(t)$ are defined by $M(t)$.

The uniqueness of this correspondence depends on how we restrict the properties of redistributions $R_1, R_2, \ldots$. First, in Sect. 3.2, we prove the unique correspondence between $R(t)$ and $M(t)$ for the so-called uni-point redistributions – partial transformations the domain of which is a singleton set, i.e., these transformations are defined for only one initial money distribution. The main motivation for this approach is the simplicity of the proof. In Sect. 3.3, we generalize the result for the so-called shift redistributions that much more precisely model real life payments.

### 3.2 Uni-Point Redistributions

A uni-point redistribution $R$ on $\mathbb{M}$ is defined for *only* a single money distribution $M$. This is equivalent to saying "$A$ has \$10, and $B$ has \$2, and the redistribution makes it so that $A$ has \$8 and $B$ has \$4."

**Definition 8 (Uni-Point Redistribution).** A redistribution $R$ is called *uni-point* if its domain is a singleton set, i.e., $\operatorname{dom} R = \{M\}$ for a certain money distribution $M$.

Therefore, $R$ is completely described as a pair $(M, M')$, where $M' = R(M)$.

**Lemma 1.** *Assume that:*

- *$M(t)$ is represented by the sequence $(M_0, t_0), (M_1, t_1), (M_2, t_2), (M_3, t_3), \ldots$*
- *$R(t)$ is represented by the sequence $(R_1, t_1), (R_2, t_2), (R_3, t_3), \ldots$*
- *$R'(t)$ is represented by the sequence $(R'_1, t'_1), (R'_2, t'_2), (R'_3, t'_3), \ldots$*

*where $R_i$ and $R'_i$ are uni-point redistributions. Now, if:*

- *$M(t)$ and $R(t)$ satisfy equation (2), i.e., $M(t) = (R_1 \circ R_2 \circ \ldots \circ R_n)(M_0)$, where $n$ is the largest natural number such that $t_n \leqslant t$ and*
- *$M(t)$ and $R'(t)$ satisfy equation (2), i.e., $M(t) = (R'_1 \circ R'_2 \circ \ldots \circ R'_n)(M_0)$, where $n$ is the largest natural number such that $t'_n \leqslant t$*

*then $R(t) = R'(t)$, i.e., $R_1 = R'_1, R_2 = R'_2, \ldots$, and $t_1 = t'_1, t_2 = t'_2, \ldots$.*

**Proof:** Assume that $R(t) \neq R'(t)$ and $k$ be the smallest index such that $(R_k, t_k) \neq (R'_k, t'_k)$, i.e. $(R_1, t_1) = (R'_1, t'_1), \ldots, (R_{k-1}, t_{k-1}) = (R'_{k-1}, t'_{k-1})$. First, we show that $t_k = t'_k$. Indeed, if $t_k < t'_k$, then for $t_k < t < t'_k$, we have that, on one hand, $M(t) = (R_1 \circ R_2 \circ \ldots \circ R_k)(M_0)$, but on the other hand, $M(t) = (R'_1 \circ R'_2 \circ \ldots \circ R'_{k-1})(M_0)$. This implies:

$$\begin{aligned} M(t) &= (R_1 \circ R_2 \circ \ldots \circ R_k)(M_0) \\ &= R_k((R_1 \circ R_2 \circ \ldots \circ R_{k-1})(M_0)) \\ &= R_k((R'_1 \circ R'_2 \circ \ldots \circ R'_{k-1})(M_0)) \\ &= R_k(M(t)), \end{aligned}$$

which means that $R_k$ as a uni-point redistribution is trivial, which is a contradiction. If again $t'_k < t_k$, then $R'_k$ would be trivial and we have a similar contradiction. Hence, $t_k = t'_k$.

If $R_k \neq R'_k$, then for $t_k < t < \min t_{k+1}, t'_{k+1}$ and $M_{k-1} = M(t_{k-1})$ , we have $M(t) = R_k(M_{k-1}) \neq R'_k(M_{k-1}) = M(t)$, which again is a contradiction. □

### 3.3 Shift Redistributions

Uni-point redistributions are not good models for real-world payments as they are too restrictive, i.e., defined only for one particular money distribution and their description – the pair $(M, M')$ – involves the entire money distribution that is currently valid. Real-world payments, in contrast, tend to describe relatively small "local" changes in the money distribution, and can be represented in a much more compact form. Payments usually change just a few monetary units, and this is done independently of the other units in $M$. Hence, redistributions $R$ should be defined for many money distributions $M$ and hence, $\mathrm{dom}(R)$ is not a singleton set. A redistribution $R$ has to make the same relative changes in all money distributions $M \in \mathrm{dom}(R)$. For example, a payment such as "$A$ pays $B$ \$10" can be applied to any money distribution $M$ where $A$ is the bearer of at least ten dollars in $M$ and must decrease the money of $A$ by \$10 and increase the money of $B$ by \$10 in every such $M$ and do nothing else, no matter how much money other parties have in $M$.

Even though they are more similar, shift redistributions are still not exactly the same as payments. A payment is an economic term that describes monetary value exchanged for goods or services. A shift redistribution describes a transformation to the money distribution. For example, a bitcoin block acceptance is a shift redistribution that contains many independent payments. Likewise in the current financial system, there are end of day settlement procedures, which take into account many individual payments, some of which completely or partially offset each other, and simply apply the net effect of all payments.

The changes made by $R$ can be reconstructed if $M$ and $R(M)$ are known for a particular $M$, i.e., $R$ is completely defined if just one argument-value pair $(M, R(M))$ is known.

Real-valued functions of this type are called *shift functions*. An example of such a function is $f_\delta(x) = x+\delta$, where $\delta$ is constant. If one knows $(x, f_\delta(x))$ for an $x$, the value $\delta$ can be computed by

$$\delta = f_\delta(x) - x, \tag{3}$$

and hence the function $f_\delta$ is uniquely defined by any pair $(x, f_\delta(x))$.

Inspired by this analogy, we define shift redistributions $R_\Delta$, where $\Delta$ is called a *difference set* that describes the differences between $R_\Delta(M)$ and $M$. The difference set $\Delta$ is not itself a money distribution. We also define the subtraction operation $\ominus$ on money distributions such that the equation

$$\Delta = R_\Delta(M) \ominus M$$

analogous to (3) holds (Lemma 2).

In order to describe local changes that a redistribution $R$ does, we need to define:

- The set $U^-$ of monetary units that are deleted by $R$, and for each such unit $u \in U^-$ we have to list its value and bearer before applying $R$, i.e., we have to describe a function $\Delta^-\colon U^- \to \mathbb{N} \times \mathfrak{B}$.
- The set $U^+$ of new monetary units that are created by $R$, and for each such unit $u \in U^+$ we have to list its value and bearer after applying $R$, i.e., we have to describe a function $\Delta^+\colon U^+ \to \mathbb{N} \times \mathfrak{B}$.
- The set $U^0$ of monetary units $u$ the parameters $\nu(u)$ and $\beta(u)$ of which are changed by $R$, and for each such unit we have to list its value change (positive or negative), as well as its previous and current bearers, .e. we have to describe a function $\Delta^0\colon U^0 \to \mathbb{Z} \times \mathfrak{B} \times \mathfrak{B}$. For the compactness of representation, the units that are not changed by $R$ should not belong to $U^0$.

A difference set should describe all these changes and hence it has the next mathematical definition.

**Definition 9 (Difference Set).** A *difference set* $\Delta$ is a nested tuple $\langle\langle U^-, U^+, U^0\rangle, \langle\Delta^-, \Delta^+, \Delta^0\rangle\rangle$, where:

- $U^-$, $U^+$, and $U^0$ are non-intersecting sets of monetary units.
- $\Delta^-\colon U^- \to \mathbb{N} \times \mathfrak{B}$ is a total function.
- $\Delta^+\colon U^+ \to \mathbb{N} \times \mathfrak{B}$ is a total function.
- $\Delta^0\colon U^0 \to \mathbb{Z} \times \mathfrak{B} \times \mathfrak{B}$ is a total function so that for every $u \in U^0$, if $\Delta^0(u) = (d_u, b_u, b'_u)$, then $d_u \neq 0$ or $b_u \neq b'_u$.

**Definition 10 (Domain of a Difference Set).** The set $U^- \cup U^0$ is called the *domain* of $\Delta$ and is denoted by $\operatorname{dom} \Delta$.

**Definition 11 (Creation Set of a Difference Set).** The set $U^+$ is called the *creation* of $\Delta$ and is denoted by $\operatorname{cre} \Delta$.

Every difference set $\Delta$ uniquely defines a redistribution $R_\Delta$, see Def. 12.

**Definition 12 (Shift Redistribution $R_\Delta$).** Given a difference set $\Delta$A, a *shift redistribution* $R_\Delta$ is defined by as follows. (i) The domain $\operatorname{dom}(R_\Delta)$ of $R_\Delta$ is the set of all money distributions $M = (U, \nu, \beta)$ such that:

D0: $U^- \cup U^0 = \operatorname{dom} \Delta \subseteq U$
D1: $U^+ \cap U = \emptyset$
D2: $\forall u \in U^- \colon \nu(u) = n_u, \beta(u) = b_u$, where $\Delta^-(u) = (n_u, b_u)$.
D3: $\forall u \in U^0 \colon \nu(u) + d_u \geqslant 0, \beta(u) = b_u$, where $\Delta^0(u) = (d_u, b_u, b'_u)$.

(ii) For every $M = (U, \nu, \beta) \in \operatorname{dom}(R_\Delta)$, we define $R_\Delta(M) = M' = (U', \nu', \beta')$ as follows:

R0: $U' = (U \backslash U^-) \cup U^+$
R1: For every $u \in U'$, if $u \in U \backslash U^-$, then
- If $u \in U^0$ then $\nu'(u) = \nu(u) + d_u$ and $\beta'(u) = b'_u$, where $\Delta^0(u) = (d_u, b_u, b'_u)$.
- If $u \notin U^0$ then $\nu'(u) = \nu(u)$ and $\beta'(u) = \beta(u)$.

R2: If $u \in U^+$, then $\nu'(u) = n_u$ and $\beta'(u) = b_u$, where $\Delta^+(u) = (n_u, b_u)$.

**Definition 13 (Difference of Money Distributions).** The *difference* $M' \ominus M$ of two money distributions $M = (U, \nu, \beta)$ and $M' = (U', \nu', \beta')$ is a difference set $\langle\langle \mathbf{U}^-, \mathbf{U}^+ \rangle, \langle \mathbf{U}^0; \Delta^-, \mathbf{\Delta}^0, \mathbf{\Delta}^+ \rangle\rangle$ defined as follows:

$$
\begin{aligned}
\mathbf{U}^- &= U \backslash U' \\
\mathbf{U}^+ &= U' \backslash U \\
\mathbf{U}^0 &= \{u \in U' \cap U \colon \nu'(u) \neq \nu(u) \text{ or } \beta(u) \neq \beta'(u)\} \\
\mathbf{\Delta}^-(u) &= (\nu(u), \beta(u)) \quad \text{for every } u \in \mathbf{U}^- \\
\mathbf{\Delta}^+(u) &= (\nu'(u), \beta'(u)) \quad \text{for every } u \in \mathbf{U}^+ \\
\mathbf{\Delta}^0(u) &= (\nu'(u) - \nu(u), \beta(u), \beta'(u)) \quad \text{for every } u \in \mathbf{U}^+
\end{aligned}
$$

A schematic view of of the sets $\mathbf{U}^-, \mathbf{U}^+, \mathbf{U}^0$ is depicted in Fig. 4.

From the above, we can say that for a shift redistribution, all monetary units in $U$ are either newly created, newly destroyed, having their value or bearer changed, or are totally unchanged. Since the overall *amount* of money in the money distribution remains unchanged in a shift redistribution, we can observe that there are different ways of accomplishing the shift redistribution, using the above categories, that can effect the same payments in different ways, depending on how the money scheme works. For example, in the case of paper or coin money, the value of individual bills do not change, but the bearer changes. In the case of a bank account payment, the value of both accounts changes – the payer's account decreases in value, which is offset by a corresponding increase in the recipients account value. In a bitcoin transaction, UTXO's having a certain total value are destroyed, while new UTXO's with potentially different values and bearers are created. The new UTXO's total value equals the sum of those that are destroyed.

**Lemma 2.** *For every shift redistribution $R_\Delta$ and for every money distribution $M \in \operatorname{dom}(R_\Delta)$, we have that*

$$\Delta = R_\Delta(M) \ominus M \tag{4}$$

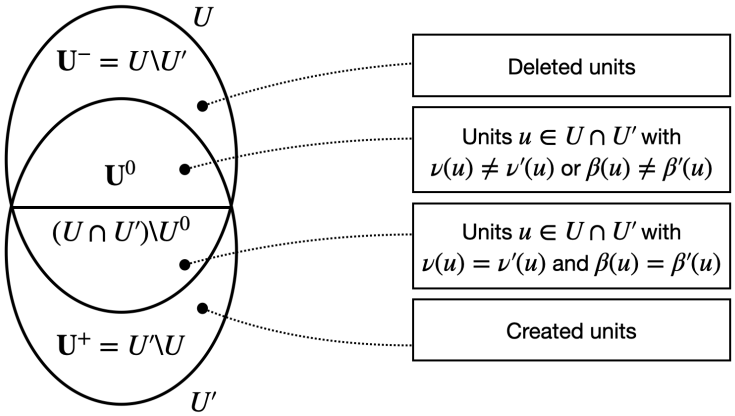


**Fig. 4.** Schematic representation of the sets $\mathbf{U}^-, \mathbf{U}^+, \mathbf{U}^0$ in the difference set $M' \ominus M$ for $M = (U, \nu, \beta)$ and $M' = (U', \nu', \beta')$.

**Proof:** Let $M = (U, \nu, \beta) \in \mathrm{dom}(R_\Delta)$ and $R_\Delta(M) = M' = (U', \nu', \beta')$. We have to prove the equalities $\mathbf{U}^- = U^-$, $\mathbf{U}^+ = U^+$, $\mathbf{U}^0 = U^0$, $\mathbf{\Delta}^- = \Delta^-$, $\mathbf{\Delta}^+ = \Delta^+$, and $\mathbf{\Delta}^0 = \Delta^0$.
$\underline{\mathbf{U}^- = U^-}$: By (R0), we have $U' = (U \backslash U^-) \cup U^+$ by definition of $R_\Delta$. Then, as $U^- \subseteq U$ and $U^+ \cap U^- = \emptyset$:

$$\mathbf{U}^- = U \backslash U' = U \backslash [(U \backslash U^-) \cup U^+] = [U \backslash (U \backslash U^-)] \backslash U^+ = U^- \backslash U^+ = U^-.$$

$\underline{\mathbf{U}^+ = U^+}$: As by (D1), $U^+ \cap U = \emptyset$. Hence:

$$\mathbf{U}^+ = U' \backslash U = [(U \backslash U^-) \cup U^+] \backslash U = \underbrace{[(U \backslash U^-) \backslash U]}_{\emptyset} \cup \underbrace{(U^+ \backslash U)}_{U^+} = U^+.$$

$\underline{\mathbf{U}^0 = U^0}$: If $u \in \mathbf{U}^0$, then $u \in U' \cap U$ and either $\nu'(u) \neq \nu(u)$ or $\beta(u) \neq \beta'(u)$ and hence, $u \notin U^+$ by (D1) in the definition of $\mathrm{dom}(R_\Delta)$. Therefore, $u \in U \backslash U^-$. If now $u \notin U^0$, then $\nu'(u) = \nu(u)$ and $\beta'(u) = \beta(u)$ by (R1) in the definition of $R_\Delta$ and we have a contradiction. Therefore, $u \in U^0$, and hence $\mathbf{U}^0 \subseteq U^0$.

Let now $u \in U^0$. Then $u \notin U^+$, $u \notin U^-$, and by (D0), $u \in U$. Then, $u \in U'$ by (R0), and hence, $u \in U^0 \cap U'$. Therefore, by (R1), we have $\nu'(u) = \nu(u) + d_u$ and $\beta'(u) = b'_u$, where $\Delta^0(u) = (d_u, b_u, b'_u)$. By the definition of $\Delta^0$, we have $d_u \neq 0$ or $b_u \neq b'_u$, which again implies that either $\nu'(u) \neq \nu(u)$ or $\beta(u) \neq \beta'(u)$ and hence $u \in \mathbf{U}^0$ by the definition of $\mathbf{U}^0$. Hence, $U^0 \subseteq \mathbf{U}^0$.
$\underline{\mathbf{\Delta}^- = \Delta^-}$: Let $u \in U^-$ and $\mathbf{\Delta}^-(u) = (n_u, b_u)$. Hence, $n_u = \nu(u)$ and $b_u = \beta(u)$, and by (D2), $\Delta^-(u) = (\nu(u), \beta(u)) = (n_u, b_u) = \mathbf{\Delta}^-(u)$.
$\underline{\mathbf{\Delta}^+ = \Delta^+}$: Let $u \in U^+$ and $\mathbf{\Delta}^+(u) = (n_u, b_u)$. Hence, $n_u = \nu'(u)$ and $b_u = \beta'(u)$, and by (R2), $\Delta^+(u) = (\nu'(u), \beta'(u)) = (n_u, b_u) = \mathbf{\Delta}^+(u)$.
$\underline{\mathbf{\Delta}^0 = \Delta^0}$: Let $u \in U^0$ and $\mathbf{\Delta}^0(u) = (\overline{d}_u, \overline{b}_u, \overline{b'}_u)$. Hence, $\overline{d}_u = \nu'(u) - \nu(u)$, $\overline{b}_u = \beta(u)$ and $\overline{b'}_u = \beta'(u)$. Let $\Delta^0(u) = (d_u, b_u, b'_u)$. By (D3), $\beta(u) = b_u$, and hence $\overline{b}_u = b_u$. As $u \in U^0$, then $u \in U'$ and $u \in U \backslash U^-$. Hence, by (R1), we have $\nu'(u) = \nu(u) + d_u$ and $\beta'(u) = b'_u$. Hence, $\overline{b'}_u = b'_u$ and $\overline{d}_u = \nu'(u) - \nu(u) = d_u$. Hence, $\Delta^0(u) = (\overline{d}_u, \overline{b}_u, \overline{b'}_u) = \mathbf{\Delta}^0(u)$. □

**Corollary 1.** *For every uni-point redistribution represented by a pair $(M, M')$, there is one and only one shift redistribution $R$ with $M' = R(M)$.*

**Theorem 1.** *Assume that:*

- *$M(t)$ is represented by the sequence $(M_0, t_0), (M_1, t_1), (M_2, t_2), (M_3, t_3), \ldots$*
- *$R(t)$ is represented by the sequence $(R_1, t_1), (R_2, t_2), (R_3, t_3), \ldots$*

– $R'(t)$ *is represented by the sequence* $(R'_1, t'_1), (R'_2, t'_2), (R'_3, t'_3), \ldots$

*where* $R_i$ *and* $R'_i$ *are shift redistributions. Then if:*

– $M(t)$ *and* $R(t)$ *satisfy equation (2), and*
– $M(t)$ *and* $R'(t)$ *satisfy equation (2),*

*then* $R(t) = R'(t)$*, i.e.,* $R_1 = R'_1, R_2 = R'_2, \ldots$*, and* $t_1 = t'_1, t_2 = t'_2, \ldots$.

**Proof:** Direct implication from Lemma 1 and Lemma 2. □

### 3.4 Velocity of Money

**Definition 14 (Money Flow).** Given two money distributions $M$ and $M'$ on a bearer set $\mathfrak{B}$, we define the *money flow* between $M$ and $M'$ as the following quantity:

$$\varphi(M', M) = \frac{1}{2} \sum_{b \in \mathfrak{B}} |\sigma(M', b) - \sigma(M, b)| \tag{5}$$

**Definition 15 (Money Flow of a Redistribution).** Given a redistribution $R$ and a money distribution $M \in \operatorname{dom} R$, the *money flow* $\psi(R)$ *of the redistribution* $R$ is the quantity:

$$\psi(R) = \varphi(R(M), M) \tag{6}$$

Let $(M(t), R(t))$ be a money evolution represented by initial money distribution $M_0$ and a time series $(R_1, t_1), (R_2, t_2), (R_3, t_3), \ldots$. Let $\sigma_0 = \sigma(M)$ be the total value of money distributions.

**Definition 16 (Velocity of Money in a Money Evolution).** By the *velocity of money* in the interval $[t_0, t]$, where $t_0 < t$, we mean a function $V(t_0, t)$ defined as follows:

$$V(t_0, t) = \frac{1}{\sigma_0 \cdot (t - t_0)} \sum_{t_0 < t_i < t} \psi(R_i). \tag{7}$$

Note that for the definition having a practical value, the distance $|t - t_0|$ should not be too small, because for a point $t_i$ in the time series the flow $V(t_i - \epsilon, t_i + \epsilon)$ approaches to infinity, if $\epsilon$ approaches to 0. Usually, $|t - t_0|$ is assumed to be a fixed unit of time such as year, month, day, etc.

We observe that the form of this velocity equation bears strong resemblance to the usual definition of money velocity seen in economic literature:

$$V = \frac{P \cdot T}{\sigma_0}. \tag{8}$$

In the above definition, $T$ represents the volume of transactions in the economy during a certain period, which is equivalent to the sum in our derivation. Additionally, the denominator, $\sigma_0$ represents the volume of money in circulation in both definitions. A notable difference is the presence of $P$ in the economic definition, which ours lacks. The price level can be seen as a function that maps the amount of monetary units to goods and services. Since we only consider the mechanics of money schemes, specifically, and not the goods and services they are exchanged for, this is a reasonable result. Additionally, since many theories of money examine the effect of increasing and decreasing the monetary supply on prices, and here we have mainly considered cases where the overall monetary supply remains fixed, derivation of a price function will be left as future work.

### 3.5 Emission and Withdrawal

In most practical money solutions, the total amount of money is not constant, but changes from time to time because of emissions and withdrawals. For example, this is the case with Central Bank controlled money.

Therefore, in the mathematical model of money, we also have to consider emission and withdrawal transformations. The time series of the system is in the form $(T_1, t_1), (T_2, t_2), \ldots$, where $T_i$ is a transformation of one of the following types:

- *Redistribution* $R_i$ that does not change the total amount $\sigma_0$ of money.
- *Emission* $E_i$ that:
  1. Extends the total amount $\sigma_0$ of money
  2. Deletes no monetary units
  3. Changes the bearer of no monetary units
  4. Reduces the value of no monetary units
- *Withdrawal* $W_i$ that:
  1. Reduces the total amount $\sigma_0$ of money
  2. Creates no monetary units
  3. Changes the bearer of no monetary units
  4. Extends the value of no monetary units

The mathematical model of money with emission and withdrawal is similar to the theory of invariant money. An emission or withdrawal transformations $T$ can also be modeled as shift transformation that are uniquely defined by the difference set $T(M) \ominus M$ for any money distribution $M$.

The unique correspondence theorem (Theorem 1) also holds for general shift transformations and hence, also for the money theory with emission and withdrawal.

The velocity of money equations (7) and (8) can also be generalized to money with emission and withdrawal. However, there are several ways of doing so because $\sigma_0$ is no more a constant but rather a function $\sigma_0(t)$.

## 4 Payments

We study the definition of allowed redistributions by certain compositions of atomic redistributions that are called payments. For formal description of payments we have to use *coproduct* – a standard concept in set theory and category theory [36,37].

The coproduct also turns out to be the most relevant composition operation when modeling decomposability of implementations of money. This is because, if payments and shift redistributions can be modeled using the coproduct operation, we will see that it becomes easier to split them apart so these smaller pieces can be handled in parallel on different computers.

### 4.1 Coproduct

Let $U$ be any set that is partitioned into to subsets $U_1$ and $U_2$, i.e., $U = U_1 \cup U_2$ and $U_1 \cap U_2 = \emptyset$. Then we write $U = U_1 \oplus U_2$ and say that $U$ is a *coproduct* of its subsets $U_1$ and $U_2$.

If $f_1 \colon U_1 \to V$ and $f_2 \colon U_2 \to V$ are arbitrary functions, then we can define a function $f \colon U \to V$ in the next way:

$$f(u) = \begin{cases} f_1(u) \text{ if } u \in U_1 \\ f_2(u) \text{ if } u \in U_2 \end{cases}$$

Such a function $f$ is called the *coproduct of functions* $f_1$ and $f_2$ and is denoted by $f_1 \oplus f_2$.

Let $\iota_1 \colon U_1 \to U$ and $\iota_2 \colon U_2 \to U$ be the inclusion maps, i.e., $\iota_1(u_1) = u_1$ and $\iota_2(u_2) = u_2$ for every $u_1 \in U_1$ and $u_2 \in U_2$.

The coproduct has the following *universal property*. For every functions $f_1 \colon U_1 \to V$ and $f_2 \colon U_2 \to V$ there is a unique function $f \colon U \to V$ denoted by $f_1 \oplus f_2$ such that $f \circ \iota_1 = f_1$ and $f \circ \iota_2 = f_2$.

The coproduct can be defined for any number of sets.

**Definition 17 (Coproduct of Sets and Functions).** If $U_1, U_2, \ldots, U_n$ are sets, then the *coproduct* $U_1 \oplus U_2 \oplus \ldots \oplus U_n$ of these sets is $(U; \iota_1, \iota_2, \ldots, \iota_n)$, where $U$ is a set and $\iota_i \colon U_i \to U$ are functions so that for every set $V$ and functions $f_i \colon U_i \to V$ (where $i \in \{1, 2, \ldots, n\}$), there is a unique function $f \colon U \to V$ so that $f \circ \iota_i = f_i$ for every $i \in \{1, 2, \ldots, n\}$. This function $f$ is denoted by $f_1 \oplus f_2 \oplus \ldots \oplus f_n$ and is called the *coproduct of* $f_1, f_2, \ldots, f_n$. (Fig. 5)

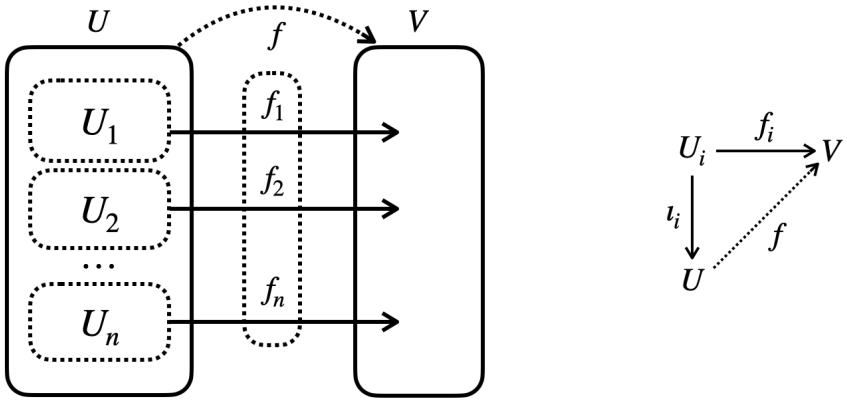


**Fig. 5.** Set diagram (left) and commutative diagram (right) for coproduct construction.

It is easy to show that for any sets $U_1, U_2, \ldots, U_n$, their coproduct always exists $U_1 \oplus U_2 \oplus \ldots \oplus U_n$. One possible construction is as follows:

$$\begin{aligned} U &= \{(u, i) \colon 1 \leqslant i \leqslant n,\ u \in U_i\} \\ \iota_i(u) &= (u, i) \end{aligned}$$

For example, if $U_1 = U_2 = \{a, b\}$, then $U_1 \oplus U_2 = (U; \iota_1, \iota_2)$, where

$$\begin{aligned} U &= \{(a, 1), (b, 1), (a, 2), (b, 2)\} \\ \iota_1(a) &= (a, 1) \\ \iota_1(b) &= (b, 1) \\ \iota_2(a) &= (a, 2) \\ \iota_2(b) &= (b, 2) \end{aligned}$$

Set partitioning is a special case of the set coproduct operation, which is what we will show to be the most important capability required to shard and scale a money scheme.

### 4.2 Payment Rules

In this section, we describe characterize payments as building blocks of redistributions. Payments are certain kind of redistributions that are defined on money subdistributions. Payments can be converted to more general redistributions via general composition and coproduct as a special (parallel) form of composition. Hence, the properties of payments define the properties of redistributions.

Payments usually change the status of just a few monetary units and are hence, defined on subdistributions. This is for the sake of having short descriptions. First, we define subdistributions, then characterize the payments and provide a few examples of payment types.

For any money distribution $M = (U, \nu, \beta)$, and for any subset $V \subseteq U$ with the corresponding inclusion map $\iota_V \colon V \to U$, the triple $V = M|_V = (V, \nu|_V, \beta|_V)$, where $\nu|_V = \iota_V \circ \nu$ and $\beta|_V = \iota_V \circ \beta$, is also a money distribution, which we denote by $M|_V$.

**Definition 18 (Sub-Distribution).** The money distribution $M|_V$ is said to be a *sub-distribution* of $M$.

**Definition 19 (Payment of Type $\Delta$).** A *payment of type* $\Delta$ is a shift redistribution $P = R_\Delta$ defined by a difference set $\Delta$ that is called a *payment type*.

If $P = R_\Delta$ is a payment and $V = \operatorname{dom} \Delta$ and $M = (U, \nu, \beta) \in \operatorname{dom} P$, then $P$ can be viewed as a shift redistribution on a sub-distribution $M|_V = (V, \nu|_V, \beta|_V)$ of $M$ that has the following properties:

- *Money preservation*: $\sigma(P(M|_V)) = \sigma(M|_V)$, which means that payments preserve the total money of the sub-distribution $M|_V$.
- *Non-redundancy*: From $\operatorname{dom}(\Delta) = V$, it follows that $V$ does not contain monetary units $u$ that do not change under $P$, i.e., are not deleted by $P$ neither the values of $\nu(u)$ and $\beta(u)$ are changed by $P$.

According to Sect. 3.3, such payments are uniquely defined by their types.
The following type of payments are some examples of payment types.

**Bill Payments** $P = R_\Delta$, where $\Delta = \langle\langle \emptyset, \emptyset, \{u\}\rangle, \langle \emptyset, \emptyset, \{(u, 0, b_1, b_2)\}\rangle\rangle$, where $u \in U$ and $b_1 \neq b_2 \in \mathfrak{B}$, i.e., $P$ changes the bearer of a single monetary unit $u$ from $b_1$ to $b_2$ and does nothing else.

**Account Payments** $P = R_\Delta$, where

$$\Delta = \big\langle \langle \emptyset, \emptyset, \{u, v\}\rangle \ , \ \langle \emptyset, \emptyset, \{(u, -n, a, a), (v, n, b, b)\}\rangle \big\rangle, \tag{9}$$

where $n \in \mathbb{N}$, $u, v \in U$ and $a \neq b \in \mathfrak{B}$, i.e., $P$ changes the value of two monetary units $u, v$ by $-n$ and $n$, respectively, and does nothing else.

**Bitcoin Payments** $P = R_\Delta$ where

$$\begin{aligned} \Delta = \langle & \\ & \langle \{u_1, \ldots, u_k\}, \{v_1, \ldots, v_\ell\}, \emptyset \rangle, \\ & \langle \{(u_1, n_1, a_1), \ldots, (u_k, n_k, a_k)\}, \{(v_1, m_1, b_1), \ldots, (v_\ell, m_\ell, b_\ell)\}, \emptyset \rangle \\ \rangle & \end{aligned} \tag{10}$$

where

$$n_1, \ldots, n_k, m_1, \ldots, m_\ell \in \mathbb{N}, \tag{11}$$

$$u_1, \ldots, u_k, v_1, \ldots, v_\ell \in U, \tag{12}$$

$$a_1, \ldots, a_k, b_1, \ldots, b_\ell \in \mathfrak{B}, \tag{13}$$

and

$$n_1 + \ldots + n_k = m_1 + \ldots + m_\ell, \tag{14}$$

i.e., $P$ destroys units $u_1, \ldots, u_k$ and creates new units $v_1, \ldots, v_\ell$ with the same total value.

### 4.3 Coproducts of Money Distributions

For developing a suitable decomposition theory for money solutions, we first have to define decomposition rules for money distributions.

If $U = U_1 \cup U_2 \cup \ldots \cup U_n$ is a partition of $U$, then every component $U_i$ defines a sub-distribution $M_i = M|_{U_i}$. Note that

$$\sigma(M) = \sigma(M_1) + \sigma(M_2) + \ldots + \sigma(M_n). \tag{15}$$

**Definition 20 (Coproduct of Money Distributions).** Given money distributions $M_1 = (U_1, \nu_1, \beta_1), \ldots, M_n = (U_n, \nu_n, \beta_n)$, the following is also a money distribution:

$$M = (U_1 \oplus \ldots \oplus U_n, \nu_1 \oplus \ldots \oplus \nu_n, \beta_1 \oplus \ldots \oplus \beta_n) \tag{16}$$

We call $M$ in (16) a *coproduct of money distributions*, that we also denote by $M = M_1 \oplus M_2 \oplus \ldots \oplus M_n$.

### 4.4 Coproducts of Payments

Payments act on money sub-distributions. If two payments $P_1$ and $P_2$ act independently on two different components $M_i$ and $M_j$ of the coproduct of money distributions, these payments can be executed independently, and in parallel.

A precise mathematical definition for parallel execution of such payments turns out to be related to the coproduct of functions construction described in Sect. 4.1.

Let $M = (U, \nu, \beta)$ be a money distribution. Let $U = U_1 \oplus U_2 \oplus \ldots \oplus U_n$. Let $P_1 = R_{\Delta_1}, \ldots, P_n = R_{\Delta_n}$ be payments such that:

- $\operatorname{dom} \Delta_1 \subseteq U_1, \ldots, \operatorname{dom} \Delta_n \subseteq U_n$, i.e., $P_i$ acts on the sub-distribution $M_i = M|_{U_i}$.
- $\operatorname{cre} \Delta_i \cap \operatorname{cre} \Delta_j$ for every $i \neq j$, i.e., no $P_i$ and $P_j$ with $i \neq j$ will create the same new monetary units.

Hence, if $M'_i = P_i(M_i) = (U'_i, \nu'_i, \beta'_i)$ (for $i \in \{1, 2, \ldots, n\}$), then the sets $U'_i$ are mutually non-intersecting and we have a coproduct:

$$P(M) = M'_1 \oplus M'_2 \oplus \ldots \oplus M'_n = P_1(M_1) \oplus P_2(M_2) \oplus \ldots \oplus P_n(M_n).$$

It is easy to see that $P(M)$ is a money distribution and hence the shift transformation $P$ is uniquely defined by the pair $(M, P(M))$. There is a unique difference set $\Delta$, such that $P = R_\Delta$, where $\Delta = P(M) \ominus M$. Moreover, from equation (15) it follows that $P$ preserves the total money, and hence, $P$ is a redistribution of $M$.

**Definition 21 (Coproduct of Payments).** Such a redistribution $P$ is called the *coproduct of* $P_1, P_2, \ldots, P_n$ and is denoted by $P = P_1 \oplus P_2 \oplus \ldots \oplus P_n$.

If $\Delta_i = \langle\langle U_i^-, U_i^+, U_i^0\rangle, \langle \Delta_i^-, \Delta_i^+, \Delta_i^0\rangle\rangle$ (for $i \in \{1, 2, \ldots, n\}$), then:

$$\begin{aligned}\Delta = \langle & \\ & \langle U_1^- \oplus \ldots \oplus U_n^-, U_1^+ \oplus \ldots \oplus U_n^+, U_1^0 \oplus \ldots \oplus U_n^0\rangle, \\ & \langle \Delta_1^- \oplus \ldots \oplus \Delta_n^-, \Delta_1^+ \oplus \ldots \oplus \Delta_n^+, \Delta_1^0 \oplus \ldots \oplus \Delta_n^0\rangle \\ \rangle &\end{aligned}$$

**Definition 22 (Coproduct of Difference Sets).** Such $\Delta$ is called the *coproduct of* $\Delta_1, \Delta_2, \ldots, \Delta_n$ and is denoted by $\Delta = \Delta_1 \oplus \Delta_2 \oplus \ldots \oplus \Delta_n$.

A commutative diagram for the payments coproduct construction is depicted in Fig. 6.

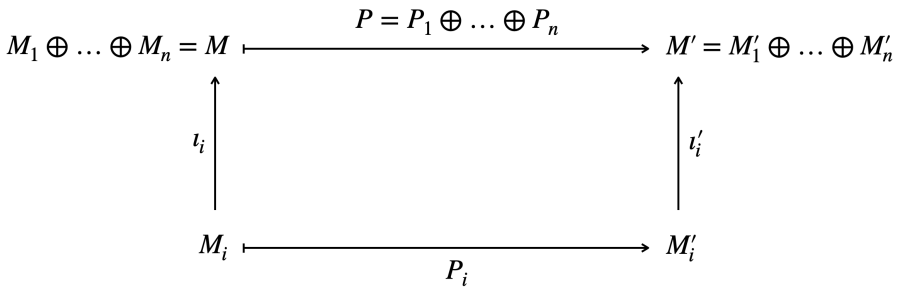


**Fig. 6.** Diagram for the construction of coproduct of payments.

### 4.5 Coproduct, Unitwise-Decomposition, and Atomicity

A redistribution is any finite composition of payments $P_i \in \mathbb{P}$:

$$R = P_1 \circ P_2 \circ \ldots \circ P_m \neq \Theta. \tag{17}$$

If $R$ is a coproduct of payments $P_1, P_2, \ldots, P_n$, i.e.,

$$R = P_1 \oplus P_2 \oplus \ldots \oplus P_n, \tag{18}$$

then this means in practice that these payments can be executed in parallel, while in the composition given by equation (17) the payments cannot, in general, be executed in parallel. Payments and their coproducts are implemented as atomic operations.

If the money system is decomposed via $U = U_1 \oplus U_2 \oplus \ldots \oplus U_n$, i.e., the set $U$ of monetary units is partitioned into non-intersecting components $U_1, U_2, \ldots, U_n$ that are stored and maintained in physically separated computers $C_1, C_2, \ldots, C_n$, then every payment $P$ can only be processed in on only one of those computers $C_i$. Both the input parameters and the output parameters $P$ have to be stored and be available in $C_i$. If there were inputs from several computers, $C_i$ would have to obtain synchronize the input data from several computers. If there were outputs of $P$ in two different computers, the system would have to implement simultaneous atomic swap operations in these computers. None of these tasks can be solved by any communication protocol in guaranteed deterministic time. Therefore, it is assumed that both the inputs and outputs of $P$ are stored in the single machine $C_i$ and hence, $P$ only changes the money distribution $M_i$ related to $U_i$.

This requirement leads to the requirement that the total money distribution $M$ is a coproduct $M = M_1 \oplus M_2 \oplus \ldots \oplus M_n$, and every atomic operation $R$ in the system has to be a coproduct of payments $R = P_1 \oplus P_2 \oplus \ldots \oplus P_n$ each $P_i$ acting on a single component $M_i$ of the total money distribution.

## 5 Money Schemes and Their Coproducts

So far, we have described money distributions and redistributions as compositions of payments. In this section, we define money schemes as mathematical abstractions of money solutions that only use certain type of payments. For example, account-based money solutions and bill-based money solutions define two different money schemes. In general, to define a money scheme, we have to define all possible money distributions, as well as all possible payments that are allowed in a particular solution.

We define the coproduct of money schemes that reflect situations where several independent money schemes are in use. For example, we have got used to having bank money in an account-based system and cash, and sometimes, we use them in parallel for payments.

We also define the universality property of money schemes that is motivated by the fact that in some money schemes (such as account-based schemes), that we call universal, we can pay arbitrary amount of

money that we have to any other bearer, while in other schemes (such as bill-based schemes) this is not always possible.

It turns out that a coproduct of universal money schemes is always universal itself, and vice versa: if a coproduct of certain money schemes is universal, also its components have to be universal. One of the consequences of this is that account-based schemes cannot be efficiently unitwise-decomposed.

### 5.1 Money Schemes

Money schemes are characterized by the set $\mathbb{M}$ of all possible money distributions and by certain shift redistributions on $\mathbb{M}$ that happen only because of *payments*. Different types of payments define different types of money schemes.

**Definition 23 (Money Scheme).** A *money scheme* is a pair $\mathcal{M} = (\mathbb{M}, \mathbb{P})$ where $\mathbb{M} \neq \emptyset$ is a set of money distributions and $\mathbb{P}$ is a set of payments that contains at least the identity payment 1.

For every money scheme $\mathcal{M} = (\mathbb{M}, \mathbb{P})$, we define the set $U_{\max}$ of all potential monetary units and the set $\mathfrak{B}$ of all potential bearers by:

$$U_{\max}(\mathbb{M}) = \{u \colon u \in U \text{ for some } M = (U, \nu, \beta) \in \mathbb{M}\}$$
$$\mathfrak{B}(\mathbb{M}) = \{b \colon b = \beta(u) \text{ and } u \in U \text{ for some } M = (U, \nu, \beta) \in \mathbb{M}\}$$

### 5.2 Coproducts of Money Schemes

For every two money schemes $\mathcal{M}_1 = (\mathbb{M}_1, \mathbb{P}_1)$ and $\mathcal{M}_2 = (\mathbb{M}_2, \mathbb{P}_2)$ one can construct a new money scheme denoted by $\mathcal{M}_1 \oplus \mathcal{M}_2 = (\mathbb{M}, \mathbb{P})$, where

$$\mathbb{M} = \{M_1 \oplus M_2 \colon M_1 \in \mathbb{M}_1, M_2 \in \mathbb{M}_2\}$$
$$\mathbb{P} = \{P_1 \oplus 1 \colon P_1 \in \mathbb{P}_1\} \cup \{1 \oplus P_2 \colon P_2 \in \mathbb{P}_2\}$$

**Definition 24 (Coproduct of Money Schemes).** The money scheme $\mathcal{M}_1 \oplus \mathcal{M}_2$ is called the *coproduct of money schemes* $\mathcal{M}_1$ and $\mathcal{M}_2$.

As $\sigma(M_1 \oplus M_2) = \sigma(M_1) + \sigma(M_2)$ for every $M_1 \in \mathbb{M}_1, M_2 \in \mathbb{M}_2$, every payment $P \in \mathbb{P}$ preserves both invariants $\sigma(M_1), \sigma(M_2)$, and hence, also the money invariant of $\mathcal{M}_1 \oplus \mathcal{M}_2$. In practice, the coproduct means that two parallel types of money are in use, while the total amounts of both types of money stay the same. It is easy to check that $\mathfrak{B}(\mathbb{M}) = \mathfrak{B}(\mathbb{M}_1) \cup \mathfrak{B}(\mathbb{M}_2)$ and $U_{\max}(\mathbb{M}) = U_{\max}(\mathbb{M}_1) \cup U_{\max}(\mathbb{M}_2)$. More generally, for $n$ money distributions $\mathcal{M}_1 = (\mathbb{M}_1, \mathbb{P}_1), \ldots, \mathcal{M}_n = (\mathbb{M}_n, \mathbb{P}_n)$ their coproduct $(\mathbb{M}, \mathbb{P}) = \mathcal{M}_1 \oplus \ldots \oplus \mathcal{M}_n$, where

$$\mathbb{M} = \{M_1 \oplus \ldots \oplus M_n \colon M_1 \in \mathbb{M}_1, \ldots, M_n \in \mathbb{M}_n\}$$
$$\mathbb{P} = \bigcup_{i=1}^{n} \{\underbrace{1 \oplus \ldots \oplus 1}_{i-1} \oplus P_i \oplus \underbrace{1 \oplus \ldots \oplus 1}_{n-i} \colon P_i \in \mathbb{P}_i\}$$

The key aspect of efficient and scalable implementability of a money scheme is related to coproduct decomposability. The storage complexity of a money scheme $\mathcal{M}$ is proportional to the number $|U|$ of monetary units. If $|U|$ gets too large, it will be necessary to decompose the money scheme:

$$\mathcal{M} = \mathcal{M}_1 \oplus \mathcal{M}_2 \oplus \ldots \oplus \mathcal{M}_n,$$

where every $\mathcal{M}_i$ manages money distributions $M_i = (U_i, \nu_i, \beta_i)$ with $|U_i| \ll |U|$, ideally, $|U_i| \sim \frac{|U|}{n}$. In practice, for an electronic money scheme, this means that different money units are stored on different databases.

### 5.3 Universal Money Schemes

Universal money schemes allow for any bearer $a$ in any money distribution to pay arbitrary amount of money to any other bearer $b$, assuming only that $a$ has that much money.

**Definition 25 (Universal Money Scheme).** A money scheme $\mathcal{M} = (\mathbb{M}, \mathbb{P})$ is called *universal* on a bearer set $\mathfrak{B}$, if for every money distribution $M = (U, \nu, \beta) \in \mathbb{M}$, for every two bearers $a, b \in \mathfrak{B}$ and for any amount $n \leqslant \sigma(M, a)$, there is a coproduct $R = P_1 \oplus P_2 \oplus \ldots \oplus P_m$ of payments, such that:

$$\begin{aligned}
&\sigma(R(M), a) = \sigma(M, a) - n \\
&\sigma(R(M), b) = \sigma(M, b) + n \\
&\forall u \in U : \text{if } u \in U \backslash U' \text{ or } \nu(u) > \nu'(u) \text{ or } \beta(u) \neq \beta'(u), \text{ then } \beta(u) = a \\
&\text{where} R(M) = (U', \nu', \beta')
\end{aligned}$$

(i.e., any bearer $a$ whose money is at least $n$ can pay $n$ units to any other bearer $b$ without changing the money of other bearers, whereas the corresponding redistribution $R$ does not delete, reduce a value or change the bearer of any monetary unit $u$ with $\beta(u) \neq a$.)

This implies that $\sigma(R(M), c) = \sigma(M, c)$ of any user $c \notin \{a, b\}$. Indeed, the third condition implies that $R$ does not reduce the money of any user except $a$, and as the reduced money of $a$ is completely compensated by the increased money of $b$, there is no room to change the money of any other users, because $R$ does not change the total money of the money scheme.

Not all money schemes have to be universal. For example, physical cash is not universal because, for example, if a bearer $a$ has a single 20 dollar bill, it is not possible for $a$ to pay 10 dollars to $b$ with a single payment transaction. However, $a$ is able to pay 20 dollars to any other bearer. More generally, if $a$ has a set $V$ of bills with total value of $n$ dollars, it is possible for $a$ to pay $n$ dollars to any other user. This property is formally defined as the weak universality.

**Definition 26 (Weakly Universal Money Scheme).** A money scheme $\mathcal{M} = (\mathbb{M}, \mathbb{P})$ is called *weakly universal* on a bearer set $\mathfrak{B}$, if and for every money distribution $M = (U, \nu, \beta) \in \mathbb{M}$, for every two bearers $a, b \in \mathfrak{B}$ and for any amount $n = \sum_{u \in V} \nu(u)$, where $V \subseteq U$ and $\beta(V) = \{a\}$, there is a coproduct $R = P_1 \oplus P_2 \oplus \ldots \oplus P_m$ of payments, such that:

$$\begin{aligned}
&\sigma(R(M), a) = \sigma(M, a) - n, \\
&\sigma(R(M), b) = \sigma(M, b) + n, \text{and} \\
&\forall u \in U : \text{if } u \in U \backslash U' \text{ or } \nu(u) > \nu'(u) \text{ or } \beta(u) \neq \beta'(u), \text{ then } \beta(u) = a, \\
&\text{where} R(M) = (U', \nu', \beta').
\end{aligned}$$

(i.e., any bearer $a$ can pay to a bearer $b$ any $n$ units that is a total value of a subset $U$ of value units owned by $a$, whereas the corresponding redistribution $R$ does not delete or reduce a value, ore change the bearer of any monetary unit $u$ with $\beta(u) \neq a$.)

We will show that money schemes $\mathcal{M} = (\mathbb{M}, \mathbb{P})$ with account payments are universal on $\mathfrak{B}(\mathbb{M})$, but are not universal on any proper superset $\mathfrak{B} \supset \mathfrak{B}(\mathbb{M})$. The schemes with bill payments are weakly universal on $\mathfrak{B}(\mathbb{M})$ but not necessarily universal.

The universal properties may seem trivial and be taken for granted. However, they have important consequences for unitwise-decomposability.

### 5.4 Unitwise-Decomposability of Universal Money Schemes

In order to understand which money schemes can be decomposed, we need to draw some conclusions about their composability.

**Theorem 2.** *If $\mathcal{M}_1 = (\mathbb{M}_1, \mathbb{P}_1)$ and $\mathcal{M}_2 = (\mathbb{M}_2, \mathbb{P}_2)$ are [weakly] universal on $\mathfrak{B}$ with $|\mathfrak{B}| \geqslant 2$, then their coproduct $\mathcal{M}_1 \oplus \mathcal{M}_2$ is also [weakly] universal on $\mathfrak{B}$.*

**Proof:** We prove the weakly universal case. The proof in the universal case is completely analogous.

Let $M = M_1 \oplus M_2 \in \mathbb{M}$ with $M_1 = (U_1, \nu_1, \beta_1)$ and $M_2 = (U_2, \nu_2, \beta_2)$ be a money distribution. Let $a, b \in \mathfrak{B}$ be two bearers, $V \subseteq U = U^1 \oplus U^2$, $\beta(V) = \{a\}$, and $n = \sum_{u \in V} \nu(u) = \sum_{u \in V} \nu_1(u)$, where $\nu = \nu_1 \oplus \nu_2$. Obviously, $n = n_1 + n_2$. Let $U_1 = V \cap U^1$ and $U_2 = V \cap U^2$, $n_1 = \sum_{u \in U_1} \nu_1(u)$, $n_2 = \sum_{u \in U_2} \nu_2(u)$. Due to the universality of $\mathcal{M}_1$, there is a redistribution $R_1 = P_1^1 \oplus \ldots \oplus P_1^{m_1}$ in $\mathcal{M}_1$, such that:

$$\begin{aligned}
&\sigma(R_1(M_1), a) = \sigma(M_1, a) - n_1,\\
&\sigma(R_1(M_1), b) = \sigma(M_1, b) + n_1,\\
&\forall u \in U_1 : \text{if } u \in U_1 \backslash U_1' \text{ or } \nu_1(u) > \nu_1'(u) \text{ or } \beta_1(u) \neq \beta_1'(u), \text{then } \beta_1(u) = a,\\
&\text{where } R_1(M_1) = (U_1', \nu_1', \beta_1').
\end{aligned}$$

Due to the universality of $\mathcal{M}_2$, there is a redistribution $R_2 = P_2^1 \oplus \ldots \oplus P_2^{m_2}$ in $\mathcal{M}_2$, such that:

$$\begin{aligned}
&\sigma(R_2(M_2), a) = \sigma(M_2, a) - n_2,\\
&\sigma(R_2(M_2), b) = \sigma(M_2, b) + n_2,\\
&\forall u \in U_2 : \text{if } u \in U_2 \backslash U_2' \text{ or } \nu_2(u) > \nu_2'(u) \text{ or } \beta_2(u) \neq \beta_2'(u), \text{then } \beta_2(u) = a,\\
&\text{where } R_2(M_2) = (U_2', \nu_2', \beta_2').
\end{aligned}$$

Hence, for the redistribution $R_1 \oplus R_2 = P_1^1 \oplus \ldots \oplus P_1^{m_1} \oplus P_2^1 \oplus \ldots \oplus P_2^{m_2}$, we have

$$\begin{aligned}
\sigma(R(M), a) &= \sigma(R_1(M_1), a) + \sigma(R_2(M_2), a)\\
&= \sigma(M_1, a) + \sigma(M_2, a) - n_1 - n_2\\
&= \sigma(M, a) - n,\\
\sigma(R(M), b) &= \sigma(R_1(M_1), b) + \sigma(R_2(M_2), b)\\
&= \sigma(M_1, b) + \sigma(M_2, b) + n_1 + n_2\\
&= \sigma(M, b) + n,\\
&\forall u \in U : \text{if } u \in U \backslash U' \text{ or } \nu(u) > \nu'(u) \text{ or } \beta(u) \neq \beta'(u), \text{ then } \beta(u) = a,\\
&\text{where } R(M) = (U', \nu', \beta') = (U_1' \oplus U_2', \nu_1' \oplus \nu_2', \beta_1' \oplus \beta_2').
\end{aligned}$$

Hence, $\mathcal{M}$ is [weakly] universal. □

Note that if $\mathcal{M}_1$ is [weakly] universal on $\mathfrak{B}(\mathbb{M}_1)$ and $\mathcal{M}_2$ is [weakly] universal on $\mathfrak{B}(\mathbb{M}_2)$, then this does not mean that $\mathcal{M}_1 \oplus \mathcal{M}_2$ is [weakly] universal on $\mathfrak{B}(\mathbb{M}_1) \cup \mathfrak{B}(\mathbb{M}_2)$.

**Theorem 3.** *If $\mathcal{M}_1 = (\mathbb{M}_1, \mathbb{P}_1)$ and $\mathcal{M}_2 = (\mathbb{M}_2, \mathbb{P}_2)$ are money schemes so that their coproduct $\mathcal{M} = \mathcal{M}_1 \oplus \mathcal{M}_2 = (\mathbb{M}, \mathbb{P})$ is [weakly] universal on $\mathfrak{B}$ with $|\mathfrak{B}| \geqslant 2$, then $\mathcal{M}_1$ and $\mathcal{M}_2$ are also [weakly] universal on $\mathfrak{B}$.*

**Proof:** We prove that $\mathcal{M}_1$ is [weakly] universal. The proof for $\mathcal{M}_2$ is analogous. Let $a, b \in \mathfrak{B}$ be two bearers.

We first prove that there is $M_2^0 \in \mathbb{M}_2$, where $\sigma(M_2^0, a) = 0$. Indeed, due to the [weak] universality of $\mathcal{M}$, for every money distribution $M = M_1 \oplus M_2 \in \mathbb{M}$ there is a redistribution $R_0$ such that $\sigma(R_0(M), a) = \sigma(M, a) - \sigma(M, a) = 0$. Note that due to $\mathbb{M} \neq \emptyset$, there is at least one $M \in \mathbb{M}$ to which one can apply $R_0$. Let $R_0(M) = M_1^0 \oplus M_2^0$. As $\sigma(R_0(M), a) = \sigma(M_1^0, a) + \sigma(M_2^0, a) = 0$, we also have $\sigma(M_2^0, a) = 0$, and hence, such $M_2^0 \in \mathbb{M}_2$ exists.

Let $M_1 = (U_1, \nu_1, \beta_1) \in \mathbb{M}_1$ be a money distribution. Let $U_1 \subseteq U_1$ where $\beta_1(U_1) = \{a\}$. Let $M = M_1 \oplus M_2^0$.

Due to the universality of $\mathcal{M}$, there is a redistribution $R = P_1 \oplus \ldots \oplus P_m$ with $R(M) = (U', \nu', \beta')$.such that:

$$\begin{aligned}
&\sigma(R(M), a) = \sigma(M, a) - n,\\
&\sigma(R(M), b) = \sigma(M, b) + n,\\
&\forall u \in U : \text{if } u \in U \backslash U' \text{ or } \nu(u) > \nu'(u) \text{ or } \beta(u) \neq \beta'(u), \text{ then } \beta(u) = a.
\end{aligned}$$

Due to the coproduct properties, we can assume w.l.o.g. that $R = R_1 \oplus R_2$, where $R_1 = P_1 \oplus \ldots \oplus P_k$ is a redistribution in $\mathcal{M}_1$ and $R_2 = P_{k+1} \oplus \ldots \oplus P_m$ is a redistribution in $\mathcal{M}_2$. As $\sigma(M, a) = \sigma(M_1, a) + \sigma(M_2^0, a)$ and $\sigma(M_2^0, a) = 0$, we have that $\sigma(M, a) = \sigma(M_1, a)$. Note also that $R(M) = R_1(M_1) \oplus R_2(M_2^0)$ and $\sigma(R(M), a) = \sigma(R_1(M_1), a) + \sigma(R_2(M_2^0), a)$.

As $R_2$ does not delete, reduce the value, or change the bearer of any monetary unit not beared by $a$, we conclude that $\sigma(R_2(M_2^0), a) = \sigma(M_2^0, a) = 0$. Hence, $\sigma(R(M), a) = \sigma(R_1(M_1), a)$ and

$$\sigma(R_1(M_1), a) = \sigma(R(M), a) = \sigma(M, a) - n = \sigma(M_1, a) - n.$$

As $R_2$ must preserve the total value of $M_2^0$ and it does not decrease the money of any other bearer, we have that $\sigma(R_2(M_2^0), b) = \sigma(M_2^0, b)$ and hence $\sigma(R_1(M_1), b) = \sigma(M_1, b) + n$ and the [weak] universality of $\mathcal{M}_1$ follows. □

These decomposability results imply (shown in Sect. 6) that:

- Universal account schemes do not allow efficient unitwise-decompositions.
- [Weakly] universal bill schemes allow ideal unitwise-decompositions with $|U_i| \sim \frac{|U|}{n}$.

## 6 Examples of Money Schemes

Now we will offer some examples of common and important money schemes. We assume the same universe $U_{\max}$ in all examples.

We also assume that the total money $\sigma(M)$ of every money distribution $M \in \mathbb{M}$ is the same, i.e., $\sigma(M) = \sigma_0$ for a constant $\sigma_0$.

### 6.1 The Bill Scheme

In the bill scheme, the monetary units are called *bills*. All payments are bill payments, as described in Sect. 4.2, i.e., the shift distributions $P = R_\Delta$ with

$$\Delta = \big\langle \langle \emptyset, \emptyset, \{u\} \rangle \, , \, \langle \emptyset, \emptyset, \{(u, 0, b_1, b_2)\} \rangle \big\rangle,$$

where $u \in U$ and $b_1 \neq b_2 \in \mathfrak{B}$, i.e., $P$ changes the bearer of a single monetary unit $u$ from $b_1$ to $b_2$ and does nothing else.

The bill scheme is the oldest money scheme, starting from seashells and stones and ending with coins and bills. A bill is a value unit that has a fixed nominal value that does not change during payments. Only the bearers of bills change during redistributions. We note that functionally, bills and coins are equivalent, and we refer to both forms using the word "bill". In physical bill schemes, payments occur when physical control over a bill changes hands at a unique moment in time. Payments may be effected by changing the bearer of several bills, ie. A payment of $15 may be composed of the transfer of one $5 bill and one $10 bill.

Further, In electronic bill schemes, it may be that multiple payments may be combined into a single, atomic redistribution, such that multiple multi-bill payments are combined into a single atomic redistribution.

In some bill schemes, all bills may have equal nominal values. In other schemes, bills of different nominal values are used. Bills never have nominal value zero. In physical bill schemes, bill issuers usually select regular, physically convenient denominations for their issued bills, ie. $1, $5, and $10 or €1, €2, and €5. In electronic bill schemes, regular denominations are not required, and it may be more convenient to allow irregular values for bills.

It is easy to see that for any composition $P$ of bill payments preserves the set and values of monetary units, i.e., for every money distribution $M = (U, \nu, \beta)$:

$$U_{P(M)} = U_M$$
$$\nu_{P(M)} = \nu_M.$$

By a single bill money scheme we mean a bill scheme that describes a one single bill, i.e., in which all allowed money distributions are in the form $M = (\{u\}, \nu, \beta)$, where $\nu = \{(u, v)\}$ and $\beta = \{(u, b)\}$, where $v \in \mathbb{N}$ and $b \in \mathfrak{B}$.

**Theorem 4.** *Every bill scheme $\mathcal{M}$ is a coproduct of single bill money schemes.*

**Proof:** The set $U$ of monetary units is constant in the bill scheme and can be represented as the coproduct of singleton subsets $U = \bigoplus_{u \in U} \{u\}$. As every payment $P$ only changes the bearer of one single bill $u$, it can be considered as a payment in the single bill money distribution $(\{u\}, \nu_u, \beta_u)$, where $\nu_u = \{(u, \nu(u))\}$ and $\beta_u = \{(u, \beta(u))\}$. If we define $\mathbb{M}_u$ to be the set of all money distributions in the form $(\{u\}, \{(u, \nu(u))\}, \{u, \beta(u)\})$, $\mathbb{P}_u$ to be the set of all payments $P$ with type $\Delta = \langle\langle \emptyset, \emptyset, \{u\}\rangle, \langle \emptyset, \emptyset, \{(u, 0, b_1, b_2)\}\rangle\rangle$, and $\mathcal{M}_u = (\mathbb{M}_u, \mathbb{P}_u)$, we have $\mathcal{M} = \bigoplus_{u \in U} \mathcal{M}_u$. □

Theorem 4 means that bill schemes are arbitrarily unitwise-decomposable – every partition of $U$ induces a coproduct decomposition.

**Theorem 5.** *Every universal bill scheme $(\mathbb{M}, \mathbb{P})$ on a bearer set $\mathfrak{B}$ (where $|\mathfrak{B}| \geqslant 2$) has only bills with unit value, i.e., $\beta(u) = 1$ for every $M = (U, \nu, \beta)$ and $u \in U$.*

**Proof:** Let $u$ be a bill with $\nu(u) > 1$. Let $M$ be a money scheme where $b \in \mathfrak{B}$ is the bearer of $u$ and has no more bills. It is not possible for $b$ to pay 1 unit to any other bearer, and hence, the scheme is not universal. □

The bill scheme with only unit-value bills is not very efficient as the number of bills has to be large and for making a payment of value $n$, one has to execute (in parallel) $n$ separate payments. Efficient bill schemes have to use bills with various denominations but then, we lose universality. Still, the lack of universality can be compensated for by using an exchange mechanism, and indeed we have already shown that we all have and use weakly universal money schemes. If we can design a practical bill scheme, we know that, because it is unitwise-decomposable, it can be made to scale by processing more transfers in parallel.

### 6.2 The Account Scheme

In account schemes, the monetary units are called *accounts*. All payments are account payments, as described in Sect. 4.2, i.e., shift distributions $P = R_\Delta$ with

$$\Delta = \big\langle \langle \emptyset, \emptyset, \{u, v\}\rangle \ , \ \langle \emptyset, \emptyset, \{(u, -n, a, a), (v, n, b, b)\}\rangle \big\rangle,$$

where $n \in \mathbb{N}$, $u, v \in U$ and $a \neq b \in \mathfrak{B}$, i.e., $P$ changes the value of two monetary units $u, v$ by $-n$ and $n$, respectively, and does nothing else. In this case, $a$ is called the payer, $u$ is called the payer account, $b$ is called the receiver and $v$ the receiver account.

It is easy to see that for any composition $R$ of account payments preserves the set and bearers of monetary units, i.e., for every money distribution $M = (U, \nu, \beta)$:

$$U_{R(M)} = U_M$$
$$\beta_{R(M)} = \beta_M.$$

It turns out that universal account schemes cannot be efficiently unitwise-decomposed.

**Theorem 6.** *If an account scheme $(\mathbb{M}, \mathbb{P})$ is universal on $\mathfrak{B}$, then $|U| \geqslant |\mathfrak{B}|$ for every $M = (U, \nu, \beta) \in \mathbb{M}$.*

**Proof:** For the universality to hold, every bearer $b \in \mathfrak{B}$ must have an account. □

**Corollary 2.** *Universal account schemes do not have efficient unitwise-decompositions, i.e., for every decomposition $\mathcal{M} = \mathcal{M}_1 \oplus \mathcal{M}_2$ of an account scheme $\mathcal{M}$ universal on $\mathfrak{B}$, then by Thm. 3, also $\mathcal{M}_1$ and $\mathcal{M}_2$ are universal on $\mathfrak{B}$ and hence operate on the sets of monetary units $U_1, U_2$ with size at least $|\mathfrak{B}|$.*

To put it more practically, in every unitwise-decomposition of a universal account scheme, all users must have accounts in every component. This is equivalent to all users having an account at every bank.

Such a decomposition is hence by value not by the number of monetary units. But representing a bigger value is not a problem in today's machines, as for example, representation of a twice bigger values just need one additional bit.

The only reason why decomposition by value makes sense is load balancing, i.e., even if every user has account in every component, the requests served by the system may be equally distributed between components. From a banking standpoint, it would mean that if one bank's payment system was overloaded, one could simply make your payment at a less busy bank, where your payee also, conveniently, has an account. In this way we avoid the banks needing to talk to each other to process a payment.

### 6.3 The Extended Account Scheme

In the extended account scheme, the monetary units are called *accounts*. All payments are either account payments (Sect. 6.2) or bill payments (Sect. 6.1). The payments either change the value of two accounts as in the account scheme, or change the bearer of a one single account. This simply means that payments can be effected using extended accounts by changing their bearer as well as by reducing their value.

It is easy to see that for any composition $R$ of extended account payments preserves the set of monetary units, i.e., for every money distribution $M = (U, \nu, \beta)$:

$$U_{R(M)} = U_M.$$

Extended account scheme offers some flexibility compared to the account scheme. It is no more true that in a composition of a universal extended account scheme, every bearer has to have account in every component, because accounts can be created on demand by using the bill payments. However, after sufficiently long time, there will most probably be accounts for most of the bearers in every component, if the components are large enough in value.

If there are many components of low value, the scheme will resemble the bill scheme.

### 6.4 The Bitcoin Scheme

In the Bitcoin scheme, the monetary units are called *UTXOs* (*Unspent Transaction Outputs*). All payments are shift distributions $P = R_\Delta$, compare with (10) – (14), i.e., $P$ destroys UTXOs $u_1, \ldots, u_k$ and creates new UTXOs $v_1, \ldots, v_\ell$ with the same total value.

Bitcoin payments are effected by creating a transaction that spends one or more Unspent Transaction Outputs and, in their place, creating one or more new Unspent Transaction Outputs with potentially different individual values – though the total remains the same. These new UTXO's may have the same or different bearers.

In case the set $\{u_1, \ldots, u_k\}$ in $\Delta$ can contain arbitrary UTXOs, then such a full-scale Bitcoin scheme is not decomposable.

**Theorem 7.** *Full-scale Bitcoin scheme is not unitwise-decomposable.*

**Proof:** Assume that we have a decomposition. Let $u_1$ and $u_2$ be two UTXOs in different components. As the scheme is full-scale, there is a payment that deletes $u_1$ and $u_2$ and creates a new coin $v$ with $\nu(v) = \nu(u_1) + \nu(u_2)$. As such a payment deletes coins in different components, it cannot be defined as a payment in one component. □

**Theorem 8.** *Full-scale Bitcoin scheme is universal.*

**Proof:** Let $M = (U, \nu, \beta)$ be a money distribution. Let $a$ be a bearer that has (bears) the coins $u_1, \ldots, u_m$ and these are all the coins that $a$ has. Hence, $\sigma(M, a) = \nu(u_1) + \ldots + \nu(u_m)$. Let $0 < n \leqslant \sigma(M, a)$ and $b$ be any other bearer. As the scheme is full-scale, there is a payment $P$ transforming $M$ to $M' = (U', \nu', \beta')$ that

deletes the coins $u_1, \ldots, u_m$, creates two coins $v_1$ and $v_2$ with $\nu'(v_1) = n$, $\beta'(v_1) = b$, $\nu'(v_2) = \sigma(M, a) - n$, and $\beta'(v_2) = a$, and does nothing else. For such a payment:

$$\sigma(P(M), a) = \sigma(M, a) - n$$
$$\sigma(P(M), b) = \sigma(M, b) + n,$$

which proves that the money scheme is universal. □

By Theorems 2 and 8, the coproduct of full-scale Bitcoin schemes is universal. However, by Theorem 7, it follows that such a coproduct is not full-scale.

Therefore, the only way of making the Bitcoin type money scheme to scale is to use many independent Bitcoin type schemes in parallel. This means that in every component, the total amount of money stays the same. Limited value payment transactions can then be localized, but for larger payments, one has to use parallel payments in several components.

In an extreme cases of decomposition, where the total values of the components become very small, the scheme becomes similar to the bill scheme. In fact, if the total value of every component is 1, every payment can only delete one coin and create one coin of the same value, and we have a scheme that is equivalent to the universal bill scheme.

## 7 Classification of Money Schemes

Now we have all the tools needed to classify all possible money schemes. The basis of a classification is the properties of the payments of the money scheme. First, in Section 7.1, we present a classification of money schemes based on certain invariants.

### 7.1 Classification by Invariance

In this section, we present a full list $(\mathbb{M}, \mathbb{P})$, based on the invariance of components $U$, $\nu$, and $\beta$ of money distributions under the payments $P \in \mathbb{P}$ of the money scheme.

**Table 1.** Map of money schemes.

| **Scheme** | $U$ | $\nu$ | $\beta$ | **Comment** |
|---|---|---|---|---|
| Trivial scheme | const | const | const | Only identity payment allowed |
| Bill scheme | const | const | var | |
| Account scheme | const | var | const | |
| Extended account scheme | const | var | var | Account payments + account ownership change |
| - | var | const | const | If $U$ changes, the so do $\nu$ and $\beta$ |
| - | var | const | var | If $U$ changes, the so do $\nu$ and $\beta$ |
| - | var | var | const | If $U$ changes, the so do $\nu$ and $\beta$ |
| Hybrid schemes | var | var | var | Bitcoin Scheme |

From a purely combinatorial viewpoint, there are eight classes of schemes as presented in Tab. 1:

1. If all three parameters $U$, $\nu$, and $\beta$ are invariant, then the payments do not change the money distribution, which means that money does not flow, and hence, this class of schemes is not interesting.
2. There exist no schemes, in which only $U$ changes, because, by changing the domain of a function, we also change the bearer and value functions, since every monetary unit must have a value and a bearer.

So, only five of these eight types are of practical interest:

1. Schemes in which only the bearer function $\beta$ changes, i.e., bill schemes (Sect. 6.1).
2. Schemes in which only the value function $\nu$ changes, i.e., account schemes (Sect. 6.2).
3. Schemes in which only $\beta$ and $\nu$ change, one example of which is the extended account scheme (Sect. 6.3).
4. Schemes in which all parameters may change, i.e., the hybrid schemes, an example of which is the the Bitcoin scheme (Sect. 6.4).

### 7.2 Descriptional Complexity of Payments

As the measure of *descriptional complexity* of a payment $P = R_\Delta$, we use the storage size of $\Delta$ represented as a data structure.

Intuitively, the complexity has to be proportional to the number of monetary units that $P$ changes. Also, the more units $P$ creates, the more complex it is. If $\Delta = (U^+, U^-, U^0; \Delta^+, \Delta^-, \Delta^0)$ is the difference set of $P$, then for every $u \in U^+ \cup U^- \cup U^0$, the storage space needed for $\Delta^+(u)$, $\Delta^-(u)$, and $\Delta^0(u)$ is constant, and hence, the value $|\operatorname{dom}\Delta| + |\operatorname{cre}\Delta| = |U^+| + |U^-| + |U^0|$ is indeed proportional to the descriptional complexity.

**Definition 27 (Descriptional Complexity of a Payment).** *By the descriptional complexity of a payment $P = R_\Delta$, we mean the value $\|P\| = |\operatorname{dom}\Delta| + |\operatorname{cre}\Delta| = |U^+| + |U^-| + |U^0|$, where*

$$\Delta = (U^+, U^-, U^0; \Delta^+, \Delta^-, \Delta^0)$$

.

**Lemma 3 (Descriptional Complexity of Coproduct).** *For every payments $P_1, P_2$, the descriptional complexity of $P = P_1 \oplus P_2$ is the sum of the descriptional complexities of $P_1$ and $P_2$, i.e., $\|P_1 \oplus P_2\| = \|P_1\| + \|P_2\|$.*

**Proof:** Direct consequence of the definitions in Sect. 4.4. □

### 7.3 Equivalence of Money Distributions and Payments

As our goal in this section is to present a general classification of payment types, we do not want to be too specific about the details of the payments. For example, if a payment creates a new monetary unit $v$, the only things that matter are the value of $v$ and the bearer of $v$. If there is another payment $P'$ that does the same things as $P$, except it creates a new unit $v' \neq v$ that has the same value and bearer, then $P$ and $P'$ are considered equivalent.

In order to give a precise mathematical definition for the equivalence of payments, we start from defining equivalent money distributions.

Intuitively, two money distributions are equivalent if, for any bearer $b$, the sets of monetary units beared by $b$ in the money distributions contain the monetary units of similar values.

**Definition 28 (Equivalent Money Distributions).** Two money distributions $M_1 = (U_1, \nu_1, \beta_1)$ and $M_2 = (U_2, \nu_2, \beta_2)$ are *equivalent* ($M_1 \sim M_2$) if there is a bijection $f\colon U_1 \to U_2$, such that $\nu_2(f(u)) = \nu_1(u)$ and $\beta_2(f(u)) = \beta_1(u)$ for every $u \in U_1$.

Equivalently, we can define the equivalence of money distributions through a special kind of redistributions called unit permutations.

**Definition 29 (Unit Permutation).** For any bijection $f\colon \mathbb{U}_{\max} \to \mathbb{U}_{\max}$, there is a redistribution $S_f$ called a *unit permutation* that transforms every money distribution $M = (U, \nu, \beta)$ to a money distribution

$$S_f(M) = (f(U), \nu \circ f, \beta \circ f) \tag{19}$$

**Definition 30 (Equivalent Money Distributions – Alternative Definition).** Money distributions $M_1$ and $M_2$ are *equivalent* ($M_1 \sim M_2$) if there is a unit permutation $S_f$ so that $M_1 = S_f(M_2)$.

Two payments are considered equivalent if they are defined on the same domain $D$ of money distributions and for every money distribution $M \in D$ the produced money distributions are equivalent.

**Definition 31 (Equivalent Payments).** Payments $P_1$ and $P_2$ are *equivalent* ($P_1 \sim P_2$) if $\mathrm{dom}(P_1) = \mathrm{dom}(P_2) = D$ and $P_1(M) \sim P_2(M)$ for every money distribution $M \in D$.

For example, a single bill payment $P_1$ that changes the bearer of a bill $u$ from $a$ to $b$ is equivalent to a payment that deletes $u$ and creates a new bill $v$ with the same value and the bearer $b$.

Two payments are defined to be of equivalent type, if in addition to arbitrary permutation of monetary units, we may also apply arbitrary permutation of the bearer set.

**Definition 32 (Unit-Bearer Permutation).** For any two bijections $f\colon \mathbb{U}_{\max} \to \mathbb{U}_{\max}$ and $g\colon \mathfrak{B} \to \mathfrak{B}$, there is a redistribution $S_{f,g}$ called a *unit-bearer permutation* that transforms every money distribution $M = (U, \nu, \beta)$ to a money distribution

$$S_{f,g}(M) = (f(U), \nu \circ f, g \circ \beta \circ f) \tag{20}$$

**Definition 33 (Payments of Equivalent Type).** Two payments $P_1$ and $P_2$ are of *equivalent type* if there is a unit-bearer permutation $S_{f,g}$ so that:

$$\mathrm{dom}(P_2) = S_{f,g}(\mathrm{dom}(P_1)) \tag{21}$$

$$P_2 \circ S_{f,g} = S_{f,g} \circ P_1 \tag{22}$$

### 7.4 Classification of ∘-Irreducible Payments

In this section, we study how arbitrary payments can be decomposed into a general composition of simpler payments, i.e., payments with smaller descriptional complexity.

**Definition 34 (∘-Irreducible Payment).** A payment $P$ is *∘-irreducible (composition-irreducible)* if it cannot be represented as a composition $P = P_1 \circ P_2$ of $P_1$ and $P_2$ with $\|P_1\| < \|P\|$ and $\|P_2\| < \|P\|$.

There are the following equivalent classes of ∘-irreducible payments, where $a$ pays $b$ an amount $n$:

1. *Single unit transfer*, where a single monetary unit $u$ with value $n$ owned by $a$ is converted to a single unit $v$ with value $n$ owned by $b$ (Fig. 7).
2. *Two-unit swap*, where two monetary units $u_1$ with value $n_1$ owned by $a$, and $u_2$ with value $n_2$ owned by $b$ are converted to two units $v_1$ with value $n_1 - n$ owned by $a$, and $v_2$ with value $v_2 + n$ owned by $b$ (Fig. 7).
3. *Two-unit split*, where single monetary unit $u$ with value $m$ owned by $a$ is converted to two units $v_1$ with value $m - n$ owned by $a$, and $v_2$ with value $n$ owned by $b$
4. *Two-unit join*, where two monetary units $u_1$ with value $n_1$ owned by $a$,and $u_2$ with value $n_2$ owned by $a$ are converted to a single unit $v$ with value $n = n_1 + n_2$ owned by $b$ (Fig. 7).

Every payment is a composition of ∘-irreducible payments.

### 7.5 Classification of ⊕-Irreducible Payments

In this section, we study how arbitrary payments can be decomposed into a coproduct of simpler payments, i.e., payments with smaller descriptional complexity.

**Definition 35 (⊕-Irreducible Payment).** A payment $P$ is *⊕-irreducible (coproduct-irreducibly)* if it cannot be represented as a coproduct $P = P_1 \oplus P_2$ of $P_1$ and $P_2$ with $\|P_1\| < \|P\|$ and $\|P_2\| < \|P\|$.

As every ∘-irreducible payment is also ⊕-irreducible, the four types depicted in Fig. 8 are ⊕-irreducible. We have three additional ⊕-irreducible payment types:

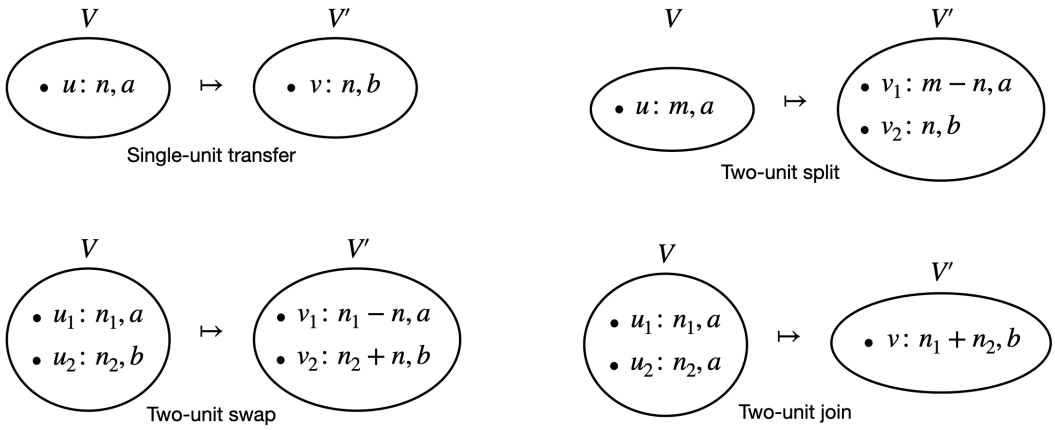


**Fig. 7.** Composition irreducible payments.

1. *General split*, where a single monetary unit $u$ with value $n$ and bearer $a$ is converted into a set of units $v_1, v_2, \ldots, v_k$ with values $n_1, n_2, \ldots, n_k$, and bearers $b_1, b_2, \ldots, b_k$, respectively, where $n = n_1 + \ldots + n_k$ (Fig. 8).
2. *General join*, where a set of units $u_1, u_2, \ldots, u_k$ with values $n_1, n_2, \ldots, n_k$, and bearers $a_1, a_2, \ldots, a_k$, respectively, is converted to a single unit $v$ with value $n_1 + n_2 + \ldots + n_k$ and bearer $a$ (Fig. 8).
3. *General swap*, where a set of units $u_1, u_2, \ldots, u_k$ with values $n_1, n_2, \ldots, n_k$, and bearers $a_1, a_2, \ldots, a_k$, respectively, is converted to a set of units $v_1, v_2, \ldots, v_\ell$ with values $m_1, m_2, \ldots, m_\ell$, and bearers $b_1, b_2, \ldots, b_\ell$, respectively, where $n_1 + \ldots + n_k = m_1 + m_2 + \ldots + m_\ell$ (Fig. 8).

General swap $P$ is irreducible only if $\sigma(W') \neq \sigma(W)$ for every non-empty proper subsets $W \subset V$ and $W' \subset V'$.

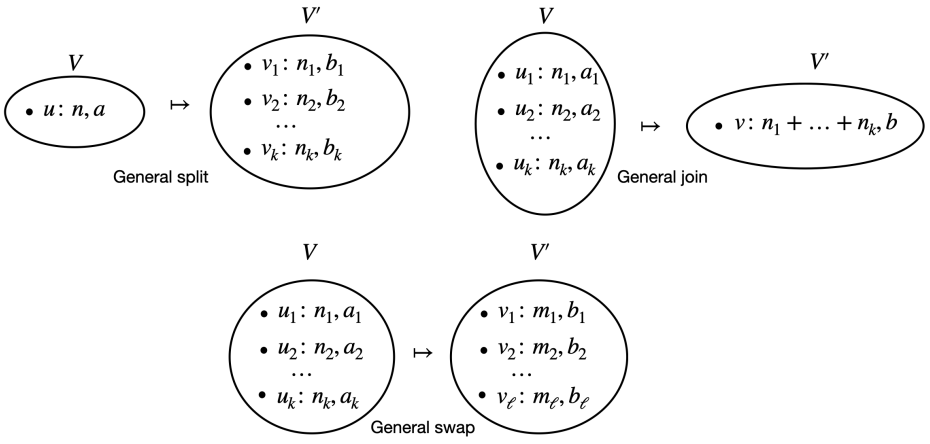


**Fig. 8.** Coproduct irreducible payments.

## 8 Parallel Decomposability of e-Money

So far, we have studied unitwise-decomposability. In this section, we study a more general form of decomposability of e-money, where a money scheme is implemented as a parallel decomposition of two state machines. The states of the component machines are not necessarily money distributions. The only assumption we make is that the pair of states can somehow be interpreted as a money distribution.

The two component machines model two physically separated machines. Every payment is executed by sending two separate commands $t_1$ and $t_2$ to corresponding components through communication channels. As physical communication channels may delay and lose information, we assume that it is not possible to guarantee that both commands are received.

If only $t_1$ is received, then the other component does nothing, i.e. executes the identity operation 1. We want the decomposition to be organized in a way that executing $(t_1, 1)$ or $(1, t_2)$ instead of $(t_1, t_2)$ also corresponds to a legal payment of the implemented money scheme. We call such a property as *atomicity* of the decomposition.

### 8.1 Decomposability of State Machines

**Definition 36 (State Machine (Standard Definition)).** A *state machine* is a triple $(S, I, \delta)$, where $S$ is a non-empty set of states, $I$ is a set of inputs, and $\delta\colon I \times S \to S$ is a state transition function. There is one and only one input $1 \in I$ such that $\delta(1, s) = s$ for every $s \in S$.

**Definition 37 (State Machine (Standard Definition)).** A *state machine* is a pair $(S, T)$, where $S$ is a non-empty set of states, $T$ is a set of state transitions. Every transition $t \in T$ is a function $t\colon S \to S$. There is one and only one identity transition $1 \in T$, such that $1(s) = s$ for every $s \in S$.

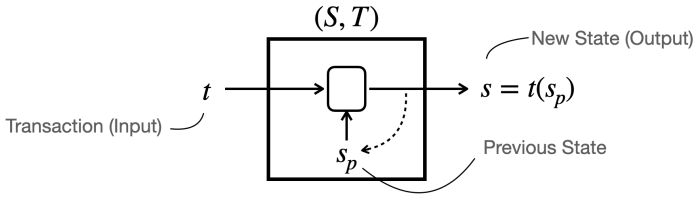


**Fig. 9.** State machine as a state transition system.

Definitions 36 and 37 are equivalent. Given $(S, T)$, we define $(S, I, \delta)$ with $I = T$ and $\delta(i, s) = i(s)$ for every $i \in I$ and $s \in S$. Given $(S, I, \delta)$, we define $T = I$ and $t(s) = \delta(t, s)$. In this work, we will use the alternative definition Def. 37, because, in definitions and proofs, it allows for algebraic expressions that are easier to handle. Every statement and result about state machines with the alternative definition can be translated to the corresponding statements and results about conventional state machines and vice versa.

**Definition 38 (Generating Set).** Given a state machine $(S, T)$, a subset $T_0 \subseteq T$ is a *generating set*, if for every transition $1 \neq t \in T$ there exist $t_1, t_2, \ldots, t_k \in T_0$ so that $t = t_1 \circ t_2 \circ \ldots \circ t_k$.

**Definition 39 (Anti-Commutator Graph).** Given a state machine $(S, T)$, an *anti-commutator graph of* $T_0$ is a graph $(T_0, \wr)$, where $\wr$ is a binary relation on $T_0$ so that $t \wr t'$ iff $t \circ t' \neq t' \circ t$.

**Definition 40 (Implementation of a State Machine).** An *implementation of a state machine* $(S, T)$ *by a state machine* $(S', T')$ is a pair $(\pi, \psi)$ of maps $\pi\colon S' \to S$ (surjective) and $\psi\colon T \to T'$ so that for every $t \in T$:

$$t \circ \pi = \pi \circ \psi(t) \ , \tag{23}$$

i.e., $t(\pi(s')) = \pi(\psi(t)(s'))$ for every $s' \in S'$.

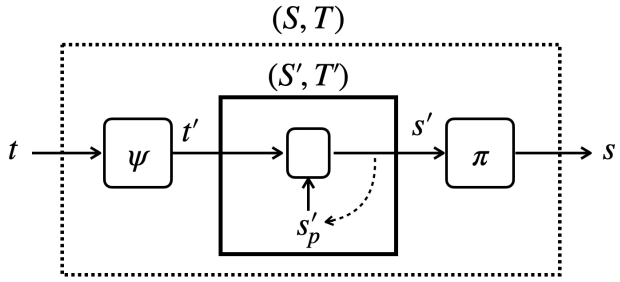


**Fig. 10.** Implementation of $(S, T)$ by $(S, T)$.

**Definition 41 (Direct Product of State Machines).** A *direct product of state machines* $(S_1, T_1)$ *and* $(S_2, T_2)$ is a state machine $(S, T)$, where $S = S_1 \times S_2$, $T = T_1 \times T_2$, and

$$(t_1, t_2)(s_1, s_2) = (t_1(s_1), t_2(s_2))$$

for every $s_1 \in S_1, s_2 \in S_2, t_1 \in T_1, t_2 \in T_2$.

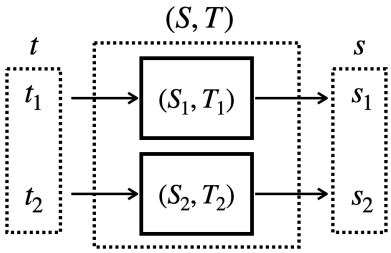


**Fig. 11.** Direct product $(S, T)$ of $(S_1, T_1)$ and $(S_2, T_2)$.

For a direct product of state machines, the identity transition is $(1, 1)$.

Algebraic decomposition theory of state machines was developed by Hartmanis and Stearns [38]. The definition of parallel composition is equivalent to that of *parallel full-decomposition* presented by Jóźwiak [39].

**Definition 42 (Parallel Decomposition of a State Machine).** A *parallel decomposition of a state machine* $(S, T)$ is an implementation $(\pi, \psi)$ of $(S, T)$ by the direct product of state machines $(S_1, T_1)$ and $(S_2, T_2)$.

Let us denote $\psi(t) = (\psi_1(t), \psi_2(t))$.

**Definition 43 (Trivial Decomposition of a State Machine).** A parallel decomposition $(\pi, \psi)$ of a state machine $(S, T)$ is *trivial*, if either:

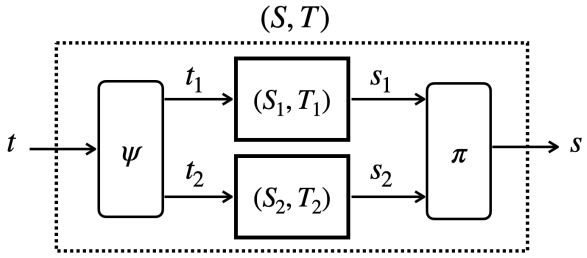


**Fig. 12.** Parallel decomposition $(\pi, \psi)$ of $(S_1, T_1)$ and $(S_2, T_2)$.

- $t \circ \pi = \pi \circ (\psi_1(t), 1)$ for all $t \in T$, or
- $t \circ \pi = \pi \circ (1, \psi_2(t))$ for all $t \in T$.

**Theorem 9 (Indecomposability).** *If $(\pi, \psi)$ is a parallel decomposition of a state machine $(S, T)$ by the direct product of state machines $(S_1, T_1)$ and $(S_2, T_2)$ such that $(S, T)$ has a generating set $T_0$ with connected anti-commutator graph, and for every $t \in T_0$, either $t \circ \pi = \pi \circ (\psi_1(t), 1)$ or $t \circ \pi = \pi \circ (1, \psi_2(t))$, then the decomposition is trivial.*

**Proof:** If $t_1, t_2 \in T_0$, $t_1 \circ \pi = \pi \circ (\psi_1(t_1), 1)$, and $t_2 \circ \pi = \pi \circ (1, \psi_2(t_2))$, then $t_1$ and $t_2$ commute, i.e. $t_2 \circ t_1 = t_1 \circ t_2$. Indeed:

$$\begin{aligned} t_2 \circ t_1 \circ \pi &= t_2 \circ \pi \circ (\psi_1(t_1), 1) = \pi \circ (1, \psi_2(t_2)) \circ (\psi_1(t_1), 1) \\ &= \pi \circ (\psi_1(t_1), 1) \circ (1, \psi_2(t_2)) = t_1 \circ \pi \circ (1, \psi_2(t_2)) \\ &= t_1 \circ t_2 \circ \pi, \end{aligned}$$

and hence, $t_2 \circ t_1 = t_1 \circ t_2$ because $\pi$ is surjective. Therefore, if $t_1, t_2 \in T_0$ and $t_1 \wr t_2$ , then either:

- $t_1 \circ \pi = \pi \circ (\psi_1(t_1), 1)$ and $t_2 \circ \pi = \pi \circ (\psi_1(t_2), 1)$, or
- $t_2 \circ \pi = \pi \circ (1, \psi_2(t_2))$ and $t_2 \circ \pi = \pi \circ (1, \psi_2(t_2))$,

but the case of $t_1 \circ \pi = \pi \circ (\psi_1(t_1), 1)$ and $t_2 \circ \pi = \pi \circ (1, \psi_2(t_2))$ is impossible because of the argument above.

As the anti-commutator graph of $T_0$ is connected, either $t \circ \pi = \pi \circ (\psi_1(t), 1)$ for all $t \in T_0$, or $t \circ \pi = \pi \circ (1, \psi_2(t))$ for all $t \in T_0$, and hence, the decomposition is trivial. □

### 8.2 Error-Tolerant Implementation and Atomicity

Let $(\pi, \psi)$ be an implementation of a state machine $(S, T)$ by a state machine $(S', T')$. We model errors in a way that instead of executing a given transistion $t \in T$, a somewhat different transistion is executed. Design goals for the implementation of $(S, T)$ define which types of errors are possible. For every transistion $t \in T$ a set $\mathcal{E}(t) \subseteq T$ of acceptable erroneous transistions is defined. Though, $t$ itself is not erroneous, we assume that $t \in \mathcal{E}(t)$. This suggests the following definition (Def. 44).

**Definition 44 (Acceptable Error).** An *acceptable error* is a function $\mathcal{E} \colon T \to 2^T$.

The source and cause of errors are implementation errors, i.e. erroneous behaviour in $(S, T)$ is caused by erroneous behavior of the implementation machine $(S', T')$. For every transistion $t' \in T'$, we define the set $\mathcal{E}'(t') \subseteq T'$ of possible erroneous transistions that may be executed instead of $t'$. Reasonable definition (specification) of $\mathcal{E}'(t')$ depends and is inspired from the ways we may try to implement $(S', T')$ as a physical machine. For example, the transistions as command messages might be lost or delayed arbitrarily. This suggests the following definition (Def. 45).

**Definition 45 (Implementation Error).** An *implementation error* is a function $\mathcal{E}' \colon T' \to 2^{T'}$.

Implementation of $(S,T)$ by $(S',T')$ is error tolerant if possible implementation errors in $(S',T')$ only cause allowed errors in $(S,T)$. This suggests the following definition (Def. 46).

**Definition 46 (Error Tolerance).** An implementation $(\pi,\psi)$ is $(\mathcal{E},\mathcal{E}')$-*error tolerant* if for every transition $t \in T$:

$$\pi \circ \mathcal{E}'(\psi(t)) \subseteq \mathcal{E}(t) \circ \pi \ , \tag{24}$$

i.e. for every $t \in T$ and $t' \in \mathcal{E}'(\psi(t))$, there exists $\bar{t} \in \mathcal{E}(t)$, such that $\pi(t'(s')) = \bar{t}(\pi(s'))$ for every state $s' \in S'$.

Atomicity is a form of error tolerance when the state machine $(S,T)$ implemented by the direct product of state machines $(S_1,T_1)$ and $(S_2,T_2)$, considering that for any pair $(t_1,t_2) \in T_1 \times T_2$ of transitions it may happen that only one of these transistions is executed in the corresponding component machine. This is inspired by potential physical implementation, where $(S_1,T_1)$ and $(S_2,T_2)$ are implemented as separate machines accessible via communication lines with message delay or loss. This suggests the following definition (Def. 47).

**Definition 47 (Atomic Decomposition).** A parallel decomposition $(\pi,\psi)$ of a state machine $(S,T)$ by the direct product of state machines $(S_1,T_1)$ and $(S_2,T_2)$ is *atomic with respect to acceptable error* $\mathcal{E} \colon T \to 2^T$ (or simply $\mathcal{E}$-*atomic*) if it is $(\mathcal{E},\mathcal{E}')$-error tolerant, where $\mathcal{E}'(t_1,t_2) = \{(1,t_2),(t_1,1)\}$ for every $(t_1,t_2) \in T_1 \times T_2$.

The implementation error model $\mathcal{E}'(t_1,t_2) = \{(1,t_2),(t_1,1)\}$ is considered to be the same in all cases of atomic decompositions studied in this paper.

The model $\mathcal{E}$ of allowed errors varies from case to case but for decomposition of money schemes, it follows the same principle. It $t \in T$ represents a payment where $a \in \mathfrak{B}$ pays the total value $n$ to $b \in \mathfrak{B}$, then any $\bar{t} \in \mathcal{E}(t)$ may only represent a partial payment where somewhat smaller value $n' \leqslant n$ is payed by $a$ to $b$. This means, that communication losses or delays must not result in larger payments or payments to non-intended payees, etc.

### 8.3 Indecomposability of the Account Money Scheme

- *States*: tuples $(v_1,v_2,\ldots,v_m) \in \mathbb{N}^m$ with $v_1+v_2+\ldots+v_m = v$, where $v \geqslant 1$ is total amount of money.
- *Transitions*: all transformations $t_{i,j,n} \colon \mathbb{N}^m \to \mathbb{N}^m$ such that $i,j \in \{1,\ldots,m\}$, $i \neq j$, $n \in \mathbb{N}$, and $t_{i,j,n}(v_1,v_2,\ldots,v_m) = (v'_1,v'_2,\ldots,v'_m)$ , where:
  - $v'_i = v_i - n$ , if $v_i \geqslant n$, otherwise $v'_i = v_i$
  - $v'_j = v_j + n$ , if $v_i \geqslant n$, otherwise $v'_j = v_j$
  - $v'_k = v_k$ , if $k \notin \{i,j\}$ .
- *Acceptable Error Mode* $\mathcal{E}_0$: $\mathcal{E}_0(t_{i,j,n}) = \{t_{i,j,n},1\}$ for every transition $t_{i,j,n}$
- *Acceptable Error Mode* $\mathcal{E}_1$: $\mathcal{E}_1(t_{i,j,n}) = \{t_{i,j,n'} \colon 0 \leqslant n' \leqslant n\}$ for every transition $t_{i,j,n}$
- *Elementary Transitions*: transitions of type $t_{i,j,1}$. Obviously, elementary transitions form a generating set $T_0$. For elementary transitions, $\mathcal{E}_0(t_{i,j,1}) = \mathcal{E}_1(t_{i,j,1}) = \{t_{i,j,1},1\}$.

**Lemma 4.** *The anti-commutator graph of the generating set $T_0$ of an account-based money scheme is connected.*

**Proof:** Elementary transitions $t_{i,j,1}$ and $t_{j,k,1}$ do not commute. Indeed, let $s_i$ be a state where $v_i = 1$ and $v_j = 0$. Let $s_j = t_{i,j,1}(s_i)$. Hence, $t_{j,k,1}(s_i) = s_i$ and $t_{j,k,1}(s_j) \neq s_j$ which implies:

$$t_{j,k,1}(t_{i,j,1}(s_i)) \neq s_j = t_{i,j,1}(s_i) = t_{i,j,1}(t_{j,k,1}(s_i)) \ .$$

Hence, for every two elementary transitions $t_{i,j,1}, t_{k,l}$ there is a two-step chain $t_{i,j,1} \wr t_{j,k,1} \wr t_{k,l,1}$ in the anti-commutator graph, and hence, the graph is connected. □

**Lemma 5.** *Let $(\pi, \psi)$ be an $\mathcal{E}_1$-atomic parallel decomposition of the account-based money scheme $(S, T)$ by the direct product of state machines $(S_1, T_1)$ and $(S_2, T_2)$. Then for every elementary transition $t = t_{i,j,1} \in T_0$ , either $t \circ \pi = \pi \circ (\psi_1(t), 1)$ or $t \circ \pi = \pi \circ (1, \psi_2(t))$.*

**Proof:** If $\pi \circ (\psi_1(t), 1) = \pi \circ (1, \psi_2(t)) = \pi$ , then:

$$t \circ \pi = \pi \circ (\psi_1(t), \psi_2(t)) = \pi \circ (\psi_1(t), 1) \circ (1, \psi_2(t)) = \pi \circ (1, \psi_2(t)) = \pi$$

and hence, $t = 1$ because $\pi$ is surjective, a contradiction. By atomicity, $\{\pi \circ (\psi_1(t), 1), \pi \circ (1, \psi_2(t))\} \subseteq \mathcal{E}_0(t_{i,j,1}) = \{t \circ \pi, \pi\}$, and hence, either $t \circ \pi = \pi \circ (\psi_1(t), 1)$ or $t \circ \pi = \pi \circ (1, \psi_2(t))$. □

**Corollary 3.** *Every $\mathcal{E}_1$-atomic parallel decomposition of the account-based money scheme by the direct product of state machines $(S_1, T_1)$ and $(S_2, T_2)$ is trivial.*

**Proof:** Immediate Corollary from Lemmas 4 and 5, and Theorem 9. □

### 8.4 Indecomposability of the Bitcoin Money Scheme

- *Supporting data structures*: A universe $\mathcal{U}$ of potential UTXOs that is assumed to be a totally ordered infinite set. By total ordering, we mean that every non-empty subset has the smallest element.
- *States*: Every state is a money distribution, i.e. consists of a finite set $U \subset \mathcal{U}$, a value function $\nu \colon U \to \mathbb{N}$, and a bearer function $\beta \colon U \to \mathbb{N}$.
- *Transition*: transformation $t$ that deletes from $U$ a subset $D \subseteq U$ and adds a subset $C \subset \mathcal{U} \backslash U$ to $U$, so that every element of $C$ is larger (in terms of the total ordering of $\mathcal{U}$) that any element of $U$. It is also assumed that:
$$\sigma(C) = \sum_{c \in C} \nu(c) = \sum_{d \in D} \nu(d) = \sigma(D) \ , \tag{25}$$
i.e. the total money in the system is preserved. The transition also defines the bearers of new UTXOs. If $C \not\subseteq U$, then nothing is done, i.e. then $t$ behaves like the identity transistion 1.
- *Acceptable Error Mode $\mathcal{E}_0$*: $\mathcal{E}_0(t) = \{t, 1\}$ for every transition $t$
- *Acceptable Error Mode $\mathcal{E}_1$*: $\mathcal{E}_1(t)$ consists of all partial transitions that delete a subset $D' \subseteq D$ and creates a subset $C' \subseteq C$ so that $\sigma(C') = \sigma(D')$ in terms of the function $\nu$ as updated by the transition (equation (25)).
- *Elementary Transitions (Set $T_0$)*: $\oplus$-irreducible Bitcoin payments (Sect. 7.5) – transitions with $D \neq \emptyset$ and $C$ such that there exist no proper subsets $\emptyset \neq D' \subset D$ and $\emptyset \neq C' \subset C$ with $\nu(D') = \nu(C')$. For elementary transitions $t$, $\mathcal{E}_0(t) = \mathcal{E}_1(t) = \{t, 1\}$.

**Lemma 6.** *The anti-commutator graph of the generating set $T_0$ of a Bitcoin money scheme is connected.*

**Proof:** Let $t, t' \in T_0$ such that $t$ deletes a set $D$ and creates a set $C$, and $t'$ deletes a set $D'$ and creates a set $C'$. Let $d \in D$ and $d' \in D'$ be two arbitrary elements. We define a third transistion $t''$ as a two-unit join (Sect. 7.5): $t''$ deletes the subset $\{d, d'\}$ and creates a new UTXO $c \in \mathcal{U}$ that is greater (in terms of the total order) that any UTXOs in $U, C, D, C', D'$.

It is easy to see that $t \wr t''$ because as the sets $D$ and $\{d, d'\}$ contain a common element $d$, only one of these transistions can be executed. Hence, $t(t'(s)) = t'(s)$ and $t'(t(s)) = t(s)$ for every state $s$. Analogously, $t'' \wr t'$.

Hence, for every two elementary transitions $t, t'$ there is a two-step chain $t \wr t'' \wr t'$ in the anti-commutator graph, and hence, the graph is connected. □

**Lemma 7.** *Let $(\pi, \psi)$ be an $\mathcal{E}_1$-atomic parallel decomposition of the Bitcoin money scheme $(S, T)$ by the direct product of state machines $(S_1, T_1)$ and $(S_2, T_2)$. Then for every elementary transition $t \in T_0$ , either $t \circ \pi = \pi \circ (\psi_1(t), 1)$ or $t \circ \pi = \pi \circ (1, \psi_2(t))$.*

**Proof:** Follows proof of Lemma 5, please see there. □

**Corollary 4.** *Every $\mathcal{E}_1$-atomic parallel decomposition of the Bitcoin money scheme by the direct product of state machines $(S_1, T_1)$ and $(S_2, T_2)$ is trivial.*

**Proof:** Immediate Corollary from Lemma 6 and 7, and Theorem 9.. □

## 9 Weak Parallel Decomposability of e-Money

In this section, we generalize the indecomposability results presented in Sect. 8 to the so-called weak parallel decomposition, where the input function $\psi$ depends on the states $s_1$ and $s_2$ of the component machines.

In the context of money scheme implementations, such generalization takes into account the possibility that a user's wallet may have information about how much money the user has in both components of the system, and this might influence the choice of transitions $t_1, t_2$ (outputs of $\psi$) enacted at the components.

We show that in a wide class of money schemes (quasi-complete money schemes) that follow atomic weak parallel decompositions, the payments must preserve the total monetary value of the components, i.e., there is no flow of money from one component to another. For example, there exist atomic weak parallel decompositions of the account scheme, but only in such a way that every account $a$ is represented as a pair of sub-accounts $a_1, a_2$ in the components so that the value $\nu(a)$ of $a$ is the sum $\nu(a_1) + \nu(a_2)$ of the value of sub-accounts.

### 9.1 Weak Implementations

Weak implementations generalize the notion of implementations by allowing the input function $\psi$ to use information about the previous state $s'_p$ of the implementing state machine.

**Definition 48 (Weak Implementation).** A *weak implementation of a state machine $(S, T)$ by a state machine $(S', T')$* is a surjective map $\pi\colon S' \to S$ such that for all $t \in T$ and $s' \in S'$, there exists $t' \in T'$ such that $\pi(t'(s')) = t(\pi(s'))$, i.e.:

$$\forall t \in T\, \forall s' \in S'\, \exists t' \in T'\colon \pi(t'(s')) = t(\pi(s'))$$

Similar to the definition of implementation, we can express $t'$ via a function $t' = \psi(t, s'_p)$ (Fig. 13).

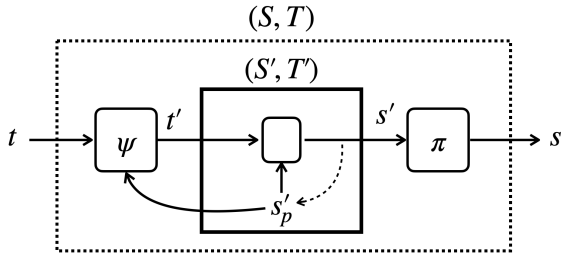


**Fig. 13.** Weak implementation.

As we do not have the universal transition map $\psi\colon T \to T'$ in weak implementations, we cannot directly use Def. 46. Therefore, we have to provide a more general definition for error tolerance of weak implementations (Def. 49).

**Definition 49 (Error Tolerance of a Weak Implementation).** A weak implementation $\pi$ is $(\mathcal{E}, \mathcal{E}')$-*error tolerant* if for all transitions $t \in T$, state $s' \in S'$, $t' \in T'$ and $\overline{t'} \in \mathcal{E}'(t')$ such that

$$t(\pi(s')) = \pi(t'(s'))$$

there exists $\overline{t} \in \mathcal{E}(t)$ such that

$$\overline{t}(\pi(s')) = \pi(\overline{t'}(s')).$$

**Definition 50 (Weak Parallel Decomposition).** A *weak parallel decomposition of a state machine* $(S, T)$ is a weak implementation $\pi$ of $(S, T)$ by the direct product of state machines $(S_1, T_1)$ and $(S_2, T_2)$. (Fig. 14)

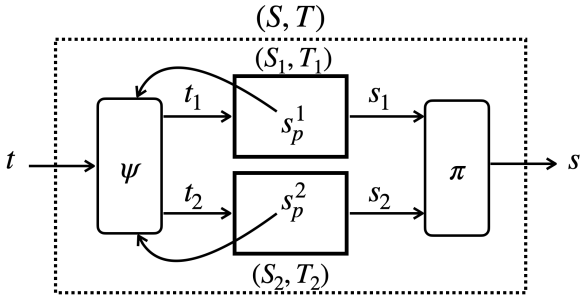


**Fig. 14.** Weak parallel decomposition.

The atomicity condition for weak parallel decomposition also needs some adjustment compared to the atomicity of parallel decomposition (Def. 47). If $t$ is a non-trivial transition that is implemented in the state $(s_1, s_2)$ as a single-component transition such that $t(\pi(s_1, s_2)) = \pi(t_1(s_1), s_2)$, its action does not depend on the state $s_2$ of the other component. This introduces two additional requirements (A2, A3) in Def. 51.

**Definition 51 (Atomic weak parallel decomposition).** A weak parallel decomposition $\pi$ of a state machine $(S, T)$ by the direct product of state machines $(S_1, T_1)$ and $(S_2, T_2)$ is *atomic with respect to an acceptable error* $\mathcal{E} \colon T \to 2^T$ (or simply $\mathcal{E}$-atomic) if:

A1: $\pi$ is $(\mathcal{E}, \mathcal{E}')$-error tolerant, where $\mathcal{E}'(t_1, t_2) = \{(1, t_2), (t_1, 1)\}$ for all $t_1 \in T_1$ and $t_2 \in T_2$.
A2: if $\pi(s_1, s_2) \neq t(\pi(s_1, s_2)) = \pi(t_1(s_1), s_2)$, then

$$\forall s_2' \colon \quad \pi(s_1, s_2') \neq t(\pi(s_1, s_2')) = \pi(t_1(s_1), s_2').$$

A3: if $\pi(s_1, s_2) \neq t(\pi(s_1, s_2)) = \pi(s_1, t_2(s_2))$, then

$$\forall s_1' \colon \quad \pi(s_1', s_2) \neq t(\pi(s_1', s_2)) = \pi(s_1', t_2(s_2)).$$

**Lemma 8.** *Given an atomic weak parallel decomposition $\pi$ of a state machine $(S, T)$ by the direct product of state machines $(S_1, T_1)$ and $(S_2, T_2)$. Then, for all $t \in T$, $t_1 \in T_1$, $t_2 \in T_2$, $s_1 \in S_1$, $s_2 \in S_2$, such that $t(\pi(s_1, s_2)) = \pi(t_1(s_1), t_2(s_2))$, there exist $t^1, t^2 \in T$ so that:*

$$t^1(\pi(s_1, s_2)) = \pi(t_1(s_1), s_2) \quad \textit{and} \quad t^2(\pi(s_1, s_2)) = \pi(s_1, t_2(s_2)).$$

**Proof:** As $(t_1(s_1), s_2) = (t_1, 1)(s_1, s_2)$ and $(s_1, t_2(s_2)) = (1, t_2)(s_1, s_2)$, it follows from A1 that there exist $t^1, t^2 \in \mathcal{E}(t)$ so that:

$$t^1(\pi(s_1, s_2)) = \pi(t_1(s_1), s_2) \quad \text{and} \quad t^2(\pi(s_1, s_2)) = \pi(s_1, t_2(s_2)).$$

The statement follows due to $\mathcal{E}(t) \subseteq T$. □

### 9.2 Quasi-Complete Money Schemes

The main decomposition theorem presented in this section is a about a wide class of money schemes, so called *quasi-complete* money schemes. Intuitively, a quasi-complete money scheme is a money scheme where every payment is associated with a payer $a$ and an amount $n$, and where every account at any state can be emptied via a finite sequence of payments. Also, if a payer has not enough money for executing a payment, then nothing happens. A formal definition is given in Def. 52. Quasi-completeness is an intuitive property that most of the practical money schemes are expect to have. Hence, assuming quasi-completeness does not practically reduce the class of money schemes the theory applies to.

**Definition 52 (Quasi-Complete Money Scheme).** A money scheme $(\mathbb{M}, \mathbb{P})$ with a bearer set $\mathfrak{B}$ is *quasi-complete* if for all $P \in \mathbb{P}$ and $a \in \mathfrak{B}$:

- there is $n \in \mathbb{N}$ so that for all $M \in \mathbb{M}$ with $P(M) \neq M$:
  - $\sigma(M, a) - \sigma(P(M), a) = n$
  - $\sigma(P(M), b) - \sigma(M, b) \geqslant 0$ for all $a \neq b \in \mathfrak{B}$
- for all $M \in \mathbb{M}$, there is a finite composition $\Pi = P^1 \circ \ldots \circ P^m$ of $P^i \in \mathbb{P}$ such that $\sigma(\Pi(M), a) = 0$.

Hence, in a quasi-complete money scheme, each payment $P$ has parameters $a$ and $n$. By $T_{a,n}$, we denote the set of all payments with the parameters $a, n$. By definition, we have that $T_{a,0} = \{1\}$.

By $\Pi_{a,n,M}$, we denote the set of all finite compositions $\Pi = P^1 \circ \ldots \circ P^m$ of payments $P^i \in \mathbb{P}$ such that $\sigma(M, a) - \sigma(\Pi(M), a) = n$.

For implementations, the allowed error mode $\mathcal{E}_1$ of a quasi-complete money scheme is defined as follows.

**Definition 53 (Allowed Errors for Quasi-Complete Money Schemes).** For each $t \in T_{a,n}$, the set $\mathcal{E}_1(t)$ is the set of all payments $t' \in T_{a,n'}$ such that $0 \leqslant n' \leqslant n$, and for all bearers $b \neq a$ and $M \in \mathbb{M}$:

$$0 \leqslant \sigma(t'(M), b) - \sigma(M, b) \leqslant \sigma(t(M), b) - \sigma(M, b).$$

**Lemma 9.** *Given a money scheme $(\mathbb{M}, \mathbb{P})$ with bearer set $\mathfrak{B}$, $M \in \mathbb{M}$, $a \in \mathfrak{B}$, $t \in \Pi_{a,n,M}$, $t' \in \Pi_{a,n',M}$, and $t(M) = t'(M)$. Then, $n = n'$.*

**Proof:** By assumptions, $n = \sigma(M, a) - \sigma(t(M), a)$ and $n' = \sigma(M, a) - \sigma(t'(M), a)$. As $t(M) = t'(M)$, it follows that $n = n'$. □

### 9.3 Weak Parallel Decomposition of Quasi-Complete Money Schemes

**Lemma 10.** *Let $\pi$ be an $\mathcal{E}_1$-atomic weak parallel decomposition of a state machine $(S, T)$ that implements a quasi-complete money scheme with bearer set $\mathfrak{B}$ by the direct product of state machines $(S_1, T_1)$ and $(S_2, T_2)$. Let $a \in \mathfrak{B}$, $s_1 \in S_1$, $s_2 \in S_2$, $t \in T$, $t_1 \in T_1$, and $t_2 \in T_2$ such that $t(\pi(s_1, s_2) = \pi(t_1(s_1), t_2(s_2)))$. Then:*

- *If $\sigma(\pi(s_1, s_2), a) - \sigma(\pi(t_1(s_1), s_2), a) = n_1 > 0$, then:*

$$\forall s_2' \in S_2 \colon\ \sigma(\pi(s_1, s_2'), a) - \sigma(\pi(t_1(s_1), s_2'), a) = n_1.$$

- *If $\sigma(\pi(s_1, s_2), a) - \sigma(\pi(s_1, t_2(s_2)), a) = n_2 > 0$, then:*

$$\forall s_1' \in S_1 \colon\ \sigma(\pi(s_1', s_2), a) - \sigma(\pi(s_1', t_2(s_2)), a) = n_2.$$

**Proof:** Let $\sigma(\pi(s_1, s_2), a) - \sigma(\pi(t_1(s_1), s_2), a) = n_1 > 0$. Due to Lemma 8, there exists $t^1 \in T$, such that $t^1(\pi(s_1, s_2)) = \pi(t_1(s_1), s_2)$. From $n_1 > 0$ it follows that $\pi(t_1(s_1), s_2) \neq \pi(s_1, s_2)$. Due to atomicity (A2):

$$\forall s_2' \in S_2 \colon\ \pi(s_1, s_2') \neq t^1(\pi(s_1, s_2')) = \pi(t_1(s_1), s_2'). \tag{26}$$

Hence, by taking $s = \pi(s_1, s_2)$, we have that $t^1(s) \neq s$ and $\sigma(s, a) - \sigma(t^1(s), a) = n_1$. By quasi-completeness, there exists $n \in N$ such that $\sigma(s', a) - \sigma(t^1(s'), a) = n$ for all $s' \in S$ such that $t^1(s') \neq s'$.

Hence, $n = n_1$. Therefore, due to (26), we have that $t^1(\pi(s_1, s_2')) \neq \pi(s_1, s_2')$ for all $s_2' \in S_2$ and hence, by taking $s' = \pi(s_1, s_2')$, we have that $\sigma(\pi(s_1, s_2'), a) - \sigma(\pi(t_1(s_1), s_2'), a) = n_1$. The proof of the second if-then statement is similar. □

**Remark:** If $\sigma(\pi(s_1, s_2), a) - \sigma(\pi(t_1(s_1), s_2), a) = 0$, but $\sigma(\pi(s_1, s_2'), a) - \sigma(\pi(t_1(s_1), s_2'), a) = n_1 > 0$ for some $s_2'$, then by applying Lemma 10 to the last inequality, we also have that $\sigma(\pi(s_1, s_2), a) - \sigma(\pi(t_1(s_1), s_2), a) = n_1 > 0$, a contradiction. The same applies for $n_2$. Hence, we have the following Corollary 5.

**Corollary 5.** *Let $\pi$ be an $\mathcal{E}_1$-atomic weak parallel decomposition of a state machine $(S, T)$ that implements a quasi-complete money scheme with bearer set $\mathfrak{B}$ by the direct product of state machines $(S_1, T_1)$ and $(S_2, T_2)$. Let $a \in \mathfrak{B}$ $s_1 \in S_1$, $s_2 \in S_2$, $t \in T$, $t_1 \in T_1$, and $t_2 \in T_2$, so that $t(\pi(s_1, s_2)) = \pi(t_1(s_1), t_2(s_2))$. Then:*

- *If $\sigma(\pi(s_1, s_2), a) - \sigma(\pi(t_1(s_1), s_2), a) = n_1$, then:*

$$\forall s_2' \in S_2 \colon \ \sigma(\pi(s_1, s_2'), a) - \sigma(\pi(t_1(s_1), s_2'), a) = n_1.$$

- *If $\sigma(\pi(s_1, s_2), a) - \sigma(\pi(s_1, t_2(s_2)), a) = n_2$, then:*

$$\forall s_1' \in S_1 \colon \ \sigma(\pi(s_1', s_2), a) - \sigma(\pi(s_1', t_2(s_2)), a) = n_2.$$

**Proof:** See the remark above. □

**Lemma 11.** *Let $\pi$ be an $\mathcal{E}_1$-atomic weak parallel decomposition of a state machine $(S, T)$ that implements a quasi-complete money scheme by the direct product of state machines $(S_1, T_1)$ and $(S_2, T_2)$. Let $s_1 \in S_1$, $s_2 \in S_2$, $t \in T$, $t_1 \in T_1$, and $t_2 \in T_2$ such that $t(\pi(s_1, s_2)) = \pi(t_1(s_1), t_2(s_2))$ and*

$$\sigma(\pi(s_1, s_2), a) - \sigma(t(\pi(s_1, s_2)), a) = n > 0. \tag{27}$$

*Then there exist $n_1, n_2 \in \mathbb{N}$ with $n = n_1 + n_2$, so that:*

- $\sigma(\pi(s_1, s_2), a) - \sigma(\pi(t_1(s_1), s_2), a) = n_1$
- $\sigma(\pi(s_1, s_2), a) - \sigma(\pi(s_1, t_2(s_2)), a) = n_2$

**Proof:** By quasi-completeness and Lemma 8, there exist $n_1, n_2 \in \mathbb{N}$, $t^1 \in T_{a,n_1}$, and $t^2 \in T_{a,n_2}$ such that $t^1(\pi(s_1, s_2)) = \pi(t_1(s_1), s_2)$ and $t^2(\pi(s_1, s_2)) = \pi(s_1, t_2(s_2))$. Note that it is not possible that $\pi(t_1(s_1), s_2) = \pi(s_1, s_2) = \pi(s_1, t_2(s_2))$. Indeed, then by (27), $\pi(s_1, s_2) \neq t(\pi(s_1, s_2)) = \pi(t_1(s_1), t_2(s_2))$, it follows also that $\pi(s_1, t_2(s_2)) \neq \pi(t_1(s_1), t_2(s_2))$. Denote $s_2' = t_2(s_2)$. Then by Lemma 8, there is $\bar{t}^1$ such that $\pi(s_1, s_2') \neq \bar{t}^1(\pi(s_1, s_2')) = \pi(t_1(s_1), s_2')$, which by atomicity (A2) implies

$$\forall s_2'' \colon \pi(s_1, s_2'') \neq \bar{t}^1(\pi(s_1, s_2'')) = \pi(t_1(s_1), s_2'').$$

Now, by taking $s_2'' = s_2$, we get $\pi(s_1, s_2) \neq \bar{t}^1(\pi(s_1, s_2)) = \pi(t_1(s_1), s_2)$, a contradiction. Hence, we have three possibilities:

- $\underline{\pi(t_1(s_1), s_2) = \pi(s_1, s_2) \neq \pi(s_1, t_2(s_2))}$: Then $\pi(s_1, s_2) \neq t^2(\pi(s_1, s_2)) = \pi(s_1, t_2(s_2))$, which by atomicity implies $\forall s_1'' \colon \ \pi(s_1'', s_2) \neq t^2(\pi(s_1'', s_2)) = \pi(s_1'', t_2(s_2))$. By taking $s_1'' = t_1(s_1)$, we get

$$\pi(s_1, s_2) \neq t(\pi(s_1, s_2)) = \pi(t_1(s_1), t_2(s_2)) = t^2(\pi(t_1(s_1), s_2)) = t^2(\pi(s_1, s_2)).$$

  Hence, by denoting $s = \pi(s_1, s_2)$ as $t \in \Pi_{a,n,s}$, $t^2 \in \Pi_{a,n_2,s}$ and $t(s) = t^2(s)$, we have $n = n_2$ by Lemma 9. Hence, $\sigma(\pi(s_1, s_2), a) - \sigma(\pi(s_1, t_2(s_2)), a) = n = n_2$. As $\pi(s_1, s_2) = \pi(t_1(s_1)) = \pi(s_1, s_2)$, we have $\sigma(\pi(s_1, s_2), a) - \sigma(\pi(t_1(s_1), s_2), a) = 0$ and we can take $n_1 = 0$.
- $\underline{\pi(t_1(s_1), s_2) \neq \pi(s_1, s_2) = \pi(s_1, t_2(s_2))}$: Similar to the previous case, $\sigma(\pi(s_1, s_2), a) - \sigma(\pi(s_1, t_2(s_2)), a) = 0$ and $\sigma(\pi(s_1, s_2), a) - \sigma(\pi(t_1(s_1), s_2), a) = n$.

– $\pi(t_1(s_1), s_2) \neq \pi(s_1, s_2) \neq \pi(s_1, t_2(s_2))$: As $\pi(s_1, s_2) \neq t^1(\pi(s_1, s_2)) = \pi(t_1(s_1), s_2)$ and $\pi(s_1, s_2) \neq t^2(\pi(s_1, s_2)) = \pi(s_1, t_2(s_2))$, then by $t^1 \in T_{a,n_1}$ and $t^2 \in T_{a,n_2}$, we have $n_1, n_2 > 0$, and:

$$\sigma(\pi(s_1, s_2), a) - \sigma(\pi(t_1(s_1), s_2), a) = n_1$$
$$\sigma(\pi(s_1, s_2), a) - \sigma(\pi(s_1, t_2(s_2)), a) = n_2$$

By Lemma 10, $\sigma(\pi(t_1(s_1), s_2), a) - \sigma(\pi(t_1(s_1), t_2(s_2)), a) = n_2$. Hence,

$$\begin{aligned} n &= \sigma(\pi(s_1, s_2), a) - \sigma(\pi(t_1(s_1), t_2(s_2)), a) \\ &= \sigma(\pi(s_1, s_2), a) - \sigma(\pi(t_1(s_1), s_2), a) + \sigma(\pi(t_1(s_1), s_2), a) - \sigma(\pi(t_1(s_1), t_2(s_2)), a) \\ &= n_1 + n_2 \end{aligned}$$

□

Let $(S, T)$ be a state machine that implements a quasi-complete money scheme with bearer set $\mathfrak{B}$ (with $|\mathfrak{B}| \geqslant 2$) and $a \in \mathfrak{B}$. Let $\pi$ be an $\mathcal{E}_1$-atomic weak decomposition of $(S, T)$ by the direct product of state machines $(S_1, T_1)$ and $(S_2, T_2)$. Let $s_1 \in S_1$, $s_2 \in S_2$. We define $\sigma_1(s_1, s_2; a)$ as the maximum total amount of money that $a$ can pay to other accounts in state $(s_1, s_2)$ via payments of type $(t, 1)$, see Def. 54. Similarly, we define $\sigma_2(s_1, s_2; a)$ as the maximum total amount of money that $a$ can pay to other accounts in state $(s_1, s_2)$ via payments of type $(1, t)$, see Def. 55.

**Definition 54.** We define $\sigma_1(s_1, s_2; a)$ as the maximum number $n_1$ such that there is a finite composition $t_1 = t_1^1 \circ \ldots \circ t_1^m$ of $t_1^i \in T_1$, such that $\sigma(\pi(s_1, s_2), a) - \sigma(\pi(t_1(s_1), s_2), a) = n_1$.

**Definition 55.** We define $\sigma_2(s_1, s_2; a)$ as the maximum number $n_2$ such that there is a finite composition $t_2 = t_2^1 \circ \ldots \circ t_2^m$ of $t_2^i \in T_2$, such that $\sigma(\pi(s_1, s_2), a) - \sigma(\pi(s_1, t_2(s_2)), a) = n_2$.

**Lemma 12.** *Given be a state machine $(S, T)$ that implements a quasi-complete money scheme with bearer set $\mathfrak{B}$ (with $|\mathfrak{B}| \geqslant 2$) and $a \in \mathfrak{B}$. Then, $\sigma_1(s_1, s_2; a) + \sigma_2(s_1, s_2; a) = \sigma(\pi(s_1, s_2), a)$.*

**Proof:** Let $\sigma(\pi(s_1, s_2), a) = n$. Due to quasi-completeness, there is a composition $t = t_{a,n^m} \circ \ldots \circ t_{a,n^2} \circ t_{a,n^1}$ with $t_{a,n^i} \in T_{a,n^i}$ such that $\sigma(t(\pi(s_1, s_2)), a) = 0$. Hence, from the weak parallel composition condition, for any state $(s_1', s_2')$ and for every $t_{a,n^i}$, there are $t_1^i \in T_1$ and $t_2^i \in T_2$ such that $t_{a,n^i}(\pi(s_1', s_2')) = \pi(t_1^i(s_1'), t_2^i(s_2'))$. We now apply this recursively as follows. For every $i \in \{1, \ldots, m\}$, let:

$$\pi(s_1^{i-1}, s_2^{i-1}) \neq t_{a,n^i}(\pi(s_1^{i-1}, s_2^{i-1})) = \pi(t_1^i(s_1^{i-1}), t_2^i(s_2^{i-1})) \tag{28}$$

where $s_1^0 = s_1$, $s_2^0 = s_2$, $s_1^i = t_1^i(s_1^{i-1})$, and $s_2^i = t_2^i(s_2^{i-1})$. We can assume the inequality in (28) without loss of generality, because whenever $\pi(s_1^{i-1}, s_2^{i-1}) = t_{a,n_i}(\pi(s_1^{i-1}, s_2^{i-1}))$, we can just remove $t_{a,n_i}$ from the composition. From (28) and quasi-completeness it follows that $\sigma(\pi(s_1^{i-1}, s_2^{i-1}), a) - \sigma(t_{a,n^i}(\pi(s_1^{i-1}, s_2^{i-1})), a) = n^i > 0$ and hence, by Lemma 11, there are $n_1^i, n_2^i \in \mathbb{N}$, with $n_i = n_1^i + n_2^i$ so that:

$$\sigma(\pi(s_1^{i-1}, s_2^{i-1}), a) - \sigma(\pi(t_1^i(s_1^{i-1}), s_2^{i-1})), a) = n_1^i$$
$$\sigma(\pi(s_1^{i-1}, s_2^{i-1}), a) - \sigma(\pi(s_1^{i-1}, t_2^i(s_2^{i-1}))), a) = n_2^i.$$

It is easy to see, by telescoping, that $n = n^1 + \ldots + n^m = n_1 + n_2$, where $n_1 = n_1^1 + \ldots + n_1^m$ and $n_2 = n_2^1 + \ldots + n_2^m$. Applying Corollary 5, we can show that

$$\sigma(\pi(s_1, s_2), a) - \sigma(\pi((t_1^m \circ t_1^1)(s_1), s_2), a) = n_1,$$
$$\sigma(\pi(s_1, s_2), a) - \sigma(\pi(s_1, (t_2^m \circ t_2^1)(s_2)), a) = n_2,$$

which implies $n_1 \leqslant \sigma_1(s_1, s_2; a)$ and $n_2 \leqslant \sigma_2(s_1, s_2; a)$. Hence,

$$\sigma(\pi(s_1, s_2), a) = n = n_1 + n_2 \leqslant \sigma_1(s_1, s_2; a) + \sigma_2(s_1, s_2; a).$$

Let $\sigma_1(s_1, s_2; a) = n_1$ and $\sigma_2(s_1, s_2; a) = n_2$. Due to Def. 54, we have that there is a composition $t_1 = t_1^m \circ \ldots \circ t_1^1$ with $t_1^i \in T_1$ such that $\sigma(\pi(s_1, s_2), a) - \sigma(\pi(t_1(s_1), s_2), a) = n_1$. Due to Def. 55, there is a composition $t_2 = t_2^{m'} \circ \ldots \circ t_2^1$ with $t_2^i \in T_2$ such that $\sigma(\pi(s_1, s_2), a) - \sigma(\pi(s_1, t_2(s_2)), a) = n_2$.

Let $s_2^0, s_2^1, s_2^2, \ldots, s_2^{m'} \in S_2$ be states and $n_2^1, n_2^2, \ldots, n_2^{m'} \in \mathbb{N}$ such that $n_2^1 + \ldots + n_2^{m'} = n_2$, so that $s_2^0 = s_2$, $s_2^i = t_2^i(s_2^{i-1})$,

$$\sigma(\pi(s_1, s_2^{i-1}), a) - \sigma(\pi(s_1, s_2^i), a) = \sigma(\pi(s_1, s_2^{i-1}), a) - \sigma(\pi(s_1, t_2^i(s_2^i)), a) = n_2^i$$

and $t^2_{a,n_2^i} \in T_{a,n_2^i}$ be a transition such that $t^2_{a,n_2^i}(\pi(s_1, s_2^{i-1})) = \pi(s_1, t_2^i(s_2^{i-1}))$ for all $i = 1, 2, \ldots, m'$. Hence, $\sigma(\pi(s_1, s_2^{i-1}), a) - \sigma(t^2_{a,n_2^i}(\pi(s_1, s_2^{i-1})), a) = n_2^i$ for all $i = 1, 2, \ldots, m'$. Hence, for the composition $t^2 = t^2_{a,n_2^{m'}} \circ \ldots \circ t^2_{a,n_2^1}$, we have $t^2(\pi(s_1, s_2)) = \pi(s_1, t_2(s_2))$, by telescoping, we obtain:

$$\sigma(\pi(s_1, s_2), a) - \sigma(t^2(\pi(s_1, s_2)), a) = n_2 \tag{29}$$

Similarly, we have a composition $t^1 = t^1_{a,n_1^m} \circ \ldots \circ t^1_{a,n_1^1}$ with $t^1(\pi(s_1, s_2)) = \pi(t_1(s_1), s_2)$ and

$$\sigma(\pi(s_1, s_2), a) - \sigma(t^1(\pi(s_1, s_2)), a) = n_1$$

By (29) and Corollary 5, we have

$$\sigma(\pi(t_1(s_1), s_2), a) - \sigma(t^2(\pi(t_1(s_1), s_2)), a) = n_2 = \sigma(t^1(\pi(s_1, s_2)), a) - \sigma(t^2(t^1(\pi(s_1, s_2)), a).$$

Hence, for the composition $t_2 \circ t_1$, we have

$$\begin{aligned}
\sigma_1(s_1, s_2; a) + \sigma_2(s_1, s_2; a) &= n_1 + n_2 \\
&= \sigma(\pi(s_1, s_2), a) - \sigma(t^1(\pi(s_1, s_2)), a) + \sigma(t^1(\pi(s_1, s_2)), a) - \sigma(t^2(t^1(\pi(s_1, s_2)), a) \\
&= \sigma(\pi(s_1, s_2), a) - \sigma((t^2 \circ t^1)(\pi(s_1, s_2)), a) \\
&\leqslant \sigma(\pi(s_1, s_2), a).
\end{aligned}$$

□

**Corollary 6.** *Given be a state machine $(S, T)$ that implements a quasi-complete money scheme with bearer set $\mathfrak{B}$ (with $|\mathfrak{B}| \geqslant 2$) and $a \in \mathfrak{B}$. Then, $\sigma_1(s_1, s_2; a) = \sigma_1(s_1, s_2'; a)$ and $\sigma_2(s_1, s_2; a) = \sigma_2(s_1', s_2; a)$ for all states $s_1, s_1' \in S_1$ and $s_2, s_2' \in S_2$.*

**Proof:** Let $t_1 = t_1^1 \circ \ldots \circ t_1^m$ of $t_1^i \in T_1$ be a finite composition such that $\sigma(\pi(s_1, s_2), a) - \sigma(\pi(t_1(s_1), s_2), a) = \sigma_1(s_1, s_2; a)$. From Corollary 5, it follows that $\sigma(\pi(s_1, s_2'), a) - \sigma(\pi(t_1(s_1), s_2'), a) = \sigma_1(s_1, s_2; a)$. Hence, $\sigma_1(s_1, s_2; a) \leqslant \sigma_1(s_1, s_2'; a)$. Similarly we can prove that $\sigma_1(s_1, s_2'; a) \leqslant \sigma_1(s_1, s_2; a)$. Hence, $\sigma_1(s_1, s_2; a) = \sigma_1(s_1, s_2'; a)$. The proof of $\sigma_2(s_1, s_2; a) = \sigma_2(s_1', s_2; a)$ is analogous. □

**Theorem 10.** *Given be a state machine $(S, T)$ that implements a quasi-complete money scheme with bearer set $\mathfrak{B}$ (with $|\mathfrak{B}| \geqslant 2$) and $a \in \mathfrak{B}$. Then, $\sigma_1(s_1, s_2) = \sum_{a \in \mathfrak{B}} \sigma_1(s_1, s_2; a)$ and $\sigma_2(s_1, s_2) = \sum_{a \in \mathfrak{B}} \sigma_2(s_1, s_2; a)$ are invariant under transitions, i.e. $\sigma_1(s_1, s_2) = \sigma_1(t_1(s_1), t_2(s_2))$ and $\sigma_2(s_1, s_2) = \sigma_2(t_1(s_1), t_2(s_2))$ for all transistions $t_1 \in T_1$ and $t_2 \in T_2$ for which there exists a transition $t \in T$ such that $t(\pi(s_1, s_2)) = (t_1(s_1), t_2(s_2))$.*

**Proof:** By Lemma 8, there exists a transition $t^1 \in T$ such that $t^1(\pi(s_1, s_2)) = \pi(t_1(s_1), s_2)$. As $t^1$ is a money scheme transition, it preserves total money:

$$\sum_{a \in \mathfrak{B}} \sigma(\pi(s_1, s_2), a) = \sum_{a \in \mathfrak{B}} \sigma(t^1(\pi(s_1, s_2)), a) = \sum_{a \in \mathfrak{B}} \sigma(\pi(t_1(s_1), s_2)), a)$$

From Lemma 12 and Corollary 6, it follows that:

$$\begin{aligned}\sum_{a\in\mathfrak{B}}\sigma_1(s_1,s_2;a)+\sum_{a\in\mathfrak{B}}\sigma_2(s_1,s_2;a) &= \sum_{a\in\mathfrak{B}}\sigma_1(t_1(s_1),s_2;a)+\sum_{a\in\mathfrak{B}}\sigma_2(t_1(s_1),s_2;a)\\ &= \sum_{a\in\mathfrak{B}}\sigma_1(t_1(s_1),s_2;a)+\sum_{a\in\mathfrak{B}}\sigma_2(s_1,s_2;a)\end{aligned}$$

and hence, by Corollary 6:

$$\sigma_1(s_1,s_2)=\sum_{a\in\mathfrak{B}}\sigma_1(s_1,s_2;a)=\sum_{a\in\mathfrak{B}}\sigma_1(t_1(s_1),s_2;a)=\sum_{a\in\mathfrak{B}}\sigma_1(t_1(s_1),t_2(s_2);a)=\sigma_1(t_1(s_1),t_2(s_2))$$

The proof of $\sigma_2(s_1,s_2)=\sigma_2(t_1(s_1),t_2(s_2))$ is similar. □

Therefore, the values $\sigma_1(s_1,s_2)$ and $\sigma_2(s_1,s_2)$ are invariant under payments and can be interpreted as total money in the first and the second component, respectively. Due to Lemma 12, it is also easy to see that:

$$\sigma_1(s_1,s_2)+\sigma_2(s_1,s_2)=\sigma(\pi(s_1,s_2)),$$

which means that the sum of total money in components equals the total money in the system. Hence, if a quasi-complete money scheme enables weak atomic parallel decomposition, the total money in components is invariant and there can be no transfer of value from one component to another.

## 10 Conclusion

There are many differences between the money schemes, but there are some similarities. In both account schemes and in the bitcoin scheme, there is a requirement for coordination, or consensus between multiple monetary units.

With accounts, it's important that the debiting and crediting happens correctly. When these accounts are located in different databases, it means that the parties who control those databases need to agree to the transaction, and then each must adjust their own database correctly, and nearly simultaneously, to avoid the possibility of errors.

With Bitcoin, it's possible to spend multiple UTXO's in the same transaction. This requires a similar kind of coordination, because the transaction is valid only if all of the inputs are still unspent. Bitcoin's blockchain is a single large database, containing all transactions, so Bitcoin validators do not actually perform any kind of coordination. However, if one were to create a different Bitcoin-like scheme, where UTXO's were stored in different databases, say at different banks, it would require the same kind of coordination that is required for account-based payments, in order to spend several of them in a single transaction.

Interestingly, the bill scheme does not share this characteristic. In fact, it is the only possible money scheme that does not. Bills can be stored in different databases, and no coordination between those databases is ever required in order to make payments. Arbitrary payments can be made, by decomposing them into multiple payments of different bills, and these decomposed parts can be processed in parallel. The bill scheme benefits from its simplicity, and so it is much easier to scale to large transaction volumes – just add more databases of bills. This is called horizontal sharding, and any system that can utilize it is easier to scale than one that cannot.

## References

1. Buldas, A., Draheim, D., Nagumo, T., Vedeshin, A.: Blockchain technology: Intrinsic technological and socio-economic barriers. In: Proceedings of FDSE'2020 – the 7th International Conference on Future Data and Security Engineering. LNCS 12466, Springer (2020) 3–27

2. Draheim, D.: Blockchains from an e-governance perspective: Potential and challenges. In: Proceedings of EGOSE'2020 – the 7th International Conference on Electronic Governance and Open Society: Challenges in Eurasia. Communications in Computer and Information Science 1349, Springer (2021) xi–xii
3. Williamson, O.E.: Transaction cost economics: the governance of contractual relations. The Journal of Law & Economics **22**(2) (1979) 233–261
4. Williamson, O.: Transaction cost economics: How it works; where it is headed. De Economist **146** (1998) 23–58
5. Koppenjan, J., Groenewegen, J.: Institutional design for complex technological systems. International Journal of Technology, Policy and Management **5**(3) (2005) 240–257
6. Gomber, P., Koch, J.A., Siering, M.: Digital finance and FinTech: current research and future research directions. Journal of Business Economics **87**(5) (2017) 537–580
7. Gomber, P., Kauffman, R., Parker, C., Weber, B.: On the Fintech revolution: Interpreting the forces of innovation, disruption, and transformation in financial services. Journal of Management Information Systems **35**(1) (2018) 220–265
8. Lee, I., Shin, Y.: Fintech: Ecosystem, business models, investment decisions, and challenges. Business Horizons **61**(1) (2018) 35–46
9. Gomber, P., Kauffman, R., Parker, C., Weber, B.: On the fintech revolution: Interpreting the forces of innovation, disruption, and transformation in financial services. Journal of Management Information Systems **35**(1) (2018) 220–265
10. Rikken, O., Janssen, M., Kwee, Z.: Governance challenges of blockchain and decentralized autonomous organizations. Information Polity **24**(4) (2019) 397–417
11. Filippova, E., Scharl, A., Filippov, P.: Blockchain: An empirical investigation of its scope for improvement. In: Proc. of ICBC'19 .– the 2nd Intl. Conf. on Blockchain. LNCS 11521, Springer (2019) 1–17
12. Filippova, E.: Empirical evidence and economic implications of blockchain as a general purpose technology. In: Proc. of TEMSCON'19 – the 3rd IEEE Conference on Technology, Engineering and Management Conference, IEEE (2019) 1–8
13. Janssen, M., Weerakkody, V., Ismagilova, E., Sivarajah, U., Irani, Z.: A framework for analysing blockchain technology adoption: Integrating institutional, market and technical factors. International Journal of Information Management **50** (2020) 302 – 309
14. Upadhyay, N.: Demystifying blockchain: A critical analysis of challenges, applications and opportunities. International Journal of Information Management **54** (2020) 1–26
15. Malcolm Campbell-Verduyn (ed.): Bitcoin and Beyond: Cryptocurrencies, Blockchains and Global Governance. Routledge (2017)
16. Voshmgir, S.: Disrupting governance with blockchains and smart contracts. Journal of Strategic Change **26**(5) (2017) 499–509
17. Voshmgir, S.: Token Economy – How the Web3 reinvents the Internet, 2nd. ed. BlochainHub Berlin, Berlin (2020)
18. Christine Lagarde: Central Banking and Fintech—A Brave New World? Bank of England Conference, London, 29 September 2017.
19. European Central Bank: ECB intensifies its work on a digital euro, 2nd october (2014) `https://www.ecb.europa.eu/press/pr/date/html/index.en.html`.
20. High-Level Task Force on Central Bank Digital Currency (HLTF-CBDC): Report on a Digital Euro. European Central Bank (2020)
21. European Central Bank: Eurosystem Report on the Public Consultation on a Digital Euro. European Central Bank (2021)
22. Buldas, A., Laur, S.: Knowledge-binding commitments with applications in time-stamping. In: Proc. of PKC'2007 – the 10th Intl. Conf. on Practice and Theory in Public-Key Cryptography. LNCS 4450 (2007) 150–165
23. Buldas, A., Saarepera, M.: Document Verification with Distributed Calendar Infrastructure. US Patent Application Publication No.: US 2013/0276058 A1 (2013)
24. Buldas, A., Kroonmaa, A., Laanoja, R.: Keyless signatures' infrastructure: How to build global distributed hash-trees. In: Proc. of NordSec'2013 – the 18th Nordic Conference on Secure IT Systems. LNCS 8208, Springer (2013)
25. Ansper, A., Buldas, A., Freudenthal, M., Willemson, J.: High-performance qualified digital signatures for X-Road. In: Proc. of NordSec 2013 – the 18th Nordic Conference on Secure IT Systems. LNCS 8208, Springer (2013) 123–138
26. Arne Ansper, Ahto Buldas, J.W.: Cryptographic Algorithms Lifecycle Report 2017. Technical Report Doc. A-101-9, AS Cybernetica (May 2018) Procurer: Information Systems Authority, Republic of Estonia.
27. Martinson, P.: Estonia – the Digital Republic Secured by Blockchain. PricewaterhouseCoopers (2019)

28. Narayanan, A., Clark, J.: Bitcoin's academic pedigree. Communications of the ACM **60**(12) (2017) 36–45
29. Narayanan, A., Clark, J.: Bitcoin's academic pedigree. ACM Queue Magazine **15**(4) (2017) 1–30
30. Yaga, D., Mell, P., Roby, N., Scarfone, K.: Blockchain Technology Overview – NIST.IR.8202. National Institute of Standards and Technology (2018)
31. Nakamoto, S.: Bitcoin: A Peer-to-Peer Electronic Cash System (2008) Available at: `https://bitcoin.org/bitcoin.pdf`.
32. Gavin Wood et al.: Ethereum: A secure decentralised generalised transaction ledger. Ethereum project yellow paper 151 (2014)
33. Szabo, N.: Smart Contracts: Building Blocks for Digital Markets. Nick Szabo (1996)
34. Szabo, N.: Formalizing and securing relationships on public networks. First Monday **2**(9) (1997)
35. Antonopoulos, A.M.: Mastering Bitcoin: Programming the Open Blockchain. O'Reilly (2017)
36. MacLane, S.: Categories for the Working Mathematician. Graduate Texts in Mathematics 5. Springer (1994)
37. Barr, M., Wells, C.: Category Theory for Computing Science. Prentice-Hall (1990)
38. Hartmanis, J., Stearns, R.: Algebraic Structure Theory of Sequential Machines. Prentice-Hall (1966)
39. Jóźwiak, L.: The full-decomposition of sequential machines with the separate realization of the next-state and output functions. Technical Report EUT Report 89-E-222, Eindhoven University of Technology. Faculty of Electrical Engineering. Research Report (1989)