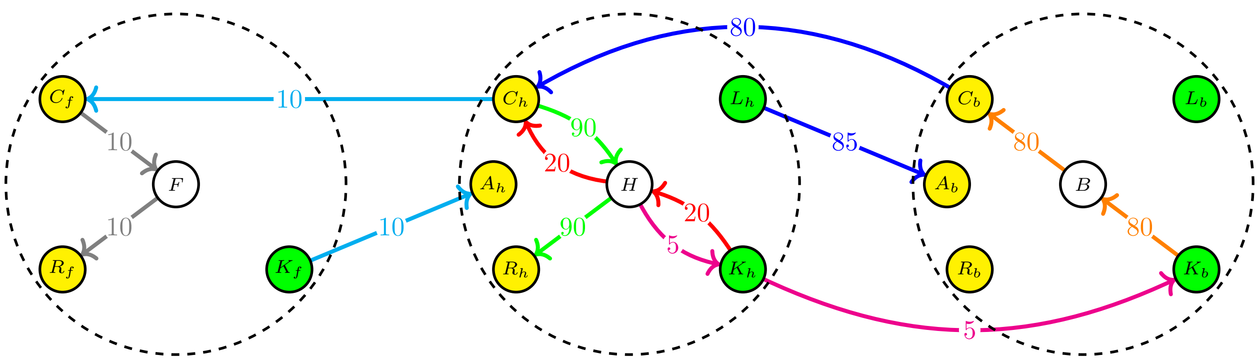
Open Risk White Paper

Connecting the Dots:

Accounting Graphs for Double and Quadruple-Entry Bookkeeping

Authors: Philippos Papadopoulos

November 11, 2025



www.openriskmanagement.com

The open future of risk management

Summary

We develop the concept of an accounting graph as an intuitive and flexible tool that can faithfully represent widely used accounting concepts. Weighted directed multigraphs representation are applied to model the standard double-entry bookkeeping of an individual accounting entity but also consistent accounts of larger economic networks that feature multiple transacting entities (so-called quadruple-entry accounting). We illustrate the concepts with stylized accounting examples. Accounting graphs are promising both as educational tools and towards the application of computational tools on accounting data.

Keywords: Double-entry bookkeeping, Quadruple-entry bookkeeping, Balance Module, Pacioli group, Graph Theory, Graph Algorithms, Network Theory, Multigraph

Further Resources

- The Open Risk Manual is an open online knowledge base covering diverse domains of risk management.
- The Open Risk Academy offers a range of online courses around risk, portfolio management and sustainable finance, which utilize the latest in interactive eLearning tools. Please inquire at **info openriskmanagement com** about eLearning possibilities.
- The Open Risk Commons is a meeting place and primary venue for discussing open source risk models.
- Sustainability.Town a demo of the open source platform Equinox that supports the holistic risk management of diverse sustainability portfolios.
- Open Source Risk Repository is our online repository of libraries, tools and frameworks that support quantitative analysis of diverse risk and portfolio management tasks. Check out in particular the sample project Accounting Graph Environment.

About Open Risk

Open Risk is an independent provider of training and risk analysis tools to the broader financial services community. Our mission is captured by the motto: *The open future of risk management.*
Learn more about our mission at: `www.openriskmanagement.com`

Contents

1 Introduction

"The Principles of Book-keeping by Double Entry constitute a theory which is mathematically by no means uninteresting: it is in fact like Euclid's theory of ratios an absolutely perfect one, and it is only its extreme simplicity which prevents it from being as interesting as it would otherwise be." - Arthur Cayley

The information intensity and sustainability challenges of modern economies taxes age-old accounting conventions and processes that have been developed historically to support management and communication objectives during simpler times. Old paper-based accounting methods have by now been largely migrated to digital re-incarnations, as part of the so-called *digital transformation* process, which creates significant opportunities for more effective management and accounting methodologies. While some relevant background technologies, e.g., XBRL are already seeing adoption, new accounting constructs that capitalize on these possibilities remain largely a task for the future. In the meantime, in this white paper we continue exploring a small corner of the vast space of digital accounting possibilities, the potential for generalized concepts of mathematical accounting, with an emphasis on the visual representation as *accounting graphs*, and an eye towards the deeper integration of financial and non-financial accounts both within and across organizations.

The modeling of economic agents as *graph nodes* in graph networks is a well developed practice in many different domains. A large range of possible economic relations can be represented by edges encoding a variety of financial or other information. In [1, 2] we discussed such specific technical choices using *property graph* representations and we developed an approach focused on capturing the *contractual relationships* between agents. We focused in particular on deriving *accounting states* and transaction-induced economic *flows* from contractual information. Contracts are rather complex objects to model from an information-theoretic perspective and property graphs (along with their realizations as concrete *databases*) offer sufficient computational structure to enables a wide range of analyses. Formally, the objects we explored there were *directed, labeled and attributed multi-graphs.* Property graphs are flexible from an information modeling perspective: they allow diverse data pieces to be stored in both edges and nodes. Yet that versatility comes at a cost: analytical work using property graphs in general requires *software tools* and algorithms encoded in programming languages. This means it is, for example, not easy to use them to express computational operations concisely using standard mathematical notation. While the abundance of computational resources and growing adoption of *open source software* means the availability of computational tools is no longer the constraint it once was but there are still significant benefits in expressing accounting concepts in more concise mathematical form.

In this work we explore more stylized mathematical graph frameworks as a toolkit to represent the accounting state and mutations of economic networks. We explore a class of graphs that is more restricted as to what quantitative and qualitative information it can store. Compared to the underlying economic exchanges and contractual links, accounting data offer *distilled* representations of economic phenomena of various types. In accounting, diverse economic transactions with any combination of *physical manifestation* (e.g. transfers of goods from one place to another, or from one owner to another, the production or decay of physical artifacts etc.) and *social reality* (keeping count of obligations and promises between individuals or groups of individuals and organizations of various types, representing ownership interests, financial claims, debts or other liabilities etc.) are all reduced to a uniform numerical encoding scheme. Our objective is explore a graph toolkit that retains sufficient expressivity to model the classic double and quadruple-entry accounting use cases, even while having available standard *graph theory* mathematical tools. In this context the loss of flexibility versus more versatile property graphs is compensated by enabling concise representations using the powerful machinery of matrices, linear algebra and associated algorithms. This journey is illustrated pictorially in Figure 1.

More concretely, in this paper we investigate the mathematical representation of double and quadruple-entry accounting systems using graph structures. We do not even attempt to formalize the accounting notions mathematically. Historically there have been efforts to organize accounting in all its bewildering collection of phenomena in semi-axiomatic manner. It would take us too far afield to review these foundational aspects here[1]. Instead, we will jump right into our task of representing accounting via mathematical graphs, taking most of the conventions and practices of accounting as a given.

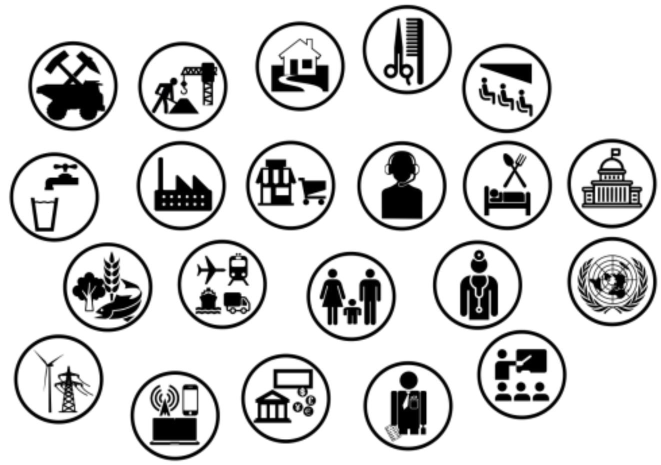
2 The accounting graphs of double-entry bookkeeping

2.1 Algebraic description of double-entry bookkeeping (DEB)

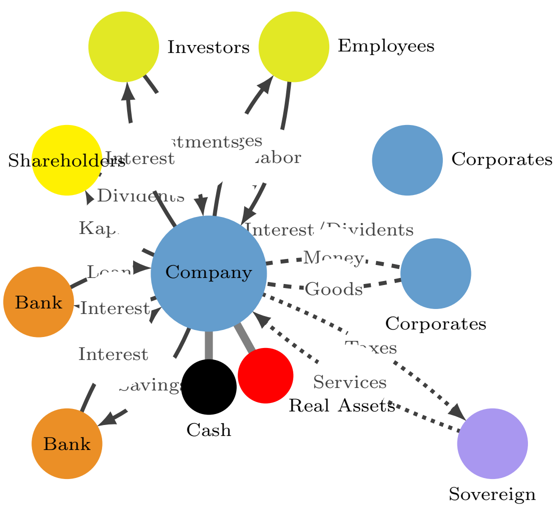
A defining feature of accounting that distinguishes it from other forms of information management is that the encoding of all transactions (which are understood as the mutations of accounting state) must *balance*, which numerically means they

[1] For a good historical review of abstract accounting models see the introduction of [3]

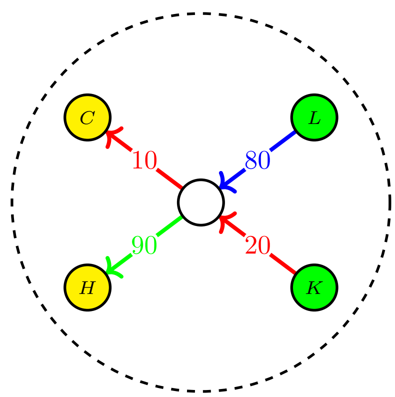
(a) Complex modern economies pose ever growing demands for adapted information management tools.



(b) Digital processes increasingly capture large amounts of information as data points: generating large numbers of *dots that must be connected*.



(c) Property graph abstractions are a very general toolkit to perform connect-the-dots exercises.



(d) Accounting graph abstractions are *simpler* graph constructs, but they offer increased conciseness, rigor and verifiability - for a subset of the available economic information.

Figure 1: An accounting graph can be understood as an end-product of a long sequence of abstractions that start with the actual economy and end up with stylized representations of economic phenomena. Intermediate constructs that offer more fidelity but sacrifice mathematical conciseness are based on property graphs.

must sum to zero. This requirement aims to introduce a *consistency* that is not automatic if, e.g., we account using a simple *catalog* of exchanges. The core mechanism for achieving this consistency is the adoption (first fully documented by Pacioli during the Italian Renaissance) of **double-entry accounting** (or bookkeeping, **DEB**). Axiomatic approaches to DEB accounting theory have a long pedigree [4]. Yet the mathematical properties of DEB systems have been elucidated only relatively recently, first through the introduction of the Pacioli group by Ellerman [5], and the related *Balance Module* by Rambaud et al. [6, 3]. It has been shown [7] that these two representations are related via an isomorphism (a mapping that establishes that these mathematical structures are equivalent). The mathematical structures developed in these works, and in particular the notion of *balance vectors* and associated vector spaces to represent transactions and balance sheets will be central to our approach. *Graph-theoretic* aspects of accounting have on occasion been discussed in the literature: Representing DEB in terms of directed graphs (digraphs) has been discussed in Arya et al.[8] and Rambaud et al.[3]. The graph aspects of quadruple-entry accounting principles are not widely explored, with a notable exception the graph representations of stock-flow consistent macro-economic models [9].

We detail next a graph framework that expresses an interesting range of accounting problems.[2]. We utilize throughout the well-known *graph-matrix duality*, where certain matrices can represent graphs numerically and, vice-versa, mathematical graphs can help visualize the information captured in matrix objects. This duality is quite empowering for analysis, as it allows the application of *linear algebra* techniques to solve graph problems and understand complex relationships.

An *accounting entity* or unit for our purposes will be any individual or organization that acts in an economic network as a distinct *agent*, with their own economic state, which is actually *accounted for*, i.e., it is expressed in terms of quantified, numerical values that satisfy certain closure relations. Organizational entities in scope can range from small or large and extremely complicated. As a special limiting case we include in the definition of an accounting entity the *entirety* of an economic network.

In practice accounting entities make use different set of *accounts*, that is a set of distinct numerical representations of their economic state. We will discuss in more detail what an account means mathematically later on. For simplicity of exposition we will assume that all entities in the economic network employ the same set of accounts but different entity types need not make use of all available accounts. The economic substance of an entity is indeed correlated with the set of accounts it typically uses.

In the accounting graph framework nodes or vertices of the graph are not the accounting entities themselves but rather *representations of their individual accounts*. Accounting entities are thus a *collection of accounts* (within a certain control perimeter) and the totality of accounts including all entities is the *accounting network*. Account nodes within and between entities might be connected using graph edges that can be either *internal* or *external* to the entities. These edges express in *simplified form* underlying economic operations, contractual relations etc.

Important complications that we will ignore in the present journey (but which do not invalidate the picture developed) include the following:

- We will ignore detailed procedures for the *valuation* of assets, liabilities, contracts etc. This so-called *measurement* process) converts a stream of diverse information into monetary values (numerical scalars). It is not unique but based on context, practicality, conventions, regulations etc.
- We will not cover the *permissibility* or feasibility of transactions, e.g., budget constraints resulting from the availability of funds in cash accounts etc.
- We do not discuss the generation of aggregations and reports (such as financial statements) from the underlying accounting states and mutations. In particular we will not discuss at all the auxiliary accounts that are constructed primarily towards that purpose
- We will not be particularly concerned with the accurate tracking of the *temporal sequence* of transactions, sidestepping differences between cost-based, accrual and market based accounting and different rules as to when and how revenue is recognized etc. We simply imagine an ordered sequence of transactions.
- We skip complex transactions that may need more elaborate edges (hypergraphs) to describe. We will instead assume that all phenomena can be described as composites of elementary transactions affecting two accounts.
- Finally, while extending to *integrated accounting* and multi-dimensional accounts and transactions that track environmental impact as per the framework discussed in [10, 11] is straight-forward, a detailed description is left for future work.

Clearly all the above simplifications put some distance between the graph tools we will presently discuss and actual accounting practice but they are necessary at this stages to help clearly describe the core ideas.

[2]There is no attempt at mathematical rigor, out focus is on making a fruitful link with the mathematical toolkit

2.2 The class of weighted directed multigraphs

The simplest possible class of *directed graphs* where two vertices can only be connected by a single edge is, alas, not sufficient to express the relations encountered in accounting. The need to capture numerically *multiple transactions* between the same sets of accounts (e.g., multiple purchases using cash) means we need to adopt a slightly richer construct: the category of *Weighted Directed Multigraphs*. A directed weighted multigraph is a directed graph which allows for multiple (parallel) edges between any two vertices. In addition, these edges are assumed to have *numerical weights*. Let us discuss those three definitional components and their implications in more detail:

In graph theory nomenclature a *multigraph* is a graph that can have *multiple edges* between the same start and end nodes. This encapsulates the concept of multiple transactions between accounts and it is an essential requirement in accounting context. While entirely intuitive, the implication of allowing edge multiplicity is that the better known *adjacency matrix* which indicates vertex-to-vertex connections ceases being the natural tool to represent such graphs. Instead we need to work with *incidence matrices* that indicate vertex-to-edge connections. In certain circumstances an adjacency matrix can be distilled from an incidence matrix.

Further, we need *directed* multigraphs (sometimes denoted multi-digraphs or *quivers*). The directionality of the edges is needed to express the from-to (or debit-credit) convention of accounting, the implication being that the same value is subtracted from one account and added to another.

In the simplest abstraction economic transactions are either exchanges (trading something of value for something else of value) or unilateral transfers that do not require anything in exchange. Both types of transactions require at least two entries. The main characteristic of double-entry bookkeeping is indeed that each transaction leads to at least two corresponding entries. Traditionally these are referred to as a credit entry and a debit entry.

In many cases (but by no means always), transactions are *reversible*: an edge with the opposite direction between the same nodes precisely negates (annuls) the first one. There are some transactions where this reversal is not feasible: for example an inventory of manufactured widgets cannot transform back into the raw materials that were used for their production.

We are interested in *weighted* graph edges, in other words, the edges are associated with a single *scalar* numerical value which is the measured value. Simpler graphs that only use the values $(-1, 0, 1)$ are potentially useful, but only represent a limited subset of the accounting information. We will stick to scalar weights (signed decimal numbers), but as noted, the extension to multi-dimensional accounting using *vector weights* is conceptually straight-forward [10] and enables the simultaneous tracking of multiple faces of economic activities (e.g., environmental impact).

As a more technical remark, the accounting graphs considered here will not involve *self-loops*. A self-loop would be a transaction that affects only one account, which is not allowed under DEB principles. Such loops may arise naturally in the process of *accounting graph aggregations* across a wider economic network, such as in input-output economic models.

2.3 Definition of the DEB accounting graph

Formally, a DEB *accounting graph* is a weighted multidigraph G that is an ordered 4-tuple $G := (V, E, f, w)$ with:

- V a set of ordered vertices $V = V_1, V_2, \ldots, V_i, \ldots, V_N$, indexed by index $i \in [1, N]$, where $|V(G)| = N$ denotes the total number of vertices (distinct accounts),
- E a set of edges E_k, indexed by index $k \in [1, M]$ where $|E(G)| = M$ denotes the number of edges of G (the number of transactions),
- A function $f : E \to V \times V$ that assigns to each edge an ordered pair of *adjacent* vertices (s, t) (representing the existence of a transfer of value between a source s and a target t account).
- A function $w : E \to W$, called a weight function, which maps each edge in E to a value from a set of possible weights, W (in general decimal numbers). This function is assigning the actual *measurement value* to a transaction between accounts.

The above definition is somewhat abstract but can be made very concrete using matrices.

2.3.1 The weighted incidence matrix

A weighted directed multigraph is represented by its *weighted incidence matrix* B, a rectangular, $N \times M$ matrix, with N being the number of nodes/vertices (accounts) $N = |V|$ and M being the number of edges/arcs $M = |E|$ (transactions).

The entries of the weighed incidence matrix B_{ik}, with i running over accounts and k running over transactions are defined as follows: If the transaction k originates from account i, the entry B_{ik} is the negative of the value $(-w)$ of the transaction (value is subtracted). If the edge k terminates at account i, the entry B_{ik} is the positive weight w (added to the account).

The incidence matrix is related to a central concept in accounting, the *journal of transactions*. A journal is a time-ordered list of transactions $J = \{T\}$ that occur during a defined period of time (between $[t_{start}, t_{end}]$). Actual accounting journals include contextual information beyond numerical values (e.g., including a time stamp). As noted, here we focus on the purely numerical elements and the role of time is played by the sequential ordering of transactions in the matrix. Thus, mathematically an accounting journal is encoded in the incidence matrix of an accounting graph.

The general form of an incidence matrix with N accounts and M transactions might look like the following, choosing the standard mapping of the first index of B to the row index (ranging over accounts) and the second index over columns (transactions).

$$B = \begin{array}{c c} & \begin{array}{ccccccc} E_1 & E_2 & E_3 & \cdots & E_k & \cdots & E_M \end{array} \\ \begin{array}{c} V_1 \\ V_2 \\ V_3 \\ \vdots \\ V_i \\ \vdots \\ V_N \end{array} & \begin{bmatrix} -w_1 & +w_2 & 0 & \cdots & -w_k & \cdots & +w_M \\ +w_1 & 0 & -w_3 & \cdots & 0 & \cdots & \vdots \\ 0 & -w_2 & +w_3 & \cdots & 0 & \cdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & +w_k & \cdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & \cdots & -w_M \end{bmatrix} \end{array} \tag{1}$$

2.3.2 Balance vectors

Mathematically the double-entry accounting or double-entry bookkeeping (DEB) principle is based on requiring the *balance* of all transactions:

$$\sum_{i=1}^{N} B_{ik} = 0, \forall k \in [1, M] \tag{2}$$

We see that the *zero column-sum property* of the incidence matrix that simply follows from its definition *automatically* encodes the balancing principle between debits and credits in DEB. In fact each column of the incidence matrix (each transaction) is an instance of a *balance vector*, a vector where the sum of all elements is zero. In the terminology of [3], the *balance module* is an algebraic structure comprising vectors whose component sums are zero. For an accounting set of N accounts, it can be thought as a hyperplane of dimension $N - 1$, embedded in N-dimensional space. When we are dealing with just two non-zero entries per column, one of which is the negative of the other such vectors are called *simple balance vectors*. Further, if these two values are always either $(+1, -1)$ we are talking about *elementary transactions* (and vectors).

In other words (and remarkably), the utilization of a weighted directed multigraph to represent a DEB accounting system naturally implements the required balance conditions, there is no need to impose them as additional requirements.

2.3.3 Allowable and feasible transactions, chart of accounts

The incidence matrix B_{ik} defined as above is a very general object. When does a random collection of numbers that follows this definition represent an real accounting system? Concrete accounting systems place additional constraints of two conceptual types [3]: The impose constraints on *allowable transactions* (i.e., what type of edges or column vectors can exist, specifically the patterns of non-zero weights) and *feasible transactions* (what range of weights are feasible for allowable transactions e.g., which accounts can go negative etc.).

Constraints on allowable transactions emerge from the nature of different accounts. The economic substance of these restrictions follows from the *chart of accounts*. Standard classifications introduce specific account labels such as Assets, Liabilities, Equity, Revenues, Expenses etc. Accounts might be further characterized as *Current* or *Capital* types, there might be a *liquidity* ordering, different types of cash equivalents, credit and of equity instruments. There might be auxiliary accounts to model depreciation, production processes, complex tax relationships with the State and any number of other complications. Finally, accountants might utilize a variety of *intermediate accounts* that are useful towards distilling relevant information and/or towards the construction of financial reports. These specializations assign to accounts/vertices and transactions/edges specific properties, which in turn define and constrain the specific types of activities that can be recorded. Mathematically such *additional graph structure information* might be encoded as a set of *binary matrices* indicating which accounts can be connected with edges. Since our objective here is not a detailed reproduction of practical realizations of accounting we will not load the discussion and leave these constraints implicit.

2.3.4 Ledgers and balance sheet vectors

The *initial ledger* is an empty graph, a set of indexed accounts V_i with no transactions. We might represent it as the null and trivially balanced vector $b^0 = 0$. A journal of transactions is *applied* to accounts by processing (posting) all its entries (transactions) in a temporal sequence (from oldest to newest, or from left to right in the incidence matrix representation). At any point in the sequence, the *ledger* is a vector b that contains the row-sums of all transactions up to that point, starting with the initial ledger:

$$b_i = \sum_{k=1}^{M} B_{ik}, \ \forall i \in [1, N] \tag{3}$$

More generally,

$$b_i^n = \sum_{k=m+1}^{n} B_{ik} + b_i^m, \ \forall i \in [1, N] \tag{4}$$

Processing a transaction (journal entry) transforms one set of valid bookkeeping statements (the ledger up to transaction E_m) into another set of valid bookkeeping statements (the updated ledger after transaction E_n). Other terms used for this updating procedure are *roll-forward* or *movements* of accounts. In words, beginning account balances, plus additions, minus subtractions become the end account balances.

It follows from the individual transaction balance property 2 that the sum of transactions balances as well.

$$\sum_{i=1}^{N} b_i = 0\,. \tag{5}$$

2.3.5 Horizontal balance

We saw that the vertical DEB balance requirement is automatic. Another type of balance that is not automatic but is occasionally observed is horizontal balance. In graph language terminology, the in-degree and out-degree of any node in V, denoted $d_{in}(V)$ and $d_{out}(V)$ represent the total amounts added or subtracted from accounts. These are obtained by summing horizontally along each row the positive and negative weights respectively.

$$d_{in}(V) = \sum_{k=1}^{M} \max(0, B_{ik}) \tag{6}$$

$$d_{out}(V) = \sum_{k=1}^{M} \max(0, -B_{ik}) \tag{7}$$

$$\tag{8}$$

When the in and out-degrees of a node are equal, the corresponding account is termed *horizontally balanced.* Horizontal balance, in contrast to the automatically enforced vertical balance may imply that there is some underlying form of *conservation law* at work.

2.3.6 (Non)elementary transactions and hypergraphs

Not every transaction can be decomposed into bilateral account mutations. A classic example would be production activities that transform defined fractions of input inventories into output inventory. In production processes the inventory of input materials can only be reduced and the inventory of products (and any byproducts) can only be increased in *fixed proportions* (typically following the weights of a stoichiometry matrix). Such transactions would be represented as edges from multiple nodes (inputs) into a single node (outputs). Many-to-many edges are also potentially present, e.g., when there is production of byproducts. Such edges are termed hyper-edges and the corresponding graph is termed a hypergraph. In the terms of their incidence matrix, such transactions will have non-zero values on more than one elements of a transaction column, drawing from within a pool of allowed transaction patterns.

2.4 Reification of the DEB accounting graph

We describe now a useful conceptual transformation of the DEB accounting graph we just introduced. This procedure is sometimes termed reification. *Reification* is term used in knowledge representation to denote a process of turning a *statement about a relationship* between two concepts into a *standalone addressable object*. A typical example of reification in graph context is to switch from connecting two nodes with an edge to *creating a new node* which represents that relationship.

In our context, reification of the accounting graph means that a single transaction between two accounts will be represented by *two transactions*, each with the same intermediating *virtual node*. Reification using such a central node can be illustrated in Figure 2.

(a) Direct (simple) representation of transactions as a weighted digraph.

(b) Reified representation of the same transactions after introducing a central node.

Figure 2: The result of introducing a central node V_0 is that all transactions are now routed via this node instead of being direct edges between accounts. Thus, the statement "node V_4 links to node V_3" becomes "node V_4 links to node V_0" and "node V_0 links to node V_3". For each one of the original transactions E_k, we now have a pair of edges (E_k, E^k), with the same weights, that *indirectly* link the transacting accounts. There is no longer any direct edge between the original accounting nodes. The vertical balance condition still applies to all transactions. A horizontal balance condition reflects the conservation of value flows through the central node.

Mathematically we have a modified and expanded incidence matrix B' that now has $N+1$ rows (an additional zero-th row being by convention the new virtual node) and $2M$ columns, representing the duplication of all transactions. The reified matrix would look like something structured as follows:

$$B' = \begin{array}{c} \\ V_0 \\ V_1 \\ V_2 \\ \vdots \\ V_N \end{array} \begin{array}{c} \begin{array}{ccccccc} E_1 & E^1 & E_2 & \cdots & E_k & \cdots & E_{2M} \end{array} \\ \left[\begin{array}{ccccccc} -w_1 & +w_1 & -w_2 & \cdots & -w_k & \cdots & -w_{2M} \\ 0 & 0 & +w_2 & \cdots & 0 & \cdots & 0 \\ +w_1 & 0 & 0 & \cdots & 0 & \cdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & +w_{2M} \\ 0 & -w_1 & 0 & \cdots & +w_k & \cdots & \vdots \end{array}\right] \end{array} \tag{9}$$

Explicitly this matrix is obtained by introducing for each k-th transaction introduce two columns $(2k, 2k+1)$ such that:

$$B'_{0,2k} = B_{ik}, \; : B_{ik} > 0 \tag{10}$$
$$B'_{i,2k} = B_{ik}, \; : B_{ik} < 0 \tag{11}$$
$$B'_{i,2k} = 0, \; : B_{ik} > 0 \tag{12}$$
$$B'_{0,2k+1} = B_{ik}, \; : B_{ik} < 0 \tag{13}$$
$$B'_{i,2k+1} = B_{ik}, \; : B_{ik} < 0 \tag{14}$$
$$B'_{i,2k+1} = 0, \; : B_{ik} > 0 \tag{15}$$
$$\tag{16}$$

As before, the balance sheet vector sums up all transactions:

$$b'_i = \sum_{k=1}^{2M} B'_{ik}, \ \forall i \in [1, N+1] \tag{17}$$

NB: The reified balance sheet vector is the trivial augmentation of the original vector b with a zero value in the first element.

Why consider such the reification of the accounting graph in the first place? What do we gain from the extra verbosity (all transactions being represented twice etc.)? It turns out this is useful both in the double-entry context we currently discuss and the quadruple-entry accounting context we will cover later: It helps elucidate the state of accounts but also with certain algebraic manipulations. Let us discuss here the first aspect:

In the direct graph representation (see Figure 2a) the vertical balance condition 5 is satisfied by the balance-sheet vector b but is not associated with any particular node in the graph. The aggregated DEB balance is rather a *global property* of the graph. While it ensures that all recorded transactions conform to accounting principles this is not particularly *visible*. Yet balance equations do have sometimes a natural interpretation as *conservation laws*, e.g., Kirchhoff's Law for electric circuits, where they reflect the conservation of charge entering and leaving (via currents) any node of the circuit. Is there any way to more visually represent the DEB accounting balance as such a conservation law? In general there is no privileged account that can serve as the node on which the conservation law could be centered. The cash account might be singled out for this role, but this can be problematic too, as cash is not always involved in transactions and anyway cash is rather loosely defined.

We notice that the first row of the incidence matrix (which documents the in and outflows of the central node) satisfies the horizontal balance condition (which is to be expected since by construction all ingoing edges have a mirror outgoing edges). Mathematically for the central node holds:

$$d_{in}(V_0) = \sum_j^M \max(0, B'_{0j}) = \sum_j^M \max(0, -B'_{0j}) = d_{out}(V_0) \tag{18}$$

$$\tag{19}$$

A key benefit of the reified graph in DEB context will be the more intuitive interpretation and linkage with the classic balance sheet, to which we turn next.

2.5 The balance sheet graph

The classic expression for a balance sheet would be a table where the balances of accounts (classified as assets and liabilities) are listed along some conventional order (along with the respective sums at the bottom).

Asset Accounts	Value	Liability Accounts	Value
V_1^A V_2^A ... V_N^A	v_1^A v_2^A ... v_N^A	V_1^L V_2^L ... V_N^L	v_1^L v_2^L ... v_N^L
Total Assets	v^A	Total Liabilities	v^L

Remarkably the above representation can be seen as the tabulation of the edge values of a reified graph, which we call the *reified balance sheet graph*. This is a directed weighted graph that is obtained from the actual multi-graph incidence matrix aggregated via *edge aggregation*. A weighted multi-digraph expressed via the reified incidence matrix B' has a unique *aggregated adjacency matrix* associated with it. To construct this adjacency matrix we must compress (sum up) the information encoded in multiple edges between nodes.

Using the weighted incidence matrix B' as input, the algorithm for generating an adjacency matrix is straightforward:

Alternatively and more concisely, the balance sheet graph of a DEB system can be created by constructing the adjacency matrix from the balance sheet vector b' as follows:

$$v_i^A = A'_{0i} = \max(0, b'_i)$$
$$v_i^L = A'_{i0} = \max(0, -b'_i)$$

Algorithm 1 From reified incidence matrix to aggregated adjacency matrix

Require: Input is the reified weighted incidence matrix B', of size $(N+1 \times 2M)$, where $N+1$ is the number of vertices and $2M$ is the number of directed edges.
Ensure: Output is the aggregated weighted adjacency matrix, A', of size $(N+1 \times N+1)$.
  Initialize an adjacency matrix, A' of the required dimension with zero entries.
  for each column E from 1 to $2M$ in B' **do**
    Find the source vertex i by identifying the row with a negative entry, $-w$.
    Find the destination vertex j by identifying the row with a positive entry, $+w$.
    Let the weight be w.
    Increment the adjacency matrix entry A'_{ij} by w.
  end for
  return A'

For all rows of the balance vector, an edge of weight b'_i between account V_i and the virtual account V_0 exists ($A'_{0i} = b'_i$) if the row element has non-zero value. If it is positive, the edge points to the account V_i, otherwise it points to the central node.

These equations make explicit that the aggregated adjacency matrix A' can be thought as the *balance sheet graph* of the DEB system. Notice that while the balance sheet information *is* available in the original balance vector b_i, the non-reified adjacency matrix A_{ij} is by construction linking accounts V_i and V_j and thus cannot enable a graph representation of the balance sheet.

2.6 The inverse problem

As transactions are being added to the initial graph and its incidence matrix expands with more columns, edges between accounts accumulate. In the horizontal aggregation of transaction data, in general there will be a *canceling out* of values which implies information loss. In general it is thus not possible to recover the full graph and incidence matrix B'_{ik} from the aggregated adjacency matrix A'_{ij}. What *is* possible (and potentially useful) is to derive a *compacted incidence matrix* C', from the adjacency matrix A'.

A compacted incidence matrix is a matrix with the same number of rows as the number of accounts $N+1$ and two columns for each edge present in the adjacency matrix. Thus its total number of columns equals twice the number of non-zero elements of the balance sheet vector b'. Again the algorithm for producing it is simple:

Iterating through the adjacency matrix A'_{ij} we identify all non-zero entries (edges). For each edge k from vertex V_i (the starting vertex) to vertex V_j (the terminating vertex) we set the entry $C'_{i,k}$ in the column k and the row for vertex V_i to $-w$ and the zero-th row $C'_{0,k}$ to w, the entry $C'_{j,k+1}$ to w and the entry $C'_{0,k+1}$ to $-w$.

3 Quadruple-entry bookkeeping

We move on to describe accounting graphs in a context where *multiple distinct economic entities* pursue double-entry accounting while transacting with each other, and there is a desire to ensure a level of consistency in their respective accounts. Loosely speaking, this the domain of *quadruple-entry accounting* or bookkeeping (**QEB**), which aims to create a consistent framework of accounting balance conditions in the presence of *multiple accounting entities.*

In contrast to DEB accounting that is practiced heavily in day-to-day financial accounting across the world, QEB is not an actual practice but rather a conceptual framework. Coordinated QEB accounting between *independent* entities is possible to envision (and might have various benefits), but it is not currently practiced. Some realization of the QEB concepts exists in the accounting of complex organizations with multiple related entities producing their own accounts. A notable use of QEB is towards building *stock-flow consistent macroeconomic models* [12]. Current use of quadruple-entry accounting concepts is mostly seen in the construction of national accounts by statistical agencies [14] where the possibility of consistency of macro/micro accounts is something that has been raised, see [13].

In contrast to the standard DEB business bookkeeping we sketched already, compilation of a QEB graph representation would have to deal with documenting interactions among multiple units. In such a QEB context we want that the accounts for an entity are still linked to their double-entry bookkeeping basis where necessary, but the subset of accounts that record accounting transactions with external entities must be modeled explicitly as such. A quadruple-entry accounting system specification thus aims to deal in a coherent way with multiple entities, each of which can be assumed to practice standard double-entry bookkeeping.

3.1 The QEB accounting graph

How are we to model the above consistency requirements as a graph? We can build the answer as a special case of the already discussed DEB graphs and incidence matrices. Namely we construct an accounting graph where a set of accounts is grouped as belonging to distinct entities. We call this the QEB accounting graph and the corresponding incidence matrix. A *QEB accounting graph* is a DEB accounting graph with *additional properties.*

Formally a QEB graph is a weighted multidigraph G that is an ordered 5-tuple $G := (V = \cup_{a=1}^{A} U_a, E, f, w)$ where

- V an ordered set of vertices across an entire economic network (all modeled accounts) with a *balanced partition* into A disjoint subsets (blocks) U_a which express the distinct accounting entities. It holds that $V = \cup_{a=1}^{A} U_a$ and $|U_a| = N$ implies the same number of accounts for all entities.
- E is a set of all edges (all transactions). As before, there is function $f : E \to V \times V$, that assigns to each edge an ordered pair of *adjacent* vertices (s, t) (transactions between accounts).
- The edge set E is split into two disjoint sets of *external versus internal edges* $E = E_X \cup E_I$, where $E_X = \{(u, v) \in E | u \in U_a, v \in U_b, a \neq b\}$ and $E_I = \{(u, v) \in E | u \in U_a, v \in U_a\}$.
- There is a function $w : E \to W$ that maps each edge in E to a value in a set of possible weights, W.

When the partition of accounts involves only one entity ($A = 1$), the QEB graph reduces to a DEB graph. Thus quadruple-entry accounting is a special case of double-entry accounting, where distinct sets of accounts are identified (representing multiple entities) and treated simultaneously.

In QEB frame, an incidence matrix, Q is adapted in straightforward fashion to represent all transactions in the economic network. For simplicity of expositions we assume all entities utilize the same set of accounts. Depending on the nature of the entity some accounts may not be applicable and thus never populated. For clarity the accounts belonging to different accounting entities are grouped in sequentially and denoted $V_{i,a}$, where i runs over the accounts index $i \in [1, N]$ and a runs over the entity index $a \in [1, A]$. Transactions (edges) that link nodes belonging to different entities are *external transactions*, otherwise they are internal transactions.

In general the Q matrix would look as follows:

$$Q = \left[\begin{array}{cccccccc}
 & E_1 & E_2 & E_3 & \cdots & E_k & \cdots & E_M \\
V_{1,1} & -w_1 & +w_2 & 0 & \cdots & -w_k & \cdots & w_M \\
V_{2,1} & +w_1 & 0 & -w_3 & \cdots & 0 & \cdots & \vdots \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
V_{N,1} & 0 & +w_2 & +w_3 & \cdots & 0 & \cdots & \vdots \\
\hline
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
V_{i,a} & 0 & 0 & 0 & \cdots & w_k & \cdots & \vdots \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
\hline
V_{1,A} & 0 & 0 & 0 & \cdots & 0 & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
V_{N,A} & 0 & 0 & 0 & \cdots & 0 & \cdots & -w_M
\end{array}\right] \tag{20}$$

We see the same overall shape as the DEB incidence matrix B but the partition naturally splits columns into the internal and external classes. E.g., in the above matrix we see transactions E_1, E_2, E_3 being internal to entity V_1 whereas there are two external transactions E_j and E_M, between V_1 and V_i and V_N. The vertical DEB balance automatically applies to all columns irrespective of whether it is an internal or external transaction.

Maybe useful to note that while quadruple-enty accounting sometimes invokes the concept of horizontal balance to denote the balancing of external transactions, this is simply an artifact of representation. In the incidence matrix representation all balances, whether internal or external are vertical.

3.1.1 Internal Transactions

An *internal transaction* refers to the representation of economic events that occur within the perimeter of single entity and do not involve the accounts of external parties. Mathematically internal transactions are the column of a Q' matrix with both non-zero entries within a block. Such transactions will generally not affect the entity's cash account (cash only changes hands via external transactions). They also do not affect an entity's liabilities (debt or equity contracts) as any mutations to these by necessity affect the corresponding counterparty accounts. Internal transactions may well reflect notable mutations of economic state which are important to account for. Internal transactions may transfer value from one internal account to another and thus modify allowable future transactions. Examples would be consumption of raw materials in production, depreciation are internal transactions.

3.1.2 External Transactions

External transactions are in the simplest form bilateral exchanges or transfers of value between entities. Mathematically they are edges that span two different blocks of the incidence matrix. They will typically occur in sets of four. E.g.,

- A decrease in the inventory account of entity A
- An increase in the cash account in entity A
- An increase in the inventory account in entity B
- A decrease in the cash account in entity B

Such external exchanges imply that a *bundle of associated transactions* is to be recorded, for the accounts in question and the two entities involved.

3.1.3 Reified accounting network graph

The corresponding reified form Q' of the incidence matrix is obtained analogously with the DEB case:

$$Q' = \begin{array}{cc}
 & \begin{array}{ccccccc} E_1 & E_2 & E_3 & \cdots & E_k & \cdots & E_{2M} \end{array} \\
\begin{array}{c} V_{0,1} \\ V_{1,1} \\ V_{2,1} \\ \vdots \\ V_{N,1} \\ \hline V_{0,a} \\ \vdots \\ V_{i,a} \\ \vdots \\ \hline V_{0,A} \\ V_{1,A} \\ \vdots \\ V_{N,A} \end{array} &
\left[\begin{array}{ccccccc}
-w_1 & +w_2 & -w_3 & \cdots & 0 & \cdots & 0 \\
0 & 0 & 0 & \cdots & -w_k & \cdots & -w_k \\
+w_1 & 0 & 0 & \cdots & 0 & \cdots & \vdots \\
\vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
0 & +w_2 & +w_3 & \cdots & 0 & \cdots & \vdots \\
\hline
0 & 0 & 0 & \cdots & 0 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & \cdots & w_k & \cdots & \vdots \\
\vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
\hline
0 & 0 & 0 & \cdots & 0 & \cdots & 0 \\
0 & 0 & 0 & \cdots & 0 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & \cdots & 0 & \cdots & -w_M
\end{array}\right]
\end{array} \tag{21}$$

For each block of accounts a belonging to a distinct accounting entity, the first row will $V_{0,a}$ has non-zero elements from the reification of internal transactions but zero elements in the external transaction columns (the central node of an entity never interacts directly with external accounts). The other rows (corresponding regular accounts) can have negative or positive entries as usual.

3.1.4 Accounting network balance sheet vector

The graph concepts we discussed already in DEB context (balance sheet vectors, adjacency matrices, compactified incidence matrices etc.) apply here in principle as well. E.g., the balance sheet vector of the entire accounting network is simply the summation of all transactions:

$$q_i = \sum_{k=1}^{M} Q_{ik}, \; \forall i \in [1, N] \tag{22}$$

and satisfies by construction the balance equation:

$$\sum_{i=1}^{N} q_i = 0 \tag{23}$$

3.2 From QEB to individual DEB representations and back

Now we can ask the question of how we can obtain from the QEB graph the individual DEB graphs. Recovering individual entity DEB accounts from a QEB representation is actually straightforward. To recover individual double-entry bookkeeping for each one of the entities, the transactions with external parties must be *mirrored* as internal transactions. What is required is that for each one of the external transactions, the incidence matrix representation replaces them with the corresponding internal edges. To achieve this using the incidence matrix Q' we need to *permute* the external transaction vectors (columns), using permutation matrices that move the non-zero values from one external party block to the other and vice versa.

In the reified version of the graph the permutation matrices are trivial: we swap the legs of external transactions with the corresponding central node. More generally we can follow either path of the following commutation diagram.

$$\begin{tikzcd}
\text{QEB Graph} \arrow[r, "\text{Permute}"] \arrow[d, "\text{Horizontal Sum}"'] & \text{A x DEB Graphs} \arrow[d, "\text{A x Horizontal Sums}"] \\
\text{QEB Balance Sheet} \arrow[r, "\text{Split}"'] & \text{A x DEB Balance Sheets}
\end{tikzcd}$$

Figure 3: The commutation diagram shows two equivalent paths to obtain individual entity balance sheets. We can either obtain first the entire accounting network balance sheet and split it into entity balance sheets or, compute first decoupled (internalized) DEB graphs which then can be aggregated afterwards.

The opposite operation (starting with a set of DEB graphs and obtaining a consistent QEB graph) is not similarly straightforward. Technically it is again a matter of applying suitable permutation matrices between distinct entity blocks of the incidence matrix. The difficulty concerns specifying the correct indices (rows) upon which the permutation matrices will act. This requires, minimally, to have an explicit specification of allowable transactions (which restricts the range of possible indices). Yet even after conditioning on this information being available, it is conceivable that multiple accounts are eligible. In other words, in contrast to the QEB to DEB reduction, the inverse DEB to QEB problem does not have a unique solution.

4 A worked out example

Up to now the DEB/QEB accounting graph framework was laid out in abstract terms: we described a minimal graph-based mathematical scaffolding that captures essential elements of both DEB and QEB accounting methodologies. A substantial part of the practice of accounting derives from the representation of *specific* economic behaviors and phenomena. As discussed in the introduction, the conventions and detailed machinery used in current accounting practice are so ornate they can seriously obscure the underlying conceptual logic of accounting. In order to clarify the mathematical description and make it more concrete it will be useful to explore the nature of these abstract graph representations with concrete examples. To this effect we will trace a number of stylized accounting transactions. We'll elaborate accounting graph representations for a drastically simplified scenario, following a household entity as the main economic actor as they pursue a number of transactions with other external parties.

4.1 The Initial Graph State

We introduce first a chart of accounts describing the household entity:

- V^0, representing the household entity (in the reified version of the graph)
- V^1/C_h, representing the cash held by the household (financial asset)
- V^2/R_h, owned real estate (tangible asset)
- V^3/A_h, other owned financial assets
- V^4/L_h, loans (a liability),
- V^5/K_h, the entity's net worth (nominally a liability)

One can spend books to describe the accounting principles and conventions behind the innocent facade of the above. We only discuss a few key aspects.

4.1.1 About the Cash Account

Modern cash, whether in paper of digital form is a essentially a *financial token*, a claim that can be exchanged for a variety of other tokens or for intrinsically valuable assets in the economy. In concrete accounting scenarios the cash account may refer to cash in hand, but most commonly to a current account with a financial intermediary. In digital form the cash account is quite literally an external account with another entity. That entity can be a private bank (in which case we talk about *private money*) or a central bank, in which case we talk about *sovereign money*. For the non-bank section of the economic network cash is a conserved quantity: it can be moved around by it cannot be destroyed or generated. Cash in the role of money is simultaneously a medium of exchange (one leg of external transactions), a unit of account (all accounting graph weights are expressed in money terms), and a financial asset (a store of value). The versatility of money accounts for accounting purposes is such that a good fraction of accounting can indeed be based on simply keeping track of cash transactions. In our treatment here we do not grant the cash account a privileged status.

4.1.2 About Equity and Net Worth Accounts

Similar to cash, the equity or net worth account is in principle an external facing, account of great importance. It represents a claim of residual value by the ultimate owners behind the accounting entity. While equity is a term associated with corporate entities, the net worth of an entity is always defined as the residual interest and in practice this is what guarantees the balance of DEB accounts. The beneficiary of that residual value might be explicitly represented in the economic network description (for example as a financial asset of another entity) or be left unspecified.

Similar accounts (indexed appropriately) will be populating the other accounting entities as we encounter them down the line. Initially the balance sheet of the household (and all other entities) is a clean slate (empty), with all accounts being zero.

Asset Accounts	Value	Liability Accounts	Value
Cash C_h	0	Loan L_h	0
Financial Assets A_h	0		
Real Estate R_h	0	Net Worth K_h	0
Total Assets	0	Total Liabilities	0

Visually, the partitioning of accounts V in blocks is illustrated as in Figure 4, namely several grouped accounts, belonging to the distinct entities.

4.2 Transaction 1: Endowment

In the first transaction we imagine a cash infusion of 20 CU's (= currency units). Think e.g., of an unexpected inheritance. This is the simplest possible transaction, as a result of which the household is assumed to possess an amount of cash which comprises its entire net worth. As the transaction does not involve an identifiable external party we can represent it as a DEB graph. The two versions of the accounting graph incidence matrix (B and B') are as follows:

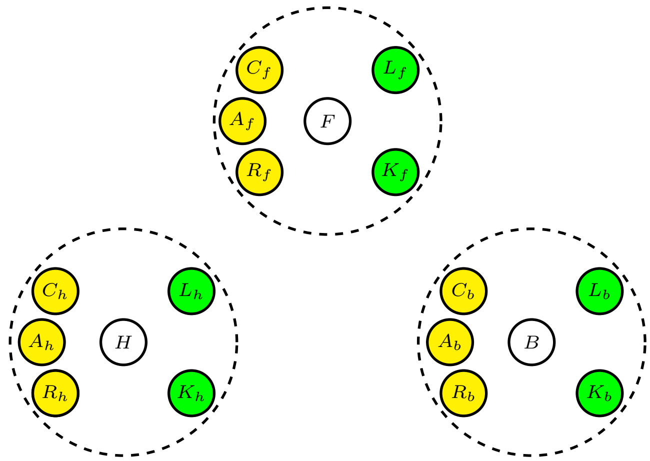
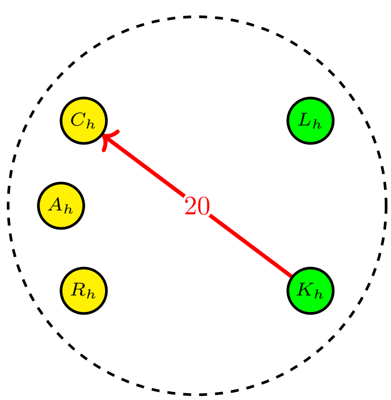


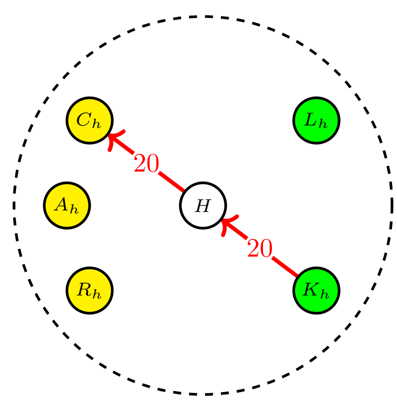
Figure 4: A chart of accounts and initial state of a hypothetical quadruple-entry accounting graph that features three distinct accounting entities: a household H, a firm F and a bank B. The accounting nodes of each entity are enclosed in a dashed perimeter that denotes the *control boundary* of each entity. Yellow-colored nodes denote accounts that are assets. Green-colored accounts denote liabilities.

$$B = \begin{bmatrix} 20 \\ 0 \\ 0 \\ 0 \\ -20 \end{bmatrix}, \; B' = \begin{bmatrix} 20 & -20 \\ 0 & 20 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ -20 & 0 \end{bmatrix}$$

Notice that B' encodes two transactions (in the redundant way following reification) and has an additional node (V_0). Both matrices satisfy the vertical balance condition (the column-sums are zero). Graphically this is represented in the two panels of Figure 5:



(a) Graph representation of the initial balance sheet. A weighted edge from the net worth account to the cash account captures all available information.



(b) Graph representation using a central node. There are two edges: one indicating the a positive cash account balance and one the corresponding positive net worth account balance.

Figure 5: A cash infusion transaction and two ways to represent it as a DEB graph. Edges leading *into* asset (yellow) accounts indicate positive balance. Edges leading *away* from liability (green) nodes will indicate positive balance.

The stylized household balance sheet (was initially blank) after this accounting transaction is as follows:

Asset Accounts	Value	Liability Accounts	Value
C_h	20	L_h	0
A_h	0		
R_h	0	K_h	20
Total Assets	20	Total Liabilities	20

At this stage we don't yet see an appreciable difference between the two DEB graph approaches. In both cases the negative value emanating from the net worth (liability) account is balanced by the positive standing of the cash account. In the case of the B matrix this is encoded as a single transaction between two accounts with no further conditions. In the case of the B' matrix this is encoded with two transactions between three accounts. The additional condition of the reified representation is that the central node is always balanced (inflow equals outflow), which means the row-sum of the first row is zero.

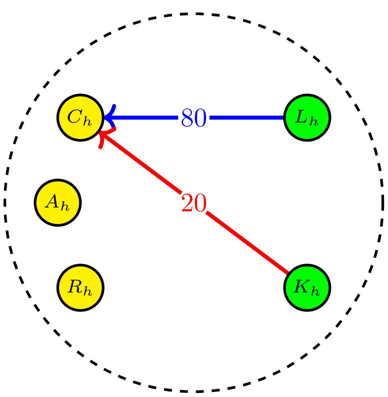
At this stage there is no external party involved (the source of the cash is obscure). The owners of the household net worth are also left unspecified. The implication is that the QEB representation does not provide any additional information. Any other blocks of the Q' matrix is unknown / zero.

4.3 Transaction 2. Borrowing from a Bank

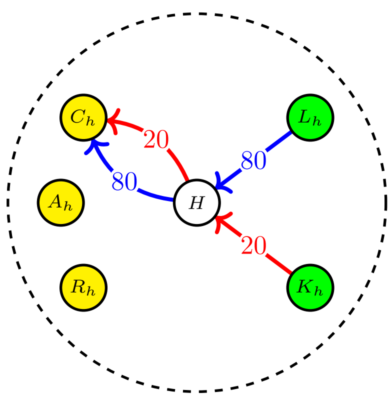
We build on the previous accounting state by introducing a transaction that models typical borrowing activity. Utilizing the standard machinery of DEB we create an internal representation of a liability account L_h (for a Loan). We imagine the household pursuing a bank loan to the tune of 80 CU's. The two versions of the graph incidence matrix are now updated (augmented with the new transaction) as follows:

$$B = \begin{bmatrix} 20 & 80 \\ 0 & 0 \\ 0 & 0 \\ 0 & -80 \\ -20 & 0 \end{bmatrix}, \; B' = \begin{bmatrix} 20 & -20 & 80 & -80 \\ 0 & 20 & 0 & 80 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -80 & 0 \\ -20 & 0 & 0 & 0 \end{bmatrix}$$

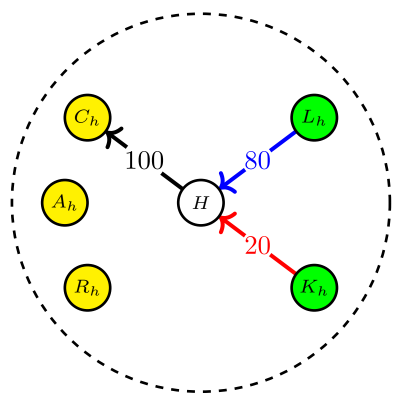
The accounting graphs corresponding to these incident matrices are now:



(a) The (direct) transaction graph



(b) The reified graph.



(c) The balance sheet graph.

Figure 6: With a second transaction added to the graph the cash account is augmented. Now two liability accounts have positive balances (outgoing edges). Once we add the second transaction, the qualitative display difference between the two graph representations starts becoming more apparent. Notice that we can mentally sum the multiple edges going from the central node to the cash account and create an explicit visual representation of the balance sheet (figure c). While we can count the in-out degrees of nodes in the direct graph representation (figure a) it is not as intuitive.

In both representations the horizontal row-sum of all transactions produces the respective balance sheets vectors:

$$b = \begin{bmatrix} 100 \\ 0 \\ 0 \\ -80 \\ -20 \end{bmatrix}, \; b' = \begin{bmatrix} 0 \\ 100 \\ 0 \\ 0 \\ -80 \\ -20 \end{bmatrix}$$

In Figure 6 we see how after multiple transactions the two representations start offering different visual aids: the direct graph representation is *minimalist* (uses the least amount of visual content), whereas the reified version offers more intuitive representation of the *aggregated state*. In particular the aggregation of the multigraph edges provides a convenient visual representation of the balance sheet.

After this transaction is posted we have an updated balance sheet:

Asset Accounts	Value	Liability Accounts	Value
C_h	100	L_h	80
A_h	0		
R_h	0	K_h	20
Total Assets	100	Total Liabilities	100

For the adjacency matrix corresponding to this balance sheet we have explicitly from equations **??**:

$$\begin{aligned} C_h &= v_1^A = A'_{01} = \max(0, b'_1) = 100 \\ L_h &= v_4^L = A'_{40} = \max(0, -b'_4) = 80 \\ K_h &= v_5^L = A'_{50} = \max(0, -b'_5) = 20 \end{aligned}$$

which is, as expected, is the adjacency matrix of the balance sheet graph in Figure 6c.

$$A' = \begin{bmatrix} 0 & 100 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 80 & 0 & 0 & 0 & 0 & 0 \\ 20 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The compactified incidence matrix is:

$$C' = \begin{bmatrix} -100 & 80 & 20 \\ 100 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & -80 & 0 \\ 0 & 0 & -20 \end{bmatrix}$$

For brevity from now on we will only display the reified form of the evolving accounting graph.

4.3.1 Borrowing from a Bank - the QEB graph

As we now have a transaction that is actually with an external party, it is useful to illustrate the above accounting state in the context of an QEB graph that includes that external counterparty. Representing a borrowing transaction in QEB involves two edges that connect accounts belonging to distinct entities (thus external), in place of one internal edge in the case of DEB. NB: Two bootstrap transactions are included under E_1 and E_2 to populate the initial balance sheets of household and bank respectively. The QEB incidence matrix of these four transactions (bootstrap plus two borrowing legs) is as follows.

$$
Q' = \begin{array}{cc}
 & \begin{array}{cccc} E_1 & E_2 & E_3 & E_4 \end{array} \\
\begin{array}{c} V_h \\ C_h \\ A_h \\ R_h \\ L_h \\ K_h \\ V_b \\ C_b \\ A_b \\ R_b \\ L_b \\ K_b \end{array} &
\left[\begin{array}{cccc}
-20 & +20 & 0 & 0 \\
+20 & 0 & 80 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & -80 \\
0 & -20 & 0 & 0 \\
\hline
+80 & -80 & 0 & 0 \\
0 & +80 & -80 & 0 \\
0 & 0 & 0 & +80 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
-80 & 0 & 0 & 0
\end{array}\right]
\end{array} \tag{24}
$$

Notice that while the column-sums per each transaction of the QEB incidence matrix are zero, the column-sums per transaction for the entity-specific block of accounts of each individual entity *do not balance* in the case of external transaction columns (E_3, E_4). When external transactions are presented as such, it is only the entire network that balances: The multi-entity QEB incidence matrix is not simply the stacking of multiple DEB incidence matrices.

Visually the above Q' matrix produces graph 7

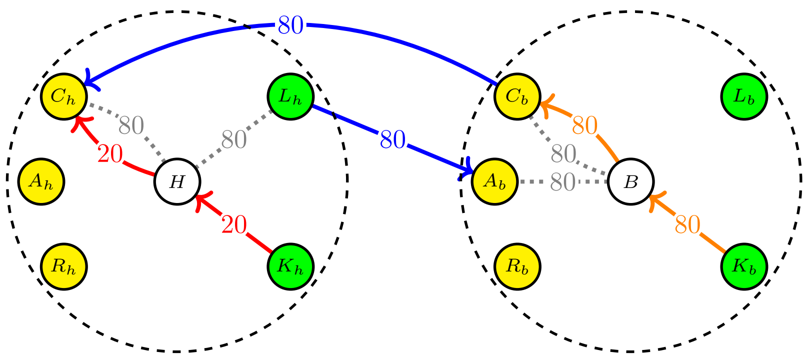


Figure 7: QEB transactions representing the interest-free loan we saw already in a DEB graph incarnation. The internal graph edge (gray) is now replaced by two external edges (blue) connecting to the bank entity / accounts. The edge from the bank account C_b to the household account C_h represents the cash transfer whereas the edge from the L_h household liability node to the bank asset A_b represents the loan contract. For good measure the bank balance sheet is bootstrapped to make the cash amount available but this is not essential for the example.

The first external transaction, E_3, encodes the transfer of cash between the two cash accounts. The second transaction E_4 creates a debt liability for the household and a debt financial asset for the bank. For easy comparison, the same transactions are shown with gray/dotted edges in the respective internal representations.

4.3.2 Borrowing from a Bank - but with interest

We assumed so far that the bank loan was interest-free, which means that the liability generated was exactly of the same amount as the cash received. Accounting for realistic commercial lending operations is quite complicated but it is useful to indicate how this would work and what kind of complexities it introduces. To emulate interest-bearing lending we assume that the disbursed cash is less than the required loan repayment amount, e.g., the cash transfer is 80 CU versus a 85 CU liability. How to account for this subtle difference properly is of course subject to elaborate accounting standards. In our stylized illustration we assume that we can immediately translate the difference between cash and generated asset into the bank *booking a profit* (thus a flow to equity). But to maintain overall balanced QEB accounts the unbalanced actual flows between bank and household should be matched by a *negative delta* (loss) for the net worth of the household. Clearly since QEB is not practiced, no such cross-entity adjustment are made. In fact households typically do not keep formally accounts at all.

The incident matrix includes now a *virtual* external transaction E_5.

$$
Q' = \left[\begin{array}{c|ccccc}
 & E_1 & E_2 & E_3 & E_4 & E_5 \\
\hline
V_h & -20 & +20 & 0 & 0 & 0 \\
C_h & +20 & 0 & 0 & +80 & 0 \\
A_h & 0 & 0 & 0 & 0 & 0 \\
R_h & 0 & 0 & 0 & 0 & 0 \\
L_h & 0 & 0 & -85 & 0 & 0 \\
K_h & 0 & -20 & 0 & 0 & +5 \\
\hline
V_b & +80 & -80 & 0 & 0 & 0 \\
C_b & 0 & +80 & 0 & -80 & 0 \\
A_b & 0 & 0 & +85 & 0 & 0 \\
R_b & 0 & 0 & 0 & 0 & 0 \\
L_b & 0 & 0 & 0 & 0 & 0 \\
K_b & -80 & 0 & 0 & 0 & -5
\end{array}\right] \tag{25}
$$

Pictorially this graph involves a slightly modified set of transactions is represented in Figure 8

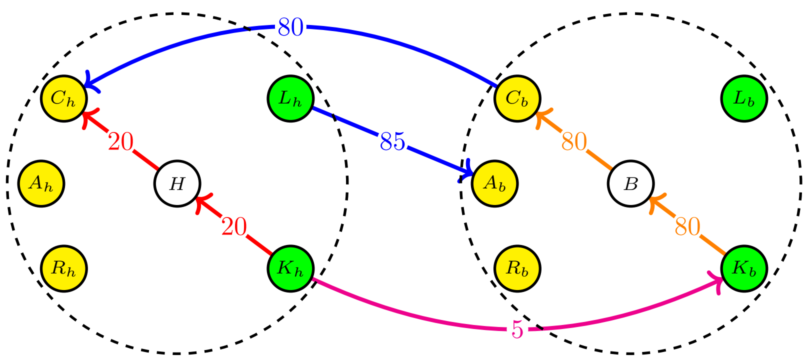


Figure 8: QEB transactions representing an interest-bearing loan. The cash transfer is not balanced by the newly generated household liability. There is an implied wealth transfer from the household to the bank. NB: The display of the bank accounts / transactions is only partial.

Since we do not yet have multiple edges between nodes, the adjacency matrix for this accounting network can be derived directly (without a need to first aggregate). Notice that the adjacency matrix is in this instance very sparse.

$$
A' = \begin{bmatrix}
0 & 20 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 85 & 0 \\
20 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 5 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 80 & 0 & 0 \\
0 & 80 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 80 & 0 & 0 & 0
\end{bmatrix} \tag{26}
$$

The decoupled (DEB) version of the graph is obtained by swapping in the incidence matrix Q' the non-zero legs of external transactions with the respective central nodes of each block, symbolically creating a matrix $P(Q')$:

$$
P(Q') = \begin{array}{c|ccccc}
 & E_1 & E_2 & E_3 & E_4 & E_5 \\
\hline
V_h & -20 & 20 & +85 & -80 & -5 \\
C_h & 20 & 0 & 0 & 80 & 0 \\
A_h & 0 & 0 & 0 & 0 & 0 \\
R_h & 0 & 0 & 0 & 0 & 0 \\
L_h & 0 & 0 & -85 & 0 & 0 \\
K_h & 0 & -20 & 0 & 0 & +5 \\
\hline
V_b & 80 & -80 & -85 & +80 & +5 \\
C_b & 0 & 80 & 0 & -80 & 0 \\
A_b & 0 & 0 & +85 & 0 & 0 \\
R_b & 0 & 0 & 0 & 0 & 0 \\
L_b & 0 & 0 & 0 & 0 & 0 \\
K_b & -80 & 0 & 0 & 0 & -5
\end{array} \tag{27}
$$

Notice that for each external transaction two internal mirrors are required to decouple the individual accounting blocks. The accounting network balance sheet vector is:

$$
q' = \begin{bmatrix} 0 \\ 100 \\ 0 \\ 0 \\ -85 \\ -15 \\ 0 \\ 0 \\ 85 \\ 0 \\ 0 \\ -85 \end{bmatrix}
$$

As mentioned in the commutation diagram discussion, the balance sheet vector is the same whether obtained before or after applying the permutation matrices. The representation of the decoupled DEB graph is given in Figure 9.

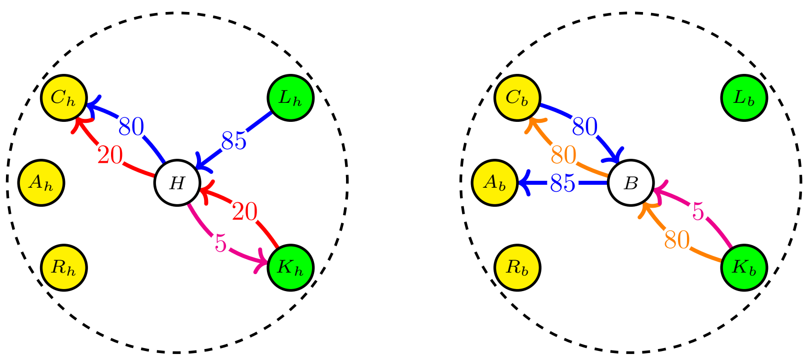


Figure 9: Two internal DEB graphs obtained from the network QEB graph

The corresponding individual balance sheets are:

Asset Accounts	Value	Liability Accounts	Value
C_h	100	L_h	85
A_h	0		
R_h	0	K_h	15
Total Assets	100	Total Liabilities	100

and

Asset Accounts	Value	Liability Accounts	Value
C_b	0	L_b	0
A_b	85		
R_b	0	K_b	85
Total Assets	85	Total Liabilities	85

4.4 Transaction 2: Buying a House

Let us now examine a further transaction, namely the purchase of a house. This will encode in our graph representation the act of buying a tangible asset in exchange for available cash. We thus model the transaction as a simple *exchange* with another party. Let us assume the house costs 90 CU. For this transaction we are given scarcely an information about the counterparty that sells the house. For this reason we have to model it as an internal transaction for the household. Fortunately this does not compromise the overall structure of the QEB graph.

In matrix form the Q' matrix including the above internal household transaction adds columns E_6 and E_7:

$$Q' = \left[\begin{array}{c|ccccccc} & E_1 & E_2 & E_3 & E_4 & E_5 & E_6 & E_7 \\ \hline V_h & -20 & +20 & 0 & 0 & 0 & 90 & -90 \\ C_h & +20 & 0 & 0 & +80 & 0 & -90 & 0 \\ A_h & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ R_h & 0 & 0 & 0 & 0 & 0 & 0 & 90 \\ L_h & 0 & 0 & -85 & 0 & 0 & 0 & 0 \\ K_h & 0 & -20 & 0 & 0 & +5 & 0 & 0 \\ \hline V_b & +80 & -80 & 0 & 0 & 0 & 0 & 0 \\ C_b & 0 & +80 & 0 & -80 & 0 & 0 & 0 \\ A_b & 0 & 0 & +85 & 0 & 0 & 0 & 0 \\ R_b & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ L_b & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ K_b & -80 & 0 & 0 & 0 & -5 & 0 & 0 \end{array}\right] \tag{28}$$

The corresponding network balance sheet vector is:

$$q' = \begin{bmatrix} 0 \\ 10 \\ 0 \\ 90 \\ -80 \\ -20 \\ 0 \\ 0 \\ 85 \\ 0 \\ 0 \\ -85 \end{bmatrix}$$

From the point of view of the household the balance sheet after this transaction captures the shift from a cash asset to a real estate asset (the bank balance sheet remains unchanged):

Asset Accounts	Value	Liability Accounts	Value
C_h	10	L_h	85
A_h	0		
R_h	90	K_h	15
Total Assets	100	Total Liabilities	100

The graph representation of the incidence matrix and balance sheet including the last transaction is given in Figure 10. Notice that the balance sheet vector can be read of directly from Figure 10b (i.e, we can associate an edge emanating or ending to each account with the corresponding element of the balance sheet vector element).

(a) Transactions Graph

(b) Balance Sheet Graph

Figure 10: The overlay of three distinct transactions helps further elucidate the visual display characteristics of the reified graph representation. The aggregated reified graph is a visual representation of the balance sheet.

4.5 Transaction 3: Setting up a business

We finally imagine that with the left-over cash the household becomes entrepreneurial and wants to setup a business. Setting up a firm in our stylized representation involves the following steps: An amount of cash will be transferred to a (new) legal entity (10 CU). This will be the initial capital of that entity. In turn this net worth becomes a financial asset of the household. In turn the business will spend the cash to acquire production assets.

The complete incident matrix including the last transaction is:

$$Q' = \left[\begin{array}{c|ccccccccccc}
 & E_1 & E_2 & E_3 & E_4 & E_5 & E_6 & E_7 & E_8 & E_9 & E_{10} & E_{11} \\
\hline
V_h & -20 & +20 & 0 & 0 & 0 & +90 & -90 & 0 & 0 & 0 & 0 \\
C_h & +20 & 0 & 0 & +80 & 0 & -90 & 0 & -10 & 0 & 0 & 0 \\
A_h & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & +10 & 0 & 0 \\
R_h & 0 & 0 & 0 & 0 & 0 & 0 & 90 & 0 & 0 & 0 & 0 \\
L_h & 0 & 0 & -85 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
K_h & 0 & -20 & 0 & 0 & +5 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
V_b & +80 & -80 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
C_b & 0 & +80 & 0 & -80 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
A_b & 0 & 0 & +85 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
R_b & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
L_b & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
K_b & -80 & 0 & 0 & 0 & -5 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
V_f & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10 & -10 \\
C_f & 0 & 0 & 0 & 0 & 0 & 0 & 0 & +10 & 0 & -10 & 0 \\
A_f & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
R_f & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & +10 \\
L_f & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
K_f & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -10 & 0 & 0
\end{array}\right] \tag{29}$$

which is visually represented in Figure 11.

References

[1] P. Papadopoulos. WP8: Connecting the Dots: Economic Networks as Property Graphs. *Open Risk White Papers*, 2019. Online Link.

[2] P. Papadopoulos. WP10: Connecting the Dots: Concentration, diversity, inequality and sparsity in economic networks. *Open Risk White Papers*, 2021. Online Link.

[3] Salvador Cruz Rambaud et al. *Algebraic Models for Accounting Systems*. World Scientific, 2010.

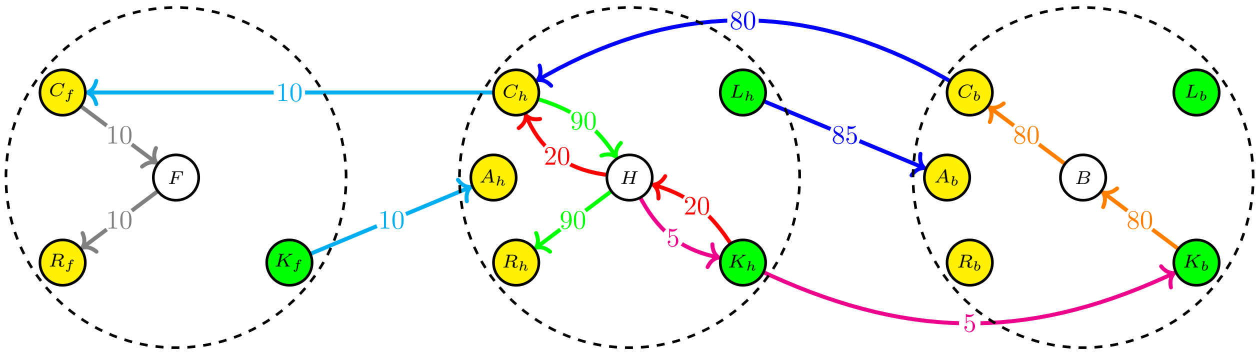


Figure 11: After setting up a firm in three stylized transactions (provide cash, obtain rights, acquire production assets) we have a fully populated QEB graph featuring explicitly three accounting entities. Several salient features of the accounting graph framework as here in display: The recording of multiple transactions (that can be optionally compacted). The presence of both external transactions (when the information is available) and internal transactions when it concerns an intrinsically internal activity or the external counterparty is not modeled. The DEB/QEB accounting balance is automatic. The central nodes always satisfy a local balance which in aggregated form is simply the depiction of the balance sheet. Other nodes (here the three cash accounts) may also satisfy such a local balance. The incidence matrix and graph are generally sparse, which suggests specific ways of storing and working with them.

[4] Wolfgang Balzer and Richard Mattessich. An axiomatic basis of accounting: a structuralist reconstruction. *Theory and Decision*, (30):213–243, 1991.

[5] D. Ellerman. *Economics, Accounting, and Property Theory.* Lexington MA: D.C. Heath., 1982.

[6] Robert A. Nehmer and Derek Robinson. An algebraic model for the representation of accounting systems. *Annals of Operations Research*, (71):179–198, 1997.

[7] Keito Ozawa. Isomorphism Between Two Algebraic Structures in Mathematical Models of Double-Entry Accounting Systems: The Pacioli Group and the Balance Module. *Preprint*, 2025.

[8] Anil Arya et al. Inferring Transactions from Financial Statements. *Contemporary Accounting Research*, (17):365–385, 2000.

[9] Miguel Carrión Álvarez and Dirk Ehnts. The Roads Not Taken: Graph Theory and Macroeconomic Regimes in Stock-flow Consistent Modeling. *Levy Economics Institute of Bard College Working Paper*, (854), 2015.

[10] P. Papadopoulos. WP12: Deep Linking Financial and Energy Accounting. *Open Risk White Papers*, 2022. Online Link.

[11] P. Papadopoulos. WP14: energyLedger: Integrated energy accounting using relational databases. *Open Risk White Papers*, 2023. Online Link.

[12] W. Godley and M. Lavoie. *Monetary Economics: An Integrated Approach to Credit, Money, Income, Production and Wealth.* Palgrave Macmillan, 2007.

[13] Harry H. Postner. Microbusiness Accounting And Macroeconomic Accounting: The Limits To Consistency. *Review of Income and Wealth*, 32(3):217–244, 1986.

[14] International Monetary Fund. Balance of Payments and International Investment Position Manual Sixth Edition (BPM6). 2009.