Formal Specification of Multi-Agent e-barter Systems ⋆

Manuel Núñez, Ismael Rodríguez, Fernando Rubio

Dept. Sistemas Informáticos y Programación
Facultad de Informática
Universidad Complutense de Madrid, E-28040 Madrid. Spain.
e-mail:{mn,isrodrig,fernando}@sip.ucm.es

Abstract

An *e-barter* multi-agent system consists of a set of agents that exchange goods. These agents may perform multilateral exchanges involving several goods. In particular, money can be one of these goods. Each agent is endowed with a *utility function* indicating the preferences of the respective user. In order to improve the performance, these barter systems are structured in a hierarchical form. Initially agents are grouped, according to localities, into local markets. Once a local market gets *completed*, that is, no more exchanges are possible, the local market itself becomes a new agent. The preferences of this agent, given by a new utility function, represent the individual preferences of its former customer agents. Then, local markets exchange goods in a higher order market until it gets completed. The process is iterated, in a bottom-up fashion, until the global market embracing all the agents in the system gets completed as well.

We provide a formal language, based on classical process algebras, to specify and analyze e-barter systems. We also study properties of e-barter systems represented in our notation. In particular, we show that the final distribution of goods in a hierarchical e-barter system is a *Pareto optimum*. In other words, we will be able to prove that economic efficiency is not lost by considering our hierarchical structure instead of a single market.

Keywords: *e-barter, formal methods, process algebras, Pareto optimum.*

⋆ Research partially supported by the Spanish *Ministerio de Ciencia y Tecnología* project MASTER (TIC2003-07848-C02-01), the *Junta de Comunidades de Castilla-La Mancha* project DISMEF (PAC-03-001), and the Marie Curie Research and Training Network *TAROT* (MRTN-CT-2003-505121).

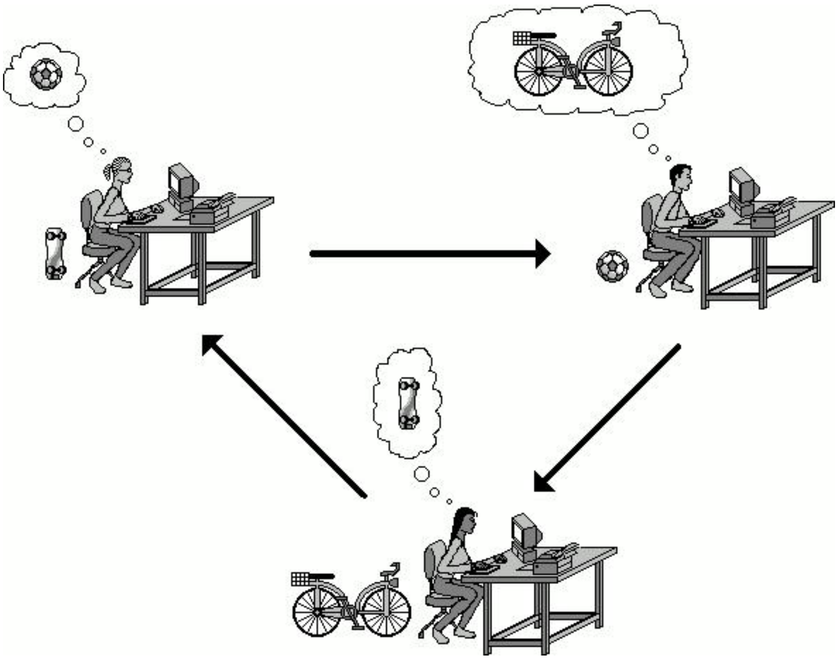

Fig. 1. Exchange of items in the presence of circular dependencies.

1 Introduction

The creation and study of communities where intelligent (electronic) agents replace their (human) owners is a topic that has attracted a lot of interest. In particular, there is ongoing research in technologies able to model users by means of agents which autonomously perform electronic transactions (see [9] for a survey on the topic). In order to increase the power of these agents they must know the preferences of the corresponding user. In this line, the concept of *utility function* is very useful. Essentially, a *utility function* returns a real number for each possible basket of goods: The bigger this number is, the happier the owner is with this basket.

Intuitively, agents should act as the customer that they are representing by considering the utility function that the corresponding user has in mind (see e.g. [25,7,6,14,19,11]). In fact, there exists several proposals showing how agents can be trained to learn the preferences of users (see e.g. [1,6,26]). Besides, a formal definition of the preferences of the user provides the agent with some negotiation capacity when interacting with other agents [12,26,15]. Let us remark that, in most cases, utility functions take a very simple form. For instance, they may indicate that a customer C is willing to exchange the item a by the items b and c. Obviously, a customer must have the possibility of changing the utility function that the agent uses to represent his preferences.

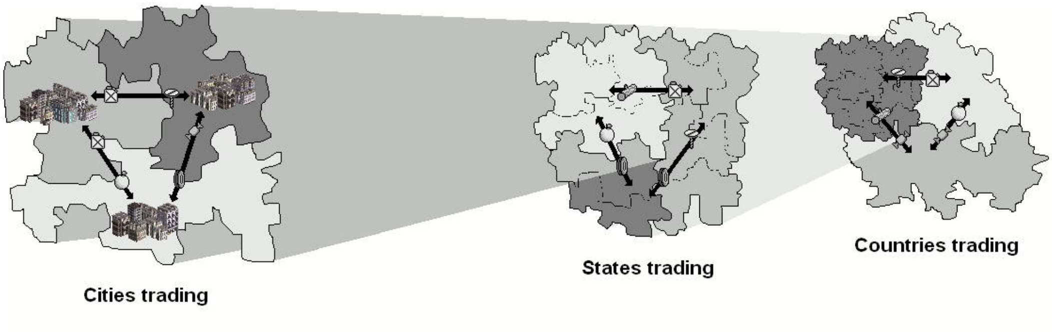

Fig. 2. Example of a hierarchical market structure.

Once the agent is notified of these modifications, it will change its negotiation strategy accordingly.

An *e-barter* system consists of a set of agents performing exchanges of products. The notion of e-barter that we will use in this paper was initially introduced in [16,17]. This work has been continued in [4] where a test derivation framework for e-barter systems has been introduced. In order to formalize e-barter systems we borrow from Microeconomics the concepts of *utility function*, *fair exchange*, and *Pareto optimum* (see [18] for a very formal and rigorous presentation of microeconomic theory). We will elaborate on these concepts in the next section. In contrast with the usual understanding of e-commerce, e-barter does not necessarily reduce all the transactions to money: An exchange is performed if the involved parts are *happier* with their new items. In our framework we consider that money is just another possible resource: An agent may be willing to exchange the good a by (m units of the good) money. Let us remark that this is not the usual treatment of money in general equilibrium theory where money is simply used to set *relative* prices.

As a matter of fact, e-barter allows a rich structure of exchanges. For example, let us suppose a very simple circular situation where for each $1 \leq i \leq r$ we have that the agent A_i owns the good a_i and desires the good $a_{(i \bmod r)+1}$. Such a situation is graphically depicted in Figure 1 for the case of 3 agents and 3 goods. This multi-agent transaction can be easily performed within our framework so that each of the agents obtains the desired item. On the contrary, it would not be so easy to perform it if these items must be first *converted* into money. For example, if the agent A_2 requests money to A_1 as *payment* for the good a_2 then it may happen that A_2 has no money to give away. As a result, the whole circular exchange could not be concluded, in spite that all agents would get happier if it were performed. Let us also note that if exchanges are restricted to be bilateral then we would also lose some good deals. For example, the agent A_2 will not be willing to give away his item since it is not interested in the item owned by A_1.

In e-barter systems, agents are grouped according to the localities of the corresponding customers (see Figure 2). First, agents are combined into *local* markets (e.g. customers living in the same city). Once this market gets *completed*, that is, when no more exchanges can be performed, an agent representing the interests of all the agents in the market is created. The new agents will be again grouped into markets (e.g. agents are grouped by counties). This situation is repeated until a *global* market is created. This hierarchical structure presents at least two advantages. First, shipping costs are diminished because agents will exchange resources as close to the location of the customer as possible. Second, by creating new (representative) agents once a market is completed and by combining them into higher order markets, we keep a small number of agents belonging to a certain market. This is very relevant if we take into account that a big number of agents would make very difficult to find the products that they are looking for. This is so because the number of messages that agents send to communicate with each other dramatically increases with the number of agents in the market. Finally, let us note that if an agent does not find the product that it is looking for in a local market, there will be a new agent looking for the same product (and taking into account the preferences of the original agent) in a *bigger* market.

In order to formally specify e-barter systems we will use a process algebraic notation based on the language `PAMR` [21,22]. This language is very suitable for our purposes because it was specially developed to deal with the specification and analysis of concurrent and distributed systems where resources play a fundamental role. Nevertheless, `PAMR` does not provide a *higher order* constructor as the one needed in e-barter systems, so the language has to be extended. In addition to a syntax, we will provide an operational semantics for the new language. By doing so, every stage of the creation of an e-barter system may be formally specified, avoiding ambiguities and providing a clear structure of the system. It is worth to point out that designers of e-barter systems do not need to go through all the semantic machinery. In fact, it is enough if they understand how the syntax of our language works, so that they can define the systems that they are interested in.

The rest of the paper is structured as follows. In Section 2 we introduce some auxiliary notation and the microeconomic concepts that we will use in this paper. Section 3 represents the bulk of the paper. First, we give an informal description of the behavior of e-barter systems. Next, we present a formalization of all the necessary concepts to specify e-barter systems. In Section 4 we discuss some issues concerning the practical implementation of our framework. Afterwards, in Section 5, we study the main theoretical properties of e-barter systems. In particular, we show that by using a hierarchical structure we do not lose economic efficiency. Next, in Section 6 we present our conclusions. Finally, in the appendix of this paper we present the proofs of some auxiliary results that are used in Section 5.

2 Basic Concepts

In this section we introduce some concepts that we will use during the rest of this paper. Specifically, we present the notions of utility function and we explain how operational rules for a process algebra are defined. First we present some mathematical notation.

Definition 1 We consider $\mathbb{R}_+ = \{x \in \mathbb{R} \mid x \geq 0\}$. We will usually denote *vectors* in $\mathbb{R}^n$ (for $n \geq 2$) by $\overline{x}, \overline{y}, \ldots$ Given $\overline{x} \in \mathbb{R}^n$, x_i denotes its *i-th* component. We extend to vectors some usual arithmetic operations. Let $\overline{x}, \overline{y} \in \mathbb{R}^n$. We define the addition of vectors $\overline{x}$ and $\overline{y}$, denoted by $\overline{x}+\overline{y}$, simply as $(x_1 + y_1, \ldots, x_n + y_n)$. We write $\overline{x} \leq \overline{y}$ if for any $1 \leq i \leq n$ we have $x_i \leq y_i$.

We will usually denote *matrices* in $A^{n\times m}$ (for $n, m \geq 2$, and a set A) by calligraphic letters $\mathcal{E}, \mathcal{E}_1, \ldots$ □

The relevant characteristics of the customers of an e-barter system are their *baskets of resources* (indicating the items that they own) and their *utility functions* (indicating preference among different baskets of resources).

Definition 2 We will suppose that there exist $p > 0$ different kinds of resources. *Baskets of resources* are defined as vectors $\overline{x} \in \mathbb{R}^p_+$. A *utility function* is a function $u : \mathbb{R}^p_+ \longrightarrow \mathbb{R}$. □

In microeconomic theory there are some restrictions that are usually imposed on utility functions (mainly strict monotonicity, convexity, and continuity). Intuitively, given a utility function u we have that $u(\overline{x}) < u(\overline{y})$ means that the basket $\overline{y}$ is preferred to $\overline{x}$.

The main feature of e-barter systems is that agents exchange resources. Let us suppose a system with n agents where p different types of products can be exchanged. Each agent has as information from the customer a pair $(\overline{x}, u)$, with $\overline{x} \in \mathbb{R}^p_+$ and $u : \mathbb{R}^p_+ \rightarrow \mathbb{R}_+$. The first component of the pair denotes the amounts that the customer owns of each kind of product. The second component is the utility function indicating the preferences of the customer with respect to the different products. A subset of agents will be willing to exchange resources if none of them decreases its utility and at least one of them improves. These exchanges are called *fair*. Formally, let us consider a set of indexes $\mathcal{A} = \{i_1, \ldots, i_m\} \subseteq \{1, \ldots, n\}$ and let us take, for any $i \in \mathcal{A}$, the pair $(\overline{x_i}, u_i)$. Let us suppose that after the exchange we have that the information associated with the agents belonging to $\mathcal{A}$ is given by the pairs $(\overline{y_i}, u_i)$, that is, the corresponding utility function remains while the basket of resources vary from $\overline{x_i}$ to $\overline{y_i}$. We will say that the exchange is fair if for any $i \in \mathcal{A}$ we have $u_i(\overline{x_i}) \leq u_i(\overline{y_i})$ and there exists $j \in \mathcal{A}$ such that $u_j(\overline{x_j}) < u_j(\overline{y_j})$. Let us remark that a necessary condition for an exchange is that no products are created/destroyed, that is, $\sum_{i \in \mathcal{A}} \overline{x_i} = \sum_{i \in \mathcal{A}} \overline{y_i}$. Eventually, the system will reach a situation where no more exchanges can be performed. In other words, it is not possible to improve the situation of one agent without deteriorating another one. Such a situation is called *Pareto optimum*. In order to determine the set of optimums of a system, techniques inherited from game theory can be used (see e.g. [28,24,27]).

We consider that customers have to pay a fee depending on the products that they exchange. Besides, shipping costs will be collected as well. In order to compute shipping costs we have to take into account not only the products that the customer receives. We also need to consider the distance between the sender and the receiver. Actually, these costs may have a strong influence in the behavior of systems. In particular, some fair exchanges will not be performed due to the additional costs. Moreover, we have to adapt the notion of Pareto optimum. [1] Let us illustrate this situation by means of a simple example.

Example 1 Let us consider a system with two users and two products. In addition, we consider money as the third product. Let us suppose that the initial distributions are $(0, 3, 5)$ and $(2, 1, 10)$, respectively, while the corresponding utility functions are defined as $u_1(x_1, x_2, x_3) = 30 \cdot x_1 + 10 \cdot x_2 + x_3$ and $u_2(x_1, x_2, x_3) = 10 \cdot x_1 + 10 \cdot x_2 + x_3$, respectively. Intuitively, the first user is indifferent between one unit of the first product and three units of the second product. If the first user gives two units of the second product in exchange for one unit of the first product, both users improve. That is, $u_1(0, 3, 5) < u_1(1, 1, 5)$ and $u_2(2, 1, 10) < u_2(1, 3, 10)$. However, this exchange could be disallowed if we consider transaction and shipping costs. In this case, we would have to decide whether we have both $u_1(0, 3, 5) \leq u_1(1, 1, 5 - t_1 - c_1)$ and $u_2(2, 1, 10) \leq u_2(1, 3, 10 - t_2 - c_2)$, where t_i denotes the transaction costs associated with the goods received by i while c_i denotes the shipping costs according to the distance between the users. Besides, in order to have a fair exchange, one of the previous inequalities must be strict. If the exchange is performed, the manager/owner of the system will increase its amount of money by $t_1 + t_2$ units. Thus, the total amount of money owned by the users is reduced in $t_1 + t_2 + c_1 + c_2$ units. $\square$

Process Algebras (see [10,20,2] for the classical notions and [3] for a recent overview on the *hot topics*) are formal languages used to specify and verify distributed and concurrent systems. As we pointed out in the introduction of this paper, we will use such a language to formalize e-barter systems. The syntax of these languages is given as an EBNF expression. In order to assign

[1] In microeconomics terms, the problem is that we partially lose the notion of *contract curve* because the induced generalized Edgeworth box shrinks after an exchange. Specifically, the problem is that money is taken out from the system due to the mentioned costs. We assume that these costs are collected by a generic agent that it is not explicitly represented.

meaning to syntactic terms, an operational semantics is usually introduced. Operational rules are defined as deduction rules. That is, a rule

$$\frac{\text{Premise}_1 \ \wedge \ \text{Premise}_2 \ \wedge \ \ldots \ \wedge \ \text{Premise}_n}{\text{Conclusion}}$$

must be interpreted as: If all of the premises hold then we can deduce the conclusion. Premises usually indicate individual behavior of components of a system; conclusions indicate how the system behaves according to individual performances. Let us remark that if a rule has no premises then the conclusion trivially holds.

3 Formalizing e-barter Systems

In this section we present how e-barter systems are organized. First, we show the basic algorithm underlying the definition of e-barter systems. Next we introduce the formal framework to specify e-barter systems.

3.1 Basic Behavior of e-barter Systems

Customers willing to participate in an e-barter system are *represented* by (electronic) agents. [2] These agents are provided with two parameters: The *basket of resources* that the customer is willing to exchange and a *utility function*. Once an agent has reached a (possibly multilateral) deal, it must be notified to the customer. If all the customers give their approval then the deal will be effectively performed, transaction fees will be added, and shipping costs will be computed according to both the amount of received items and the distance between the involved customers.

From now on we concentrate on the behavior of the different electronic entities. Even though we will formally define the behavior of e-barter systems, it is convenient to start by giving an informal explanation of how these systems work. Essentially, the behavior of an e-barter system follows this algorithm:

(1) Each agent generates the barters that its customer would be willing to perform (according to the corresponding basket of resources and utility functions).

[2] In terms of [30], our agents present as information attitude belief (vs. knowledge), while as pro-attitudes we may consider commitment and choice (vs. intention and obligation).

(2) Agents exchange goods inside their local market. A multilateral exchange will be performed if (at least) one of the involved agents improves its utility and none of them decreases its utility. This is repeated until no more exchanges are possible. In this case we say that the local market is *completed*.
(3) Once a market is completed, their agents are combined to create a new agent. This agent behaves as a representative of the combined agents. The new agent will have as basket of resources the union of the baskets corresponding to each agent. Its utility function will encode the utilities of the combined agents. *First order* agents will be combined again into markets, according to *proximity* reasons.
(4) Higher order agents trade until their market gets completed.
(5) Once a (higher order) market gets completed, the agents start to allocate the resources in a top-down way through the tree of markets until the resources arrive to the leaves of the tree (i.e. the *original* agents). Then, a new agent is created (as indicated in step 3).
(6) Once their markets get completed, new markets are created by combining agents until there exists a unique market. Once this last market gets completed, and the resources are conveniently allocated, the whole tree of agents is reset, and we start again at the first step.

As we have already indicated in the introduction of the paper, the previous behavior ensures some good properties. In particular, exchanges are made between agents located as near as possible, that is, shipping costs are minimized. Besides, the hierarchical structure improves the computational efficiency of the system. This is so because the communication overload would make the efficiency to fall in a system where all the agents are connected to a single market. In such scenario, a single market would be responsible for all the transactions and communications. Thus, its delays would become the bottleneck of the system performance. Moreover, if the single market crashes then no exchange will be possible in the system anymore (no matter whether the exchange involves nearby agents or not). Thus, the hierarchical structure is safer and more efficient from the computational point of view. In addition, as we will show in Section 5, the *economic* efficiency of a hierarchical market matches that of a non hierarchical market.

3.2 e-barter Systems: Syntax and Semantics

In this section we provide a formal syntax and semantics for the definition of e-barter systems. Even though we use a process algebraic notation (mainly when defining the operational rules) we do not need most of the usual operators appearing in this kind of languages (choice, restriction, etc). In fact, our constructions remind a parallel operator as the one presented, for example, in the process algebra CCS [20].

Definition 3 A *market system* is given by the following EBNF:

$$\begin{aligned} MS &::= ms(M) \\ M &::= A \mid \texttt{uncomp}((M,\ldots,M),sh,pr) \mid \sigma(M) \\ A &::= (S,u,\overline{x},sh,pr) \\ S &::= [\,] \mid [A,\ldots,A] \end{aligned}$$

□

First, in order to avoid ambiguity of the grammar, we annotate market systems with the terminal symbol ms. Intuitively, the market $M = (S,u,\overline{x},sh,pr)$ (that is, $M = A$) represents a *completed* market, that is, a market where no more exchanges can be performed among its agents. Let us note that in this case the market represents an *agent* that will be able to make transactions with other agents in a higher market. In the previous expression, u denotes the utility function of M and $\overline{x}$ represents the basket of resources owned by M. We consider that there are p different commodities,[3] that is $\overline{x} \in \mathbb{R}^p_+$, and that the amount of *money* is placed in the last component of the tuple. Besides, sh is the *shipping function* indicating the shipping cost of each possible transaction in this market. In turn, pr is the profit collected by the market due to transaction costs. We will assume the existence of another function, the *transaction function*, denoted by tr. This last function computes the transaction costs for each of the agents involved in an exchange by taking into account the goods that each agent receives. Let us remark that while shipping costs will depend on the market in which the transaction is performed, transaction costs do not. By doing so we can formally specify that shipping costs increase with the distance between customers.

Regarding the first argument of M there are two possible situations. Either S is an empty list or not. In the first case we have that M represents an *original* agent, that is, a direct representative of a customer (note that a single agent is trivially completed since there is nobody to deal with). In the second case, if $S = [A_1,\ldots,A_n]$ then we have that M represents an agent associated with the (possible higher order) agents $A_1,\ldots,A_n$ belonging to a completed market.

The second possible syntactic form of M, $\texttt{uncomp}((M_1,\ldots,M_n),sh,pr)$, represents an *uncompleted* market consisting of the markets $M_1,\ldots,M_n$, the shipping function sh and the profit value pr. Let us remark that in this case some

[3] We are assuming that all the items are *goods*. Nevertheless, agents could also trade *bads*. For example, a customer would be willing to give an apple pie if he *receives* minus s brown leaves in his garden. However, bads are usually not considered in microeconomic theory, as they can be easily turned into goods: Instead of considering the amount of leaves, one may consider the absence of them.

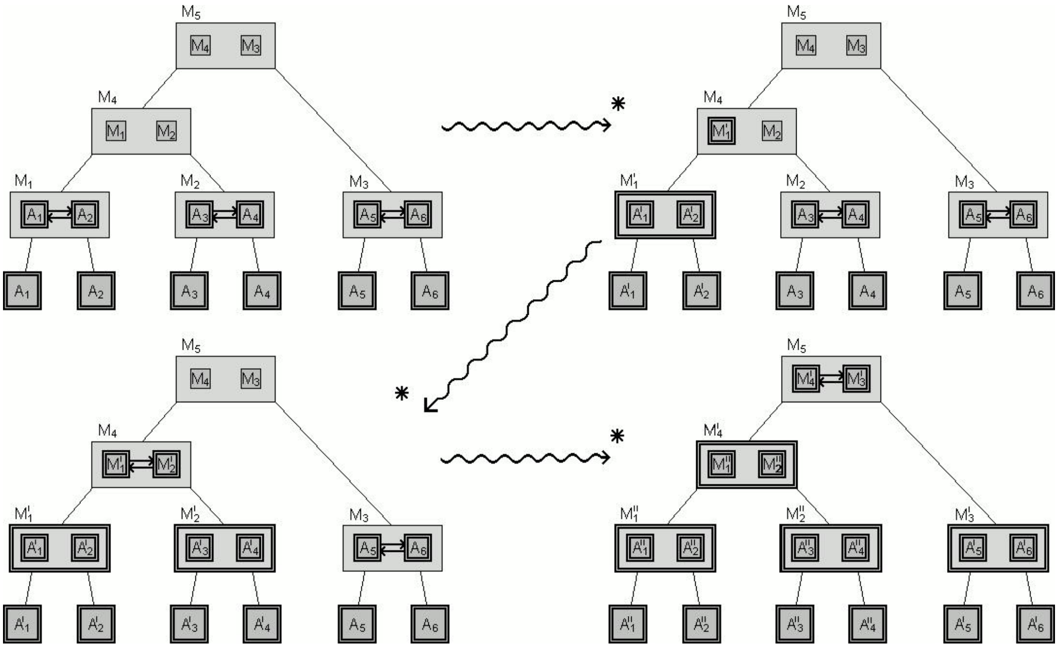

Fig. 3. A market system and some operational transitions.

of the sub-markets may be completed. Once all the markets of the system get completed, the whole system is turned again into uncompleted. The last form of M, $\sigma(M)$, represents that such operation must be performed on M.

Next we present an example showing how an e-barter system may be constructed. In this example we will also (informally) introduce the operational transitions of the language.

Example 2 Let us consider a system with six agents $A_i = ([\,], u_i, \overline{x_i}, sh_0, 0)$, for $1 \leq i \leq 6$, where sh_0 denotes a dummy shipping function. We suppose that these agents are grouped into three different markets. Initially, these markets are uncompleted (uncompleted markets are represented by a single square in the figure), so we make the following definitions:

$$M_1 = \texttt{uncomp}((A_1, A_2), sh_1, 0)$$

$$M_2 = \texttt{uncomp}((A_3, A_4), sh_2, 0)$$

$$M_3 = \texttt{uncomp}((A_5, A_6), sh_3, 0)$$

Let us consider that the first two markets are linked, and that the resulting market is also linked to the remaining market M_3. We should add the following definitions:

$$M_4 = \texttt{uncomp}((M_1, M_2), sh_4, 0)$$

$$M_5 = \texttt{uncomp}((M_4, M_3), sh_5, 0)$$

Finally, the global market is defined as $M = ms(M_5)$. This hierarchical structure is graphically presented in Figure 3, top-left.

Following the philosophy explained in the previous section, transactions will be made within a market only among completed sub-markets. So, initially only M_1, M_2, and M_3 are allowed to perform transactions (as we remarked before, original agents are trivially completed).

We will use the symbol $\rightsquigarrow$ to denote exchange of resources. Let us suppose that, after some exchanges, M_1 gets completed. That is, there exists a sequence of exchanges $M_1 \rightsquigarrow M_1^1 \rightsquigarrow M_1^2 \cdots \rightsquigarrow M_1^n = M_1'$ such that $M_1' \not\rightsquigarrow$. We will usually write $M_1 \rightsquigarrow^* M_1'$ to denote that such a sequence of exchanges can be performed. In this case, the market grouping the first two agents should be labeled as completed. So, the agents effectively perform all the achieved transactions becoming A_1' and A_2', respectively. Then, the first market will be turned into $([A_1', A_2'], f(u_1, u_2), \overline{x_1} + \overline{x_2} - (0, 0, \ldots, costs_1), sh_1, pr_1)$, where f is a function combining utility functions (such a function will be formally defined later), pr_1 is the amount of money that the system has obtained due to the fees applied to the exchange of goods, and $costs_1$ denotes the transaction and shipping costs associated with the products exchanged in the market M_1. In parallel, M_2 will have a similar behavior.

Once both M_1 and M_2 get completed, transactions between them will be allowed. Note that these transactions (inside the market M_4) will be performed according to the new utility functions, $f(u_1, u_2)$ and $f(u_3, u_4)$, and to the new baskets of resources, $\overline{x_1} + \overline{x_2} - (0, 0, \ldots, costs_1)$ and $\overline{x_3} + \overline{x_4} - (0, 0, \ldots, costs_2)$.

The process will iterate until M_5 gets completed. At this point, we will have a market as $\sigma(M_5')$. Then, the global market is structurally *reset*. $\square$

In order to simplify forthcoming operational rules we introduce the following notation to deal with utility functions. Utility functions associated with original agents (that is, $A = ([\,], u, \overline{x}, sh, 0)$) will behave as explained in Definition 2. That is, $u(\overline{z})$ indicates the relative preference shown by A towards the basket of resources $\overline{z}$. Nevertheless, if $A = ([A_1, \ldots, A_n], u, \overline{x}, sh, pr)$ then we will consider that, in addition to its usual meaning, the utility function also keeps track of how a basket of resources is distributed among the (possible higher order) agents $A_1, \ldots, A_n$. That is, $u(\overline{z}) = (r, \overline{z_1}, \ldots, \overline{z_n})$, where r still represents the utility, while $\sum \overline{z_i} = \overline{z}$ and $\overline{z_i}$ denotes the portion of the basket $\overline{z}$ assigned to A_i. Overloading the notation, if we simply write $u(\overline{z})$ we are referring to the first component of the tuple, while $u(\overline{z}).i$ denotes the $(i+1)$-*th* component of the tuple. Later we will see how the utility functions following this pattern are actually defined.

In the next definition we present the *anchor case* of our operational semantics. In order to perform complex exchanges, agents should first indicate the barters they are willing to accept.

Definition 4 Let $A = (S, u, \overline{x}, sh, pr)$ be a completed market. The *exchanges* the agent A would perform are given by the following operational rules:

$$\frac{u(\overline{x}+\overline{y}) \geq u(\overline{x}) \ \wedge \ (\overline{x}+\overline{y}) \geq \overline{0}}{(S, u, \overline{x}, sh, pr) \xrightarrow{\overline{y}} (S, u, \overline{x}+\overline{y}, sh, pr)}$$

$$\frac{u(\overline{x}+\overline{y}) > u(\overline{x}) \ \wedge \ (\overline{x}+\overline{y}) \geq \overline{0}}{(S, u, \overline{x}, sh, pr) \stackrel{\overline{y}}{\longmapsto} (S, u, \overline{x}+\overline{y}, sh, pr)}$$

where $\overline{y} \in \mathbb{R}^p$, being p the number of different commodities. □

Let us remark that $\overline{y}$ may have negative components. Actually, these tuples will contain the barters offered by the agent. For example, if $\overline{y} = (1, -1, 0, -3)$ fulfills the premise then the agent would accept a barter where it is offered one unit of the first product in exchange of one unit of the second good and three units of money. Regarding the rules, the first premise simply indicates that the agent would not decrease (resp. would increase) its utility. The second premise indicates that the agent does not run into *red numbers*, that is, an agent cannot offer a quantity of an item if it does not own it. Thus, a transition as $\longrightarrow$ denotes that the agent does not worsen; a transition as $\longmapsto$ denotes that the agent does improve.

Even though transaction and shipping costs do not explicitly appear in the previous rules, they are implicitly reflected in the last component of the corresponding tuples $\overline{y}$. That is, they are considered in the amount of money that the agents are willing to give away to perform a given exchange. As we mentioned before, these costs may have an important influence in the exchanges that an agent will be willing to perform: High shipping/transaction costs will produce that some exchanges are less attractive.

Example 3 Let us consider an agent that valuates the price of an apple in 1\$. This amount includes any additional costs that could be associated with the acquisition of an apple. In fact, the agent should not care about whether the proportion of shipping and transaction costs is high or not, provided that the final cumulated price is not higher than 1\$. Let us suppose another agent who has apples and it is willing to sell them by any amount bigger than 0.9\$. These two agents can indeed exchange apples by money at any price $0.9 \leq price \leq 1$. However, if shipping plus transaction costs are equal to 0.15\$ then the deal will not be possible. □

We will later formalize how additional costs are assigned to the *owner* of the system and to the shipping companies. Next we show how offers are combined.

Definition 5 Let $M = \texttt{uncomp}((M_1, \ldots, M_n), sh, pr)$ be an uncompleted market and $I = \{s_1, \ldots, s_r\} \subseteq \{1, \ldots, n\}$ be a set of indexes denoting the

$$\frac{\exists\, k \in I : M_k \overset{\overline{y_k}}{\longmapsto} M'_k \ \wedge\ \forall\, i \in I, M_i \overset{\overline{y_i}}{\longrightarrow} M'_i \ \wedge\ \texttt{valid}(M, \mathcal{E}, (cost_1, \ldots, cost_n))}{M \overset{\mathcal{E}}{\rightsquigarrow} \texttt{uncomp}((M'_1, \ldots, M'_n), sh, pr + \sum_i tr(\sum_j \mathcal{E}_{ji}))}$$

where • $M = \texttt{uncomp}((M_1, \ldots, M_n), sh, pr)$

• $\mathcal{E} \in (\mathbb{R}^p_+)^{n \times n}$ and $I = \{s_1, \ldots, s_r\} \subseteq \{1, \ldots, n\}$

• $M'_i = \begin{cases} M_i & i \notin I \\ (S_i, u_i, \overline{x_i} + \overline{y_i}, sh_i, pr_i) & \text{otherwise} \end{cases}$

• $\overline{y_i} = \sum_j \mathcal{E}_{ji} - \sum_j \mathcal{E}_{ij} - (0, \ldots, 0, cost_i)$, for any $i \in I$

• $cost_i = tr(\sum_j \mathcal{E}_{ji}) + sh(\sum_j \mathcal{E}_{ji})$, for any $i \in I$

Fig. 4. Operational rule for the exchange of resources in an uncompleted market.

completed markets belonging to M (that is, for any $i \in I$ we have that $M_i = (S_i, u_i, \overline{x_i}, sh_i, pr_i)$). We say that the matrix $\mathcal{E} \in (\mathbb{R}^p_+)^{n \times n}$ is a *valid exchange matrix for M under the cost tuple $\overline{c}$*, denoted by $\texttt{valid}(M, \mathcal{E}, \overline{c})$, if the following conditions hold:

- For any $1 \leq i \leq n$ we have $\sum_j \mathcal{E}_{ij} \leq \overline{x_i} - (0, \ldots, 0, c_i)$,
- for any $1 \leq i \leq n$ we have $\mathcal{E}_{ii} = \overline{0}$, and
- for any $1 \leq i, k \leq n$ such that $k \notin I$ we have $\mathcal{E}_{ki} = \overline{0}$ and $\mathcal{E}_{ik} = \overline{0}$.

□

First, let us note that the notion of *valid* matrix is considered only in the context of uncompleted markets: If a market is already completed then no more exchanges can be performed. Second, only completed markets belonging to an uncompleted one may perform exchanges among them. This restriction allows to give priority to transactions performed by *closer* agents belonging to uncompleted sub-markets. Regarding the definition of *valid* matrix, the components of matrixes $\mathcal{E}$ are baskets of resources (that is, elements belonging to $\mathbb{R}^p_+$). Thus, $\mathcal{E}_{ij}$ represents the basket of resources that the market M_i would give to M_j. In the tuple $\overline{c}$, the component c_i denotes the transaction and shipping costs that the agent M_i will have to afford. So, for any market M_i, the condition $\sum_j \mathcal{E}_{ij} \leq \overline{x_i} - (0, \ldots, 0, c_i)$ indicates that the total amount of resources given by this market to other markets must be less than or equal to the basket of resources owned by that market minus the money paid by the transaction. Finally, let us comment that an exchange does not need to include all of the completed markets. That is, if we have an exchange where only r' markets participate, then the rows and columns corresponding to the remaining $r - r'$ completed markets will be filled with $\overline{0}$. Besides, the rows and columns corresponding to the $n-r$ uncompleted markets will be also filled with $\overline{0}$.

Next we introduce the rules defining the exchange of resources. Intuitively, if we have a valid exchange matrix where at least one of the involved agents improves and no one worsens then the corresponding exchange can be performed.

Definition 6 Let $M = \texttt{uncomp}((M_1, \ldots, M_n), sh, pr)$ be an uncompleted market and $I = \{s_1, \ldots, s_r\} \subseteq \{1, \ldots, n\}$ be a set of indexes denoting the completed markets belonging to M (that is, for any $i \in I$ we have that $M_i = (S_i, u_i, \overline{x_i}, sh_i, pr_i))$. The operational transitions denoting the exchange of resources that M may perform are given by the rule shown in Figure 4. We say that M is a *local optimum*, denoted by $M \not\rightsquigarrow$, if there do not exist M' and $\mathcal{E}$ such that $M \overset{\mathcal{E}}{\rightsquigarrow} M'$. □

The operational rule presented in Figure 4 is applied under the same conditions appearing in the definition of a valid exchange matrix: It is applied to uncompleted markets and the exchange is made among a subset of the completed sub-markets. The premises indicate that at least one completed market improves after the exchange and that none deteriorates. Let us remind that, in general, a market may generate both $M_i \xrightarrow{\overline{y}} M_i'$ and $M_i \overset{\overline{y}}{\longmapsto} M_i'$. So, the previous rule also considers situations where more than one market improves (we only require that at least one improves). Besides, let us remark that $M_i \xrightarrow{\overline{0}} M_i'$ always holds. So, a market not involved in the current exchange does not disallow the exchange. The tuple of costs appearing in the condition ensuring the validity of the exchange matrix will be computed from both transaction and shipping costs. Regarding the conclusion, sub-markets belonging to M are modified according to both the corresponding exchange matrix and the costs of the exchange, while uncompleted sub-markets do not change. Let us remark that the costs of each exchange will be paid by the receiver. Besides, only the transaction costs will be added to the cumulated profit of the market. The following result easily follows from the previous definition. It indicates that exchanges allowed by the previous rule are fair.

Proposition 1 Let $M = \texttt{uncomp}((M_1, \ldots, M_n), sh, pr)$ be an uncompleted market and let $I = \{s_1, \ldots, s_r\} \subseteq \{1, \ldots, n\}$ be the set of indexes denoting the completed markets belonging to M (that is, for any $i \in I$ we have that $M_i = (S_i, u_i, \overline{x_i}, sh_i, pr_i))$. Let us suppose that there exists a valid exchange matrix $\mathcal{E}$ such that we have a transition

$$\texttt{uncomp}((M_1, \ldots, M_n), sh, pr) \overset{\mathcal{E}}{\rightsquigarrow} \texttt{uncomp}((M_1', \ldots, M_n'), sh, pr')$$

where $M_i' = (S_i, u_i, \overline{x_i'}, sh_i, pr_i))$. Then, for any $i \in I$ we have $u_i(\overline{x_i}) \leq u_i(\overline{x_i'})$. In addition, there exists $j \in I$ such that $u_j(\overline{x_j}) < u_j(\overline{x_j'})$. □

In addition to the previous inference rule we need to consider the following two exchanging rules:

$$\frac{M_k \overset{\mathcal{E}}{\rightsquigarrow} M_k'}{\texttt{uncomp}((M_1,\ldots,M_k,\ldots,M_n),sh,pr) \overset{\mathcal{E}}{\rightsquigarrow} \texttt{uncomp}((M_1,\ldots,M_k',\ldots,M_n),sh,pr)}$$

$$\frac{M \overset{\mathcal{E}}{\rightsquigarrow} M'}{ms(M) \overset{\mathcal{E}}{\rightsquigarrow} ms(M')}$$

The first rule indicates that if an uncompleted sub-market produces an exchange then the market must take that situation into account. The second rule reflects modifications within the scope of the constructor ms.

If a market reaches an optimum then we need to modify the attribute of the market by replacing a term such as $\texttt{uncomp}((M_1,\ldots,M_n),sh,pr)$ by a term such as $(S,u,\overline{x},sh,pr')$. Once a market gets completed, the money collected in the different sub-markets as transaction costs will be transferred to it. In addition, resources are recursively moved from the corresponding agents to the leaves of the tree. Let us remark that a market gets completed when all of its sub-markets are completed. The following rule uses two auxiliary notions that will be formally presented in the forthcoming Definition 8.

Definition 7 Let $M = \texttt{uncomp}((M_1,\ldots,M_n),sh,pr)$ be a market, where for any $1 \le i \le n$ we have $M_i = (S_i,u_i,\overline{x}_i,sh_i,pr_i)$. The following rule modifies the market from uncompleted to completed:

$$\frac{M \not\rightsquigarrow}{M \rightsquigarrow ([M_1',\ldots,M_n'],u,\sum \overline{x_i},sh,pr+\sum_i pr_i)}$$

where $u = \texttt{CreateUtility}(u_1,\ldots,u_n,\overline{x_1},\ldots,\overline{x_n})$ and for any $1 \le i \le n$ we have $M_i' = (S_i',u_i,\overline{x_i},sh_i,0)$ and $S_i' = \texttt{Deliver}(S_i,u_i,\overline{x_i})$. □

Let us remark that in the previous rule the transition $\rightsquigarrow$ is not labelled. These transitions play a role similar to internal transitions in classical process algebras. The presence of a negative premise in the previous rule deserves some comment. In fact, negative premises can induce inconsistent transition systems. Fortunately, we do not have this problem. Intuitively, let us note that no transition $\rightsquigarrow$ is a premise of a rule deriving a $\overset{\mathcal{E}}{\rightsquigarrow}$ transition, for some valid exchange matrix $\mathcal{E}$. Thus, we cannot construct a cycle such as $M \not\rightsquigarrow$ implies $M \overset{\mathcal{E}}{\rightsquigarrow} M'$, for some M'. Actually, it is very easy to give a *stratification* to ensure that the transitions systems are well defined (the interested reader can check [8] to see how stratifications are defined).

We need to add two more rules, as in the previous case, to record transformations given by a $\rightsquigarrow$ transition in the context of different constructors:

$$\frac{M_k \rightsquigarrow M'_k}{\texttt{uncomp}((M_1, \ldots, M_k, \ldots, M_n), sh, pr) \rightsquigarrow \texttt{uncomp}((M_1, \ldots, M'_k, \ldots, M_n), sh, pr)}$$

$$\frac{M \rightsquigarrow M'}{ms(M) \rightsquigarrow ms(M')}$$

In the following definition we present the pending functions. Intuitively, the function $\texttt{Deliver}(S, u, \overline{x})$ distributes the basket of resources $\overline{x}$ among the original agents, which are located in the leaves of the tree S. This distribution considers both the utility functions of the agents and the quantities of resources contributed by each of the agents. Besides, the application of the function $\texttt{CreateUtility}(u_1, \ldots, u_n, \overline{x_1}, \ldots, \overline{x_n})$ computes a combined utility function from the ones provided as arguments. Let us remind that, in our context, utility functions of higher order agents do not only reveal preference. In addition, they also record how resources will be distributed among agents. Thus, if we are considering an agent representing n agents, a new utility function returning a tuple with $n+1$ components will be created. The first component contains the value of the utility function. It will return the smallest utility (that is, 0) if any of the represented agents worsens after the new distribution of goods. So, it is guaranteed that the market does not perform a exchange which deteriorates any of its clients. In the general case, the value of the utility will be the addition of the individual utilities applied to the distribution of resources which maximizes this value. The next components contain the portion of the resources assigned to each of the agents.

Definition 8 Let $A = (S, u, \overline{x}, sh, pr)$ be an agent. The *allocation* of the basket of resources $\overline{x}$ among the agents belonging to S with respect to the utility function u, denoted by $\texttt{Deliver}(S, u, \overline{x})$, is recursively defined as:

$$\texttt{Deliver}(S, u, \overline{x}) = \begin{cases} [\,] & \texttt{if } S = [\,] \\ [M'_1, \ldots, M'_n] & \texttt{if } S = [M_1, \ldots, M_n] \end{cases}$$

where for any $1 \leq i \leq n$ we have that if $M_i = (S_i, u_i, \overline{x_i}, sh_i, pr_i)$ then $M'_i = (\texttt{Deliver}(S_i, u_i, u(\overline{x}).i), u_i, u(\overline{x}).i, sh_i, pr_i)$.

Let us consider n pairs $(u_i, \overline{x_i})$. The *utility function* constructed from the utility functions $u_1, \ldots, u_n$ with respect to the baskets of resources $\overline{x_1}, \ldots, \overline{x_n}$, denoted by $\texttt{CreateUtility}(u_1, \ldots, u_n, \overline{x_1}, \ldots, \overline{x_n})$, is defined as:

$$\max\{(\textstyle\sum_i u_i(\overline{x'_i}), \overline{x'_1}, \ldots, \overline{x'_n}) \mid \textstyle\sum_i \overline{x'_i} = \overline{x} \ \wedge \ \forall\, 1 \leq i \leq n : u_i(\overline{x'_i}) \geq u_i(\overline{x_i})\}$$

We consider that the previous maximization is performed over the first argument (representing the *utility*) and we assume $\max \emptyset = (0, \overline{0}, \ldots, \overline{0})$. □

$$\frac{}{ms((S,u,\overline{x},sh,pr)) \Rightarrow_{pr} ms(\sigma((S,u,\overline{x},sh,0)))}$$

$$\frac{}{\sigma(([\,],u,\overline{x},sh,pr)) \hookrightarrow ([\,],u,\overline{x},sh,pr)}$$

$$\frac{S = [A_1,\ldots,A_n]}{\sigma((S,u,\overline{x},sh,pr)) \hookrightarrow \mathtt{uncomp}((\sigma(A_1),\ldots,\sigma(A_n)),sh,pr)}$$

$$\frac{M_k \hookrightarrow M_k'}{\mathtt{uncomp}((M_1,\ldots,M_k,\ldots,M_n),sh,pr) \hookrightarrow \mathtt{uncomp}((M_1,\ldots,M_k',\ldots,M_n),sh,pr)}$$

$$\frac{M \hookrightarrow M'}{ms(M) \hookrightarrow ms(M')}$$

Fig. 5. Rules to reset a global market.

Next, in order to define how markets evolve, we compose sequences of transitions.

Definition 9 We say that a market M *evolves into* a market M', denoted by $M \rightsquigarrow^* M'$, if there exist markets $M_1,\ldots,M_{n-1}$ such that

$$M \overset{a_1}{\rightsquigarrow} M_1 \overset{a_2}{\rightsquigarrow} M_2 \overset{a_3}{\rightsquigarrow} \cdots M_{n-1} \overset{a_n}{\rightsquigarrow} M'$$

where for any $1 \leq i \leq n$ we have that a_i is either an empty label or an exchange matrix. □

Finally, we provide a mechanism to reset a global market. If the root of the tree becomes a completed market then the whole tree of markets is created again. This is done by considering the five rules shown in Figure 5. The first one initiates the process of turning all the markets back to uncompleted mode, provided that the global market has become completed. In this case, the transition is labelled by the global amount of money collected as transaction fees. The other rules define how to recursively reset the tree from the root to the leaves (original customers).

4 Practical Implementation of an e-barter System

In spite that the main scope of our framework is theoretical, we would like to briefly and informally discuss some issues that should be addressed in order to implement an e-barter system. First of all, we need to have in mind that there must be an underlying hierarchical structure to appropriately represent the system. Next, the most important decision consist in choosing one of the different possibilities for the architecture of an e-barter system. The main point guiding the design of the architecture is whether (higher-order or basic) agents, connected to some market in the hierarchical structure, should be able to negotiate and make transactions in a pairwise way, without the monitoring of a third party. If we consider that this is the case then the negotiation process would be decentralized and the architecture of an e-barter system would be close to use a *peer to peer* system for each of the represented markets. Previous work shows how, under some specific conditions, a pairwise interaction of agents following a given set of rules can yield an overall optimal distribution of resources (see [13] for one of the first proposals in this line). These architectures have the advantage of eliminating the necessity of solving the optimization problem in a centralized way. However, the necessity of a third party is not completely removed. In particular, it will be still needed to put in contact agents with somehow complementary necessities. Besides, there exists a constraint on the utility functions since they are required to have specific forms to guarantee the convergence to an optimum. In fact, as we showed in the introduction of this paper, there are simple situations where bilateral fair exchanges, that is, where none of the two involved agents worsens, are not enough to guarantee that an optimum is reached.

Since the use of a hierarchical structure allows to bind the maximal number of agents participating in each optimization subproblem, the delegation of optimization tasks to third parties is not a big disadvantage. In fact, in contrast to the previous *decentralized* alternative, we propose an architecture where a (logical) server is allocated to represent each of the markets represented in the structural tree. Initially, each basic agent sends its utility function to a local market. This market will optimize the distribution of resources according to the received utility functions. Once this market is completed, it becomes a higher-order agent, and it sends its (representative) utility function to a higher-order market, and so on. Let us note that agents must trust that the markets will perform the optimizations in a fair manner. So, activities of markets should be verified and certified by some certification institution.

Regarding fairness, it is very important that utility functions are kept *private*. That is, no other agent should know a utility function but its owner. If an agent knew the utility functions of other agents trading within its market, it could modify its own utility function so that the set of possible optimums were more favorable to its *true* interests. Hence, secure channels should be used to communicate utility functions to market servers. Since guessing the utility functions of other users can be profitable, agents have an incentive to speculate on the behavior of other agents, which opens the possibility of complex strategic behaviors. In general, using rules that promote complex behaviors is undesirable since computational effort will be wasted in the overall. Some schema, like the Generalized Vickrey Auction (see for example [29,18]), impose special rules to guarantee that users do not have any incentive to *lie* when communicating their utility functions. Basically, each agent must pay an additional fee that corresponds to the amount of utility lost by other agents because of the participation of this agent. Let us note that this approach requires to perform $n + 1$ optimizations, being n the number of participants.

In order to resolve each of the implicit optimization problems, we can still use techniques inherited from game theory. In particular, several algorithms that converge to optimal distributions through the iteration of successive adjustments have been proposed (see for example [13,5,23]). Let us remark that resolving the problem of optimizing the addition of functions is, in general, a very (computationally) complex problem. This is the reason why these iterative algorithms usually carry a constraint on the specific form of utility functions. Thus, a tradeoff has to be taken between the expressivity to denote preferences and the computational power to resolve the problem.

Finally, since reaching an optimum distribution of resources can be difficult, we might remove the necessity of finding optimal distributions. In fact, this is the way in which human economic interaction is usually performed. In this case, suboptimal algorithms could be provided so that the trade among agents stops when a *good enough* distribution of resources is reached. Among the most relevant alternatives we may mention the following:

- To iterate until a certain number of transactions has been performed.
- To iterate while the amount of exchanged goods in each transaction surpasses some minimal threshold.
- To iterate while the variation of utility of the involved agents is higher than a minimal threshold.

5 Properties of e-barter Systems

In this section we formally study the main economic properties of our e-barter systems. In particular, we will show that the economic efficiency of a hierarchical e-barter system matches (at least) that corresponding to a non-hierarchical system where all the agents are directly linked to one single market embracing all the agents. This means that the final configurations reached in the hierarchical system will be as good as those we would reach in a non-hierarchical system. In particular, any final distribution of goods reached in a hierarchical e-barter system would actually be a Pareto optimum in a system where a (unique) centralized market were used. Actually, due to the fact that our e-barter systems encourage exchanges between nearby users, we have that shipping costs are reduced. In fact, if some clients find what they desire in a local market then they will be able to satisfy their necessities at low shipping costs. So, the utility lost by clients in terms of shipping costs will be low. Transactions will be made in higher-order markets only if some necessities could not be satisfied in local markets. Thus, the economic efficiency of a hierarchical structure, from the clients point of view, could be higher than that of a centralized market. Our formal analysis will focus on checking that all distributions reached by our systems are Pareto optimal, which ensures that, *at least*, they match the efficiency of a non-hierarchical system. In addition to the previously commented drawbacks of having a unique market, hierarchical systems are more efficient from a computational point of view. In particular, the number of messages exchanged between agents strongly decreases.

Next, we introduce some preliminary definitions. The first one is used to identify the set of *basic agents* of a market (i.e. the customer agents located in the leaves of the structural tree). This definition applies to both completed and uncompleted markets. The basic agents will be given in the form of pairs (utility function, tuple of resources).

Definition 10 Let M be a market. The set of *basic agents* of M, denoted by $\texttt{Basic}(M)$, is given by the set of pairs recursively defined as follows:

$$\texttt{Basic}(M)=\begin{cases}\{(u,\overline{x})\} & \texttt{if}\ M = ([\,], u, \overline{x}, sh, pr) \\ \bigcup_i \texttt{Basic}(M_i) & \texttt{if} \begin{pmatrix} M = ([M_1,\ldots,M_n], u, \overline{x}, sh, pr) \\ \vee \\ M = \texttt{uncomp}((M_1,\ldots,M_n), sh, pr) \end{pmatrix} \wedge n \geq 1 \end{cases}$$

□

Next we formally define what a *Pareto optimum* is. In short, a Pareto optimum is a configuration where no more fair exchanges can be performed, that is, exchanges improving some agents but without worsening any other agent. This concept considers all the basic agents *together* in the sense that we can *ignore* the actual hierarchical structure of markets.

In the following we are interested in determining whether the final distributions reached by a hierarchical system are as good as those given by Pareto optimums. This will allow us to assess the economic performance of hierarchical systems. In our first approach the effect of transaction and shipping costs is not considered. As a result, it is irrelevant to know *who* gives each item to *whom*, provided that the total amount of given items equals the total amount of received items. Thus, in order to consider possible exchanges of resources, the only requirements are that the exchange neither creates nor destroys resources in the global market and that no agent has *debts* after the exchange. Later we will introduce a complementary concept to take into account additional costs (we will call it $\alpha-$*Pareto optimum*).

Definition 11 Let $S = \{(u_1, \overline{x_1}), \ldots, (u_m, \overline{x_m})\}$ be a set of pairs. We say that S is a *Pareto optimum* if there do not exist tuples of resources $\overline{y_1}, \ldots, \overline{y_m}$ such that all of the following conditions hold:

- $\sum_i \overline{y_i} = \overline{0}$,
- for any $1 \leq i \leq m$ we have $\overline{x_i} + \overline{y_i} \geq \overline{0}$,
- for any $1 \leq i \leq m$ we have $u_i(\overline{x_i} + \overline{y_i}) \geq u_i(\overline{x_i})$, and
- there exists $1 \leq j \leq m$ such that $u_j(\overline{x_j} + \overline{y_j}) > u_j(\overline{x_j})$.

□

In order to define a new notion of Pareto optimality where additional costs are considered, we need a formalism to properly abstract these costs. We provide a function to denote the additional costs associated with an exchange of resources. In this function, the first parameter is an index whose purpose is to identify the agent that must pay the additional costs. The subsequent parameters contain the tuples of resources that this agent receives from the other agents. As we did before, we assume again that there are p different kinds of resources.

Definition 12 Let $S = \{(u_1, \overline{x_1}), \ldots, (u_m, \overline{x_m})\}$ be a set of pairs. A *function of additional costs* for S is a function $\alpha : \mathbb{N} \times \underbrace{\mathbb{R}_+^p \times \ldots \times \mathbb{R}_+^p}_{m} \longrightarrow \mathbb{R}_+$. □

Additional costs will affect the assessment of whether there exists a good exchange. In fact, an exchange will be *good* only if it is so *in spite of* these costs (i.e. after these costs are added). In contrast with the previous notion of Pareto optimum, we will have to identify who gives each item to whom since additional costs will depend on it. In order to do so, exchanges will be formalized by using exchange matrixes.

Definition 13 Let $S = \{(u_1, \overline{x_1}), \ldots, (u_m, \overline{x_m})\}$ be a set of pairs and α be a function of additional costs for S. We say that S is an $\alpha-$*Pareto optimum* if there do not exist $\mathcal{E} \in (\mathbb{R}_+^p)^{m \times m}$, with $\overline{y_i} = \sum_j \mathcal{E}_{ji} - \sum_j \mathcal{E}_{ij}$ for $1 \leq i \leq m$, such that all of the following conditions hold:

- $\sum_i \overline{y_i} = \overline{0}$,

- for any $1 \leq i \leq m$ we have $\overline{x_i} + \overline{y_i} - (0, \ldots, 0, \alpha(i, \mathcal{E}_{1i}, \ldots, \mathcal{E}_{mi})) \geq \overline{0}$ and $u_i(\overline{x_i} + \overline{y_i} - (0, \ldots, 0, \alpha(i, \mathcal{E}_{1i}, \ldots, \mathcal{E}_{mi}))) \geq u_i(\overline{x_i})$, and
- there exists $1 \leq j \leq m$ such that $u_j(\overline{x_j} + \overline{y_j} - (0, \ldots, 0, \alpha(j, \mathcal{E}_{1j}, \ldots, \mathcal{E}_{mj}))) > u_j(\overline{x_j})$.

□

Next we provide a mechanism to compute a function of additional costs which matches the additional costs incurred by the agents of a given hierarchical market. This construction will allow us to apply the notion of $\alpha-$Pareto optimum to a hierarchical market in such a way that the additional costs of the market are properly considered. In order to compute the function of additional costs we must take into account both shipping and transaction costs. For each possible exchange in which the agent i receives from the other agents the tuples of resources $\overline{y_1}, \ldots, \overline{y_m}$, the following recursive definition computes, for each level, the additional costs associated with those resources that cannot be exchanged in a *lower* level of the hierarchy (and so they have to be exchanged in the current level). Then, we remove those resources and we perform a recursive call for the lower level. Recursive calls stop when all the subagents of the market are actually basic agents, which is ensured in the following definition by the condition $n = m$.

Definition 14 Let $M = \texttt{uncomp}((M_1, \ldots, M_n), sh, pr)$ be an uncompleted market having as transaction function tr. Let us consider the set of basic agents $\texttt{Basic}(M) = \{(u_1, \overline{x_1}), \ldots, (u_m, \overline{x_m})\}$. The *function of additional costs associated with* M is a function α_M such that for any $1 \leq i \leq n$ and tuples of resources $\overline{y_1}, \ldots, \overline{y_m}$ we have

$$\alpha_M(i, \overline{y_1}, \ldots, \overline{y_m}) = \begin{cases} \sum_k (sh(\overline{y_k}) + tr(\overline{y_k})) & \texttt{if } n = m \\ \alpha_{M'}(i, \overline{v_1}, \ldots, \overline{v_r}) + \sum_k (sh(\overline{z_k}) + tr(\overline{z_k})) & \text{otherwise} \end{cases}$$

where M' is the (unique) agent belonging to the set $\{M_1, \ldots, M_n\}$ such that $(u_i, \overline{x_i}) \in \texttt{Basic}(M')$. Since $\texttt{Basic}(M') \subseteq \texttt{Basic}(M)$, we have considered $\texttt{Basic}(M') = \{(u'_1, \overline{v_1}), \ldots, (u'_r, \overline{v_r})\} \subseteq \{(u_1, \overline{x_1}), \ldots, (u_m, \overline{x_m})\} = \texttt{Basic}(M)$. Besides, for any $1 \leq k \leq m$ we have that $\overline{z_k}$ is given by

$$\overline{z_k} = \begin{cases} \overline{y_k} & \texttt{if } (u_k, \overline{x_k}) \notin \texttt{Basic}(M') \\ \overline{0} & \text{otherwise} \end{cases}$$

□

The following result states that the utility function of each market can be defined in terms of the utilities of the basic agents located in the leaves of the hierarchical tree. This is an alternative vision since this function was formerly defined in terms of the utilities of each of the (higher order) subagents directly linked to the corresponding market. More precisely, the result shows that the utility of any market is the addition of the utilities of its basic agents, provided that some additional constraints, regarding the minimal utility in each lower level, hold. However, these constraints can be rewritten in terms of the additions of utilities for some subsets of basic agents. The proof of Theorem 1 uses three auxiliary results (Lemmas 3, 4, and 5) that are given in the appendix of the paper.

Theorem 1 *(Utility Transformation Theorem)* Let us consider a completed market $A = ([A'_1, \ldots, A'_n], u, \overline{x}, sh, pr)$ such that for any $1 \leq j \leq n$ we have $A'_j = (L'_j, u'_j, \overline{x'_j}, sh'_j, pr'_j)$. Let $S = \texttt{Basic}(A) = \{(u_1, \overline{x_1}), \ldots, (u_m, \overline{x_m})\}$, and for any $1 \leq j \leq n$ let $S'_j = \texttt{Basic}(A'_j)$. Then, there exists $\mathcal{H} \in \mathcal{P}(\mathcal{P}(\mathbb{N}) \times \mathbb{R}_+)$ such that

$$u(\overline{z}) = \max \left\{ \left(\sum_{(u_j,\overline{x_j}) \in S} u_j(\overline{z''_j}), \sum_{(u_j,\overline{x_j}) \in S'_1} \overline{z''_j}, \ldots, \sum_{(u_j,\overline{x_j}) \in S'_n} \overline{z''_j} \right) \middle| \begin{array}{c} \sum\limits_{(u_j,\overline{x_j}) \in S} \overline{z''_j} = \overline{z} \\ \wedge \\ \forall\, (H, v) \in \mathcal{H} : \\ \sum\limits_{j \in H} u_j(\overline{z''_j}) \geq v \end{array} \right\} \quad (1)$$

Proof. By Definition 8 we have

$$u(\overline{z}) = \max \left\{ \left(\sum_j u'_j(\overline{z'_j}), \overline{z'_1}, \ldots, \overline{z'_n} \right) \middle| \begin{array}{c} \sum_j \overline{z'_j} = \overline{z} \\ \wedge \\ \forall\, 1 \leq j \leq n : u'_j(\overline{z'_j}) \geq w_j \end{array} \right\} \quad (2)$$

for some constant values $w_1, \ldots, w_n$. We will prove, by structural induction over A, that the expressions (1) and (2) coincide.

In the anchor case we have $n = 0$, that is, a market $A = ([], u, \overline{x}, sh, pr)$. Then, $S = \{(u, \bar{x})\}$, since A is already a basic agent. Thus, the tuple returned by $u(\bar{z})$ in expression (2) contains a single component, which gives the utility of A for the basket $\bar{z}$. Let us check whether this value coincides with that returned by the expression (1). This expression returns the utility of A for some tuple $\bar{z}''_1$, because the only agent ranged in the addition is A. The constraint $\sum_{(u_j,\overline{x_j}) \in S} \overline{z''_j} = \overline{z}$ imposed by the wording of the result implies $\overline{z''_1} = \overline{z}$. Thus, by setting $\mathcal{H} = \emptyset$ we have that the utility returned by $u(\bar{z})$ in each of the expressions coincide, so the result holds.

Let us consider now the inductive case, that is, the set $\{A'_1, \ldots, A'_n\}$ is non-empty. Let $Q = \{i \mid 1 \leq i \leq n \ \wedge \ L'_i \neq [\,]\}$. For any $i \in Q$ let us consider the set $S'_i = \{(u_{r_1}, \overline{x_{r_1}}), \ldots, (u_{r_{m_i}}, \overline{x_{r_{m_i}}})\}$ and the list $L'_i = [A^i_1, \ldots, A^i_{k_i}]$, where for any $1 \leq l \leq k_i$ we have $A^i_l = (L^i_l, u^i_l, \overline{x^i_l}, sh^i_l, pr^i_l)$. Finally, for any $1 \leq j \leq k_i$ let $S^i_j = \mathtt{Basic}(A^i_j)$. Obviously, if $i \notin Q$ then u_i represents the utility of the basic agent A'_i. Otherwise, by induction hypothesis, we have that for any $i \in Q$

$$u'_i(\overline{z}) = \max \left\{ \left(\sum_{(u_j,\overline{x_j}) \in S'_i} u_j(\overline{z''_j}), \sum_{(u_l,\overline{x_l}) \in S^i_1} \overline{z''_l}, \ldots, \sum_{(u_l,\overline{x_l}) \in S^i_{k_i}} \overline{z''_l} \right) \left| \begin{array}{c} \sum_{(u_j,\overline{x_j}) \in S'_i} \overline{z''_j} = \overline{z} \\ \wedge \\ \forall\ (H,v) \in \mathcal{H}'_i : \\ \sum_{j \in H} u_j(\overline{z''_j}) \geq v \end{array} \right. \right\}$$

for some sets $\mathcal{H}'_i \in \mathcal{P}(\mathcal{P}(\mathbb{N}) \times \mathbb{R}_+)$. Let us show that by substituting in expression (2) all the terms u'_i, with $i \in Q$, by the previous expression we get the expression (1). Let us remark that for any functions f_{ij} and boolean conditions C_i and C'_{ij} we have that the following auxiliary property holds:

$$\max\{\textstyle\sum_i \max(\{\sum_j f_{ij}(\overline{x_{ij}}) \mid \sum_j \overline{x_{ij}} = \overline{x_i} \ \wedge \ \forall_j C'_{ij}\}) \mid \sum_i \overline{x_i} = \overline{x} \ \wedge \ \forall_i C_i\}$$
$$\|$$
$$\max\{\textstyle\sum_{ij} f_{ij}(\overline{x_{ij}}) \mid \sum_{ij} \overline{x_{ij}} = \overline{x} \ \wedge \ \forall_i C_i \ \wedge \ \forall_{ij} C'_{ij}\}$$

We got the previous result by successively applying Lemma 3, to exchange the positions of the first addition and the second maximization, and Lemma 4, to eliminate the second maximization. Let us remark that the quantifier $\exists\ y'$ required in Lemma 4 to embrace the conditions introduced by the maximization operators is superfluous since the equality $\sum_{ij} \overline{x_{ij}} = \overline{x}$ holds iff $\exists\ (\overline{x_1}, \ldots, \overline{x_n}) : \sum_i \overline{x_i} = \overline{x} \ \wedge \ \forall\ i : \sum_j \overline{x_{ij}} = \overline{x_i}$. This is so because such expression allows each $\overline{x_i}$ to take a unique value. Though not given explicitly, we assume that any occurrence of a term $\overline{x_i}$ in C_i is substituted by $\sum_j \overline{x_{ij}}$ in the right hand side of the equality. The reason is that the maximization carried out by the outermost maximization operator in the left hand side is performed over $(\overline{x_1}, \ldots, \overline{x_n})$, while the right hand side operator maximizes over $(\overline{x_{11}}, \ldots, \overline{x_{nm}})$. So, in the formula obtained after substituting all terms u'_i in expression (2) we can both transfer the boolean conditions to the upper level and make the internal maximization operator to collapse with the external one.

Now we will relate the previous expression and our formulae. The conditions C_i and C'_{ij} match the constraints over the minimal addition of utilities of subagents for each specific agent. Thus, the term $\forall_i C_i$ is in fact instantiated as $\forall\ 1 \leq j \leq n : u'_j(\overline{z'_j}) \geq w_j$ while the term $\forall_{ij} C'_{ij}$ is instantiated as the quantification $\forall\ 1 \leq i \leq n, (H,v) \in \mathcal{H}'_i : \sum_{j \in H} u_j(\overline{z''_j}) \geq v$. Let us remark that

$\forall_{ij} C'_{ij}$ already follows the pattern $\forall\ (H, v) \in \mathcal{H} : \sum_{j \in H} u_j(\overline{z''_j}) \geq v$ because we can create a single set $\mathcal{H}$ by performing the union of all the sets $\mathcal{H}'_i$. Let us consider the term $\forall_i C_i$. After applying induction hypothesis to substitute each u'_i by the expression we gave above, we can remove the new maximization operators. This is so because for any functions f_{ij} and boolean conditions $D', D'_1, \ldots, D'_n$, with $D' \Rightarrow \forall_i D'_i$, the following auxiliary property holds:

$$\max\{\textstyle\sum_{ij} f_{ij}(\overline{x_{ij}}) \mid D' \wedge \forall_i\ \max(\{\sum_j f_{ij}(\overline{x'_{ij}}) \mid \sum_j \overline{x'_{ij}} = \sum_j \overline{x_{ij}} \wedge D'_i\}) \geq w_i\}$$
$$\|$$
$$\max\{\textstyle\sum_{ij} f_{ij}(\overline{x_{ij}}) \mid D' \ \wedge\ \forall_i \sum_j f_{ij}(\overline{x_{ij}}) \geq w_i\}$$

We obtain this equality since, by applying Lemma 5, we have that the first maximization operator already maximizes the term $\sum_j f_{ij}(\overline{x'_{ij}})$ for each i. Let us note that the quantifier $\exists\ z'$ required in Lemma 5 to embrace the conditions introduced by the innermost maximization can be easily removed. In fact, we will show that the condition

$$D' \wedge \forall_i\ \exists(\overline{x'_{i1}}, \ldots, \overline{x'_{im}}) : (\sum_j f_{ij}(\overline{x'_{ij}}) \geq w_i \wedge \sum_j \overline{x'_{ij}} = \sum_j \overline{x_{ij}} \wedge D'_i)$$

can be indeed replaced by the condition $D' \ \wedge\ \forall_i \sum_j f_{ij}(\overline{x_{ij}}) \geq w_i$.

Let us suppose that $(\overline{x_{11}}, \ldots, \overline{x_{nm}})$ returns the maximal value of $\sum_{ij} f_{ij}(\overline{x_{ij}})$ such that $D' \wedge \forall_i\ \exists(\overline{x'_{i1}}, \ldots, \overline{x'_{im}}) : (\sum_j f_{ij}(\overline{x'_{ij}}) \geq w_i \wedge \sum_j \overline{x'_{ij}} = \sum_j \overline{x_{ij}} \wedge D'_i)$ holds. Since both $\sum_{ij} f_{ij}(\overline{x_{ij}}) = \sum_i \sum_j f_{ij}(\overline{x_{ij}})$ and $\forall_i\ \sum_j f_{ij}(\overline{x'_{ij}}) \geq w_i$ hold, we have $\forall_i\ \sum_j f_{ij}(\overline{x_{ij}}) \geq w_i$. Let us note that if $\exists_i\ \sum_j f_{ij}(\overline{x_{ij}}) < w_i$ then, by setting $\overline{x_{i1}}, \ldots, \overline{x_{im}}$ equal to $\overline{x'_{i1}}, \ldots, \overline{x'_{im}}$, the value of $\sum_{ij} f_{ij}(\overline{x_{ij}})$ would be higher than the maximal, which is a contradiction. Hence, $(\overline{x_{11}}, \ldots, \overline{x_{nm}})$ fulfills the condition $D' \ \wedge\ \forall_i \sum_j f_{ij}(\overline{x_{ij}}) \geq w_i$. The reverse implication is trivially obtained by setting $(\overline{x'_{11}}, \ldots, \overline{x'_{nm}}) = (\overline{x_{11}}, \ldots, \overline{x_{nm}})$ and taking into account that since $D' \Rightarrow \forall_i D'_i$ the tuple $(\overline{x'_{11}}, \ldots, \overline{x'_{nm}})$ fulfills $\forall_i D'_i$.

In the context of our formula, we have $D' \equiv \sum_{ij} \overline{x_{ij}} = \overline{x} \ \wedge\ \forall_{ij} C'_{ij}$ and $D'_i \equiv \forall_j C'_{ij}$. The additional conditions $D'_1, \ldots, D'_n$ appearing in the term $\forall_i C_i$ are redundant since they are the same as those imposed by D'. Thus, $\forall_i C_i$ can be substituted by $\forall_i \sum_j f_{ij}(\overline{x_{ij}}) \geq w_i$ or, by using the notation of our formula, by the expression $\forall_i \sum_{(u_j, \overline{x_j}) \in \mathtt{Basic}(A'_i)} u_j(\overline{z''_j}) \geq w_i$. It easily follows that the terms representing $\forall_i C_i$ and $\forall_{ij} C'_{ij}$ can be joint together to create one single set $\mathcal{H}$ including both conditions. Finally, let us remark that the elements that do not represent utility in the tuple returned by the function u (i.e. those specifying how the resources should be delivered among subagents) are exactly the inputs for each function u'_i. Since the expression defining u'_i, by induction hypothesis, states $\sum_{(u_j, \overline{x_j}) \in S'_i} \overline{z''_j} = \overline{z}$, where $\overline{z}$ is that input, we have that these elements match those given in the wording of the result. □

The previous result shows that, under certain conditions, the utility of a market is just the addition of the utilities of the subagents. The following lemma (whose proof is given in the appendix) states that if an exchange is either neutral or positive for all the basic agents of a market then it will also be so for all the subagents that are directly linked to the market.

Lemma 1 *(Propagation Lemma)* Let us consider a completed market $A = ([A'_1, \ldots, A'_n], u, \overline{x}, sh, pr)$, with $A'_i = (L'_i, u'_i, \overline{x'_i}, sh'_i, pr'_i)$, and let $\overline{y_1}, \ldots, \overline{y_m}$ be tuples of resources. Let $\texttt{Basic}(A) = \{(u_1, \overline{x_1}), \ldots, (u_m, \overline{x_m})\}$. We have

$$\forall\, 1 \leq i \leq m : u_i(\overline{x_i} + \overline{y_i}) \geq u_i(\overline{x_i})$$

$$\Downarrow$$

$$\forall\, 1 \leq i \leq n : u'_i\left(\overline{x'_i} + \sum_{(u_j,\overline{x_j}) \in \texttt{Basic}(A'_i)} \overline{y_j}\right) \geq u'_i(\overline{x'_i})$$

□

In the following result we go one step further. We show that if an exchange is neutral or positive for all the basic agents of a market, being strictly positive for some of them, then this exchange is neutral or positive for all of the subagents that are directly linked to the market, being strictly positive for some of these subagents. The proof is also given in the appendix of the paper.

Lemma 2 *(Improvement Propagation lemma)* Let us consider a completed market $A = ([A'_1, \ldots, A'_n], u, \overline{x}, sh, pr)$, with $A'_i = (L'_i, u'_i, \overline{x'_i}, sh'_i, pr'_i)$, and let $\overline{y_1}, \ldots, \overline{y_m}$ be tuples of resources. Let $\texttt{Basic}(A) = \{(u_1, \overline{x_1}), \ldots, (u_m, \overline{x_m})\}$. Finally, let us consider $R \subseteq \{1, \ldots, m\}$. Then,

$$\forall\, 1 \leq i \leq m : u_i(\overline{x_i} + \overline{y_i}) \geq u_i(\overline{x_i}) \ \wedge\ \forall\, i \in R : u_i(\overline{x_i} + \overline{y_i}) > u_i(\overline{x_i})$$

$$\Downarrow$$

$$\forall\, 1 \leq i \leq n : \left(\begin{array}{c} u'_i\left(\overline{x'_i} + \displaystyle\sum_{(u_j,\overline{x_j}) \in \texttt{Basic}(A'_i)} \overline{y_j}\right) \geq u'_i(\overline{x'_i}) \\ \wedge \\ \left(\begin{array}{c} \{j \mid (u_j, \overline{x_j}) \in \texttt{Basic}(A'_i)\} \cap R \neq \emptyset \\ \Downarrow \\ u'_i\left(\overline{x'_i} + \displaystyle\sum_{(u_j,\overline{x_j}) \in \texttt{Basic}(A'_i)} \overline{y_j}\right) > u'_i(\overline{x'_i}) \end{array} \right) \end{array} \right)$$

□

If what is good for the basic agents is also good for the subagents of a market, then we can prove that the final configuration of a hierarchical system must be a Pareto optimum, provided that the additional costs are set to 0. This result, formally presented in the forthcoming Theorem 2, is important for our framework because it implies that the economic efficiency in a hierarchical system matches that of a single centralized market. Afterwards we will provide a similar result for the case when shipping and transaction costs are not necessarily nil.

Theorem 2 *(Optimality Theorem)* Let M, M' be markets and A be an agent such that $M \rightsquigarrow^* M' \hookrightarrow ms(\sigma(A))$. If transaction and shipping costs are set to zero then the distribution of resources provided by $\texttt{Basic}(A)$ represents a Pareto optimum.

Proof. We prove the result by structural induction over A. As anchor case we have $A = ([\,], u, \overline{x}, sh, pr)$. In this case the result trivially holds since any market conformed by a single agent is a Pareto optimum.

As inductive case we consider a market $A = ([A_1, \ldots, A_n], u, \overline{x}, sh, pr)$, with $A_i = (L_i, u_i, \overline{x_i}, sh_i, pr_i)$. By induction hypothesis we assume that each of the sets $\texttt{Basic}(A_i)$ represents a Pareto optimum. Let us suppose that the set $\texttt{Basic}(A) = \{(u_1, \overline{x_1}), \ldots, (u_m, \overline{x_m})\}$ is not a Pareto optimum. We will show that we get a contradiction.

If the distribution is not a Pareto optimum then there exists an exchange of resources $\overline{y_1}, \ldots, \overline{y_m}$ in $\texttt{Basic}(A)$ such that the following conditions hold:

- For any $1 \leq i \leq m$ we have $\overline{x_i} + \overline{y_i} \geq \overline{0}$ and $u_i(\overline{x_i} + \overline{y_i}) \geq u_i(\overline{x_i})$,
- $\sum_{1 \leq i \leq m} \overline{y_i} = \overline{0}$, and
- there exists $1 \leq j \leq m$ such that $u_j(\overline{x_j} + \overline{y_j}) > u_j(\overline{x_j})$.

There are two possibilities: Either all of the (basic) agents involved in that exchange are represented in A by the same agent A_k (i.e. for any $1 \leq l \leq m$ such that $\overline{y_l} \neq \overline{0}$ we have $(u_l, \overline{x_l}) \in \texttt{Basic}(A_k)$) or not. In the former case we get a contradiction since, by induction hypothesis, we assume that $\texttt{Basic}(A_k)$ represents a Pareto optimum. Let us consider the latter case. By Lemma 2, the modification of the resources of each subagent A_i by $\sum_{(u_j, \overline{x_j}) \in \texttt{Basic}(A_i)} \overline{y_j}$ yields a configuration where the utility of some subagents improves and no utility worsens. However, this implies that $M' \not\rightsquigarrow$ does not hold because there are more valid exchanges to perform. Thus, a transition such as $M' \hookrightarrow ms(\sigma(A))$ is not possible because it requires M' to be a completed market while it is not. Therefore, in both cases we have a contradiction by assuming that $\texttt{Basic}(A)$ is not a Pareto optimum. □

Regarding the general situation, that is transaction and shipping costs are greater than zero, the classical Pareto optimum concept does not apply and we have to use the notion of $\alpha-$Pareto optimum (see Definition 13). For instance, a desirable exchange in a free or low transaction fee environment could not be a desirable exchange in an expensive one. Then, the set of optimums is different because of the effect of the fees. The next result states that a completed market will be an $\alpha-$Pareto optimum for the set of basic agents. This means that a hierarchical market presents the same economic efficiency as a single market where the additional costs are those of the hierarchical market.

Theorem 3 *(Theorem of $\alpha-$optimality)* Let M, M' be markets and A be an agent such that $M \rightsquigarrow^* M' \hookrightarrow ms(\sigma(A))$. Let α be the function of additional costs associated to M. We have that the distribution of resources provided by $\texttt{Basic}(A)$ represents an $\alpha-$Pareto optimum.

Proof. The proof, again by *reductio ad absurdum*, follows the same pattern as that of Theorem 2. Let us remark that Theorem 1 and Lemmas 1 and 2 are still applicable because they are concerned only with the utility functions of agents. The main difference with respect to the proof of Theorem 2 is that, in this case, the assumption considers that there exists a possible good exchange *in spite of* the additional costs. However, if there were a good exchange, by Lemma 2, the utility of agents would improve after the exchange, so M' is not completed and we get a contradiction. □

6 Conclusions

The main objective of this paper has been to introduce a formal framework to specify and analyze e-barter systems. These systems allow customers to exchange products through the use of electronic agents representing their preferences. In order to allow the precise specification of e-barter systems we have introduced a formal syntax, based on classical process algebras, to define them. We have also presented a semantics for syntactic terms based on inference rules. One of the main advantages of using a formal framework is that we may study properties of these systems.

In this paper we have concentrated ourselves in studying general properties that any e-barter system fulfills. In particular, we have proved that a hierarchical structure may reach the same optimums as a centralized market. We have also shown that this kind of hierarchical structure presents additional advantages. Specifically, it allows to diminish shipping costs and it reduces the computational power (in terms of exchanged messages) needed to implement the market(s).

Acknowledgements

We would like to thank the anonymous reviewers of this paper for the careful reading and the useful suggestions. We would also like to thank Natalia López for her collaboration in the first stages of this line of research.

References

[1] F. Bacchus and A. Grove. Graphical models for preference and utility. In *Uncertainty in Artificial Intelligence (UAI-95)*, pages 3–10. Morgan Kaufmann, 1995.

[2] J.C.M. Baeten and W.P. Weijland. *Process Algebra.* Cambridge Tracts in Computer Science 18. Cambridge University Press, 1990.

[3] J.A. Bergstra, A. Ponse, and S.A. Smolka, editors. *Handbook of Process Algebra.* North Holland, 2001.

[4] A. Cavalli and S. Maag. Automated test scenarios generation for an e-barter system. In *19th ACM Symposium on Applied Computing, SAC'04*, pages 795–799. ACM Press, 2004.

[5] J.Q. Cheng and M.P. Wellman. The WALRAS algorithm: A convergent distributed implementation of general equilibrium outcomes. *Computational Economics*, 12:1–24, 1998.

[6] M. Dastani, N. Jacobs, C.M. Jonker, and J. Treur. Modelling user preferences and mediating agents in electronic commerce. In *Agent Mediated Electronic Commerce, LNCS 1991*, pages 163–193. Springer, 2001.

[7] T. Eymann. Markets without makers - a framework for decentralized economic coordination in multiagent systems. In *WELCOM 2001, LNCS 2232*, pages 63–74. Springer, 2001.

[8] J. F. Groote. Transition system specifications with negative premises. *Theoretical Computer Science*, 118:263–299, 1993.

[9] R. Guttman, A. Moukas, and P. Maes. Agent-mediated electronic commerce: A survey. *The Knowledge Engineering Review*, 13(2):147–159, 1998.

[10] C.A.R. Hoare. *Communicating Sequential Processes.* Prentice Hall, 1985.

[11] J. Keppens and Q. Shen. A calculus of partially ordered preferences for compositional modelling and configuration. In *AAAI Workshop on Preferences in AI and CP: Symbolic Approaches*, pages 39–46. AAAI Press, 2002.

[12] S. Kraus. Negotiation and cooperation in multi-agent systems. *Artificial Intelligence*, 94(1-2):79–98, 1997.

[13] J.F. Kurose and R. Simha. A microeconomic approach to optimal resource allocation in distributed computer systems. *IEEE Transactions on Computers*, 38(5):705–717, 1989.

[14] J. Lang, L. van der Torre, and E. Weydert. Utilitarian desires. *Autonomous Agents and Multi-Agent Systems*, 5(3):329–363, 2002.

[15] A.R. Lomuscio, M. Wooldridge, and N.R. Jennings. A classification scheme for negotiation in electronic commerce. In *Agent Mediated Electronic Commerce, LNAI 1991*, pages 19–33. Springer, 2001.

[16] N. López, M. Núñez, I. Rodríguez, and F. Rubio. A formal framework for e-barter based on microeconomic theory and process algebras. In *Innovative Internet Computer Systems, LNCS 2346*, pages 217–228. Springer, 2002.

[17] N. López, M. Núñez, I. Rodríguez, and F. Rubio. A multi-agent system for e-barter including transaction and shipping costs. In *18th ACM Symposium on Applied Computing, SAC'03*, pages 587–594. ACM Press, 2003.

[18] A. Mas-Colell, M.D. Whinston, and J.R. Green. *Microeconomic Theory.* Oxford University Press, 1995.

[19] M. McGeachie and J. Doyle. Utility functions for ceteris paribus preferences. In *AAAI Workshop on Preferences in AI and CP: Symbolic Approaches*, pages 33–38. AAAI Press, 2002.

[20] R. Milner. *Communication and Concurrency.* Prentice Hall, 1989.

[21] M. Núñez and I. Rodríguez. `PAMR`: A process algebra for the management of resources in concurrent systems. In *FORTE 2001*, pages 169–185. Kluwer Academic Publishers, 2001.

[22] M. Núñez and I. Rodríguez. Encoding `PAMR` into (timed) `EFSMs`. In *FORTE 2002, LNCS 2529*, pages 1–16. Springer, 2002.

[23] D.C. Parkes and L.H. Ungar. Iterative combinatorial auctions: Theory and practice. In *17th National Conference on Artificial Intelligence (AAAI-2000)*, pages 74–81. AAAI Press/The MIT Press, 2000.

[24] S. Parsons and M. Wooldridge. Game theory and decision theory in multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 5(3):243–254, 2002.

[25] L. Rasmusson and S. Janson. Agents, self-interest and electronic markets. *The Knowledge Engineering Review*, 14(2):143–150, 1999.

[26] T. Sandholm. Agents in electronic commerce: Component technologies for automated negotiation and coalition formation. In *CIA'98, LNCS 1435*, pages 113–134. Springer, 1998.

[27] W.C. Stirling, M.A. Goodrich, and D.J. Packard. Satisficing equilibria: A non-classical theory of games and decisions. *Autonomous Agents and Multi-Agent Systems*, 5(3):305–328, 2002.

[28] M. Tennenholtz. Game theory and artificial intelligence. In *Foundations and Applications of Multi-Agent Systems, LNAI 2403*, pages 49–58. Springer, 2002.

[29] H.R. Varian. Economic mechanism design for computerized agents. In *USENIX Workshop on Electronic Commerce*, 1995.

[30] M. Wooldridge and N.R. Jennings. Intelligent agents: Theory and practice. *The Knowledge Engineering Review*, 10(2):115–152, 1995.

Appendix: Auxiliary results used in Section 5

In this appendix we give the proofs of some results that are used in Section 5. First, we introduce a formal definition of the maximization operator. This notation is useful only to facilitate the formal formulation of the forthcoming lemmas. In the following definition we distinguish between the names of the variables (denoted by x and y) and their actual values (denoted by x' and y', respectively).

Definition 15 Let Σ be a signature, $\mathcal{T}$ be a set of data types, and $\alpha : \Sigma \longrightarrow T$ be a typing function. Let $x = (x_1, \ldots, x_n) \in \Sigma^n$ and $y = (y_1, \ldots, y_m) \in \Sigma^m$ be two tuples of symbols such that for any $1 \leq i \leq n$ we have $\alpha(x_i) = T_i \in \mathcal{T}$ and for any $1 \leq i \leq m$ we have $\alpha(y_i) = T'_i \in \mathcal{T}$. Let $x' \in T_1 \times \ldots \times T_n$, $f : T'_1 \times \ldots \times T'_m \longrightarrow \mathbb{R}$, and $L : T_1 \times \ldots \times T_n \times T'_1 \times \ldots \times T'_m \longrightarrow \texttt{Bool}$.

We define the *maximal* f such that L given x, x', and y, denoted by $\texttt{max}_y^{x=x'}(f, L)$, as the value of $f(y')$, with $y' \in T'_1 \times \ldots \times T'_m$, such that

- $L(x', y')$ holds and
- for any $y'' \in T'_1 \times \ldots \times T'_m$ we have that $L(x', y'')$ implies $f(y') \geq f(y'')$.

□

For instance, if $f(y) = 2 \cdot y$ and $L(x, y) = (y \leq 3 \cdot x)$ then $\texttt{max}_y^{x=10}(f, L) = 60$. This is so because the value y' that maximizes $f(y')$ while fulfilling $L(10, y')$ is 30. Let us note that it is straightforward to transform expressions using the new operator into expressions without any occurrence of it. For example, $\texttt{max}_y^{x=10}(f, L)$ can be formulated simply as $\max\{2 \cdot y \mid y \leq 30\}$.

Next we present the three lemmas that were used in the proof of Theorem 1. The first result shows that, under certain conditions, we can exchange the positions of an addition and a maximization operator.

Lemma 3 Let $g : T^k \to \mathbb{R}$, $f_1, \ldots, f_k : T \to \mathbb{R}$, $L_1, \ldots, L_k : T^2 \to \texttt{Bool}$, and $L' : T^{2k} \to \texttt{Bool}$ be functions such that for any $x'_1, \ldots, x'_k, y'_1, \ldots, y'_k \in T$

we have $L'(x'_1,\ldots,x'_k,y'_1,\ldots,y'_k) = L_1(x'_1,y'_1) \wedge \ldots \wedge L_k(x'_k,y'_k)$ and $g(y'_1,\ldots,y'_k) = \sum_i f_i(y'_i)$. Then,

$$\sum_{i=1}^{k}(\mathtt{max}_{y_i}^{x_i=x'_i}(f_i,L_i)) = \mathtt{max}_{(y_1,\ldots,y_k)}^{(x_1,\ldots,x_k)=(x'_1,\ldots,x'_k)}(g,L')$$

Proof. Let us suppose that for some values $b_1,\ldots,b_k$ we have that $L_i(x'_i,b_i)$ holds and that $\mathtt{max}_{y_i}^{x_i=x'_i}(f_i,L_i) = f_i(b_i)$, for any $1 \leq i \leq k$. First, we easily get $\sum_{i=1}^{k}(\mathtt{max}_{y_i}^{x_i=x'_i}(f_i,L_i)) = \sum_{i=1}^{k} f_i(b_i)$. We only need to prove that the expression $\mathtt{max}_{(y_1,\ldots,y_k)}^{(x_1,\ldots,x_k)=(x'_1,\ldots,x'_k)}(g,L')$ is also equal to $\sum_{i=1}^{k} f_i(b_i)$. Let us suppose that this is not the case. Then, by taking into account the way g and L' are defined, we have that there exists a tuple $(b'_1,\ldots,b'_k) \neq (b_1,\ldots,b_k)$ such that $\mathtt{max}_{(y_1,\ldots,y_k)}^{(x,\ldots,x_k)=(x'_1,\ldots,x'_k)}(g,L') = \sum_{i=1}^{k} f_i(b'_i)$, with $L'(x'_1,\ldots,x'_k,b'_1,\ldots,b'_k) = \mathtt{True}$. Thus, we have that for any $1 \leq i \leq k$, $L_i(x'_i,b'_i)$ holds. Besides, since L' is the conjunction of $L_1,\ldots,L_k$, we have that $L'(x'_1,\ldots,x'_k,b_1,\ldots,b_k)$ holds. So, $(b_1,\ldots,b_k)$ is one of the cases to be considered to perform the maximization $\mathtt{max}_{(y_1,\ldots,y_k)}^{(x_1,\ldots,x_k)=(x'_1,\ldots,x'_k)}(g,L')$. However, if $\sum_{i=1}^{k} f_i(b'_i)$ is the maximal value obtained by considering all the possible cases then $\sum_{i=1}^{k} f_i(b'_i) > \sum_{i=1}^{k} f_i(b_i)$. Hence, it must exist $1 \leq r \leq k$ such that $f_r(b'_r) > f_r(b_r)$ holds. Nevertheless, we have that $f_i(b_i)$ is the maximal value with $L_i(x'_i,b_i) = \mathtt{True}$. Therefore, $\sum_{i=1}^{k} f_i(b'_i) > \sum_{i=1}^{k} f_i(b_i)$ holds only if for some index $1 \leq w \leq k$ we have that $L_w(x'_w,b'_w)$ does not hold. Thus, we conclude that $L'(x'_1,\ldots,x'_k,b'_1,\ldots,b'_k)$ is also false since the predicate L' is the conjunction of the predicates $L_1,\ldots,L_k$. So, we get a contradiction. □

Next we show some conditions so that the maximal of the maximal can be expressed in terms of a single maximization operator.

Lemma 4 Let $\beta, f : T \to \mathbb{R}$ and $L', L_1, L_2 : T^2 \to \mathtt{Bool}$ be functions such that for any $q, x', y', v' \in T$ we have $L'(x',v') = \exists\, y' : L_1(y',v') \wedge L_2(x',y')$ and $\beta(q) = \mathtt{max}_v^{z=q}(f,L_1)$. Then,

$$\mathtt{max}_y^{x=x'}(\beta,L_2) = \mathtt{max}_v^{x=x'}(f,L')$$

Proof. Let us suppose that $\mathtt{max}_v^{x=x'}(f,L') = a$, that is, a is the highest value of $f(v')$ such that there exists y' fulfilling $L_1(y',v')$ and $L_2(x',y')$. We will show that such a condition matches the condition over which $\mathtt{max}_y^{x=x'}(\beta,L_2)$ maximizes. This fact would make both expressions to return the same value.

Let us suppose that $\mathtt{max}_y^{x=x'}(\beta,L_2) = b$, that is, b is the maximal value of $\mathtt{max}_v^{z=y''}(f,L_1)$ such that $L_2(x',y'')$ holds. Then, there also exists v' such that $L_1(y'',v')$ holds and $f(v')$ is the maximal value such that $L_1(y'',v')$ holds. By construction we have $f(v') = b$. Looking at the requirements over y'' we deduce that y'' is forced to fulfill both $L_1(y'',v')$ and $L_2(x',y'')$. Besides, the maximization returns the maximal value of $f(v')$ for which v' fulfills such a condition. Since this is exactly the constraint that L' requires in $\mathtt{max}_v^{x=x'}(f, L')$ (renaming y' as y'') we have that both expressions return the same value. $\square$

In the following result we proof that, under some conditions, a maximization that contains a certain requirement expressed in terms of a certain maximal value can also be expressed without using the maximization operator in the requirement.

Lemma 5 Let $\gamma, L, L_1, L_2 : T^2 \to \mathtt{Bool}$ and $f, g : T \to \mathbb{R}$ be functions such that for any $x', y', z' \in T$ we have $\gamma(x', y') = L_2(x', y') \wedge (\mathtt{max}_z^{v=y'}(g, L_1) \geq K)$ and $L(x', y') = \exists\, z' : ((g(z') \geq K) \wedge L_1(y', z')) \wedge L_2(x', y')$. Then,

$$\mathtt{max}_y^{x=x'}(f, \gamma) = \mathtt{max}_y^{x=x'}(f, L)$$

Proof. We have that $\mathtt{max}_y^{x=x'}(f, \gamma)$ returns the maximal value $f(y')$ such that $\gamma(x', y')$ holds. In addition, $\gamma(x', y')$ implies that $L_2(x', y')$ holds and that we have $\mathtt{max}_z^{v=y'}(g, L_1) \geq K$. Besides, if $\mathtt{max}_z^{v=y'}(g, L_1) \geq K$ then the maximal $g(z')$ such that $L_1(y', z')$ holds fulfills $g(z') \geq K$. That is, the values y' of y considered in the maximization $\mathtt{max}_y^{x=x'}(f, \gamma)$ are those for which $L_2(x', y')$ holds and the maximal z' such that $L_1(y', z')$ holds also fulfills $g(z') \geq K$. Let us note that this is equivalent to search among those values y' such that $L_2(x', y')$ holds and for some z' we have that $L_1(y', z')$ holds, with $g(z') \geq K$. This is so because if these conditions hold for the value z' that returns the maximum, then they will also hold for some z' (in particular, the same). Besides, if these conditions hold for some z', then they will also hold for the value returning the maximum such that $L_1(y', z')$ holds. This is so because the condition $g(z') \geq K$ cannot become false for some z' that returns the highest value. Since these requirements match those imposed by the function L, we deduce that the value of $\mathtt{max}_y^{x=x'}(f, \gamma)$ must be equal to that of $\mathtt{max}_{(y,z)}^{x=x'}(f, L)$. $\square$

Finally, we present the proofs of Lemmas 1 and 2.

Lemma 1. *(Propagation Lemma)* Let us consider a completed market $A = ([A'_1, \ldots, A'_n], u, \overline{x}, sh, pr)$, with $A'_i = (L'_i, u'_i, \overline{x'_i}, sh'_i, pr'_i)$, and let $\overline{y_1}, \ldots, \overline{y_m}$ be tuples of resources. Let $\mathtt{Basic}(A) = \{(u_1, \overline{x_1}), \ldots, (u_m, \overline{x_m})\}$. We have

$$\forall\, 1 \leq i \leq m : u_i(\overline{x_i} + \overline{y_i}) \geq u_i(\overline{x_i})$$

$$\Downarrow$$

$$\forall\, 1 \leq i \leq n : u'_i\left(\overline{x'_i} + \sum_{(u_j, \overline{x_j}) \in \mathtt{Basic}(A'_i)} \overline{y_j}\right) \geq u'_i(\overline{x'_i})$$

Proof: Let $P = \{A'_i | L'_i = [\,]\}$, that is, P is the set of basic agents directly linked to A. We consider the following two cases. First, if $P = \{A'_1, \ldots, A'_n\}$ then the previous result trivially holds since both inequalities coincide. Second, let us suppose that $P \neq \{A'_1, \ldots, A'_n\}$. Let us consider an agent $A'_i \notin P$. Given the fact that L'_i is not empty, let us suppose that $L'_i = [A^i_1, \ldots, A^i_{k_i}]$, with $A^i_l = (L^i_l, u^i_l, \overline{x^i_l}, sh^i_l, pr^i_l)$. Let $S'_i = \texttt{Basic}(A'_i)$, with $S'_i = ((u_{r_1}, \overline{x_{r_1}}), \ldots, (u_{r_{m_i}}, \overline{x_{r_{m_i}}}))$, and for any $1 \leq j \leq k_i$ let $S^i_j = \texttt{Basic}(A^i_j)$. By Definition 8 we have

$$u'_i(\overline{z}) = \max\{(\textstyle\sum_j u^i_j(\overline{z'_j}), \overline{z'_1}, \ldots, \overline{z'_{k_i}}) \mid \textstyle\sum_j \overline{z'_j} = \overline{z} \ \wedge \ \forall\, 1 \leq j \leq k_i : u^i_j(\overline{z'_j}) \geq w_j\}$$

for some constant values $w_1, \ldots, w_{k_i}$. Now, by applying Theorem 1, we also obtain

$$u'_i(\overline{z}) = \max \left\{ \left(\sum_{(u_j,\overline{x_j}) \in S'_i} u_j(\overline{z''_j}), \sum_{(u_l,\overline{x_l}) \in S^i_1} \overline{z''_l}, \ldots, \sum_{(u_l,\overline{x_l}) \in S^i_{k_i}} \overline{z''_l} \right) \middle| \begin{array}{c} \displaystyle\sum_{j \in \{r_1, \ldots, r_{m_i}\}} \overline{z''_j} = \overline{z} \\ \wedge \\ \forall\, (H, v) \in \mathcal{H}'_i : \\ \sum_{j \in H} u_j(\overline{z''_j}) \geq v \end{array} \right\}$$

for some $\mathcal{H}'_i \in \mathcal{P}(\mathcal{P}(\mathbb{N}) \times \mathbb{R}_+)$. Given the fact that $A'_1, \ldots, A'_n$ are all completed we have that the modifications of resources have been already propagated to the basic agents. Thus, $\sum_{(u_j,\overline{x_j}) \in \texttt{Basic}(A'_i)} u_j(\overline{x_j}) = u'_i(\overline{x'_i})$. If for any $1 \leq i \leq m$ we have $u_i(\overline{x_i} + \overline{y_i}) \geq u_i(\overline{x_i})$ then it is clear that we also have $\sum_{(u_j,\overline{x_j}) \in \texttt{Basic}(A'_i)} u_j(\overline{x_j} + \overline{y_j}) \geq u'_i(\overline{x'_i})$. Besides, we can also deduce that $u'_i(\overline{x'_i} + \sum_{(u_j,\overline{x_j}) \in \texttt{Basic}(A'_i)} \overline{y_j}) \geq \sum_{(u_j,\overline{x_j}) \in \texttt{Basic}(A'_i)} u_j(\overline{x_j} + \overline{y_j})$. This is so because the modification $\sum_{(u_j,\overline{x_j}) \in \texttt{Basic}(A'_i)} \overline{y_j}$ can be propagated among the basic agents of A'_i in such a way that the addition of their new utilities (that is, the new utility obtained by A'_i) matches at least that in the right hand side of the inequality. This can be done by modifying the resources of each basic agent $(u_i, \overline{x_i}) \in \texttt{Basic}(A'_i)$ according to $\overline{y_i}$. Let us remark that the constraints $\forall\, (H, v) \in \mathcal{H}'_i : \sum_{j \in H} u_j(\overline{z''_j}) \geq v$ hold in u'_i because if for any $1 \leq i \leq m$ we have $u_i(\overline{x_i} + \overline{y_i}) \geq u_i(\overline{x_i})$ then for any subset of basic agents $S' \subseteq \texttt{Basic}(A'_i)$ we may also conclude that $\sum_{(u_j,\overline{x_j}) \in S'} u_j(\overline{x_j} + \overline{y_j}) \geq \sum_{(u_j,\overline{x_j}) \in S'} u_j(\overline{x_j})$. Thus, the result holds. □

Lemma 2. *(Improvement Propagation lemma)* Let us consider a completed market $A = ([A'_1, \ldots, A'_n], u, \overline{x}, sh, pr)$, with $A'_i = (L'_i, u'_i, \overline{x'_i}, sh'_i, pr'_i)$, and let $\overline{y_1}, \ldots, \overline{y_m}$ be tuples of resources. Let $\texttt{Basic}(A) = \{(u_1, \overline{x_1}), \ldots, (u_m, \overline{x_m})\}$.

Finally, let us consider $R \subseteq \{1, \ldots, m\}$. Then,

$$\forall\, 1 \le i \le m : u_i(\overline{x_i} + \overline{y_i}) \ge u_i(\overline{x_i}) \;\wedge\; \forall\, i \in R : u_i(\overline{x_i} + \overline{y_i}) > u_i(\overline{x_i})$$

$$\Downarrow$$

$$\forall\, 1 \le i \le n : \left(\begin{array}{c} u'_i\left(\overline{x'_i} + \displaystyle\sum_{(u_j,\overline{x_j}) \in \texttt{Basic}(A'_i)} \overline{y_j}\right) \ge u'_i(\overline{x'_i}) \\ \wedge \\ \left(\begin{array}{c} \{j \mid (u_j, \overline{x_j}) \in \texttt{Basic}(A'_i)\} \cap R \neq \emptyset \\ \Downarrow \\ u'_i\left(\overline{x'_i} + \displaystyle\sum_{(u_j,\overline{x_j}) \in \texttt{Basic}(A'_i)} \overline{y_j}\right) > u'_i(\overline{x'_i}) \end{array} \right) \end{array} \right)$$

Proof: The proof is similar to that of Lemma 1. Thus, let us consider only the case of agents whose utility increases after the exchange. Let A'_i be an agent such that $\{j \mid (u_j, \overline{x_j}) \in \texttt{Basic}(A'_i)\} \cap R \neq \emptyset$. Let us note that the utility of A'_i is strictly higher after the exchange because there exists $(u_k, \overline{x_k}) \in \texttt{Basic}(A'_i)$ such that $u_k(\overline{x_k} + \overline{y_k}) > u_k(\overline{x_k})$ and for any $(u_j, \overline{x_j}) \in \texttt{Basic}(A'_i)$ we have that $u_j(\overline{x_j} + \overline{y_j}) \ge u_j(\overline{x_j})$. Since there is a configuration of resources where no subagent of A'_i worsens and some of them improve, we may conclude that the addition of utilities of subagents of A'_i in the best configuration is strictly higher after the exchange. □