Towards Multi-Agent Knowledge Allocation

Madalina Croitoru, Sebastian Rudolph

Towards Multi-Agent Knowledge Allocation

Madalina Croitoru[1] Sebastian Rudolph[2]
[1]University of Montpellier II, France
[2] KIT, Germany

Abstract

The classical setting of query answering either assumes the existence of just one knowledge requester, or the knowledge requests from different parties are treated independently from each other. This assumption does not always hold in practical applications where requesters often are in direct competition for knowledge. We provide a formal model for this type of scenario scenario by proposing the Multi-Agent Knowledge Allocation (MAKA) setting which combines the fields of query answering in information systems and multi-agent resource allocation. We define a bidding language based on exclusivity-annotated conjunctive queries and succinctly translate the allocation problem into a graph structure allowing to employ a wide range of constraint solving techniques for optimal allocation.

1 Introduction

Conjunctive query answering (between knowledge requester and knowledge provider) constitutes the de-facto standard of interacting with resources of structured information. It has been widely addressed in the literature, starting from the databases area (see (see [8, 1], but later being adopted for ontological information systems as well [6, 5]. The classical setting in query answering is focused on the case where just one knowledge requester is present. In case multiple requesters are present, the queries posed by different parties are processed and answered as independent from each other, thus making the multi-requester scenario a straightforward extension of the individual case. If at some time, n knowledge requesters asked for (eventually overlapping) information, this information would (potentially duplicately) be distributed to everybody according to their query.

While the above practice is natural in some cases, the assumption that queries can be processed independently clearly does not always hold in practical applications where the requesters are in direct competition for information. Let us consider for instance a multi-agent setting, with requester agents concurrently demanding information from a provider agent (example scenarios include military applications, news agencies, intelligence services, etc.). Of course, in this context, requester agents will not be willing to share "sensitive" information with other agents.

A structurally related problem is the multi-agent resource allocation (MARA) setting [9]. However, in such a setting (i) the agents ask for resources (not knowledge) and (ii) agents *a priori* know the pool of available resources. Work in this field is either aimed towards bidding language expressiveness (what preferences over subsets of resources can be correctly represented by the language) or algorithmic aspects of the allocation problem (see for instance [11, 2, 10] and others). The notion of multiplicity of resources, or resources used exclusively or shared has also been recently investigated in a logic-based language [13].

In the proposed multi-agent knowledge allocation (MAKA) setting, the n requester agents, at some given time (in a single-step), ask for knowledge (and not resources). They express their requests in the form of conjunctive queries that are endowed with exclusivity constraints and valuations, which indicate the subjective value of potentially allocated answers. Knowledge allocation poses interesting inherent problems not only from a bidding and query answering viewpoint, but also in terms of mechanism design.

The contributions of this paper can be summarized as follows: On the conceptual side, the aim of this paper is

- to motivate and formally define the novel problem of Multi-Agent Knowledge Allocation,
- to define the notion of exclusivity-enabled queries to account for a setting where knowledge requesters compete for information and
- to lay down future work directions opened by this novel setting: increased expressivity, dynamic allocations, fairness, multiple providers etc.

On the technical side, drawing from the fields of query answering in information systems and multi-agent resource allocation, we

- define syntax and semantics of a bidding language featuring exclusivity-annotated conjunctive queries as well as valuation functions and
- show a way to succinctly translate the multi-agent knowledge allocation problem into a network flow problem allowing to employ a wide range of constraint solving techniques to find the optimal allocation of answers to agents.

The paper is structured as follows: In Section 2 we conceptually extend the classical setting of query answering by ways of handling exclusivity demands. Section 3 introduces and formally defines the Multi-Agent Knowledge Allocation problem. Section 4 informally argues for a flow-network-based representation of bids of knowledge requesters. Consequently, Section 5 defines Knowledge Allocation Networks and shows that the knowledge allocation problem can be reduced to a maximum flow problem in these networks. Section 7 concludes and discusses avenues for future work.

2 Querying with Exclusivity Constraints

We first introduce our framework of exclusivity-aware querying as a basis for an adequate MAKA bidding formalism. For illustration purposes, we accompany the notions introduced herein by an example about celebrity news. As a starting point, we define the term of a *knowledge base* to formally describe a pool of information (or knowledge) that a *knowledge provider agent* holds in stock and that is to be delivered to *knowledge requester agents* according to their bids expressed via queries.

Definition 1 (Knowledge Base)**.** *Let C be a set of constants and $P = P_1 \cup P_2 \ldots \cup P_n$ a set of predicates of arity $i = 1, \ldots, n$. Given some $i \in \{1, \ldots, n\}$, $p \in P_i$ and $c_1, \ldots, c_i \in C$ we call $p(c_1, c_2 \ldots, c_i)$ a* ground fact*. We denote by $\mathcal{GF}$ the set of all ground facts. A* knowledge base *$\mathcal{K}$ is defined as a set of ground facts: $\mathcal{K} \subseteq \mathcal{GF}$.*

Consider the following predicates: actor, director, singer (all unary), marriage and act (binary) and five constants AJ (Angelina Jolie), BP (Brad Pitt), MMS (Mr. and Ms. Smith), JB (Jessica Biel), JT (Justin Timberlake). The knowledge base we consider consists of the following ground facts:

$\mathtt{actor}(\mathtt{AJ})$	$\mathtt{director}(\mathtt{AJ})$	$\mathtt{marriage}(\mathtt{AJ}, \mathtt{BP})$
$\mathtt{actor}(\mathtt{BP})$	$\mathtt{singer}(\mathtt{JT})$	$\mathtt{act}(\mathtt{AJ}, \mathtt{MMS})$
$\mathtt{actor}(\mathtt{JB})$		$\mathtt{act}(\mathtt{BP}, \mathtt{MMS})$

Note that in this version, the notion of knowledge base is very much alike a classical database. However, we will extend this notion in future work by allowing more general logical statements.

To describe information needs, we must provide for a sort of templates where *(query) variables* take the place of the pieces of information that a knowledge requester is interested in. Following the standard terminology of logical querying, we call these templates *atoms*.

Definition 2 (Terms, Atoms)**.** *Let V be a countably infinite set of* (query) variables*. We define the set of* terms *by $T = V \cup C$. As usual, given $i \in \{1 \ldots n\}$, $p \in P_i$ and $t_1, \ldots, t_i \in T$ we call $p(t_1, \ldots, t_i)$ an* atom*. We denote by $\mathcal{A}_{T,P}$ the set of all atoms constructed from P and T. Note that ground facts are just special atoms.*

For instance, if we consider the set of variables $V = \{x, y\}$ and the set of constants $C = \{\mathtt{AJ}, \mathtt{BP}, \mathtt{MMS}, \mathtt{JB}, \mathtt{JT}\}$, then $\mathtt{actor}(x)$, $\mathtt{act}(y, \mathtt{MMS})$, $\mathtt{marriage}(\mathtt{AJ}, \mathtt{BP})$ are all atoms over the previously defined sets P and C.

Since in the MAKA scenario requesters might be competing for certain pieces of knowledge, we have to provide them with the possibility of asking for an atom exclusively (exclusive) or not (shared). This additional information is captured by the notion of *exclusivity-annotated* atoms, ground facts and queries defined next.

Definition 3 (Exclusivity-Annotated Atoms)**.** *An* exclusivity-annotated atom *is an element from $\mathcal{A}^{e}_{P,T} := \mathcal{A}_{P,T} \times \{\mathsf{shared}, \mathsf{exclusive}\}$. Let $\{\mathsf{shared}, \mathsf{exclusive}\}$ be ordered as to $\mathsf{shared} \preceq \mathsf{exclusive}$. In particular, an* exclusivity-annotated ground fact *is an element from $\mathcal{GF}^{e}_{P,C} := \mathcal{GF}_{P,T} \times \{\mathsf{shared}, \mathsf{exclusive}\}$.*

Considering our example, exclusivity-annotated atoms would for instance be: $\langle \mathtt{actor}(x), \mathsf{shared}\rangle$, $\langle \mathtt{marriage}(\mathtt{AJ}, \mathtt{BP}), \mathsf{shared}\rangle$, $\langle \mathtt{marriage}(\mathtt{AJ}, \mathtt{BP}), \mathsf{exclusive}\rangle$, $\langle \mathtt{act}(y, \mathtt{MMS}), \mathsf{shared}\rangle$.

Note that the idea of exclusivity annotation is a novel concept going beyond the classical query answering framework. The defined order defined between exclusive and shared is used, intuitively, for query answering. It allows to specify concisely that an answer delivered exclusively is suitable for a knowledge requester who demanded that information shared (but not vice-versa). The actual semantics of $\mathsf{exclusive}$ and shared will only be made explicit when defining what an allocation is. For specifying structurally complex information needs, (exclusivity-annotated) atoms need to be logically combined, giving rise to the central notion of (exclusivity-annotated) *queries*.

Definition 4 (Exclusivity-Annotated Queries)**.** *An* exclusivity-annotated conjunctive query *(EACQ) is an element of $boolexp(\mathcal{A}^{e}_{P,T})$, i.e., a positive boolean expression (an expression with boolean operators $\wedge$ and $\vee$) over exclusivity-annotated atoms.*[1]

For example, a query asking for exclusivity marriages between actors and directors (where only the "marriage" itself is required as exclusive information, but the "actor" and "director" knowledge is sharable with other knowledge requester agents) would be written as:

$$\begin{array}{l}\langle \mathtt{marriage}(x,y), \mathsf{exclusive}\rangle \wedge \\ \big((\langle \mathtt{actor}(x), \mathsf{shared}\rangle \wedge \langle \mathtt{director}(y), \mathsf{shared}\rangle) \vee \\ \ (\langle \mathtt{actor}(y), \mathsf{shared}\rangle \wedge \langle \mathtt{director}(x), \mathsf{shared}\rangle)\big).\end{array}$$

Apart from the exclusivity annotation, the presented query formalism obviously captures the core of the functionality of common querying formalisms like SQL and SPARQL [14]. Although we omit filtering for the sake of brevity, this could be easily accommodated, as well as left or right joins.

As usual, when an EACQ q is posed to a knowledge base, answers are encoded as bindings of the variables to elements from C that make q true in $\mathcal{K}$. For the further presentation, it is convenient to also identify parts W of $\mathcal{K}$ that allow to derive that μ is an answer to q; such W are called *witnesses*.[2]

Definition 5 (Query Answers & Witnesses)**.** *Let $vars(q)$ denote the set of variables occurring in any atom of q and $dup(\mathcal{K}) := \mathcal{K} \times \{\mathsf{shared}, \mathsf{exclusive}\}$ a knowledge base $\mathcal{K}$'s full enrichment with possible annotations. An* answer *to q w.r.t. $\mathcal{K}$ is a mapping $\mu : vars(q) \cup C \to C$ with $\mu(c) = c$ for all $c \in C$ such that $eval(q, \mu, dup(\mathcal{K})) = true$. Thereby, $eval : boolexp(\mathcal{A}^{e}_{P,T}) \times C^{T} \times 2^{\mathcal{GF}^{e}} \to \{true, false\}$ is the evaluation function defined as follows: For an exclusivity-annotated atom $\langle p(t_1, \ldots, t_i), e\rangle$, we let $eval(\langle p(t_1, \ldots, t_i), e\rangle, \mu, A) = true$ exactly if we have $\langle p(\mu(t_1), \ldots, \mu(t_i), f\rangle \in A$ for an $e \preceq f$; further, the truth-value assignment is lifted to boolean expressions in the usual way. Given an answer μ to a query q a* witness *for μ is a set $W \subseteq dup(\mathcal{K})$ of exclusivity-annotated ground atoms for which $eval(q, \mu, W) = true$. Moreover, W is called minimal, if for all $W' \subset W$ holds $eval(q, \mu, W') = false$. We let $\mathcal{W}^{\min}_{\mathcal{K},q,\mu}$ denote the set of all the minimal witnesses for μ and $\mathcal{W}^{\min}_{\mathcal{K},q}$ the set of all minimal witnesses for all answers to q.*

In the above definition, the use of the order $\preceq$ takes care of the fact that if some piece of knowledge is requested as shared information (i.e. without demanding exclusivity), it is acceptable to nevertheless get it assigned exclusively.

[1] Note that this actually generalizes the classical notion of conjunctive queries where only $\wedge$ is allowed.

[2] Since our querying formalism is monotone, we can be sure that μ is an answer to q w.r.t. W whenever it is an answer w.r.t. $\mathcal{K}$.

Let us consider the previous example query for marriages between actors, where the marriage information was asked for exclusively. There is only one answer μ to this query w.r.t. our previously introduced knowledge base: $\mu = \{x \mapsto \mathtt{AJ}, y \mapsto \mathtt{BP}\}$. There are four minimal witnesses for μ arising from different combinations of exclusivity annotations[3]:

atom	W_1	W_2	W_3	W_4
$\mathtt{marriage}(\mathtt{AJ}, \mathtt{BP})$	exc.	exc.	exc.	exc.
$\mathtt{director}(\mathtt{AJ})$	sh.	exc.	sh.	exc.
$\mathtt{actor}(\mathtt{BP})$	sh.	sh.	exc.	exc.

This means that the $\mathtt{marriage}(\mathtt{AJ}, \mathtt{BP})$ can only be exclusively allocated (as $\langle \mathtt{marriage}(\mathtt{AJ}, \mathtt{BP}), \mathsf{exclusive}\rangle$) but the $\mathtt{director}(\mathtt{AJ})$ and $\mathtt{actor}(\mathtt{BP})$ atoms can be either "shareably" allocated with other requesters ($\langle \mathtt{actor}(\mathtt{BP}), \mathsf{shared}\rangle$) or exclusively allocated only to one requester agent ($\langle \mathtt{director}(\mathtt{AJ}), \mathsf{exclusive}\rangle$) (cf. Definition 3).

Note that there could be an exponentially large number of witnesses introduced by conjunctions of disjunctions. This aspect will be further addressed in Section 6.

3 The Knowledge Allocation Problem Defined

Multi Agent Knowledge Allocation (MAKA) can be interpreted as an abstraction of a *market-based* centralized distributed knowledge-based system for query answering. In such a MAKA system, there is central node a, the *auctioneer* (or the *knowledge provider*), and a set of n nodes, $I = \{1, \ldots, n\}$, the *bidders* (or the *knowledge requesters*), which express their information need (including exclusivity requirements) via queries, which are to be evaluated against a knowledge base $\mathcal{K}$, held by the auctioneer. Depending on the allocation made by the auctioneer, the bidders will be provided with minimal witnesses for answers to their queries.

The auctioneer asks bidders to submit in a specified common language, the *bidding language*, their *knowledge request*.

Definition 6 (Knowledge Request)**.** *The* knowledge request *of bidder i, denoted by Q_i is a set of pairs $\langle q, \varphi\rangle$ where q is a an EACQ and $\varphi : \mathbb{N} \to \mathbb{R}_+$ is a monotonic function. Thereby, $\varphi(k)$ expresses the individual interest (value) of bidder i in obtaining k distinct answers to q.*

Following the ongoing example in the paper, a knowledge request for an exclusively known marriage between a known actor and a known director, where each such marriage information is paid 30 units for the would be singleton set $\{\langle q, \varphi\rangle\}$ with

$$\begin{aligned} q = \quad & \langle\langle \mathtt{marriage}(x, y), \mathsf{exclusive}\rangle \wedge \\ & \big((\langle \mathtt{actor}(x), \mathsf{shared}\rangle \wedge \langle \mathtt{director}(y), \mathsf{shared}\rangle) \vee \\ & \;(\langle \mathtt{actor}(y), \mathsf{shared}\rangle \wedge \langle \mathtt{director}(x), \mathsf{shared}\rangle)\big), \\ \varphi = \quad & k \mapsto 30 \cdot k. \end{aligned}$$

[3]For space reasons, exc. stands for exclusive and sh. stands for shared

Assume a fixed knowledge base $\mathcal{K}$. For an EACQ q we let $R(q)$ denote its set of answers μ. In the general case, only witnesses of an answer subset S of $R(q)$ can be allocated to the bidder who asked q. S could be empty: either there are no answers to the query or responses were given to other bidders who asked them exclusively and paid more. For the query above, there are six possible witnesses that can be allocated. Once one of the witnesses is allocated, then $\texttt{marriage}(\texttt{AJ}, \texttt{BP})$ can no longer be given to any other agent (as formally ensured in the definition of a knowledge allocation below). However, if another bidder asked the same query as above, but is willing to pay 70 units, the first bidder will get the empty set. Note that the same empty set will also be returned if the bidder will ask for marriages between two directors (the case where no answers were found in the knowledge base).

The valuation function $\varphi : \mathbb{N} \to \mathbb{R}_+$ can be defined in several ways. Assuming that $val_q^i \in \mathbb{R}_+$ denotes a bidder i's interest to obtain a single answer to a query q, standard valuation options are

- naive valuation: $\varphi^n(|S|) = |S| \cdot val_q^i$,
- threshold valuation: $\varphi^t(|S|) = |S| \cdot val_q^i$ if $|\ S\ | \leq threshold_{q_i}^i$ and $|S| \cdot (val_q^i - discount_q^i)$ otherwise,
- budget valuation: $\varphi^b(|S|) = min\{\varphi_i(|S|), budget_i\}$ where φ_i can either be φ_i^n or φ_i^t,
- exclusivity valuation: Defined as either of the valuations above except that the bidder will pay a bonus that justifies the exclusivity interest.

The naive valuation simply assigns the same value val_q^i for each answer satisfying the query. The threshold valuation also assigns the same value for each received answer but up to a limit of number of answers and then a discounted value (for the first $threshold_{q_i}^i$ answers the agent pays val_q^i and for the rest, he only pays $val_q^i - discount_q^i$). The budget valuation foresees if there are too many answers and imposes an upper limit for the price the bidder is willing to pay.

Consequently, given the knowledge request Q_j of some bidder j, the individual prize $\mathfrak{v}_j(S)$ the agent is willing to pay on receiving a portion $S \subseteq dup(\mathcal{K})$ of facts from the knowledge base endowed with exclusivity guarantees is calculated by summing up the costs for the individual query matches arising from $\langle q, \varphi \rangle \in Q_j$, which, in turn, are determined by counting the answers for q w.r.t. the partial knowledge base S and applying the function φ to that number, i.e.:

$$\mathfrak{v}_j(S) = \sum_{\langle q, \varphi \rangle \in Q_j} \varphi(|\{\mu \mid eval(q, \mu, S) = true\}|).$$

Based on bidders' valuations, the auctioneer will determine a *knowledge allocation*, specifying for each bidder her obtained knowledge bundle and satisfying the *exclusivity constraints* (expressing that exclusivity annotations associated to atoms in the respective bundle are indeed complied with).

Definition 7 (Knowledge Allocation). *Given a knowledge base $\mathcal{K}$ and a set $\{1, \ldots, n\}$ of bidders, a* knowledge allocation $\mathbf{O}$ *is defined as an n-tuple $(O_1, \ldots, O_n)$, with $O_i \subseteq dup(\mathcal{K})$ for all $i \in \{1, \ldots, n\}$ such that*

- $\{\langle a, \textsf{shared}\rangle, \langle a, \textsf{exclusive}\rangle\} \not\subseteq O_1 \cup \ldots \cup O_n$ *for all ground atoms $a \in \mathcal{K}$, and*
- $O_i \cap O_j \cap (\mathcal{K}\times\{\textsf{exclusive}\}) = \emptyset$ *for all i, j with $1 \leq i < j \leq n$.*

The first condition ensures that the same ground atom a has not been given both exclusively and shared to the same, or two different agents. This means, for instance, if in the previous example once we allocate $\langle \texttt{director}(\texttt{AJ}), \textsf{exclusive}\rangle$ to some requester, then we cannot also allocate it as shared ($\langle \texttt{director}(\texttt{AJ}), \textsf{shared}\rangle$).

The second condition ensures that two atoms have not been exclusively allocated to two different agents. Following our example, if W_4 is chosen for allocation then $\langle \texttt{director}(\texttt{AJ}), \textsf{exclusive}\rangle$ and $\langle \texttt{actor}(\texttt{BP}), \textsf{exclusive}\rangle$ will only be allocated to one single agent and cannot be allocated exclusively to other agents. Also, please note that according to the first condition we cannot allocate it shared either.

Given a knowledge allocation, one can compute its *global value* by summing up the individual prizes paid by the bidders for the share they receive. Obviously, the knowledge allocation problem aims at an *optimal allocation*, which maximizes this value.

Definition 8. *Given an allocation $\mathbf{O}$, its* global value $\mathfrak{v}(\mathbf{O})$ *is defined by $\mathfrak{v}(\mathbf{O}) = \sum_{j=1,n} \mathfrak{v}_j(O_j)$. An allocation $\mathbf{O}$ will be called* optimal *if for all allocations $\mathbf{O}'$ holds $\mathfrak{v}(\mathbf{O}') \leq \mathfrak{v}(\mathbf{O})$.*

Let us again consider the knowledge base $\mathcal{K}$ introduced in Section 2 and three agents $1, 2, 3$ with knowledge requests Q_1, Q_2, Q_3 asking for information. Assume $Q_1 = \{\langle q_1, \varphi_1\rangle\}$, where q_1 asks for marriages where there is at least one actor involved[4] and $\varphi_1(k) = 50 \cdot k$, i.e. Agent 1 is willing to pay 50 units for every answer. Further assume $Q_2 = \{\langle q_2, \varphi_2\rangle\}$ where q_2 asks for marriages between an actor and a director but requests the marriage information exclusively and $\varphi_2(k) = 120 \cdot k$. Finally let $Q_3 = \{\langle q_3, \varphi_3\rangle\}$ with q_3 asking for marriages between people acting in the same movie and $\varphi_3(k) = 100 \cdot k$. From the values above it is obvious that $\texttt{marriage}(\texttt{AJ}, \texttt{BP})$ (the ground fact that poses shareability problems) will get allocated to the two agents 1 and 3 that did not ask for exclusivity and will bring a joint revenue of 150, while allocating it exclusively to agent 2 would only bring 120. Of course, for any number of possible answers and when different types of valuations come into play (threshold, budget etc.) the problem remains the same: finding the best admissible split of answers that maximizes revenue.

The task of the auctioneer finding a maximum value allocation for a given set of bidders' knowledge requests $\{Q_1, \ldots, Q_n\}$, is called in the Combinatorial Auctions' field the *Winner Determination Problem* (WDP). This is a NP-hard problem, being equivalent to weighted set-packing. It tends to be solvable in many practical cases, but care is often required in formulating the problem to capture structure that is present in the domain [15]. Usually, the WDP is expressed as an integer linear programming problem (ILP) there are standard methods for solving this type of problems [9].

[4]Unless the exclusivity is stated explicitly, the agents are willing to share the information.

We solve the multi-agent knowledge allocation problem by reducing using a flow based representation of the TBBL language [7] for combinatorial auctions. In the next section we give a quick overview of the TBBL language and then we show how the flow representation of this language naturally captures the MAKA problem.

4 How to Represent Bids

A TBBL bid is represented as a tree, where the leaf nodes represent the goods and the non-leaf nodes are defined by the means of an IC operator that is associated with a lower bound x, and an upper bound y (written: IC_x^y). Both x and y are non-negative integers. An IC operator is defined to be satisfied depending on the satisfaction of its children: IC_x^y is satisfied if at least x and at most y of its children are satisfied.

Consequently, IC can be instantiated to become an element of the class of logical operators, such as XOR, OR, AND:

- $XOR(i_1, i_2, ..., i_x) = IC_1^1(i_1, i_2, ..., i_x)$.
- $OR(i_1, i_2, ..., i_x) = IC_x^1(i_1, i_2, ..., i_x)$.
- $AND(i_1, i_2, ..., i_x) = IC_x^x(i_1, i_2, ..., i_x)$;

We can also concisely represent more "verbose" logical formulae such as $((a \wedge i_1 \wedge i_2) \vee (a \wedge i_2 \wedge i_3) \vee (a \wedge i_1 \wedge i_3))$ (a and any two of i_1, i_2 or i_3) by $IC_2^2(a, IC_3^2(i_1, i_2, i_3))$. Please see [7] for an in-depth discussion on the TBBL language and a comparison with other bidding languages in the literature such as OR^* and $\mathcal{L}_{GB}$.

In the following, we will use TBBL and adapt it to our representational needs. Let us illustrate the deployment of TBBL for representing MAKA bids by the means of an example.

In the previous section, an agent a_1 posed an EACQ q_1 requesting marriages where there is at least one actor involved:

$$\langle \mathtt{marriage}(x,y), \mathsf{shared}\rangle \wedge \\ \big(\langle \mathtt{actor}(x), \mathsf{shared}\rangle \vee \langle \mathtt{actor}(y), \mathsf{shared}\rangle\big)$$

Assuming the same knowledge base $\mathcal{K}$ as in the previous examples before, there are two answers μ_1 and μ_2 to this query: $\mu_1 = \{x \mapsto \mathtt{AJ}, y \mapsto \mathtt{BP}\}$; $\mu_2 = \{x \mapsto \mathtt{BP}, y \mapsto \mathtt{AJ}\}$. Using the TBBL representation depicted in Figure 1, we can describe which combinations of allocatable atoms answer q_1.

TBBL representations can be equivalently expressed using *flow networks*. A flow network is an acyclic digraph where the nodes are either internal nodes or two distinguished nodes (start and end). The edges in such a network are labeled with *capacities* (upper and lower). Intuitively, the flow needs to be able to pass from the start node to the end node in a way that the capacities of the edges are not violated. Thereby, the flow has to be *conserved* (basically the incoming flow in a node needs to be equal with the outgoing flow). The edge capacities can then "direct" the flow through certain nodes. To encode TBBL representations into a flow network, the containment IC_y^x nodes are represented by virtue of edges with according upper and lower capacities.

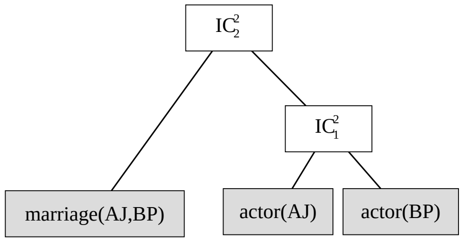

Figure 1: The TBBL depiction of a MAKA bid

In Figure 2 the network flow representation of the network depicted in Figure 1 is provided:

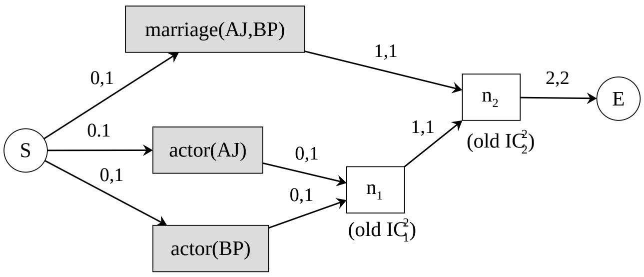

Figure 2: Flow representation of the TBBL MAKA bid

Now assume a second agent joins in, a_2, uttering a very similar request q_2: marriages with one actor involved, yet the marriage information is requested exclusively:

$$\langle \texttt{marriage}(x, y), \textsf{exclusive}\rangle \wedge \\ \big(\langle \texttt{actor}(x), \textsf{shared}\rangle \vee \langle \texttt{actor}(y), \textsf{shared}\rangle\big)$$

As before, there are two answers μ_1 and μ_2 to this query: $\mu_1 = \{x \mapsto \texttt{AJ}, y \mapsto \texttt{BP}\}$; $\mu_2 = \{x \mapsto \texttt{BP}, y \mapsto \texttt{AJ}\}$, and, as before, the according TBBL representation of this bid is the same as for a_1 shown in Figure 1. The extra constraint of the agent wanting exclusively the marriage information (not sharable with others) cannot be directly expressed in TBBL. As shown in [7] extra constraints are possible outside the TBBL representation and fully compatible with the MIPS implementation described by the authors. However, given that the exclusivity and sharable properties are key constraints within Multi-Agent Knowledge Allocation, we need to express it within the same representation as the bid. Therefore, we will privilege in this paper the flow representation which allows the encoding of the exclusivity information directly in the network.

We will do this by explicitly encoding into the network that exclusively allocated atoms cannot be given to other agents. This can be achieved by introducing another node in the network that represents the exclusive marriage information. This node will be made allocatable only to one agent by setting the capacities of the participating edges accordingly. This way, the flow (which will be ultimately used to determine admissible allocations of the knowledge), can only be directed in its entirety to one single agent, depriving the others agents of the possibility of receiving it. Flow associated to sharable information, on the other hand, will be allowed to be split and hence provide more than one agent with the requested information.

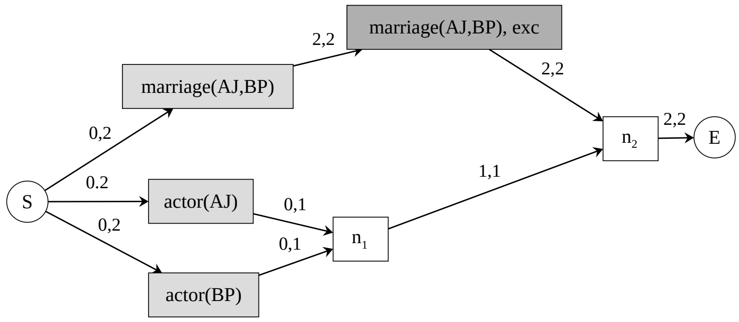

Figure 3: Flow representation of an exclusivity MAKA bid

In Figure 3 the flow network corresponding to agent a_2 is shown. The reason why there is now an initial flow of up to 2 from the start node to the ground atoms is that there are now two agents (a_1 and a_2) involved – the network of a_1 would have to be adapted accordingly.

One major advantage of the representation just introduced is that the individual bidders' networks can be integrated into a global network from which admissible allocations can be derived. The network in Figure 4 shows both agents, a_1 and a_2, bidding for information. Each of the agents have submitted their bids which have been put together in a complete network, by means of which also potential conflicts can be analyzed. One can see that indeed the exclusivity constraint is enforced by requiring an in-flow of exact 2 (and not less) to the node labeled $\langle \mathtt{marriage}(\mathtt{AJ}, \mathtt{BP}), \mathsf{exclusive} \rangle$, which would leave no flow for a_2 since the total maximal flow through the node $\langle \mathtt{marriage}(\mathtt{AJ}, \mathtt{BP}) \rangle$ is just 2.

As we have seen, the flow network encoding of agent's bids has several advantages. It does not only allow for a rather intuitive representation of exclusivity constraints, it also gives rise to a direct way of integrating the bids of all participating requesters. Moreover, this integrated representation can be directly employed to solve the allocation problem as admissible allocations correspond to admissible flows.

While this section was aimed at providing the necessary intuitions by ad-hoc examples, the next section will be devoted to defining the employed type of network formally, describing a generic method to create such a network out of a set of MAKA knowledge requests, and establishing the correspondences between network flows and optimal allocations.

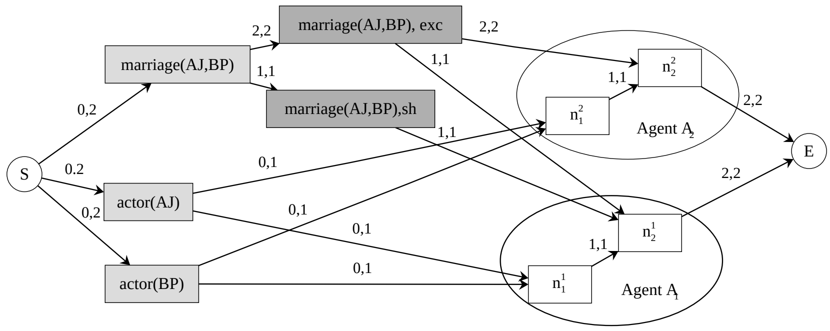

Figure 4: Flow representation of two MAKA bids

5 Defining Knowledge Allocation Networks

As demonstrated in the previous section, we propose to solve the MAKA problem using a network representation that captures the notion of containment of an atom in a witness allocated to a given agent. The above containment relation is represented using paths in the network flow (an atom belongs to a witness if and only if there is a certain path from the atom vertex to a vertex representing the witness) and a mechanism to express which path must be considered in order to instantiate a given witness. This mechanism is based on a simple extension of network flows, which is described below. We first define the generic terminology of allocation networks before we define how to create a knowledge allocation network representing a specific MAKA problem.

Definition 9. *An* allocation network *is a tuple* $\mathcal{N} = (V_\mathrm{a}, V_\mathrm{o}, E, \textsc{start}, \textsc{end}, \mathsf{cap}, \mathsf{lb}, \mathsf{flow}, \mathsf{val})$ *where:*

1. $(V_\mathrm{a} \cup V_\mathrm{o}, E)$ *is an acyclic digraph with two types of nodes, called* allocatable *(V_a) and* other *nodes (V_o) and two distinguished nodes* $\textsc{start}, \textsc{end} \in V_\mathrm{o}$*; Every node in* $\mathcal{N}$ *lies on a directed path from* $\textsc{start}$ *to* $\textsc{end}$*. We use* $V_\mathrm{int} = (V_\mathrm{a} \cup V_\mathrm{o}) \setminus \{\textsc{start}, \textsc{end}\}$ *to denote* internal nodes.
2. $\mathsf{cap}, \mathsf{lb} : E \to \mathbb{N} \cup \{+\infty\}$ *are functions defined on the set of directed edges; for a directed edge* $\langle i, j\rangle \in E$ *we call* $\mathsf{cap}(\langle i, j\rangle)$*, denoted* c_{ij}*, the* capacity *and we call* $\mathsf{lb}(\langle i, j\rangle)$*, denoted* l_{ij}*, the* lower bound *on* $\langle i, j\rangle$ *and additionally require* $l_{ij} \leq c_{ij}$*. All edges* $\langle a, \textsc{end}\rangle$ *have capacity* $+\infty$ *and lower bound* 0.
3. $\mathsf{flow} : V_\mathrm{int} \to \{\mathsf{conservation}, \mathsf{multiplier}\}$ *is a labeling function which associates each internal node* v *with a flow rule.*
4. $\mathsf{val} : V^- \times \mathbb{N} \to \mathbb{R}^+$*, a monotonic flow valuation function, where* $V^- = \{v \mid \langle v, \textsc{end}\rangle \in E\}$.

As discussed before, we now assume that flow is pushed through the network starting from the arcs leaving the start node. The flow on each arc is a non-negative integer value. If the flow f_{ij} on an arc $\langle i, j\rangle$ is positive, it must satisfy the restriction

$l_{ij} \leq f_{ij} \leq c_{ij}$, where the *lower bound* l and the *upper bound* c (since sometimes called *capacity*). Finally, the internal nodes of the network obey the flow rules associated to them.

Definition 10. *An* allocation flow *in an allocation network* $\mathcal{N} = (V_\mathrm{a}, V_\mathrm{o}, E, \textsc{start}, \textsc{end}, \mathsf{cap}, \mathsf{lb}, \mathsf{flow}, \mathsf{val})$ *is a function* $f : E \to \mathbb{N}$ *satisfying the following properties, where we let* f_{ij} *denote* $f(\langle i, j\rangle)$*:*

1. *For each* $\langle i, j\rangle \in E$ *we have* $f_{ij} = 0$ *or* $l_{ij} \leq f_{ij} \leq c_{ij}$
2. *If* $v \in V_\mathrm{int}$ *has* $\mathsf{flow}(v) = \mathsf{conservation}$ *then* $\sum_{\langle i,v\rangle \in E} f_{iv} = \sum_{\langle v,i\rangle \in E} f_{vi}$.
3. *If* $v \in V_\mathrm{int}$ *has* $\mathsf{flow}(v) = \mathsf{multiplier}$ *and* $f_{iv} > 0$ *for some* i*, then* $l_{vj} < f_{vj} < c_{vj}$ *for all* j *with* $\langle v, j\rangle \in E$.
4. *If* $v \in V_\mathrm{int}$ *has* $\mathsf{flow}(v) = \mathsf{multiplier}$ *and* $f_{iv} = 0$ *for all* i*, then* $f_{vj} = 0$ *for all* j *with* $\langle v, j\rangle \in E$.

The set of all allocation flows in $\mathcal{N}$ *is denoted by* $\mathcal{F}^{\mathcal{N}}$.

After having defined the network and flows on it, we can define how the network gives rise to a function associating values to subsets S of the network's allocatable nodes. This valuation function is derived from the maximal flow possible when only nodes from S are allowed to receive nonzero in-flow.

Definition 11. *Let* f *be an allocation flow in an allocation network:* $\mathcal{N} = (V_\mathrm{a}, V_\mathrm{o}, E, \textsc{start}, \textsc{end}, \mathsf{cap}, \mathsf{lb}, \mathsf{flow}, \mathsf{val})$. *The* value of f, $\mathsf{val}(f)$, *is defined as* $\mathsf{val}(f) = \sum_{v \in V^-} \mathsf{val}(v, f_{v\mathrm{END}})$. *The* grant *associated to* f*, denoted by* $G_f \subseteq V_\mathrm{a}$ *is defined by* $G_f = \{a \in V_\mathrm{a} \mid f_{va} > 0 \text{ for some } \langle v, a\rangle \in E\}$. *The* valuation *associated to* $\mathcal{N}$ *is the function* $\mathfrak{v}_\mathcal{N} : 2^{V_\mathrm{a}} \to \mathbb{R}_+$*, where for each* $X \subseteq V_\mathrm{a}$, $\mathfrak{v}_\mathcal{N}(X) = \max\{\mathsf{val}(f) \mid f \in \mathcal{F}^{\mathcal{N}}, G_f = X\}$.

In the following, we will describe how the framework of allocation networks can be applied to our problem of multi-agent knowledge allocation. To that end, we define two types of networks. *Single bidder allocation networks* capture the payments obtainable from bidders for a given portion of the knowledge base $\mathcal{K}$ according to their knowledge requests.

Definition 12. *Given a knowledge base* $\mathcal{K}$ *and some bidder's knowledge request* Q*, the associated* single bidder allocation network $\mathcal{N}_{\mathcal{K},Q}$ *is the allocation network* $(V_\mathrm{a}, V_\mathrm{o}, E, \textsc{start}, \textsc{end}, \mathsf{cap}, \mathsf{lb}, \mathsf{flow}, \mathsf{val})$ *where*

1. $V_\mathrm{a} = \bigcup\{W \mid W \in \mathcal{W}^{\min}_{\mathcal{K},q}, \langle q, \varphi\rangle \in Q\}$*, i.e. the allocatable nodes contain every exclusivity-annotated ground fact occurring in any minimal witness for any answer to any query posed by the bidder. We let* $\mathsf{flow}(a) = \mathsf{multiplier}$ *for every* $a \in V_\mathrm{a}$
2. $V_\mathrm{o} = Q \cup Ans \cup Wit \cup \{\textsc{start}, \textsc{end}\}$ *where:*
   - *For every* $\langle q, \varphi\rangle \in Q$ *we let* $\mathsf{flow}(\langle q, \varphi\rangle) = \mathsf{conservation}$ *as well as* $\mathsf{val}(\langle q, \varphi\rangle, n) = \varphi(n)$.
   - Ans *contains all pairs* $\langle q, \mu\rangle$ *for which* μ *is an answer to* q *and* $\langle q, \varphi\rangle \in Q$ *for some* φ*. We let* $\mathsf{flow}(\langle q, \mu\rangle) = \mathsf{multiplier}$

- *Wit contains all witnesses for Ans i.e.*
  $Wit = \bigcup_{\langle q,\mu\rangle \in Ans} \mathcal{W}^{\min}_{\mathcal{K},q,\mu}$, *we let* $\mathsf{flow}(W) = \mathsf{conservation}$ *for every* $W \in Wit$.

3. *E contains the following edges:*

   - $\langle \text{START}, a\rangle$ *for every* $a \in V_{\mathrm{a}}$*, whereby* $l_{\text{START},a} = 0$ *and* $c_{\text{START},a} = 1$
   - $\langle a, W\rangle$ *for* $a \in V_{\mathrm{a}}$ *and* $W \in Wit$ *whenever* $a \in W$*, moreover* $l_{a,W} = 1$, $c_{a,W} = 1$.
   - $\langle W, \langle q, \mu\rangle\rangle$ *for* $W \in Wit$ *and* $\langle q, \varphi\rangle \in Q$ *whenever* $W \in \mathcal{W}^{\min}_{\mathcal{K},q,\mu}$*. We let* $l_{a,W} = k$ *and* $c_{a,W} = k$ *where* $k = |\{a \mid \langle a, W\rangle \in E\}|$.
   - $\langle\langle q, \mu\rangle, \langle q, \varphi\rangle\rangle$ *for* $\langle q, \mu\rangle \in Ans$ *and* $\langle q, \varphi\rangle \in Q$*.We let* $l_{a,W} = 1$, $c_{a,W} = 1$.
   - $\langle\langle q, \varphi\rangle, \text{END}\rangle$ *for all* $\langle q, \varphi\rangle \in Q$; $l_{a,W} = 0$, $c_{a,W} = +\infty$

The following theorem guarantees, that the established network definition indeed captures the bidder's interest.

Theorem 1. *Given a knowledge base $\mathcal{K}$, some bidder's knowledge request Q_i, and the respective single bidder allocation network $\mathcal{N}_{\mathcal{K},Q_i}$, the valuation $\mathfrak{v}_{\mathcal{N}_{\mathcal{K},Q_i}}$ associated to $\mathcal{N}_{\mathcal{K},Q_i}$ coincides with the bidder's valuation $\mathfrak{v}_i$.*

Proof. (Sketch) Consider a set S of exclusivity-annotated ground facts. By definition, $\mathfrak{v}_{\mathcal{N}_{\mathcal{K},Q_i}}(S)$ equals the maximal $\mathsf{val}(f)$ among all flows f for which exactly those allocatable nodes that correspond to S receive non-zero in-flow. For such a flow f, the structure of the $\mathcal{N}_{\mathcal{K},Q_i}$ ensures that

- exactly those nodes from Wit receive full input flow which correspond to the witnesses contained in S,
- consequently, exactly those nodes from Ans receive non-zero input flow which correspond to answers for which S contains some witness,
- consequently, the flow that END receives from each $\langle q, \varphi\rangle \in Q$ equals the number of answers to q that can be derived from S.

Given these correspondences, we can conclude:

$$\begin{aligned}\mathfrak{v}_{\mathcal{N}_{\mathcal{K},Q_i}}(S) &= \textstyle\sum_{\langle q,\varphi\rangle \in Q_i} \varphi(f_{\langle q,\varphi\rangle \text{END}}) \\ &= \textstyle\sum_{\langle q,\varphi\rangle \in Q_i} \varphi(|\{\mu \mid eval(q, \mu, S) = true\}|) \\ &= \mathfrak{v}_i(S)\end{aligned}$$

□

In a second step, we conjoin the individual bidders networks into one large network which also takes care of the auctioneer's actual allocation problem including the enforcement of the exclusivity constraints. Figure 5 illustrates the overall structure of the Knowledge Allocation Network formally defined next.

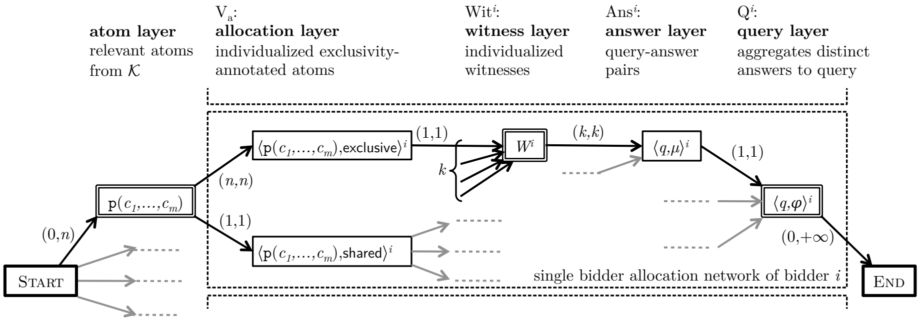

Figure 5: Structure of a Knowledge Allocation Network. Doubly boxed nodes indicate the conservation flow rule, all other internal nodes are multipliers. Edges $\langle i, j\rangle$ are associated with lower bounds and capacities by labels (l_{ij}, c_{ij}).

Definition 13. *Given a knowledge base* $\mathcal{K}$ *and bidders' knowledge requests* $Q_1, \ldots, Q_n$*, the associated* knowledge allocation network $\mathcal{N}_{\mathcal{K},Q_1,\ldots,Q_n}$ *is the structure* $(V_\text{a}, V_\text{o}, E, \text{START}, \text{END}, \mathsf{cap}, \mathsf{lb}, \mathsf{flow}, \mathsf{val})$ *constructed from the respective single bidder allocation networks* $\mathcal{N}_{\mathcal{K},Q_i} = (V_\text{a}^i, V_\text{o}^i, E^i, \text{START}^i, \text{END}^i, \mathsf{cap}^i, \mathsf{lb}^i, \mathsf{flow}^i, \mathsf{val}^i)$ *in the following way:*

1. $V_\text{a} \cup V_\text{o} = \mathit{Atoms} \cup \{\text{START}, \text{END}\} \cup \bigcup_i \{v^i \mid v \in V_\text{int}^i\}$ *i.e. except for the start and end nodes, the knowledge allocation network contains one copy of every single bidder network node, moreover we let* $\mathsf{flow}(v^i) = \mathsf{flow}^i(v)$ *and* $V_\text{a} = \bigcup_i \{v^i \mid v \in V_\text{a}^i\}$*, i.e., the allocatable nodes in the new network are exactly the allocatable nodes from all the single bidder allocation networks. As new nodes we have* $\mathit{Atoms} = \{a \mid \{\langle a, \mathsf{exclusive}\rangle, \langle a, \mathsf{shared}\rangle\} \cap \bigcup_i V_\text{a}^i \neq \emptyset\}$*, i.e., annotation-free atoms whose annotated versions are part of witnesses for answers to some of the bidder-posed queries. We also let* $\mathsf{val}(v^i) = \mathsf{val}^i(v)$ *where applicable.*
2. E *contains the following edges:*
   - $\langle v_1^i, v_2^i\rangle$ *for all* $\langle v_1, v_2\rangle \in E^i$*, i.e. all edges between internal nodes of the single bidder networks remain unchanged, the same holds for their capacity and lower bound values,*
   - $\langle \text{START}, a\rangle$ *for all* $a \in V_\text{a}$*, we let* $l_{\text{START}a} = 0$ *as well as* $c_{\text{START}a} = n$*,*
   - $\langle a, \langle a, \mathsf{shared}\rangle^i\rangle$ *for all* $\langle a, \mathsf{shared}\rangle \in V_\text{a}^i$*, we let* $l_{a,\langle a,\mathsf{shared}\rangle^i} = 1$ *and* $c_{a,\langle a,\mathsf{shared}\rangle^i} = 1$*,*
   - $\langle a, \langle a, \mathsf{exclusive}\rangle^i\rangle$ *for all* $\langle a, \mathsf{exclusive}\rangle \in V_\text{a}^i$*, and we let* $l_{a,\langle a,\mathsf{exclusive}\rangle^i} = n$ *and* $c_{a,\langle a,\mathsf{exclusive}\rangle^i} = n$*,*
   - $\langle\langle q, \varphi\rangle^i, \text{END}\rangle$ *for every* $\langle\langle q, \varphi\rangle, \text{END}\rangle \in E_i$*, as required we let* $l_{\langle q,\varphi\rangle^i,\text{END}} = 0$ *and* $c_{\langle q,\varphi\rangle^i,\text{END}} = +\infty$*.*

We arrive at the following theorem, which ensures that the MAKA problem can be reformulated as a maximal flow problem in our defined knowledge allocation network. Therefore, once the network has been constructed, any maximum value flow will represent the maximal allocation.

Theorem 2. *Given a knowledge base $\mathcal{K}$, bidders' knowledge requests $Q_1, \ldots, Q_n$, and the associated* knowledge allocation network $\mathcal{N}_{\mathcal{K},Q_1,\ldots,Q_n}$*, the valuation $\mathfrak{v}_{\mathcal{N}_{\mathcal{K},Q_1,\ldots,Q_n}}$ associated to $\mathcal{N}_{\mathcal{K},Q_1,\ldots,Q_n}$ is related with the global valuation $\mathfrak{v}$ of the knowledge allocation problem in the following way:*

1. *for every network flow $f \in \mathcal{F}^{\mathcal{N}_{\mathcal{K},Q_1,\ldots,Q_n}}$, we have that $\mathbf{O}_f := (\{a \mid a^1 \in G_f\}, \ldots, \{a \mid a^n \in G_f\})$ is a knowledge allocation (in particular, the exclusivity constraints are satisfied),*

2. *conversely, for every knowledge allocation $\mathbf{O} = (O_1, \ldots, O_n)$, there is a network flow $f \in \mathcal{F}^{\mathcal{N}_{\mathcal{K},Q_1,\ldots,Q_n}}$ with $G_f = \bigcup_{1 \leq i \leq n} \{a^i \mid a \in O_i\}$,*

3. *for any allocation $\mathbf{O} = (O_1, \ldots, O_n)$, we have*

$$\mathfrak{v}_{\mathcal{N}_{\mathcal{K},Q_1,\ldots,Q_n}}\Big(\bigcup_{1 \leq i \leq n} \{a^i \mid a \in O_i\}\Big) = \mathfrak{v}(\mathbf{O}).$$

Proof. (Sketch) To show the first claim, one has to show that $(\{a \mid a^1 \in G_f\}, \ldots, \{a \mid a^n \in G_f\})$ satisfies the two conditions from Definition 7. That this is indeed the case by construction can be argued along the same lines as the explanations in Section 4. To show the second claim, one assumes a knowledge allocation $(O_1, \ldots, O_n)$ and constructs a valid flow for $\mathcal{N}_{\mathcal{K},Q_1,\ldots,Q_n}$ along the correspondences shown in the proof of Theorem 1. Finally given the one-to-one correspondence between flows and allocations established by the two preceding claims, one can use Theorem 1 to show that the two functions indeed coincide. □

6 Succinct Representation via Witness Structures

As mentioned in Section 2, conjunctions of disjunctions could potentially introduce an exponentially large number of witnesses. In this section we address this aspect and present a succinct representation via witness structures defined as follows.

Given an EACQ q and an answer μ w.r.t. a set $\mathcal{S}$ of exclusivity annotated axioms, the corresponding *witness structure* is obtained as follows:

- Let $\mu(q)$ denote the boolean expression on exclusivity-annotated ground terms obtained by substituting all terms in q according to μ.

- Obtain $\mu(q)|_{\mathcal{S}}$ by substituting every occurrence of any exclusivity-annotated ground fact $a \notin \mathcal{S}$ in $\mu(q)$ by $\perp$.

- Let $b_{\mu,q,\mathcal{S}}$ be the boolean expression obtained by applying the recursive simplifying function s to $\mu(q)|_{\mathcal{S}}$ where

$$\mathrm{s}: \begin{cases} \bot \mapsto \bot \\ a \mapsto a \text{ for } a \in \mathcal{GF}^e \\ a_1 \wedge a_2 \mapsto \begin{cases} \bot \text{ if } \bot \in \{\mathrm{s}(a_1), \mathrm{s}(a_2)\} \\ \mathrm{s}(a_1) \wedge \mathrm{s}(a_2) \text{ otherwise,} \end{cases} \\ a_1 \vee a_2 \mapsto \begin{cases} \bot \text{ if } \bot = \mathrm{s}(a_1) = \mathrm{s}(a_2) \\ \mathrm{s}(a) \text{ if } \{\mathrm{s}(a_1), \mathrm{s}(a_2)\} = \{\bot, a\} \\ \mathrm{s}(a_1) \vee \mathrm{s}(a_2) \text{ otherwise.} \end{cases} \end{cases}$$

- Exploiting commutativity and associativity of $\wedge$ and $\vee$ rewrite the binary application of these operators into set-based ones $\bigwedge$ and $\bigvee$, obtaining $b'_{\mu,q,\mathcal{S}}$.

- Now, the term structure $b'_{\mu,q,\mathcal{S}}$ gives rise to a directed acyclic graph $\mathcal{G}_{\mu,q,\mathcal{S}}$, called *witness structure* whose nodes are (i) exclusivity-annotated ground atoms (ii) $\bigwedge$-nodes or (iii) $\bigvee$-nodes.

Noting that $\bigvee$- and $\bigwedge$-nodes can be implemented by the multiplier and the conservation flow rules together with adequate capacities and lower bounds, it is straightforward to see that $\mathcal{G}_{\mu,q,dup(\mathcal{K})}$ can be used as a part of the bidding graph of an agent posing q as a part of his bid. Thereby, the witness structures substitute the witness layer and interlink allocation layer and answer layer. This representation can then be lifted to knowledge allocation networks and is more succinct than the naïve strategy of introducing nodes for all (minimal) witnesses for μ, as only one witness structure (the size of which is linearly bounded by the query) per answer is needed.

7 Conclusion and Future Work

We have introduced and formally defined the problem of Multi-Agent Knowledge Allocation by drawing from the fields of query answering in information systems and combinatory auctions. To this end, we have defined a bidding language based on exclusivity-annotated conjunctive queries. Moreover we have shown a way to succinctly translate the allocation problem into flow networks allowing to employ a wide range of constraint solving techniques to find the optimal allocation.

While an implementation of the existing framework – combining existing query-answering with constraint solving methods – is one of the immediate next steps, conceptual future work on the subject will include the following topics:

- **Extending the bidding language**: One straightforward extension would be to allow for so-called *non-distinguished* variables, i.e. variables which need to be bound in order to make the query match, but the concrete binding itself is not of interest (comparable to variables in SQL statements which do not occur in the SELECT part). While it is straightforward to extend the language accordingly, an appropriate extension of the framework would be to allow not just for ground facts (like $\texttt{marriage}(\texttt{AJ}, \texttt{BP})$) to be delivered to the requester but also for "anonymized" facts (like $\texttt{marriage}(\texttt{AJ}, *)$

or, more formally $\exists x.\mathtt{marriage}(\mathtt{AJ}, x)$), which would in turn require an adaption of exclusivity handling.

- **Extending knowledge base expressivity**: On one hand, the knowledge base formalism could be extended to cover not just ground facts but more advanced logical statements such as Datalog rules (used in deductive databases) or ontology languages (such as RDFS [4] or OWL [12]). In that case, a distinction has to be made between propositions which are explicitly present in the knowledge base and those that are just logically entailed by it, which in turn requires to revisit our definitions: if the latter kind of propositions can be delivered to the requester, exclusivity constraints must be propagated to them. On the other hand, exclusivity annotations might be extended to allow for a more fine-grained specification: a requester might be willing to share some information with certain other requesters but not with all of them.

- **Covering Dynamic Aspects of Knowledge Allocation**: In particular in the area of news, dynamic aspects are of paramount importance: news items are annotated by time stamps and their value usually greatly depends on their timeliness. Moreover we can assume the information provider's knowledge pool to be continuously updated by incoming streams of new information. Under this aspect, the provider may not only have to choose whom to provide with a certain piece of information but also assess the likeliness that there will be a matching piece of information incoming such that the bundled information can be sold at a greater revenue. That is, it might make sense to withhold pieces of knowledge from immediate allocation.

- **Fairness**: As mentioned before, there could be that an agent receives the empty set ("0 answers to your query") given the fact that all the answers have been given to competitors who asked for the same information but paid more. This immediately raises fairness [3] problems that should be investigated under this setting.

- **Multiple Providers**: Finally, it might be useful to extend the setting to the case where multiple agents offer knowledge; in that case different auctioning and allocation mechanisms would have to be considered. This would also widen the focus towards distributed querying as well as knowledge-providing web-services and the corresponding matchmaking and orchestration problems.

References

[1] S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases*. Addison-Wesley, 1995.

[2] C. Boutilier and H. Holger. Bidding languages for combinatorial auctions. In *Proc. IJCAI'01*, pages 1211–1217, 2001.

[3] S. Brams. On envy-free cake division. *J. Comb. Theory, Ser. A*, 70(1):170–173, 1995.

[4] D. Brickley and R. V. Guha, editors. *RDF Vocabulary Description Language 1.0: RDF Schema*. W3C Recommendation, 10 February 2004. Available at `http://www.w3.org/TR/rdf-schema/`.

[5] A. Calì, G. Gottlob, and M. Kifer. Taming the infinite chase: Query answering under expressive relational constraints. In *Proc. KR'08*, pages 70–80, 2008.

[6] D. Calvanese, G. D. Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Tractable reasoning and efficient query answering in description logics: The *dl-lite* family. *J. Autom. Reasoning*, 39(3):385–429, 2007.

[7] R. Cavallo, D. Parkes, A. Juda, A. Kirsch, A. Kulesza, S. Lahaie, B. Lubin, L. Michael, and J. Shneidman. Tbbl: A tree-based bidding language for iterative combinatorial exchanges. In *Proc. IJCAI-Workshop on Advances in Preference Handling*, 2005.

[8] A. K. Chandra and P. M. Merlin. Optimal implementation of conjunctive queries in relational data bases. In *Proc. STOC'77*, pages 77–90, 1977.

[9] P. Cramton, Y. Shoham, and R. Steinberg. *Combinatorial Auctions*. MIT Press, 2006.

[10] A. Giovannucci, J. Rodriguez-Aguilar, J. Cerquides, and U. Endriss. Winner determination for mixed multi-unit combinatorial auctions via petri nets. In *Proc. AAMAS'07*, 2007.

[11] N. Nisan. Bidding and allocations in combinatorial auctions. In *Proc. EC-2000*, 2000.

[12] W. OWL Working Group. *OWL 2 Web Ontology Language: Document Overview*. W3C Recommendation, 27 October 2009. Available at `http://www.w3.org/TR/owl2-overview/`.

[13] D. Porello and U. Endriss. Modelling combinatorial auctions in linear logic. In *Proc. KR'10*, 2010.

[14] E. Prud'hommeaux and A. Seaborne, editors. *SPARQL Query Language for RDF*. W3C Recommendation, 15 January 2008. Available at `http://www.w3.org/TR/rdf-sparql-query/`.

[15] M. Rothkopf, A. Pekeąc, and R. Harstad. Computationally manageable combinatorial auctions. *Management Science*, 44:1131–1147, 1998.