Tensor-based Representation and Reasoning of Horn-$\mathcal{SHOIQ}$ Ontologies

Zhangquan Zhou

Southeast University, China, quanzz@seu.edu.cn

Abstract. Recently, tenors have been widely used to encode triples in an RDF graph, which is sometimes called a knowledge graph, for the purpose of knowledge completion. An interesting question is, can we use tensors to represent OWL ontologies, and handle logical reasoning with ontologies by tensor operations. Although there exists some work on representing logical rules by using tensors, they can hardly work for our case as they do not lead to a sound and complete reasoning. This motives us to explore a new kind of encoding method. In this paper, we take the first effort to theoretically build the connection between tensor-based representation and a Horn fragment of OWL, Horn-$\mathcal{SHOIQ}$, i.e., to study how to encode Horn-$\mathcal{SHOIQ}$ ontologies to tensors, and further consider using tensor operations to handle ontology materialization, which is an important logical reasoning service for ontology-based applications. We show that the soundness and completeness of ontology materialization can be guaranteed by using tensor operations.

Keywords: tensor, ontology, Horn-$\mathcal{SHOIQ}$, materialization

1 Introduction

Tensor has been used as a data structure in many fields for its capability of modeling different problems, and the feasibility of massive processing [8]; many applications are also built on tensor-based platforms, such as TensorFlow,[1] which is published by Google for efficiently processing large tensors. Recently, tenors have also been widely used to encode triples in an RDF graph, which is sometimes called a knowledge graph, for the purpose of knowledge completion [3,2,14,15]. An interesting question is, can we use tensor to represent the Web Ontology Language[2] (OWL), and handle logical reasoning with OWL ontologies by tensor operations?

There exists work on representing logic rules by using tensors. One line of the work is to transform logical reasoning to a statistic problem. In the work of [7,11,15,5], a logic rule is encoded to an objective function over tensors, or a constraint condition in an objective function. Logical reasoning is then transformed to a maximization problem, i.e., computing a group of arguments such that the objective function is optimal on the training inputs. One limit of the work in this line is that the given encoding methods work on facts or logic statements that are in the form of triples describing binary relations (e.g.,$\langle$*Jack hasMum Helen*$\rangle$),

[1] https://www.tensorflow.org/
[2] https://www.w3.org/OWL/

and can hardly be adapted to logic languages with complex syntaxes. Another line of the work focuses on using tensor products (i.e., *Kronecker products* and *inner products*) to represent logic languages and handle logical reasoning. We refer the readers to the two main related work [9,13]. This work studies the reasoning tasks that are provided for special purposes (i.e., query answering on real corpus), but does not aim to study a specific logic language that has formal definitions. Further, there is also no theoretical proofs of the soundness and completeness of reasoning based on tensor operations.

In this paper, we take the first effort to study the problem of using tensors to represent Horn-$\mathcal{SHOIQ}$, a Horn fragment of OWL, and further consider using tensor operations to handle ontology materialization, which is an important reasoning service in many ontology-based applications. According to [4], many real ontologies can be expressed in Horn-$\mathcal{SHOIQ}$. Thus, our work makes it possible for connecting expressive ontology reasoning and statistical reasoning based on tensors. On the other hand, we also explore the possibility of using the techniques of tensors to handle materialization such that materialization can be efficiently performed by high-performance tensor-based platforms.

The existing methods can hardly be adapted to Horn-$\mathcal{SHOIQ}$, since complex relations (e.g., *concept conjunction* and *existence restriction*) are allowed in Horn-$\mathcal{SHOIQ}$; the task of ontology materialization is also more complicated than the basic reasoning tasks studied in existing work. To address these problems, we propose a new kind of encoding method that can equivalently represent logic statements (i.e., *axioms* and *assertions*) occurring in an Horn-$\mathcal{SHOIQ}$ ontology. Based on this encoding method, we further identify a group of tensor operations to handle materialization. We also give the theoretical results about the soundness and completeness of materialization based on tenor operations. The proofs of these theoretical results can be found in the technical report which is also submitted to the conference system.

2 Ontology and Tensor Operations

In this section, we introduce some notions that are used in this paper.

Ontology and Materialization. Horn-$\mathcal{SHOIQ}$ is a Horn fragment of the *description logics* (DLs for short) that underpins OWL [1]. Let $\mathbf{CN}$, $\mathbf{RN}$ and $\mathbf{IN}$ be the sets of *atomic concepts*, *atomic roles* and *individuals* respectively, which are the basic logic elements in DLs. Let the set of roles be $\mathbf{R} := \mathbf{RN} \cup \{R^-|R \in \mathbf{RN}\}$ where R^- is the inverse role of R. For ease of discussion, we focus on the *simple forms* of *axioms* that are allowed in Horn-$\mathcal{SHOIQ}$ (see the second column of Table 1). We define a Horn-$\mathcal{SHOIQ}$ ontology $\mathcal{O}$ by a set of axioms of the forms (`A1`-`A11`), *concept assertions* of the form $A(a)$, *role assertions* of the form $R(a, b)$ and *equivalent assertions* of the form $a \approx b$. In an axiom of either of the forms (`A1`-`A11`), concepts $A_{(i)}$ and B are atomic concepts (including *top concept* ($\top$) and *bottom concept* ($\bot$)); R and S are roles in $\mathbf{R}$; a is an individual in $\mathbf{IN}$. The semantics follows the standard model theoretic semantics and can be found in [1].

Given a Horn-$\mathcal{SHOIQ}$ ontology $\mathcal{O}$, we use $\mathtt{mat}(\mathcal{O})$ to denote the result of ontology materialization with respect to $\mathcal{O}$. We assume that $\mathcal{O}$ satisfies *the tractability condition* [4], i.e., this ontology can be materialized in polynomial time. In this sense, $\mathtt{mat}(\mathcal{O})$ can be defined based on the rules in Table 1 (see the third column). Specifically, for each axiom and assertion $\alpha \in \mathcal{O}$, $\alpha \in \mathtt{mat}(\mathcal{O})$ holds; for each rule in Table 1, if the axioms and assertions occurring on the left hand side of $\vdash$ are in $\mathtt{mat}(\mathcal{O})$, then the assertions occurring on the right hand side of $\vdash$ are in $\mathtt{mat}(\mathcal{O})$. Each rule in Table 1 corresponds to the axiom in the same row. Without introducing confusions, we use Rule ($\mathtt{A}i$) ($i \in \{1, 2, .., 11\}$) to denote the rule that corresponds to the axiom ($\mathtt{A}i$). We take Rule ($\mathtt{A1}$) as an example. This rule says that if an axiom of the form $A \sqsubseteq B$ and a concept assertion of the form $A(a)$ are in $\mathtt{mat}(\mathcal{O})$, then $B(a)$ should be in $\mathtt{mat}(\mathcal{O})$, where $A \sqsubseteq B$ and $A(a)$ are called *premises*, $B(a)$ is called a *consequence*, and the common logic element among premises (A in this rule) is called a *joint*. The other rules can be explained similarly. Further, we say that a joint is an *invisible joint* if it does not occur in consequences (like A in the previous example); otherwise it is called a *visible joint*.

Tensor Operations [8]. An *order-n tensor* over the vector-space $\mathbb{R}^d$ is denoted by $\mathtt{T}^{(n)}$ (the superscript can be omitted).[3] $\mathtt{T}^{(n)}$ can be viewed as an n-dimensional array of real numbers, where each element is written as $\mathtt{T}_{\gamma_1 \dots \gamma_n}$ or $[\mathtt{T}]_{\gamma_1 : \gamma_n}$ ($\gamma_k \in \{1, 2, .., d\}$ for all $k \in \{1, .., n\}$). Specially, an order-1 tensor $\mathtt{V}^{(1)}$ over $\mathbb{R}^d$ is a d-dimensional vector; an order-2 tensor $\mathtt{M}^{(2)}$ is a matrix whose dimensions are $d \times d$. In this work, we apply three basic tensor operations, *tensor addition*, *Kronecker product* and *inner product*, which are introduced below.

Tensor addition is denoted by the operator $+$. The two tensors that participate in an addition are required to have the same order. Specifically, $\mathtt{U}^{(n)} + \mathtt{V}^{(n)} = \mathtt{T}^{(n)}$ where $\mathtt{T}_{\gamma_1 \dots \gamma_n} = \mathtt{U}_{\gamma_1 \dots \gamma_n} + \mathtt{V}_{\gamma_1 \dots \gamma_n}$. The two tensors $\mathtt{U}^{(n)}$ and $\mathtt{V}^{(n)}$ are two addends of $\mathtt{T}^{(n)}$; we say that $\mathtt{U}^{(n)}$ and $\mathtt{V}^{(n)}$ are *additive* in $\mathtt{T}^{(n)}$. Kronecker product, denoted by $\otimes$, is a kind of product operation that obtains tensors of higher orders. Kronecker product can be performed between any two tensors and is defined as $\mathtt{U}^{(n)} \otimes \mathtt{V}^{(m)} = \mathtt{T}^{(n+m)}$ where $\mathtt{T}_{\gamma_1 \dots \gamma_n \gamma'_1 \dots \gamma'_m} = \mathtt{U}_{\gamma_1 \dots \gamma_n} \mathtt{V}_{\gamma'_1 \dots \gamma'_m}$. For a tensor $\mathtt{T} = \mathtt{U}_1 \otimes \dots \otimes \mathtt{U}_k$, we say that $\mathtt{U}_i$ is the i^{th} *Kronecker term* of $\mathtt{T}$ where $i \in \{1, .., k\}$.

Inner product, denoted by $\bullet$, is defined as follows: $\mathtt{U}^{(n)} \bullet_{(i,j)} \mathtt{V}^{(m)} = \mathtt{T}^{(n+m-2)}$, where i (resp., j) points to the i^{th} (resp., j^{th}) order of $\mathtt{U}^{(n)}$ (resp., $\mathtt{V}^{(m)}$) for $i \in \{1, .., n\}$ and $j \in \{1, .., m\}$. Each element in the tensor $\mathtt{T}^{(n+m-2)}$ is written as $\mathtt{T}_{\gamma_1 \dots \gamma_{i-1} \gamma_{i+1} \dots \gamma_n \gamma'_1 \dots \gamma'_{j-1} \gamma'_{j+1} \dots \gamma'_m} = \sum_{\beta=1}^{d} \mathtt{U}_{\gamma_1 \dots \gamma_{i-1} \beta \gamma_{i+1} \dots \gamma_n} \mathtt{V}_{\gamma'_1 \dots \gamma'_{j-1} \beta \gamma'_{j+1} \dots \gamma'_m}$; we call the i^{th} order of $\mathtt{U}$ and the j^{th} order of $\mathtt{V}$ a pair of the *inner product orders* (an *IPO pair* for short). For simplicity, we apply *the Einstein Summation Convention* in inner products. That is $\mathtt{T}_{\gamma_1 \dots \gamma_{i-1} \gamma_{i+1} \dots \gamma_n \gamma'_1 \dots \gamma'_{j-1} \gamma'_{j+1} \dots \gamma'_m}$ can be directly written as $\mathtt{U}_{\gamma_1 \dots \gamma_{i-1} \beta \gamma_{i+1} \dots \gamma_n} \mathtt{V}_{\gamma'_1 \dots \gamma'_{j-1} \beta \gamma'_{j+1} \dots \gamma'_m}$, where we use the same symbol β to denote an IPO pair. For example, each element in $\mathtt{U}^{(2)} \bullet_{(1,2)} \mathtt{V}^{(3)}$ can be written as $\mathtt{U}_{\beta \gamma_2} \mathtt{V}_{\gamma'_1 \beta \gamma'_3}$. Inner products can be easily extended to general operations on two

[3] A vector-space $\mathbb{R}^d$ denotes a set of all d-dimensional vectors.

or more IPO pairs. Consider a tensor $\mathtt{T}' = \mathtt{U}^{(2)} \bullet_{(1,2)(2,3)} \mathtt{V}^{(3)}$ that performs inner product over $\mathtt{U}$ and $\mathtt{V}$ based on two IPO pairs. Each element of $\mathtt{T}'$ can be written as $[\mathtt{T}']_{\gamma_1'} = \mathtt{U}_{\alpha\beta}\mathtt{V}_{\gamma_1'\alpha\beta}$, i.e., $\sum_{\alpha=1}^{d}\sum_{\beta=1}^{d}\mathtt{U}_{\alpha\beta}\mathtt{V}_{\gamma_1'\alpha\beta}$. Without special statements, for inner products over two tensors with a same order, we omit the subscripts of $\bullet$ to denote that inner products are performed by treating all corresponding orders as IPO pairs, e.g., we use $\mathtt{U}^{(3)} \bullet \mathtt{V}^{(3)}$ to denote $\mathtt{U}^{(3)} \bullet_{(1,1)(2,2)(3,3)} \mathtt{V}^{(3)}$.

Table 1: Ontology Axioms, Rules for Materialization and Corresponding Tensor Operations

	Axioms	Rules	Tensor Operations
(A1)	$A \sqsubseteq B$	$A \sqsubseteq B, A(a) \vdash B(a)$	$[\phi_{\mathtt{A1}}(\mathfrak{O})]_{x_1:x_7} = \mathtt{T}^{\mathtt{ca}}_{x_1}(\mathfrak{O}_{y_1y_2y_3x_2x_3y_6x_6}\mathtt{T}^{\mathtt{A1}}_{y_1})(\mathfrak{O}_{z_1y_2y_3x_4x_5y_6x_7}\mathtt{T}^{\mathtt{ca}}_{z_1})$
(A2)	$A_1 \sqcap A_2 \sqsubseteq B$	$A_1 \sqcap A_2 \sqsubseteq B, A_1(a), A_2(a) \vdash B(a)$	$[\phi^{\mathtt{A2}}(\mathfrak{O})]_{x_1:x_7} = \mathtt{T}^{\mathtt{ca}}_{x_1}(\mathfrak{O}_{y_1y_2y_3y_4y_5x_2x_3}\mathtt{T}^{\mathtt{A2}}_{y_1})(\mathfrak{O}_{z_1y_2y_3z_4x_4z_6x_6}\mathtt{T}^{\mathtt{ca}}_{z_1})(\mathfrak{O}_{k_1y_4y_5z_4x_5x_6x_7}\mathtt{T}^{\mathtt{ca}}_{k_1})$
(A3)	$A \sqsubseteq \forall R.B$	$A \sqsubseteq \forall R.B, A(a), R(a,b) \vdash B(b)$	$[\phi^{\mathtt{A3}}(\mathfrak{O})]_{x_1:x_7} = \mathtt{T}^{\mathtt{ca}}_{x_1}(\mathfrak{O}_{y_1y_2y_3y_4y_5x_2x_3}\mathtt{T}^{\mathtt{A3}}_{y_1})(\mathfrak{O}_{z_1y_2y_3z_4z_5x_6x_7}\mathtt{T}^{\mathtt{ca}}_{z_1})(\mathfrak{O}_{k_1y_4y_5z_4z_5x_4x_5}\mathtt{T}^{\mathtt{ra}}_{k_1})$
(A4)	$\exists R.A \sqsubseteq B$	$\exists R.A \sqsubseteq B, R(a,b), A(b) \vdash B(a)$	$[\phi^{\mathtt{A4}}(\mathfrak{O})]_{x_1:x_7} = \mathtt{T}^{\mathtt{ca}}_{x_1}(\mathfrak{O}_{y_1y_2y_3y_4y_5x_2x_3}\mathtt{T}^{\mathtt{A4}}_{y_1})(\mathfrak{O}_{z_1y_4y_5z_4z_5x_6x_7}\mathtt{T}^{\mathtt{ca}}_{z_1})(\mathfrak{O}_{k_1y_2y_3x_4x_5z_4z_5}\mathtt{T}^{\mathtt{ra}}_{k_1})$
(A5)	$A \sqsubseteq \exists R.B$	$A \sqsubseteq \exists R.B, A(a) \vdash R(a, o^A_{R,B}), B(o^A_{R,B})$	$[\phi^{\mathtt{A5}}(\mathfrak{O})]_{x_1:x_7} = \mathtt{T}^{\mathtt{ra}}_{x_1}(\mathfrak{O}_{y_1y_2y_3x_2x_3x_6x_7}\mathtt{T}^{\mathtt{A5.1}}_{y_1})(\mathfrak{O}_{z_1y_2y_3x_4x_5z_6z_6}\mathtt{T}^{\mathtt{ca}}_{z_1}) + \mathtt{T}^{\mathtt{ca}}_{x_1}(\mathfrak{O}_{k_1k_2k_3x_2x_3x_4x_5}\mathtt{T}^{\mathtt{A5.2}}_{k_1})(\mathfrak{O}_{l_1k_2k_3l_4l_4x_6x_7}\mathtt{T}^{\mathtt{ca}}_{l_1})$
(A6)	$A \sqsubseteq \leq 1R.B$	$A \sqsubseteq \leq 1R.B, A(a), R(a,b), R(a,c)$ $, B(c) \vdash b \approx c$	$[\phi^{\mathtt{A6}}(\mathfrak{O})]_{x_1:x_7} = \mathtt{T}^{\mathtt{eq}}_{x_1}(\mathfrak{O}_{y_1y_2y_3y_4y_5y_6y_7}\mathtt{T}^{\mathtt{A6}}_{y_1})(\mathfrak{O}_{z_1y_2y_3z_4z_5z_6z_6}\mathtt{T}^{\mathtt{ca}}_{z_1})(\mathfrak{O}_{k_1k_2y_4k_4z_4x_2x_3}\mathtt{T}^{\mathtt{ra}}_{k_1})(\mathfrak{O}_{l_1k_2y_5k_4z_5l_6x_4}\mathtt{T}^{\mathtt{ra}}_{l_1})(\mathfrak{O}_{m_1y_6y_7l_6x_5x_6x_7}\mathtt{T}^{\mathtt{ca}}_{m_1})$
(A7)	$A \sqsubseteq \neg B$	$A \sqsubseteq \neg B, A(a) \vdash \neg B(a)$	$[\phi^{\mathtt{A7}}(\mathfrak{O})]_{x_1:x_7} = \mathtt{T}^{\mathtt{ng}}_{x_1}(\mathfrak{O}_{y_1y_2y_3x_2x_3x_6x_7}\mathtt{T}^{\mathtt{A7}}_{y_1})(\mathfrak{O}_{z_1y_2y_3x_4x_5z_6z_6}\mathtt{T}^{\mathtt{ca}}_{z_1})$
(A8)	$A \sqsubseteq \{a\}$	$A \sqsubseteq \{a\}, A(b) \vdash b \approx a$	$[\phi^{\mathtt{A8}}(\mathfrak{O})]_{x_1:x_7} = \mathtt{T}^{\mathtt{ng}}_{x_1}(\mathfrak{O}_{y_1y_2y_3x_4x_5x_6x_7}\mathtt{T}^{\mathtt{A8}}_{y_1})(\mathfrak{O}_{z_1y_2y_3x_2x_3z_6z_6}\mathtt{T}^{\mathtt{ca}}_{z_1})$
(A9)	$R \sqsubseteq S$	$R \sqsubseteq S, R(a,b) \vdash S(a,b)$	$[\phi^{\mathtt{A9}}(\mathfrak{O})]_{x_1:x_7} = \mathtt{T}^{\mathtt{ra}}_{x_1}(\mathfrak{O}_{y_1y_2y_3x_2x_3y_6y_6}\mathtt{T}^{\mathtt{A9}}_{y_1})(\mathfrak{O}_{z_1y_2y_3x_4x_5x_6x_7}\mathtt{T}^{\mathtt{ra}}_{z_1})$
(A10)	$R \sqsubseteq S^-$	$R \sqsubseteq S^-, R(a,b) \vdash S(b,a)$	$[\phi^{\mathtt{A10}}(\mathfrak{O})]_{x_1:x_7} = \mathtt{T}^{\mathtt{ra}}_{x_1}(\mathfrak{O}_{y_1y_2y_3x_2x_3y_6y_6}\mathtt{T}^{\mathtt{A10}}_{y_1})(\mathfrak{O}_{z_1y_2y_3x_6x_7x_4x_5}\mathtt{T}^{\mathtt{ra}}_{z_1})$
(A11)	$Tra(R)$	$Tra(R), R(a,b), R(b,c) \vdash R(a,c)$	$[\phi^{\mathtt{A11}}(\mathfrak{O})]_{x_1:x_7} = \mathtt{T}^{\mathtt{ra}}_{x_1}(\mathfrak{O}_{y_1y_2x_2y_4y_4y_5y_5}\mathtt{T}^{\mathtt{A11}}_{y_1})(\mathfrak{O}_{z_1z_2y_2x_4x_5z_6z_7}\mathtt{T}^{\mathtt{ra}}_{z_1})(\mathfrak{O}_{k_1z_2x_3z_6z_7x_6x_7}\mathtt{T}^{\mathtt{ra}}_{k_1})$

3 Basic Ideas and Challenges

Basic Ideas. The methods of transforming logical reasoning to a statistic problem cannot lead to a sound and complete reasoning, since these methods can hardly be completely adapted to logic languages from the level of syntaxes. However, one can use tensor products to syntactically encode different logic languages according to [12]. Thus, in this work, we encode ontologies based on tensor products as well, for the purpose of a sound and complete reasoning. In the following, we provide three basic ideas for encoding ontologies by using tensors.

• First, Kronecker products can describe multi-relations [12,6,13]. Entities or terms are usually encoded to order-1 tensors (namely vectors), which are also called *basic tensors*; A logic statement, as a multi-relation among entities and terms, can then be encoded to so called a *statement tensor*, which is a Kronecker product over basic tensors. We illustrate this using the following example.

Example 1. Given an ontology $\mathcal{O}_{ex1}$, it contains an axiom $A \sqsubseteq B$ and a concept assertion $A(a)$. The axiom $A \sqsubseteq B$ can be viewed as a kind of multi-relation (i.e., subsumption) between the two concepts A and B. It can be encoded to a statement tensor $\mathtt{A} \otimes \mathtt{B}$, where $\mathtt{A}$ and $\mathtt{B}$ are basic tensors uniquely corresponding to A and B respectively.[4] Similarly, $A(a)$ can be encoded to the statement tensor $\mathtt{A} \otimes \mathtt{a}$ where the basic tensor $\mathtt{a}$ corresponds to a. A consequence $B(a)$ can be derived by applying Rule (A1) (see Table 1) over $A \sqsubseteq B$ and $A(a)$ where A is a joint. Further, $B(a)$ corresponds to the statement tensor $\mathtt{B} \otimes \mathtt{a}$.

• Second, a *set* can be described using tensor additions [13]. Since an ontology is actually a set of axioms and assertions, it can be encoded to a tensor that is a summation over all statement tensors, In Example 1, the ontology $\mathcal{O}_{ex1}$ can be encoded to the tensor $\mathfrak{O}_{ex1} = \mathtt{A} \otimes \mathtt{B} + \mathtt{A} \otimes \mathtt{a}$.

• Third, the premises in a rule can be matched by performing inner products on the IPO pairs that point to joints [7,9,13]. We call these inner products *joint-based products*. Consider Example 1 again. The application of Rule (A1) can be performed by using the inner product $(\mathtt{A} \otimes \mathtt{B}) \bullet_{(1,1)} (\mathtt{A} \otimes \mathtt{a})$, where the IPO pair $(1,1)$ points to the basic tensor $\mathtt{A}$. We also say that the basic tensor $\mathtt{A}$ is a *joint* in this joint-based product. The correctness of joint-based products is guaranteed by *the orthogonality condition* [13], which says that all basic tensors should be *orthogonal* mutually, i.e., for any two basic tensors $\mathtt{U}$ and $\mathtt{V}$, if $\mathtt{U} = \mathtt{V}$, then $\mathtt{U} \bullet \mathtt{V} = 1$, otherwise $\mathtt{U} \bullet \mathtt{V} = 0$. If the basic tensors $\mathtt{A}$, $\mathtt{B}$ and $\mathtt{a}$ satisfy the orthogonality condition, then it can be checked that $(\mathtt{A} \otimes \mathtt{B}) \bullet_{(1,1)} (\mathtt{A} \otimes \mathtt{a})$ equals to $\mathtt{B} \otimes \mathtt{a}$ that is actually the statement tensor of the consequence $B(a)$. Note that, if basic tensors are of order-1, a joint-based product would eliminate the joint due to the definition of inner products, and construct the consequence through Kronecker products over the non-eliminated basic tensors (e.g., $\mathtt{B}$ and $\mathtt{a}$ in the previous example).

Challenges. We follow the above ideas to undertake this work. However, we have to face the following challenges.

• Joint-based products can hardly work for visible joints if basic tensors are of order-1. Consider Rule (A2) in Table 1. The individual a is a visible joint between the premises $A_1(a)$ and $A_2(a)$ when applying this rule, since a occurs in the consequence $B(a)$. Suppose, $\mathtt{a}, \mathtt{A}_1, \mathtt{A}_2$ and $\mathtt{B}$ are the basic tensors corresponding to a, A_1, A_2 and B respectively; $\mathtt{A}_1 \otimes \mathtt{a}$ and $\mathtt{A}_2 \otimes \mathtt{a}$ are the statement tensors of $A_1(a)$ and $A_2(a)$ respectively. If $\mathtt{a}$ is of order-1, then a joint-based product between $\mathtt{A}_1 \otimes \mathtt{a}$ and $\mathtt{A}_2 \otimes \mathtt{a}$ would eliminate $\mathtt{a}$. In this case, $\mathtt{B} \otimes \mathtt{a}$ cannot be obtained

[4] The basic tensors $\mathtt{A}$ and $\mathtt{B}$ are randomly mapped to A and B respectively from the vector-space.

by any tensor operation. Thus, the first challenge is how to adapt joint-based products to the cases where visible joints occur.

• The second challenge appears when handling Rule ($\mathtt{A5}$). Rule ($\mathtt{A5}$) is a *Skolemized* rule i.e., the individual $o^A_{R,B}$, also called a *special individual*, is a fixed individual with respect to the axiom of the form $A \sqsubseteq \exists R.B$. Since $o^A_{R,B}$ does not occur in premises and, further, joint-based products cannot introduce new basic tensors, it is difficult to directly use a joint-based product to handle the application of Rule ($\mathtt{A5}$).

• Current work on representing logic rules using tensors does not aim to study a specific logic language that has formal definitions. Further, there is no theoretical analysis, i.e., the proofs of soundness and completeness of reasoning based on tensor operations. However, Horn-$\mathcal{SHOIQ}$ is a well-defined logic language. Thus, we should give the theoretical results of the soundness and completeness of materialization based on tensor operations.

In following paragraphs, we discuss the above issues and give our methods and theoretical results.

4 A Tensor-based Representation for Ontologies

In this section, we discuss how to encode Horn-$\mathcal{SHOIQ}$ ontologies to tensors in the vector-space $\mathbb{R}^d$.

Encoding Concepts, Roles and Individuals. As discussed in the previous section, if basic tensors are of order-1, then joint-based products can hardly work for the cases where visible joints occur. This is because a joint-based product would eliminate joints.

To address the above issue, an intuitive idea is to let joints be preserved after performing a joint-based product. To this end, we consider using *symmetric order-2* tensors [8]. A symmetric order-2 tensor can be defined as a self Kronecker product over an order-1 tensor, i.e., $\mathtt{V} \otimes \mathtt{V}$ where $\mathtt{V}$ is an order-1 tensor. The reason why we choose symmetric order-2 tensors is that they have an important property: if the orthogonality condition is satisfied, the result of an inner product over a symmetric order-2 tensor on any IPO pair is still this tensor according to the definition of inner product, i.e., one can check that $(\mathtt{V} \otimes \mathtt{V}) \bullet_{(n,m)} (\mathtt{V} \otimes \mathtt{V}) = (\mathtt{V} \otimes \mathtt{V})$ where $n, m \in \{1, 2\}$. In this way, when applying Rule ($\mathtt{A2}$) by tensor operations, if the joint $\mathtt{a}$ is a symmetric order-2 tensor, $\mathtt{a}$ can be preserved after performing a joint-based product between $\mathtt{A}_1 \otimes \mathtt{a}$ and $\mathtt{A}_2 \otimes \mathtt{a}$. For other rules in Table 1, we discuss how symmetric order-2 tensors work in the next section.

In this work, we encode all atomic concepts, atomic roles and individuals to symmetric order-2 tensors. Specifically, given a set of vectors $\mathbb{V} \subseteq \mathbb{R}^d$ (we call it a *basic vector set*), each atomic concept, atomic role and individual occurring in $\mathcal{O}$ is mapped to a symmetric order-2 tensor $\mathtt{V} \otimes \mathtt{V}$ where $\mathtt{V} \in \mathbb{V}$ is uniquely corresponding to this concept, role or individual. *In what follows, we denote 'basic tensors' to such symmetric order-2 tensors.* We use the same letter in bold and Teletype font for representing the mapped tensors. For example, an atomic concept A (resp., an atomic role R, an individual a) is mapped to a basic tensor $\mathtt{A}$ (resp., $\mathtt{R}$, $\mathtt{a}$).

Encoding Axioms and Assertions. We follow the first idea (see the previous section) to encode logic statements (i.e., axioms and assertions in our case) to statement tensors. According to the second idea, we can further encode an ontology by performing a tensor addition over all statement tensors. However, to perform the tensor addition, all statement tensors should have the same order. To this end, we give a transformation function τ over all axioms and assertions in a given ontology $\mathcal{O}$ (see Table 2).

Table 2: The Transformation Function τ

$\tau(A \sqsubseteq B)$	$=$	$\mathtt{T}^{\mathtt{A1}} \otimes \mathtt{A} \otimes \mathtt{B} \otimes \boldsymbol{\theta}$	(1)
$\tau(A_1 \sqcap A_2 \sqsubseteq B)$	$=$	$\mathtt{T}^{\mathtt{A2}} \otimes \mathtt{A}_1 \otimes \mathtt{A}_2 \otimes \mathtt{B}$	(2)
$\tau(A \sqsubseteq \forall R.B)$	$=$	$\mathtt{T}^{\mathtt{A3}} \otimes \mathtt{A} \otimes \mathtt{R} \otimes \mathtt{B}$	(3)
$\tau(\exists R.A \sqsubseteq B)$	$=$	$\mathtt{T}^{\mathtt{A4}} \otimes \mathtt{R} \otimes \mathtt{A} \otimes \mathtt{B}$	(4)
$\tau(A \sqsubseteq \exists R.B)$	$=$	$\mathtt{T}^{\mathtt{A5}} \otimes \mathtt{A} \otimes \mathtt{R} \otimes \mathtt{B}$	(5)
$\tau(A \sqsubseteq \leq 1R.B)$	$=$	$\mathtt{T}^{\mathtt{A6}} \otimes \mathtt{A} \otimes \mathtt{R} \otimes \mathtt{B}$	(6)
$\tau(A \sqsubseteq \neg B)$	$=$	$\mathtt{T}^{\mathtt{A7}} \otimes \mathtt{A} \otimes \mathtt{B} \otimes \boldsymbol{\theta}$	(7)
$\tau(A \sqsubseteq \{a\})$	$=$	$\mathtt{T}^{\mathtt{A8}} \otimes \mathtt{A} \otimes \mathtt{a} \otimes \boldsymbol{\theta}$	(8)
$\tau(R \sqsubseteq S)$	$=$	$\mathtt{T}^{\mathtt{A9}} \otimes \mathtt{R} \otimes \mathtt{S} \otimes \boldsymbol{\theta}$	(9)
$\tau(R \sqsubseteq S^-)$	$=$	$\mathtt{T}^{\mathtt{A10}} \otimes \mathtt{R} \otimes \mathtt{S} \otimes \boldsymbol{\theta}$	(10)
$\tau(\mathtt{Tra}(R))$	$=$	$\mathtt{T}^{\mathtt{A11}} \otimes \mathtt{R} \otimes \boldsymbol{\theta} \otimes \boldsymbol{\theta}$	(11)
$\tau(A(a))$	$=$	$\mathtt{T}^{\mathtt{ca}} \otimes \mathtt{A} \otimes \mathtt{a} \otimes \boldsymbol{\theta}$	(12)
$\tau(R(a,b))$	$=$	$\mathtt{T}^{\mathtt{ra}} \otimes \mathtt{R} \otimes \mathtt{a} \otimes \mathtt{b}$	(13)
$\tau(a \approx b)$	$=$	$\mathtt{T}^{\mathtt{eq}} \otimes \mathtt{a} \otimes \mathtt{b} \otimes \boldsymbol{\theta}$	(14)

We use the tensors $\mathtt{T}^{\varphi}$ ($\varphi \in \{\mathtt{A1}, .., \mathtt{A11}\}$), $\mathtt{T}^{\mathtt{ca}}$, $\mathtt{T}^{\mathtt{ra}}$ and $\mathtt{T}^{\mathtt{eq}}$ for indicating the type of a statement tensor. For example, $\mathtt{T}^{\mathtt{A1}}$ indicates that a statement tensor corresponds to an axiom of the form $A \sqsubseteq B$. We call these tensors *type tensors*. We also say that a statement tensor $\mathtt{S}$ is of type $\mathtt{T}^{\varphi}$ if $\mathtt{T}^{\varphi}$ occurs as the first Kronecker term in $\mathtt{S}$, where $\varphi \in \{\mathtt{A1}, ..., \mathtt{A11}\} \cup \{\mathtt{ca}, \mathtt{ra}, \mathtt{eq}\}$. Consider the equation (2). An axiom of the form $A_1 \sqcap A_2 \sqsubseteq B$ is mapped to an order-7 tensor (one type tensor and three symmetric order-2 tensors) where three basic tensors, i.e., $\mathtt{A}_1, \mathtt{A}_2$ and $\mathtt{B}$, are needed for encoding the three concepts A_1, A_2 and B respectively. The statement tensors transformed from the axioms or assertions of the forms $A \sqsubseteq \forall R.B$, $A \sqsubseteq \exists R.B$, $\exists R.A \sqsubseteq B$, $A \sqsubseteq \leq 1R.B$ and $R(a, b)$ have orders of 7 as well (see the equations (3-6,13)). For axioms or assertions of the forms $A \sqsubseteq B$, $A \sqsubseteq \neg B$, $A \sqsubseteq \{a\}$, $R \sqsubseteq S$, $R \sqsubseteq S^-$, $\mathtt{Tra}(R)$, $A(a)$ and $a \approx b$, they can be transformed to tensors of lower orders, since we can use less than three basic tensors to encode these axioms or assertions. Thus, we introduce a new symmetric order-2 tensor $\boldsymbol{\theta} = \theta \otimes \theta$, where $\theta \in \mathbb{V}$, to make the final transformed statement tensors be at the same order (i.e., order-7, see the equations (1,7-12,14)). We use the following example to show how τ works.

Example 2. Given an ontology $\mathcal{O}_{ex2}$, it contains the axioms $A_1 \sqcap A_2 \sqsubseteq B$, $B \sqsubseteq \forall S.C$, $C \sqsubseteq \exists S.A, R \sqsubseteq S$ and the assertions $R(a, b), A_1(a), A_2(a)$. We now perform the transformation function τ on $\mathcal{O}_{ex2}$. Specifically, the axiom $A_1 \sqcap A_2 \sqsubseteq B$ is transformed to the tensor $\mathtt{T}_1 = \mathtt{T}^{\mathtt{A2}} \otimes \mathtt{A}_1 \otimes \mathtt{A}_2 \otimes \mathtt{B}$. The axiom $B \sqsubseteq \forall S.C$ is transformed to the tensor $\mathtt{T}_2 = \mathtt{T}^{\mathtt{A3}} \otimes \mathtt{B} \otimes \mathtt{S} \otimes \mathtt{C}$. The axiom $C \sqsubseteq \exists S.A$ is transformed to the tensor $\mathtt{T}_3 = \mathtt{T}^{\mathtt{A5}} \otimes \mathtt{C} \otimes \mathtt{S} \otimes \mathtt{A}$. It is also easy to transform the role inclusion axioms and the assertions to statement tensors. They are $T_4 = T^{A9} \otimes R \otimes S \otimes \theta$, $T_5 = T^{ra} \otimes R \otimes a \otimes b$, $T_6 = T^{ca} \otimes A_1 \otimes a \otimes \theta$, $T_7 = T^{ca} \otimes A_2 \otimes a \otimes \theta$.

Encoding Ontologies. Based on the above encoding method, a given ontology $\mathcal{O}$ can be transformed to an order-7 tensor as well. Specifically, we use $\mathfrak{O}$ to denote the corresponding tensor of $\mathcal{O}$, and let $\mathfrak{O} = \sum_\alpha \tau(\alpha)$ where α denotes each axiom and assertion in $\mathcal{O}$. We say that $\mathfrak{O}$ is an *ontology tensor* of $\mathcal{O}$. For the ontology $\mathcal{O}_{ex2}$ in Example 2, its corresponding ontology tensor (denoted by $\mathfrak{O}_{ex2}$) is $\mathfrak{O}_{ex2} = \mathtt{T}_1 + \mathtt{T}_2 + .. + \mathtt{T}_7$. One can check that, under the orthogonality condition, if a statement tensor $\mathtt{S}$ is additive in an ontology tensor $\mathfrak{O}$, then $\mathfrak{O} \bullet \mathtt{S} = 1$, otherwise $\mathfrak{O} \bullet \mathtt{S} = 0$. Thus, we can give the following theorem to show the correctness of our encoding method.

Theorem 1. *Given a Horn-$\mathcal{SHOIQ}$ ontology $\mathcal{O}$, $\mathfrak{O}$ is the ontology tensor of $\mathcal{O}$ based on a basic vector set $\mathbb{V}$ and the transformation function τ. We have that, for each axiom or assertion α, if $\mathbb{V}$ satisfies the orthogonality condition, then $\mathfrak{O} \bullet \tau(\alpha) = 1$ iff $\alpha \in \mathcal{O}$ and $\mathfrak{O} \bullet \tau(\alpha) = 0$ iff $\alpha \notin \mathcal{O}$.*

5 Materialization via Tensor Operations

Based on the above encoding method, we discuss in this section how to handle materialization by tensor operations, i.e., how to handle the rules in Table 1 by joint-based products.

Decomposition and Joint Eliminating Inner Products. We first consider Rule ($\mathtt{A1}$). This rule has two premises, i.e., $A \sqsubseteq B$ and $A(a)$. Intuitively, for handling this rule via tensor operations, we can just focus on the statement tensors of types $\mathtt{T}^{\mathtt{A1}}$ and $\mathtt{T}^{\mathtt{ca}}$. On the other hand, an ontology tensor $\mathfrak{O}$ is a summation of statement tensors of all types according to the encoding method given in the previous section. Thus, we consider first *decomposing* $\mathfrak{O}$ to get all tensors of types $\mathtt{T}^{\mathtt{A1}}$ and $\mathtt{T}^{\mathtt{ca}}$, and then, constructing new statement tensors of type $\mathtt{T}^{\mathtt{ca}}$. The definition of decomposition is given as follows.

Definition 1. (**Decomposition**) *Given an ontology $\mathcal{O}$, $\mathfrak{O}$ is an ontology tensor transformed from $\mathcal{O}$ by the transformation function τ. A $\mathtt{T}^\varphi$-decomposition is the tensor $\mathfrak{O} \bullet_{(1,1)} \mathtt{T}^\varphi$, denoted by $\mathtt{D}^\varphi$, where $\varphi \in \{Ai | i \in \{1, .., 11\}\} \cup \{\mathtt{ca}, \mathtt{ra}, \mathtt{eq}\}$.*

A $\mathtt{T}^\varphi$-decomposition $\mathtt{D}^\varphi$ is an order-6 tensor that is a summation of all tensors in the form of $\mathtt{T}_1 \otimes \mathtt{T}_2 \otimes \mathtt{T}_3$, which are actually statement tensors without including type tensors as the first Kronecker terms. In other words, we can use a $\mathtt{T}^\varphi$-decomposition to get all tensors of type $\mathtt{T}^\varphi$. Back to Rule ($\mathtt{A1}$). We can first get a $\mathtt{T}^{\mathtt{A1}}$-decomposition $\mathtt{D}^{\mathtt{A1}}$ and a $\mathtt{T}^{\mathtt{ca}}$-decomposition $\mathtt{D}^{\mathtt{ca}}$, and then perform a joint-based product between $\mathtt{D}^{\mathtt{A1}}$ and $\mathtt{D}^{\mathtt{ca}}$ with respect to the joint. Since symmetric order-2 tensors are used as basic tensors, we define a new kind of joint-based product as follows.

Definition 2. (**Joint Eliminating Inner (JEI) Products**) *Given two decompositions $\mathtt{D}^\varphi$ and $\mathtt{D}^\psi$, an (n,m)-JEI product between $\mathtt{D}^\varphi$ and $\mathtt{D}^\psi$ is defined as $\mathtt{D}^\varphi \bullet_{(2n-1,2m-1)(2n,2m)} \mathtt{D}^\psi$, where $n, m \in \{1, 2, 3\}$.*

An (n,m)-JEI product is an inner product performed between two decompositions $\mathtt{D}^{\varphi}$ and $\mathtt{D}^{\psi}$, where n and m indicate the positions of joints, i.e., the n^{th} (resp., m^{th}) Kronecker terms of the basic tensors additive in $\mathtt{D}^{\varphi}$ (resp., $\mathtt{D}^{\psi}$) are joints. Further, the joints in an JEI product would be eliminated, since the inner product in Definition 2 is performed on two IPO pairs (i.e., $(2n-1, 2m-1), (2n, 2m)$) that correspond to the positions of the joints. In the case of Rule (A1), since the joint is an invisible joint, it should be eliminated. Thus, we perform a (1,1)-JEI product between $\mathtt{D}^{\mathtt{A1}}$ and $\mathtt{D}^{\mathtt{ca}}$. Formally, we give an operation (denoted by $\phi^{\mathtt{A1}}$) over the ontology $\mathfrak{O}$. The formulation of each element in $\phi^{\mathtt{A1}}(\mathfrak{O})$ can be found at the fourth column and the first row of Table 1. In this equation, the contents in the first (resp., second) bracket is an element in $\mathtt{D}^{\mathtt{A1}}$ (resp., $\mathtt{D}^{\mathtt{ca}}$). The (1,1)-JEI product between $\mathtt{D}^{\mathtt{A1}}$ and $\mathtt{D}^{\mathtt{ca}}$ can be checked through the shared subscripts y_2 and y_3. Finally, we add the type tensor $\mathtt{T}^{\mathtt{ca}}$ as the first Kronecker term.

Since all of the joints in Rule (A3), Rule (A4), Rule (A7), Rule(A8), Rule (A9) and Rule (A10) are invisible joints, these rules can all be handled by JEI products. Similar to the operation $\phi^{\mathtt{A1}}$, we define other six operations for these rules. They are denoted by $\phi^{\mathtt{A3}}$, $\phi^{\mathtt{A4}}$, $\phi^{\mathtt{A7}}$, $\phi^{\mathtt{A8}}$, $\phi^{\mathtt{A9}}$ and $\phi^{\mathtt{A10}}$ respectively. One can give the formulations of these operations by following the idea of $\phi^{\mathtt{A1}}$. Thus, we do not discuss them in detail here. The formulations of each element after performing these operations can be found at Table 1.

Joint Preserving Inner Products. We now consider Rule (A2) in Table 1, where A_1 and A_2 are two invisible joints and a is a visible joint. For handling the invisible joint A_1 (resp., A_2), a (1,1)-JEI product (resp., (2,1)-JEI product) is performed between the $\mathtt{T}^{\mathtt{A2}}$-decomposition $\mathtt{D}^{\mathtt{A2}}$ and a $\mathtt{T}^{\mathtt{ca}}$-decomposition $\mathtt{D}_1^{\mathtt{ca}}$ (resp., another $\mathtt{T}^{\mathtt{ca}}$-decomposition $\mathtt{D}_2^{\mathtt{ca}}$). Recall the property of symmetric order-2 tensors (see the previous section). In order to make the visible joint a be preserved in the final results, we give another kind of joint-based product as follows.

Definition 3. (**Joint Preserving Inner (JPI) Products**) *Given two decompositions $\mathtt{D}^{\varphi}$ and $\mathtt{D}^{\psi}$, an (n,m)-JPI product between $\mathtt{D}^{\varphi}$ and $\mathtt{D}^{\psi}$ is defined as $\mathtt{D}^{\varphi} \bullet_{(2n-1,2m-1)} \mathtt{D}^{\psi}$, where $n, m \in \{1, 2, 3\}$.*

Similar to an (n,m)-JEI product, an (n,m)-JPI product is also an inner product performed between two decompositions $\mathtt{D}^{\varphi}$ and $\mathtt{D}^{\psi}$, where n and m indicate the positions of joints. The difference lies in that the joint in an (n,m)-JPI product would be preserved, since the inner product in Definition 3 is performed on only one IPO pair $(2n-1, 2m-1)$. In the case of Rule (A2), we actually perform an (2,2)-JPI product between $\mathtt{D}_1^{\mathtt{ca}}$ and $\mathtt{D}_2^{\mathtt{ca}}$. Correspondingly, we give an operation, denoted by $\phi^{\mathtt{A2}}$, over the ontology tensor $\mathfrak{O}$ (see Table 1) to handle Rule (A2).

Both Rule (A6) and Rule (A11) contain visible joints, i.e., the joint c in Rule (A6) and the joint R in Rule (A11) (see Table 1). These two rules can thus be handled using JPI products by following the idea of handling Rule (A2). We define two operations $\phi^{\mathtt{A6}}$ and $\phi^{\mathtt{A11}}$ for handling Rule (A6) and Rule (A11) respectively.

The formulations of each element after performing these two operations can be found at Table 1.

Handling Rule (A5). Finally, we consider Rule (A5). It is difficult to directly use JEI products or JPI products to handle Rule (A5). This is because that, the special individual ($o^A_{R,B}$ in Table 1) does not occur in premises, while any kind of tensor operation cannot introduce new basic tensors. To address this problem, we consider encoding special individuals to statement tensors in advance. For example, for each axiom of the form $A \sqsubseteq \exists R.B$, we can encode it to a statement tensor $\mathtt{T}^{\mathtt{A5}} \otimes \mathtt{A} \otimes \mathtt{R} \otimes \mathtt{B} \otimes \mathtt{o}^A_{R,B}$, where $\mathtt{o}^A_{R,B}$ is a newly introduced basic tensor for encoding the special individual $o^A_{R,B}$. However, this kind of statement tensor is not of order-7. In order to keep the transformed statement tensors be at order-7, we can encode each axiom of the form $A \sqsubseteq \exists R.B$ to two statement tensors: $\mathtt{T}^{\mathtt{A5.1}} \otimes \mathtt{A} \otimes \mathtt{R} \otimes \mathtt{o}^A_{R,B}$ and $\mathtt{T}^{\mathtt{A5.2}} \otimes \mathtt{A} \otimes \mathtt{B} \otimes \mathtt{o}^A_{R,B}$, where $\mathtt{T}^{\mathtt{A5.1}}$ and $\mathtt{T}^{\mathtt{A5.2}}$ are two newly introduced type tensors. In this way, the equation (5) in Table 2 is replaced by the following one:

$$\boxed{\begin{aligned}\tau(A \sqsubseteq \exists R.B) = \mathtt{T}^{\mathtt{A5.1}} &\otimes \mathtt{A} \otimes \mathtt{R} \otimes \mathtt{o}^A_{R,B} \\ &+ \mathtt{T}^{\mathtt{A5.2}} \otimes \mathtt{A} \otimes \mathtt{B} \otimes \mathtt{o}^A_{R,B} \quad (5)\end{aligned}}$$

We give an operation, denoted by $\phi^{\mathtt{A5}}$, over the ontology tensor $\mathfrak{O}$ for handling Rule (A5). The formulation of each element $\phi^{\mathtt{A5}}(\mathfrak{O})$ is shown in Table 1. In the first addend of that formulation, the contents in the first bracket denotes an element in a $\mathtt{T}^{\mathtt{A5.1}}$-decomposition $\mathtt{D}^{\mathtt{A5.1}}$, while the contents in the second bracket denotes an element in a $\mathtt{T}^{\mathtt{ca}}$-decomposition $\mathtt{D}^{\mathtt{ca}}$. A (1,1)-JEI product is performed between $\mathtt{D}^{\mathtt{A5.1}}$ and $\mathtt{D}^{\mathtt{ca}}$ (see the shared subscripts y_2 and y_3). In the second addend of the above equation, the contents in the first bracket denotes an element in a $\mathtt{T}^{\mathtt{A5.2}}$-decomposition $\mathtt{D}^{\mathtt{A5.2}}$, and the contents in the second bracket denotes an element in a $\mathtt{T}^{\mathtt{ra}}$-decomposition $\mathtt{D}^{\mathtt{ra}}$. Similarly, a (1,1)-JEI product is performed between $\mathtt{D}^{\mathtt{A5.2}}$ and $\mathtt{D}^{\mathtt{ra}}$ (see the shared subscripts k_2 and k_3).

Complete Materialization. In the previous paragraphs, we give eleven operations that handle the rules in Table 1 respectively. It should be noted that assertions in $\mathtt{mat}(\mathcal{O})$ are obtained by applying the rules in Table 1 iteratively until no more consequence can be obtained. Thus, we should also apply all the eleven operations iteratively to conduct a complete materialization. To this end, we define a new operation $\Phi(\mathfrak{O}) = \Sigma_\varphi \phi^\varphi(\mathfrak{O})$ where $\varphi \in \{\mathtt{A1}, ..., \mathtt{A11}\}$. Further, we give the following operations: $\Phi^0(\mathfrak{O}) = \mathfrak{O}$, $\Phi^{i+1}(\mathfrak{O}) = \Phi(\Phi^i(\mathfrak{O})) + \Phi^i(\mathfrak{O})$, where i is an integer and $i \geq 1$. The first addend of the operation $\Phi^{i+1}(\mathfrak{O})$ is to apply Φ once over $\Phi^i(\mathfrak{O})$, which can be viewed as an updated ontology tensor with new statement tensors computed from $\Phi^{i-1}(\mathfrak{O})$. To include old statement tensors, the addend $\Phi^i(\mathfrak{O})$ has to be added in $\Phi^{i+1}(\mathfrak{O})$.

Example 3. Consider Example 2 again. We apply Φ over the ontology tensor $\mathfrak{O}_{ex2}$. In $\Phi^1(\mathfrak{O}_{ex2})$, two new statement tensors $T_8 = T^{\mathtt{ca}} \otimes B \otimes a \otimes \theta$ and $T_9 = T^{\mathtt{ra}} \otimes S \otimes a \otimes b$ are obtained by applying $\phi^{\mathtt{A2}}$ and $\phi^{\mathtt{A9}}$ respectively. In $\Phi^2(\mathfrak{O}_{ex2})$, a new statement tensor $T_{10} = T^{\mathtt{ca}} \otimes C \otimes b \otimes \theta$ is obtained by applying $\phi^{\mathtt{A3}}$. In $\Phi^3(\mathfrak{O}_{ex2})$, two new statement tensors $T_8 = T^{\mathtt{ca}} \otimes A \otimes o^C_{S,A} \otimes \theta$ and $T_9 = T^{\mathtt{ra}} \otimes S \otimes b \otimes o^C_{S,A}$ are obtained by applying $\phi^{\mathtt{A5}}$. One can check that the result of $\Phi^3(\mathfrak{O}_{ex2})$ corresponds to the results of a complete materialization of $\mathcal{O}_{ex2}$.

We use Theorem 2 to show the soundness and completeness of Φ.

Theorem 2. *Given an ontology $\mathcal{O}$, $\mathfrak{O}$ is the corresponding ontology tensor of $\mathcal{O}$ based on a basic vector set $\mathbb{V}$ and the transformation function τ. The basic vector set $\mathbb{V}$ satisfies the orthogonality condition. For any integer $i \geq 0$ and any assertion α, we have that:*

(*Soundness*) $\Phi^i(\mathfrak{O}) \bullet \tau(\alpha) \geq 1$ *implies* $\alpha \in \mathtt{mat}(\mathcal{O})$;[5]

(*Completeness*) *there exists an integer N such that if $\alpha \in \mathtt{mat}(\mathcal{O})$ then $\Phi^N(\mathfrak{O}) \bullet \tau(\alpha) \geq 1$.*

Discussion. From the perspective of implementation, the operation Φ may suffer from large amount of computation for processing high order tensors, i.e., tensors of order-7, since we use symmetric order-2 tensors as basic tensors. To make implementations more efficient in practice, we can restrict that the basic vector set only contains *one-hot vectors* (in these vectors, one element is 1, all other elements are 0). In this way, we can use the techniques of sparse tensors for the guarantee of efficiency. On the other hand, we can use the techniques of approximate reasoning [10] to avoid the cases where visible joints appear. We can also identify a fragment of Horn-$\mathcal{SHOIQ}$ where the axioms of the forms (A2), (A6) and (A11) cannot be stated. This fragment has an enough expressivity that covers OWL 2 QL ontologies and RDFS ontologies without including transitive roles. For encoding ontologies expressed in this fragment, we can use vectors as basic tensors. The order of statement tensors can then be reduced to 4; the amount of computation is decreased by d^3 times in theory.

6 Related Work

There are mainly two lines of work on handling logic rules by tensors as mentioned in the first section. In the first line, logic reasoning is transformed to statistic problems, i.e., a maximization problem [7,11], an integer linear programming problem [15], the problem of tensor decomposition [5]. These methods can hardly handle logic languages with complex syntaxes. In another line of the work, tensor products are used to syntactically encode different logic languages. The authors of [12] and [6] study how to use tensor products to represent logic languages. However, their methods can hardly be generalized to our case, since they do not study how to use the encoding methods to handle logic reasoning. The authors of the work [9,13] investigate a kind of reasoning task (query answering for Facebook bAbI tasks), and propose to transform it to tensor products based on work of [12]. Due to the special purpose of the bAbI tasks, the reasoning task is relatively simple compared to ontology materialization. In summary, the above methods cannot be directly applied in our work, e.g., they do not work for the cases where visible joints occur. Moreover, there is no theoretical analysis in the related works, i.e., proofs for soundness and completeness of logical reasoning based tensor operations.

[5] Here, we use '≥ 1' to include the situations where redundant occurs. Specifically, if $\tau(\alpha)$ occurs as an addend in $\Phi^i(\mathfrak{O})$ more than once, it can be checked that $\Phi^i(\mathfrak{O}) \bullet \tau(\alpha) \geq 1$.

7 Conclusions and Future Work

In this paper, we provided a novel method to encode a Horn-$\mathcal{SHOIQ}$ ontology to tensors by using symmetric order-2 tensors. The new encoding method avoids the problems caused by handing visible joints and Skolemized rules. We further showed the correctness of the encoding method. We then presented a method to perform ontology materialization by using tensor operations. We showed the soundness and completeness of the proposed method. To make our work more practical, we identify a fragment of Horn-$\mathcal{SHOIQ}$ that can be represented by order-4 tensors. As one future work, we will implement a system for ontology materialization based on high-performance tensor-based platforms. As another future work, we consider incorporating the techniques of knowledge completion with our methods. Specifically, the 11 operations $\phi^{\mathtt{A1}}...\phi^{\mathtt{A11}}$ can be formulated as constraints in a linear programming problem by following the idea given in [15]. This would improve the accuracy for knowledge completion.

References

1. F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, 2003.
2. A. Bordes, N. Usunier, A. García-Durán, J. Weston, and O. Yakhnenko. Translating embeddings for modeling multi-relational data. In *Proc. of NIPS*, pages 2787–2795, 2013.
3. A. Bordes, J. Weston, R. Collobert, and Y. Bengio. Learning structured embeddings of knowledge bases. In *Proc. of AAAI*, 2011.
4. D. Carral, C. Feier, B. C. Grau, P. Hitzler, and I. Horrocks. Pushing the boundaries of tractable ontology reasoning. In *Proc. of ISWC*, pages 148–163, 2014.
5. K. Chang, W. Yih, B. Yang, and C. Meek. Typed tensor decomposition of knowledge bases for relation extraction. In *Proc. of EMNLP*, pages 1568–1579, 2014.
6. S. Clark and S. Pulman. Combining symbolic and distributional models of meaning. In *Proc. of AAAI Spring Symposium*, pages 52–55, 2007.
7. E. Grefenstette. Towards a formal distributional semantics: Simulating logical calculi with tensors. *J. CoRR*, 2013.
8. T. G. Kolda and B. W. Bader. *Tensor Decompositions and Applications*. Sandia National Laboratories, 2007.
9. M. Lee, X. He, W. Yih, J. Gao, L. Deng, and P. Smolensky. Reasoning in vector space: An exploratory study of question answering. *J. CoRR*, 2015.
10. J. Z. Pan, Y. Ren, and Y. Zhao. Tractable approximate deduction for OWL. *J. Artif. Intell.*, pages 95–155, 2016.
11. T. Rocktäschel, M. Bosnjak, S. Singh, and S. Riedel. Low-dimensional embeddings of logic. In *Proc. of ACL Workshop on Semantic Parsing*, pages 45–49, 2014.
12. P. Smolensky. Tensor product variable binding and the representation of symbolic structures in connectionist systems. *J. Artif. Intell.*, 46(1-2):159–216, 1990.
13. P. Smolensky, M. Lee, X. He, W. Yih, J. Gao, and L. Deng. Basic reasoning with tensor product representations. *J. CoRR*, 2016.
14. R. Socher, D. Chen, C. D. Manning, and A. Y. Ng. Reasoning with neural tensor networks for knowledge base completion. In *Proc. of NIPS*, pages 926–934, 2013.
15. Q. Wang, B. Wang, and L. Guo. Knowledge base completion using embeddings and rules. In *Proc. of IJCAI*, pages 1859–1866, 2015.

Appendix

A Proof of Theorem 1

Theorem 1 *Given a Horn-$\mathcal{SHOIQ}$ ontology $\mathcal{O}$, $\mathfrak{O}$ is the ontology tensor of $\mathcal{O}$ based on a basic vector set $\mathbb{V}$ and the transformation function τ. We have that, for each axiom or assertion α, if $\mathbb{V}$ satisfies the orthogonality condition, then $\mathfrak{O} \bullet \tau(\alpha) = 1$ iff $\alpha \in \mathcal{O}$ and $\mathfrak{O} \bullet \tau(\alpha) = 0$ iff $\alpha \notin \mathcal{O}$.*

Proof: This theorem can be easily proved by considering Lemma 1. □

Lemma 1 *Given a Horn-$\mathcal{SHOIQ}$ ontology $\mathcal{O}$, $\mathfrak{O}$ is the ontology tensor of $\mathcal{O}$ based on a basic vector set $\mathbb{V}$ and the transformation function τ. We have that, for any statement tensor $\mathtt{S}$, if $\mathbb{V}$ satisfies the orthogonality condition, then $\mathfrak{O} \bullet \mathtt{S} \geq 1$ iff $\mathtt{S}$ is additive in $\mathfrak{O}$, and $\mathfrak{O} \bullet \mathtt{S} = 0$ iff $\mathtt{S}$ is nonadditive in $\mathfrak{O}$.*

Proof: According to the definition of $\mathfrak{O}$, we rewrite $\mathfrak{O}$ by the formulation $\mathfrak{O} = \sum_i \mathtt{S}_i$, where $\mathtt{S}_i$ is a statement tensor transformed from some axiom or assertion in $\mathcal{O}$. In this way, $\mathfrak{O} \bullet \mathtt{S}$ can be expanded as follows:

$$\begin{aligned}\mathfrak{O} \bullet \mathtt{S} &= [\textstyle\sum_i \mathtt{S}_i]_{x_1:x_7}[\mathtt{S}]_{x_1:x_7} \\ &= \textstyle\sum_i [\mathtt{S}_i]_{x_1:x_7}[\mathtt{S}]_{x_1:x_7} \\ &= \textstyle\sum_i \mathtt{S}_i \bullet \mathtt{S}\end{aligned}$$

Since $\mathbb{V}$ satisfies the orthogonality condition, we can check that, if for some i, $\mathtt{S}_i \neq \mathtt{S}$, then $\mathtt{S}_i \bullet \mathtt{S} = 0$; otherwise, $\mathtt{S}_i \bullet \mathtt{S} = 1$. In this sense, $\mathfrak{O} \bullet \mathtt{S} \geq 1$ implies that there exists one or more than one $\mathtt{S}_i$ that equals to $\mathtt{S}$; $\mathfrak{O} \bullet \mathtt{S} = 0$ implies that for all $\mathtt{S}_i$, $\mathtt{S}_i$ does not equals to $\mathtt{S}$. On the contrary, if $\mathtt{S}$ is additive in $\mathfrak{O}$, $\mathfrak{O} \bullet \mathtt{S} \geq 1$ holds, otherwise $\mathfrak{O} \bullet \mathtt{S} = 0$. □

B Proof of Theorem 2

Theorem 2 *Given an ontology $\mathcal{O}$, $\mathfrak{O}$ is the corresponding ontology tensor of $\mathcal{O}$ based on a basic vector set $\mathbb{V}$ and the transformation function τ, and $\mathbb{V}$ satisfies the orthogonality condition For any integer $i \geq 0$ and any assertion α, we have that:*

(Soundness) $\Phi^i(\mathfrak{O}) \bullet \tau(\alpha) \geq 1$ implies $\alpha \in \mathtt{mat}(\mathcal{O})$;

(Completeness) there exists an integer N such that if $\alpha \in \mathtt{mat}(\mathcal{O})$, then $\Phi^N(\mathfrak{O}) \bullet \tau(\alpha) \geq 1$.

Proof: We first prove the soundness by an induction on $\Phi^i(\mathfrak{O})$.

(*Basic case*). When $i = 0$, $\Phi^0(\mathfrak{O}) = \mathfrak{O}$. Further $\mathfrak{O} \bullet \tau(\alpha) \geq 1$ means that $\tau(\alpha)$ is additive in $\mathfrak{O}$ according to Lemma 1. This also implies that $\alpha \in \mathcal{O}$. Obviously, $\alpha \in \mathtt{mat}(\mathcal{O})$ holds according to the definition of $\mathtt{mat}(\mathcal{O})$.

(*Inductive cases*). We have the induction hypothesis that, for some $i \geq 0$, $\Phi^j(\mathfrak{O}) \bullet \tau(\alpha) \geq 1$ implies that $\alpha \in \mathtt{mat}(\mathcal{O})$ for each integer j and $0 \leq j \leq i$.

We then should prove that $\Phi^{i+1}(\mathfrak{O}) \bullet \tau(\alpha) \geq 1$ implies that $\alpha \in \mathtt{mat}(\mathcal{O})$. Recall the formulation of $\Phi^{i+1}(\mathfrak{O})$, i.e., $\Phi^{i+1}(\mathfrak{O}) = \Phi(\Phi^i(\mathfrak{O})) + \Phi^i(\mathfrak{O})$. We expand the formulation of $\Phi^{i+1}(\mathfrak{O}) \bullet \tau(\alpha)$ as follows.

$$\begin{aligned}\Phi^{i+1}(\mathfrak{O}) \bullet \tau(\alpha) &= [\Phi(\Phi^i(\mathfrak{O})) + \Phi^i(\mathfrak{O})]_{x_1:x_7}[\tau(\alpha)]_{x_1:x_7} \\ &= [\Phi(\Phi^i(\mathfrak{O}))]_{x_1:x_7}[\tau(\alpha)]_{x_1:x_7} \\ &\quad + [\Phi^i(\mathfrak{O})]_{x_1:x_7}[\tau(\alpha)]_{x_1:x_7}\end{aligned}$$

Since $\Phi^{i+1}(\mathfrak{O}) \bullet \tau(\alpha) \geq 1$, then (1) $[\Phi(\Phi^i(\mathfrak{O}))]_{x_1:x_7}[\tau(\alpha)]_{x_1:x_7} \geq 1$; or (2) $[\Phi^i(\mathfrak{O})]_{x_1:x_7}[\tau(\alpha)]_{x_1:x_7} \geq 1$ (note that, due to the orthogonality condition, these two addends cannot be lower than 1). If (2) holds, then $\tau(\alpha)$ is additive in $\Phi^i(\mathfrak{O})$ according to Lemma 1. By the induction hypothesis, $\alpha \in \mathtt{mat}(\mathcal{O})$ holds. We next consider the case of (1). $[\Phi(\Phi^i(\mathfrak{O}))]_{x_1:x_7}[\tau(\alpha)]_{x_1:x_7}$ can be further expanded as follows.

$$\begin{aligned}[\Phi(\Phi^i(\mathfrak{O}))]_{x_1:x_7}[\tau(\alpha)]_{x_1:x_7} &= [\phi^{\mathtt{A1}}(\Phi^i(\mathfrak{O}))]_{x_1:x_7}[\tau(\alpha)]_{x_1:x_7} \\ &\quad + [\phi^{\mathtt{A2}}(\Phi^i(\mathfrak{O}))]_{x_1:x_7}[\tau(\alpha)]_{x_1:x_7} \\ &\quad \vdots \\ &\quad + [\phi^{\mathtt{A11}}(\Phi^i(\mathfrak{O}))]_{x_1:x_7}[\tau(\alpha)]_{x_1:x_7} \\ &= \phi^{\mathtt{A1}}(\Phi^i(\mathfrak{O})) \bullet \tau(\alpha) \\ &\quad + \phi^{\mathtt{A2}}(\Phi^i(\mathfrak{O})) \bullet \tau(\alpha) \\ &\quad \vdots \\ &\quad + \phi^{\mathtt{A11}}(\Phi^i(\mathfrak{O})) \bullet \tau(\alpha)\end{aligned}$$

where some addend is greater and equal than 1 due to the orthogonality condition. We use $\mathfrak{O}^i$ to denote $\Phi^i(\mathfrak{O})$, and $\mathcal{O}^i$ to denote the corresponding ontology of $\mathfrak{O}^i$. Let $\tau(\alpha) = \mathtt{T}^{\varphi} \otimes \mathtt{T}_1 \otimes \mathtt{T}_2 \otimes \mathtt{T}_3$. We mainly prove the cases of the operations $\phi^{\mathtt{A1}}$ and $\phi^{\mathtt{A2}}$. For the cases of other operations, one can use the similar idea to prove them.

(*Case* $\phi^{\mathtt{A1}}$). $\phi^{\mathtt{A1}}(\mathfrak{O}^i) \bullet \tau(\alpha) \geq 1$ implies that $\tau(\alpha)$ is additive in $\phi^{\mathtt{A1}}(\Phi^i(\mathfrak{O}))$ according to Lemma 1. Let $\mathfrak{O}^i \bullet_{(1,1)} \mathtt{T}^{\mathtt{A1}} = \sum_{\forall \mathtt{A},\mathtt{B}} \mathtt{A} \otimes \mathtt{B} \otimes \boldsymbol{\theta}$, where each $\mathtt{T}^{\mathtt{A1}} \otimes \mathtt{A} \otimes \mathtt{B} \otimes \boldsymbol{\theta}$ is additive in $\mathfrak{O}^i$; $\mathfrak{O}^i \bullet_{(1,1)} \mathtt{T}^{\mathtt{ca}} = \sum_{\forall \mathtt{A}',\mathtt{a}} \mathtt{A}' \otimes \mathtt{a} \otimes \boldsymbol{\theta}$, where each $\mathtt{T}^{\mathtt{ca}} \otimes \mathtt{A}' \otimes \mathtt{a} \otimes \boldsymbol{\theta}$ is additive in $\mathfrak{O}^i$. Then, each element in $\phi^{\mathtt{A1}}(\mathfrak{O}^i)$ can be written as follows.

$$[\phi^{\mathtt{A1}}(\mathfrak{O}^i)]_{x_1:x_7} = \sum_{\forall \mathtt{A},\mathtt{B}} \sum_{\forall \mathtt{A}',\mathtt{a}} \mathtt{T}^{\mathtt{A1}}_{x_1} \mathtt{A}_{y_2y_3} \mathtt{B}_{x_2x_3} \boldsymbol{\theta}_{y_6x_6} \mathtt{A}'_{y_2y_3} \mathtt{a}_{x_4x_5} \boldsymbol{\theta}_{y_6x_7}$$

Since $\tau(\alpha)$ is additive in $\phi^{\mathtt{A1}}(\mathfrak{O}^i)$, we have that there exists a group of tensors $\mathtt{A}, \mathtt{A}', \mathtt{B}, \mathtt{a}$ such that $\mathtt{T}^{\mathtt{A1}}_{x_1} \mathtt{A}_{y_2y_3} \mathtt{B}_{x_2x_3} \boldsymbol{\theta}_{y_6x_6} \mathtt{A}'_{y_2y_3} \mathtt{a}_{x_4x_5} \boldsymbol{\theta}_{y_6x_7} = 1$. This also implies that $\mathtt{A}' = \mathtt{A}, \mathtt{T}^{\varphi} = \mathtt{T}^{\mathtt{A1}}, \mathtt{T}_1 = \mathtt{B}, \mathtt{T}_2 = \mathtt{a}$ and $\mathtt{T}_3 = \boldsymbol{\theta}$. Further, we have that $A \sqsubseteq B \in \mathcal{O}^i, A(a) \in \mathcal{O}^i$, and that, $\tau(\alpha)$ is transformed from $B(a)$. Obviously, $B(a) \in \mathcal{O}^{i+1}$ holds. Further $B(a) \in \mathtt{mat}(\mathcal{O})$ also holds.

(*Case* $\phi^{\mathtt{A2}}$). $\phi^{\mathtt{A2}}(\mathfrak{O}^i) \bullet \tau(\alpha) \geq 1$ implies that $\tau(\alpha)$ is additive in $\phi^{\mathtt{A2}}(\Phi^i(\mathfrak{O}))$. Let $\mathfrak{O}^i \bullet_{(1,1)} \mathtt{T}^{\mathtt{A2}} = \sum_{\forall \mathtt{A}_1,\mathtt{A}_2,\mathtt{B}} \mathtt{A}_1 \otimes \mathtt{A}_2 \otimes \mathtt{B}$, where each $\mathtt{T}^{\mathtt{A1}} \otimes \mathtt{A}_1 \otimes \mathtt{A}_2 \otimes \mathtt{B}$ is additive in

$\mathfrak{O}^i$; $\mathfrak{O}^i \bullet_{(1,1)} \mathtt{T}^{\mathtt{ca}} = \sum_{\forall \mathtt{A}',\mathtt{a}} \mathtt{A}' \otimes \mathtt{a} \otimes \boldsymbol{\theta}$, where each $\mathtt{T}^{\mathtt{ca}} \otimes \mathtt{A}' \otimes \mathtt{a} \otimes \boldsymbol{\theta}$ is additive in $\mathfrak{O}^i$. Then, each element in $\phi^{\mathtt{A2}}(\mathfrak{O}^i)$ can be written as follows.

$$[\phi^{\mathtt{A2}}(\mathfrak{O}^i)]_{x_1:x_7} = \sum_{\forall \mathtt{A}_1,\mathtt{A}_2,\mathtt{B}} \sum_{\forall \mathtt{A}',\mathtt{a}} \sum_{\forall \mathtt{A}'',\mathtt{b}} \mathtt{T}^{\mathtt{A2}}_{x_1}[\mathtt{A}_1]_{y_2 y_3}[\mathtt{A}_2]_{y_4 y_5}\mathtt{B}_{x_2 x_3}\mathtt{A}'_{y_2 y_3}\mathtt{a}_{z_4 x_4}\boldsymbol{\theta}_{z_6 z_6}\mathtt{A}''_{y_4 y_5}\mathtt{b}_{z_4 x_5}\boldsymbol{\theta}_{x_6 x_7}$$

Since $\tau(\alpha)$ is additive in $\phi^{\mathtt{A1}}(\mathfrak{O}^i)$, we have that there exists a group of tensors $\mathtt{A}_1, \mathtt{A}_2, \mathtt{B}, \mathtt{A}', \mathtt{A}'', \mathtt{a}$ and $\mathtt{b}$ such that $\mathtt{T}^{\mathtt{A2}}_{x_1}[\mathtt{A}_1]_{y_2 y_3}[\mathtt{A}_2]_{y_4 y_5}\mathtt{B}_{x_2 x_3}\mathtt{A}'_{y_2 y_3}\mathtt{a}_{z_4 x_4}\boldsymbol{\theta}_{z_6 z_6}\mathtt{A}''_{y_4 y_5}\mathtt{b}_{z_4 x_5}\boldsymbol{\theta}_{x_6 x_7} = 1$. This also implies that $\mathtt{A}' = \mathtt{A}_1, \mathtt{A}'' = \mathtt{A}_2, \mathtt{a} = \mathtt{b}\mathtt{T}^{\varphi} = \mathtt{T}^{\mathtt{A2}}, \mathtt{T}_1 = \mathtt{B}, \mathtt{T}_2 = \mathtt{a}$ and $\mathtt{T}_3 = \boldsymbol{\theta}$. Further, we have that $A_1 \sqcap A_2 \sqsubseteq B \in \mathcal{O}^i, A_1(a) \in \mathcal{O}^i, A_2(a) \in \mathcal{O}^i$, and that, $\tau(\alpha)$ is transformed from $B(a)$. Obviously, $B(a) \in \mathcal{O}^{i+1}$ holds by applying Rule ($\mathtt{A2}$). Further $B(a) \in \mathtt{mat}(\mathcal{O})$ also holds.

To prove the completeness, we label each assertion in $\mathtt{mat}(\mathcal{O})$ with a unique integer. The labeled assertions are defined recursively as follows. For each assertion α originally occurring in $\mathcal{O}$, it is labeled with the integer 0, denoted by α^0. If some assertion α is derived by applying some rule in Table 1 from preconditions with labels, i.e., $\beta^{x_1}, ..., \beta^{x_n}$, then $\max\{x_1, ..., x_n\} + 1$ is a *candidate label* of α; for all candidate labels of α, the minimal one is the final label of α. The completeness can then be proved by an induction on the labeled assertions.

(*Basic case*) For all assertions of the label 0, their corresponding statement tensors are additive in the ontology tensor $\mathfrak{O}$. Obviously, the result holds.

(*Inductive cases*) We have an induction hypothesis that, for some integer $i \geq 0$ and each assertion α^j $(1 \leq j \leq i)$, we have that, if $\alpha^j \in \mathtt{mat}(\mathcal{O})$, then $\Phi^j(\mathfrak{O}) \bullet \tau(\alpha) \geq 1$ holds. We now need to prove that this result also holds for the integer $i + 1$.

For a labeled assertion $B(a)^{i+1} \in \mathtt{mat}(\mathcal{O})$, we suppose that $B(a)^{i+1}$ is derived by applying Rule ($\mathtt{A1}$) on the premises $A \sqsubseteq B$ and $A(a)^j$ where $0 \leq j \leq i$. According to the induction hypothesis, $\Phi^j(\mathfrak{O}) \bullet \tau(A(a)) \geq 1$ holds, or says $\mathtt{T}^{\mathtt{ca}} \otimes \mathtt{A} \otimes \mathtt{a} \otimes \boldsymbol{\theta}$ is additive in $\Phi^j(\mathfrak{O})$. Since the corresponding statement tensor $\mathtt{T}^{\mathtt{A1}} \otimes \mathtt{A} \otimes \mathtt{B} \otimes \boldsymbol{\theta}$ is additive in $\Phi^j(\mathfrak{O})$, by applying $\phi^{\mathtt{A1}}$ over $\Phi^j(\mathfrak{O})$, $\mathtt{T}^{\mathtt{ca}} \otimes \mathtt{B} \otimes \mathtt{a} \otimes \boldsymbol{\theta}$ has to be additive in $\Phi^{j+1}(\mathfrak{O})$. Thus $\Phi^{i+1}(\mathfrak{O}) \bullet \tau(B(a)) \geq 1$ holds. For other rules and role assertions, one can follow the similar idea to prove all inductive cases. We do not give the details here. $\square$