

Connecting state-based and team-based inquisitive logic

IVANO CIARDELLI, *Università di Padova*.

E-mail: ivano.ciardelli@unipd.it

SIMONE CONTI, *Università di Padova*.

E-mail: simone.conti@phd.unipd.it

Abstract

We describe entailment-preserving translations between two inquisitive systems: the standard system of inquisitive first-order logic InqBQ , which uses a state-based semantic framework, and its team-semantics counterpart, InqBT . By using these translations, we connect some major open problems about these logics, and transfer some recent results about fragments of InqBQ to InqBT .

Keywords: inquisitive logic, team semantics, dependence logic, translations

1 Introduction

The past decade has seen the rise of two closely related semantic approaches that permit the extension of classical predicate logic with new formulas or operators. The first approach, which may be called *state-based semantics*, proposes to interpret formulas of predicate logic, not relative to a single relational structure, but relative to an *information state*, which may be seen as a *set* of relational structures based on a common domain; this approach has been advocated especially in the literature on *inquisitive logic* [4, 7], where it is used to interpret a predicate logic language that comprises not only statements, but also questions.¹ The second approach, known as *team semantics*, proposes to interpret formulas of predicate logic, not relative to a single variable assignment, but relative to a *set* of assignments, called a *team*. This approach has been used since [19, 20] to interpret various logics that can specify relations of dependence and independence between the values of variables. A prominent example is dependence logic [25], which extends the language of predicate logic with dependence atoms of the form $=(x_1, \dots, x_n; y)$, expressing the fact that the value of y is functionally determined by the values of $x_1, \dots, x_n$. Some important results on dependence logic and its fragments can be found in [12, 21–24, 27]. Other prominent examples of team-based logics are inclusion logic [13] and independence logic [14].

The two approaches are very much in the same spirit: in each case, the key idea is to replace single indices of evaluation by sets of indices. Moreover, in each case, the behaviour of classical logic is recovered for those formulas for which satisfaction at a set of indices reduces to satisfaction at each index in the set. In light of these similarities, it is not surprising that the two approaches have tight connections, which have permitted fruitful transfers of insights and results (see, a.o., [2, 8, 28]):

¹ More recently, this approach has also been employed to build logical frameworks designed to capture logical features of disjunctions and epistemic modals [1, 18].

for instance, many operators can be introduced with analogous clauses in state-based semantics and team semantics, and their logical features are often the same. Yet, there are also some differences between the two settings: relational structures and assignments play a different role in the semantics: the former are used to interpret the non-logical vocabulary of the language, while the latter are used to interpret quantification; as a consequence, there is no general one-to-one correspondence between state-based logics and team-based logics.²

In this paper, we investigate the relations between a state-based logic and a team-based logic whose semantics is structurally parallel: these are the standard inquisitive first-order logic **InqBQ** [4, 7], and the team-based inquisitive logic **InqBT**, which was first investigated in the team semantics literature under the name of **WID** (for *weak intuitionistic dependence logic*) (see [4, 26]).

The system **InqBQ** occupies a central place in inquisitive logic, where it is regarded as the standard inquisitive extension of classical first-order logic. It may be seen as obtained by extending a classical language with two question-forming operators: the inquisitive disjunction $\vee$ (intuitively understood as generating disjunctive questions of the form ‘whether ... or’), and the inquisitive existential quantifier $\exists$ (intuitively understood as generating instance-seeking questions ‘what is an x such that ...’). In terms of $\vee$, a polar question operator (understood as ‘whether or not ...’) is defined as $? \varphi := \varphi \vee \neg \varphi$. By means of these operators, in this logic we can regiment a broad spectrum of questions. Thus, e.g. in addition to standard formulas like $\forall x Px$, capturing the statement that all objects are P , in **InqBQ** we also have formulas like $? \forall x Px$, $\forall x ? Px$ and $\exists x P$, capturing, respectively, the questions *whether or not all objects are P*, *which objects are P* and *what is an instance of a P*.

While a fair amount of work has been devoted to the study of **InqBQ** [4, 6, 15–17], some key meta-theoretic questions about it remain open. In particular, it is not known whether this system is *entailment-compact*, i.e. if any conclusion that follows from a set of premises also follows from a finite subset, and whether its set of validities is recursively enumerable (and, thus, whether a complete axiomatization is possible). In recent works [6, 16], these questions have been answered in the positive for two wide syntactic fragments of **InqBQ**: the classical antecedent fragment (Clant), where only classical formulas (i.e. formulas without inquisitive operators) are allowed as antecedents of an implication, and the restricted existential fragment (Rex), where inquisitive existentials are only allowed to appear in the antecedent of an implication.

The team-based logic **InqBT** has the same language as **InqBQ**, and its semantic clauses mirror exactly the ones for **InqBQ** in the team semantics setting. In comparison with dependence logic and related systems, **InqBT** only features a simple version of the quantifiers: although the quantifiers of dependence logic are able to assign to a variable x a range of different values within a team, the quantifiers of **InqBT** simply explore different ways of assigning a constant value to x throughout the team. As a result, at the level of sentences **InqBT** is no more expressive than classical first-order logic [26]. When it comes to open formulas, however, **InqBT** can express many interesting facts. In particular, the dependence atoms of dependence logic can be expressed in terms of implications among questions: if λx abbreviates the question $\exists y (x = y)$ asking for the value of x , the dependence atom $= (x_1, \dots, x_n; y)$ can be rendered as $\lambda x_1 \wedge \dots \wedge \lambda x_n \rightarrow \lambda y$; moreover, as discussed in detail in [4], by considering different kinds of questions, many other interesting dependence patterns can be expressed in **InqBT** as well.

Interestingly, the same fundamental questions that are open about **InqBQ** are open about **InqBT** as well: it is not known whether **InqBT** is entailment-compact, and whether the set of

²It is worth noting that these differences do not exist in the propositional case, where both state-based and team semantics interpret formulas relative to sets of assignments of truth-values to the atomic sentences. While these sets are called, respectively, *states* and *propositional teams* in the literature, they do not generate two different semantic frameworks and, therefore, no connection problem arises at the propositional level.

validities is recursively enumerable. As compared with InqBQ , however, InqBT has been much less investigated; in particular, no positive results about its fragments have been established. The exact connection between InqBT and other well-studied team-based logics is an open problem, too; in particular, it is not known whether a translation from InqBT to (a fragment of) dependence logic is possible, or whether InqBT can define team properties that are not expressible in dependence logic. The reason why the relation is complex is that, while InqBT contains only weak quantifiers, definable in dependence logic, it features a powerful implication connective that quantifies over sub-teams—an operator that is known not to be definable in dependence logic [26].

In light of the close parallel between the semantics of InqBQ and InqBT , it seems natural to ask if one can establish entailment-preserving translations between these two logics. This question has been posed explicitly in [4], where it is left as an open problem. In the present paper, we establish a positive answer to this question: we define entailment-preserving translations between InqBQ and InqBT , and we use these translations to connect the open problems concerning these logics, and to transfer the recent results about the fragments of InqBQ to the corresponding fragments of InqBT .

Our strategy is based on the idea of encoding information states by means of teams and vice versa, and on the possibility of characterizing the general properties of these encodings by means of formulas of the target logic. We believe that these ideas are not merely instrumental in connecting InqBQ and InqBT , but may find more general application in the process of relating state-based and team-based logics.

The paper is organized as follows. In Section 2, we provide some background on InqBQ and InqBT . In Section 3, we define the translations and show that they preserve entailments. In Section 4, we connect the key open problems for InqBQ and InqBT and we transfer the results about the Rex and Clant fragments of InqBQ to the corresponding fragments of InqBT . Section 5 concludes by outlining directions for future work.

2 Background

2.1 InqBQ

In this section, we introduce the system InqBQ , along with some relevant notions and results. For a more comprehensive overview of this system and for proofs of the results mentioned in this section, we refer to [4].

Syntax A signature for InqBQ is a set of symbols Σ containing predicate symbols, indicated by uppercase letters (e.g. R_n), and function symbols, indicated by lowercase letters (e.g. f_n). The subscripts specify the *arity* of each symbol and will be omitted when possible. Relation symbols of arity 0 can be seen as propositional atoms, while function symbols of arity 0 are called *constants*. Certain function symbols are additionally designated as *rigid*, and will be indicated with sans-serif font (e.g. $\mathbf{f}_n$). Identity is regarded as a particular binary predicate symbol, indicated by $=$, which may or may not be included in Σ . By a *relational* signature we will mean a signature containing only predicate symbols and no function symbols.

The set of terms $\text{Ter}[\Sigma]$ is defined in the usual way, using the functions of Σ and a countably infinite set of variables $\text{Var} = \{x_0, x_1, \dots\}$. We will use the following symbols as meta-variables: $x, y, z; y_1, y_2, \dots; z_1, z_2, \dots$. Note that we will not always use meta-variables, as we will sometimes need to reference the actual variables $x_0, x_1, \dots$ and their specific subscripts.

The set of formulas $\mathcal{L}(\Sigma)$, or simply $\mathcal{L}$, is then defined inductively as follows:

$$\varphi ::= \perp \mid R_n(t_1, \dots, t_n) \mid \varphi \wedge \varphi \mid \varphi \rightarrow \varphi \mid \varphi \vee \varphi \mid \forall x \varphi \mid \exists x \varphi,$$

where R_n is an n -ary predicate of Σ and $t_1, \dots, t_n \in \text{Ter}[\Sigma]$.

4 Connecting state-based and team-based inquisitive logic

The symbols $\vee$ and $\exists$ are two question-forming operators and they are called, respectively, *inquisitive disjunction* and *inquisitive existential quantifier*. These symbols are interpreted differently compared with their classical counterparts, which we introduce as defined operators by letting $\neg\varphi := \varphi \rightarrow \perp$, $\varphi \vee \psi := \neg(\neg\varphi \wedge \neg\psi)$, $\exists x\varphi := \neg\forall x\neg\varphi$. We also introduce an additional inquisitive operator $?$, defined by $?\varphi := \varphi \vee \neg\varphi$, and the biconditional $\leftrightarrow$, defined by $\varphi \leftrightarrow \psi := (\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi)$.

Formulas without $\vee$ or $\exists$ are called *classical formulas*, and they may be identified with the standard formulas of predicate logic, with a particular choice of primitives. The set of classical formulas is denoted by $\mathcal{L}^{cl}(\Sigma)$, or simply $\mathcal{L}^{cl}$.

The set of free variables of a formula φ or set of formulas Φ is defined in the usual way, and denoted as $FV(\varphi)$ or $FV(\Phi)$. A formula with no free variables is called *sentence* or *closed formula*. When $FV(\varphi) \subseteq X$ for some $\varphi \in \mathcal{L}(\Sigma)$ and set of variables X , we write $\varphi \in \mathcal{L}(\Sigma, X)$: in other words, $\mathcal{L}(\Sigma, X)$ is the set of formulas of $\mathcal{L}(\Sigma)$ whose set of free variables is included in X .

Semantics We start from the case of models that do not interpret the identity predicate. A relational information model is a triple $M = \langle W, D, I \rangle$ where

- W is a set, whose elements are called *possible worlds* or simply *worlds*
- D is a non-empty set, called the *domain*, whose elements are called *individuals*
- I is a function assigning to each possible world w a function I_w from Σ , satisfying the following constraints:
 - $I_w(R_n) \subseteq D^n$, for any predicate $R_n \in \Sigma$
 - $I_w(f_n) : D^n \rightarrow D$, for any function symbol $f_n \in \Sigma$
 - if $f \in \Sigma$ is rigid, then $I_w(f) = I_{w'}(f)$ for all $w, w' \in W$

We will abbreviate $I_w(R)$ as R_w and $I_w(f)$ as f_w .

This notion of a model can be extended to include an interpretation of the identity predicate. A relational information model with identity is a relational information model equipped with an identity extension function. We write these structures as quadruples $M = \langle W, D, I, \sim \rangle$. An identity extension function $\sim$ assigns to each world w a binary relation $\sim_w$, which is the extension of the identity relation at w . For $\sim_w$ to be the extension of an identity relation, it must satisfy the following constraints for any $w \in W$:

- Equivalence conditions: $\sim_w$ must be an equivalence relation on D
- Congruence conditions: if $d_1 \sim_w d'_1, \dots, d_n \sim_w d'_n$, then
 - Predicates: $\langle d_1, \dots, d_n \rangle \in I_w(R_n) \iff \langle d'_1, \dots, d'_n \rangle \in I_w(R_n)$
 - Functions: $I_w(f_n)(d_1, \dots, d_n) \sim_w I_w(f_n)(d'_1, \dots, d'_n)$

We will refer to both relational information models with identity and relational information models as *models* when no ambiguity arises.

Definition 2.1.

A model of InqBQ can be identified with a set of relational structures over a common domain. When we exclude the identity predicate, such classical counterparts can be obtained by assigning each world w with the structure $M_w = \langle D, I_w \rangle$. In the general case, we assign each world $w \in W$ to the model $\tilde{M}_w = \langle \tilde{D}_w, \tilde{I}_w \rangle$, where:

- $\tilde{D}_w = D / \sim_w$
- $\tilde{I}_w(f)([d_1]_{\sim_w}, \dots, [d_n]_{\sim_w}) = [I_w(f)(d_1, \dots, d_n)]_{\sim_w}$

$$- \langle [d_1]_{\sim_w}, \dots, [d_n]_{\sim_w} \rangle \in \tilde{I}_w(R) \iff \langle d_1, \dots, d_n \rangle \in I_w(R)$$

$\tilde{M}_w$ is well defined due to the congruence conditions imposed on $\sim$.

For a set $X \subseteq \text{Var}$ of variables, an X -assignment over a model $M = \langle W, D, I, \sim \rangle$ is a function $g : X \rightarrow D$. Note that if $X = \emptyset$, there is only one X -assignment, the empty function, which we denote by ϵ . If it is not crucial to specify the set X , we just call g an assignment. If g is an X -assignment, $y \in \text{Var}$ and $d \in D$, we let $g[y \mapsto d]$ be the $X \cup \{y\}$ -assignment defined as follows:

$$g[y \mapsto d](x) := \begin{cases} d & \text{if } x = y \\ g(x) & \text{if } x \in X \text{ and } x \neq y. \end{cases}$$

If w is a world and g an assignment whose domain includes the variables in a term t , the referent $[t]_g^w \in D$ is defined inductively in the usual way:

$$[x]_g^w = g(x), \quad [a]_g^w = I_w(a), \quad [f(t_1, \dots, t_n)]_g^w = I_w(f)([t_1]_g^w, \dots, [t_n]_g^w). \quad (2.1)$$

The semantics of InqBQ is then given by a definition of *support*.

Definition 2.2. (InqBQ support)

Let $M = \langle W, D, I, \sim \rangle$ be an InqBQ model, let $s \subseteq W$ be a state in M and let g be an X -assignment over M . The support conditions for any formula φ with $FV(\varphi) \subseteq X$ are defined as follows:

- $M, s \models_g \perp \iff s = \emptyset$
- $M, s \models_g R(t_1, \dots, t_n) \iff \text{for all } w \in s, \langle [t_1]_g^w, \dots, [t_n]_g^w \rangle \in I_w(R)$
- $M, s \models_g t_1 = t_2 \iff \text{for all } w \in s, [t_1]_g^w \sim_w [t_2]_g^w$
- $M, s \models_g \varphi \wedge \psi \iff M, s \models_g \varphi \text{ and } M, s \models_g \psi$
- $M, s \models_g \varphi \vee \psi \iff M, s \models_g \varphi \text{ or } M, s \models_g \psi$
- $M, s \models_g \varphi \rightarrow \psi \iff \text{for all } t \subseteq s, M, t \models_g \varphi \text{ implies } M, t \models_g \psi$
- $M, s \models_g \forall x \varphi \iff \text{for all } d \in D, M, s \models_{g[x \mapsto d]} \varphi$
- $M, s \models_g \exists x \varphi \iff \text{for some } d \in D, M, s \models_{g[x \mapsto d]} \varphi$

When no confusion arises, we will drop the reference to the model M and simply write $s \models_g \varphi$.³

We introduce some useful abbreviations: given a state s in a model M , an assignment g , a formula $\varphi \in \mathcal{L}$ and a set of formulas $\Phi \subseteq \mathcal{L}$ we write $M \models_g \varphi$, for $M, W \models_g \varphi$; we write $M, s \models_g \Phi$, meaning that $M, s \models_g \varphi$ for all $\varphi \in \Phi$ and we write $M, s \models \varphi$ (or simply $s \models \varphi$), meaning that, for all X -assignments g over M with $X \supseteq FV(\varphi)$, $M, s \models_g \varphi$.

Entailment is defined as the preservation of support. For $\Phi \cup \{\psi\} \subseteq \mathcal{L}$, we say that Φ entails ψ , written $\Phi \models_{\text{InqBQ}} \psi$, if for any model M , for any state s and for any assignment g with domain $X \supseteq FV(\Phi \cup \{\psi\})$, if $M, s \models_g \Phi$, then $M, s \models_g \psi$. If $\emptyset \models_{\text{InqBQ}} \varphi$, we write $\models_{\text{InqBQ}} \varphi$ and we say that φ is a validity of InqBQ.

³For the defined operators, the following clauses can be derived:

- $M, s \models_g \neg \varphi \iff \text{for all } \emptyset \neq t \subseteq s : t \not\models_g \varphi \iff \text{for all } w \in s, \{w\} \not\models_g \varphi$
- $M, s \models_g ?\varphi \iff M, s \models_g \varphi \text{ or } M, s \models_g \neg \varphi$
- $M, s \models_g \varphi \vee \psi \iff \text{for all } w \in s, \{w\} \models_g \varphi \text{ or } M, \{w\} \models_g \psi$
- $M, s \models_g \exists x \varphi \iff \text{for all } w \in s, \text{ for some } d \in D, M, \{w\} \models_{g[x \mapsto d]} \varphi$

Basic properties Support verifies the following fundamental properties:

1. **Empty state property:** for all $\varphi \in \mathcal{L}$, $\emptyset \models \varphi$.
2. **Persistency:** for all models M , states s , assignments g and formulas $\varphi \in \mathcal{L}$, if $s \models_g \varphi$ and $t \subseteq s$, then $t \models_g \varphi$.
3. **Locality:** for all models M , states s , assignments g and formulas $\varphi \in \mathcal{L}$, $M, s \models_g \varphi \iff M|_s, s \models_g \varphi$, where $M|_s = \langle s, D, I|_s, \sim|_s \rangle$.

The first property is comparable with the *ex-falso-quodlibet* principle of classical logic. It expresses the fact that the empty state, which represents the state of inconsistent information, supports any formula. Persistency, on the other hand, can be seen as a representation of the fact that formulas supported at a information state remain supported after adding information. Locality tells us that support at a state only depends on worlds in the state.

We also state an intuitive and familiar property: assignments do not affect the support of sentences. If φ is a closed formula, then for any model M , state s and assignment g ,

$$M, s \models_g \varphi \iff M, s \models \varphi.$$

Truth-conditionality and classical formulas For classical formulas, the inquisitive semantics of InqBQ is conservative over the truth-conditional semantics of classical logic. To make this connection precise, we consider how the semantics works at the level of individual worlds. We say that a formula φ is *true* at a world w if it is supported by the singleton state $\{w\}$. We write $M, w \models \varphi$ (or simply $w \models \varphi$) instead of $M, \{w\} \models \varphi$.

We say that a formula φ is *truth-conditional* if for φ , being supported at a state amounts to being true at every world in the state, i.e. if, for any model M , state s and assignment g ,

$$M, s \models_g \varphi \iff M, w \models_g \varphi \text{ for all } w \in s.$$

All classical formulas are truth-conditional in InqBQ. We get the preservation of classical semantics for classical formulas by combining their truth-conditionality with the following result (see Proposition 5.1.7 in [4]): for any classical formula $\alpha \in \mathcal{L}^{cl}$, for any model M , world w and assignment g ,

$$M, w \models_g \alpha \iff \tilde{M}_w \models_{\tilde{g}_w} \alpha \text{ in first-order classical logic,}$$

where $\tilde{g}_w$ is the assignment given by $\tilde{g}_w(x) = [g(x)]_{\sim_w}$.

Additionally (see Proposition 5.3.7 in [4]), classical formulas exhaust all truth-conditional formulas up to logical equivalence, i.e.

$$\varphi \text{ is truth-conditional} \iff \varphi \equiv \alpha \text{ for some } \alpha \in \mathcal{L}^{cl},$$

where $\varphi \equiv \alpha$ means that for all M, s and g , $M, s \models_g \varphi \iff M, s \models_g \alpha$.

Essential equivalence The definition of the models of InqBQ allows for the presence of duplicate worlds, i.e. worlds that interpret all symbols of the signature in the same way. Having duplicate worlds in a model clearly does not affect what formulas are supported by the model. This is made precise using the notion of essential equivalence.

Definition 2.3.

Given a model $M = \langle W, D, I, \sim \rangle$, we say that two worlds $w, w' \in W$ are *essentially equivalent*, written as $w \approx_e w'$, if $\tilde{M}_w = \tilde{M}_{w'}$.

We also define W^e as the quotient $W/\approx_e$. The relation $\approx_e$ is an equivalence relation and we will use $[w]$ to refer to $[w]_{\approx_e}$.

Definition 2.4.

Let $M = \langle W, D, I, \sim \rangle$ be a model. The essential quotient of M is $M^e = \langle W^e, D, I^e, \sim^e \rangle$, where, for all $w \in W$,

- $I_{[w]}^e(f)(d_1, \dots, d_n) = I_w(f)(d_1, \dots, d_n)$
- $\langle d_1, \dots, d_n \rangle \in I_{[w]}^e(R) \iff \langle d_1, \dots, d_n \rangle \in I_w(R)$
- $\sim_{[w]}^e \equiv \sim_w$

The definition of essentially equivalent worlds guarantees that M^e is well defined.

The essential quotient of a model behaves exactly as expected: it supports the same formulas as the model itself.

Proposition 2.5.

Let M be a model and g an assignment. For any $\varphi \in \mathcal{L}$,

$$M \models_g \varphi \iff M^e \models_g \varphi.$$

2.2 InqBT

In this section, we introduce the system **InqBT**, along with the relevant notions and results. For a more detailed presentation of **InqBT** we refer to §7.4 of [4].

Syntax A signature for **InqBT** is defined as a standard first-order signature. The set of formulas of **InqBT** is simply $\mathcal{L}(\Sigma)$, the same as for **InqBQ**. The definition of the set of classical formulas ($\mathcal{L}^{cl}$) and the definition of free and bound variables are also unchanged from **InqBQ**.⁴

Semantics A model $\mathcal{M}$ for **InqBT** is a relational structure $\mathcal{M} = \langle \mathcal{D}, \mathcal{I} \rangle$, where

- $\mathcal{D}$ is a non-empty set, whose elements are called individuals
- $\mathcal{I}$ is a function with domain Σ , satisfying the following constraints:
 - $\mathcal{I}(R_n) \subseteq \mathcal{D}^n$, for any predicate $R_n \in \Sigma$
 - $\mathcal{I}(f_n) : \mathcal{D}^n \rightarrow \mathcal{D}$, for any function symbol $f_n \in \Sigma$

As above, an X -assignment is a function $g : X \rightarrow \mathcal{D}$ from a set X of variables to the domain $\mathcal{D}$. An X -team is a set of X -assignments; as usual, we omit reference to X when possible. If T is an X -team, $y \in \text{Var}$ and $d \in \mathcal{D}$, we define $T[y \mapsto d]$ to be the $X \cup \{y\}$ -team defined as $T[y \mapsto d] := \{g[y \mapsto d] \mid g \in T\}$. If T is an X -team and $Y \subseteq X$, we define the restriction of T to Y as $T|_Y = \{g|_Y \mid g \in T\}$.

The referent of a term is defined as an individual $[t]_g^{\mathcal{M}} \in \mathcal{D}$ in the usual way. The semantic clauses for **InqBT** are analogous to those of **InqBQ**. The only difference is that now the role of the states is taken on by the teams.

Definition 2.6. (**InqBT** support)

Let $\mathcal{M} = \langle \mathcal{D}, \mathcal{I} \rangle$ be a model of **InqBT** and let T be an X -team over $\mathcal{M}$. We define support for a formula φ with $FV(\varphi) \subseteq X$ inductively as follows:

- $\mathcal{M} \models_T \perp \iff s = \emptyset$
- $\mathcal{M} \models_T R(t_1, \dots, t_n) \iff \text{for all } g \in T, \langle [t_1]_g^{\mathcal{M}}, \dots, [t_n]_g^{\mathcal{M}} \rangle \in \mathcal{I}(R)$

⁴The quantifiers $\forall$ and $\exists$, as defined by the semantic clauses of **InqBT** given below, are often denoted by $\forall^1$ and $\exists^1$ in the team semantics literature (see, e.g. [22] and [26]). Here, we denote them using the same symbols that we adopted for their **InqBQ** counterparts to avoid duplicating the language and to highlight the parallel between the two systems.

- $\mathcal{M} \models_T t_1 = t_2 \iff \text{for all } g \in T, [t_1]_g^{\mathcal{M}} = [t_2]_g^{\mathcal{M}}$
- $\mathcal{M} \models_T \varphi \wedge \psi \iff \mathcal{M} \models_T \varphi \text{ and } \mathcal{M} \models_T \psi$
- $\mathcal{M} \models_T \varphi \vee \psi \iff \mathcal{M} \models_T \varphi \text{ or } \mathcal{M} \models_T \psi$
- $\mathcal{M} \models_T \varphi \rightarrow \psi \iff \text{for all } T' \subseteq T, \mathcal{M} \models_{T'} \varphi \text{ implies } \mathcal{M} \models_{T'} \psi$
- $\mathcal{M} \models_T \forall x \varphi \iff \text{for all } d \in \mathcal{D}, \mathcal{M}, s \models_{T[x \mapsto d]} \varphi$
- $\mathcal{M} \models_T \exists x \varphi \iff \text{for some } d \in \mathcal{D}, \mathcal{M}, s \models_{T[x \mapsto d]} \varphi$

We write $\mathcal{M} \models \varphi$ if $\mathcal{M} \models_T \varphi$ for every team T with domain $X \supseteq FV(\varphi)$.

For $\Phi \cup \{\psi\} \subseteq \mathcal{L}$, we say that Φ entails ψ , and we write $\Phi \models_{\text{InqBT}} \psi$, if for any model $\mathcal{M}$ and for any X -team T with $X \supseteq FV(\Phi \cup \{\psi\})$, if $\mathcal{M} \models_T \Phi$, then $\mathcal{M} \models_T \psi$. If $\emptyset \models_{\text{InqBT}} \varphi$, we also write $\models_{\text{InqBT}} \varphi$ and say that φ is valid.

Basic properties As was the case for InqBQ, support in InqBT satisfies the two following fundamental properties (see Proposition 7.4.2 in [4]):

1. **Empty state property:** for all models $\mathcal{M}$ and formulas $\varphi \in \mathcal{L}$, $\mathcal{M} \models_{\emptyset} \varphi$.
2. **Persistency:** for all models $\mathcal{M}$, teams T and T' and formulas $\varphi \in \mathcal{L}$, if $\mathcal{M} \models_T \varphi$ and $T' \subseteq T$, then $\mathcal{M} \models_{T'} \varphi$.

Moreover, support conditions for a formula φ at a team only depend on the evaluation of the free variables of φ in the team (see Proposition 7.4.6 in [4]).

Proposition 2.7.

Let $\varphi \in \mathcal{L}$, let $\mathcal{M}$ be a model and let $T : X \rightarrow \mathcal{D}$ and $T' : X' \rightarrow \mathcal{D}$ be teams with $X, X' \supseteq FV(\varphi)$. If $T|_{FV(\varphi)} = T'|_{FV(\varphi)}$,

$$\mathcal{M} \models_T \varphi \iff \mathcal{M} \models_{T'} \varphi.$$

In particular, sentences receive the same evaluation under any non-empty team.

Truth-conditionality and classical formulas We say that a formula is *true* at a model under an assignment g if it is supported at that model under the team $\{g\}$. A formula φ is then called truth-conditional if for every model $\mathcal{M}$ and team T ,

$$\mathcal{M} \models_T \varphi \iff \text{for all } g \in T, \mathcal{M} \models_g \varphi.$$

Similarly to InqBQ, classical formulas are truth-conditional (Proposition 7.4.5 in [4]).

To see that InqBT is conservative over first-order classical logic, it then suffices to observe that, in InqBT, truth under an assignment at a model coincides with the classical notion of truth.

3 Translations

In this section, we define translations in both directions between InqBQ and InqBT. For the sake of clarity, we first focus on the case of a relational signature Σ (i.e. Σ contains only predicate symbols) without the identity predicate. This simplified case will allow us to better convey the central ideas behind the translation. The extension to the general case, which mostly just involves some technical adaptations, is described in Section 3.3.

3.1 Translating from InqBQ to InqBT

To define a translation $(\cdot)^\sharp$ from InqBQ to InqBT, our strategy is to simulate the semantics of InqBQ within InqBT; more precisely, we will encode a model M for InqBQ as a model $\mathcal{M}^\sharp$ for

InqBT , and a state $s \subseteq W$ as a corresponding team T_s in $\mathcal{M}^\sharp$, in such a way that the semantics of an InqBQ -sentence φ relative to M and s matches the semantics of its translation $\varphi^\sharp$ relative to $\mathcal{M}^\sharp$ and T_s .

Translating models. A model M for InqBQ includes two domains: a set W of worlds and a set D of individuals. We may encode M as a model $\mathcal{M}^\sharp$ for InqBT whose single domain is the disjoint union $W \uplus D$. To retrieve the distinction between worlds and individuals, we introduce a unary predicate ‘Ind’, which holds only for individuals. To simulate the fact that an n -ary predicate R in InqBQ is true or false relative to n -tuple of objects *at a world*, we encode R as an $n + 1$ -ary predicate R' , which holds of $n + 1$ -tuples consisting of a world and an n -tuple of individuals.

Definition 3.1.

Let Σ be a relational signature without identity. We define a signature $\Sigma^\sharp := \{R'_{n+1} \mid R_n \in \Sigma\} \uplus \{\text{Ind}_1\}$, where Ind is a fresh unary predicate.

Definition 3.2.

Let $M := \langle D, W, I \rangle$ be a model of InqBQ in the signature Σ . We define a model $\mathcal{M}^\sharp := \langle \mathcal{D}^\sharp, \mathcal{I}^\sharp \rangle$ of InqBT in the signature $\Sigma^\sharp$:

- $\mathcal{D}^\sharp := D \uplus W$
- $\mathcal{I}^\sharp(R'_{n+1}) := \{\langle d_1, d_2, \dots, d_{n+1} \rangle \in (\mathcal{D}^\sharp)^{n+1} \mid d_1 \in W, \langle d_2, \dots, d_{n+1} \rangle \in I_{d_1}(R_n)\}$
- $\mathcal{I}^\sharp(\text{Ind}) := D$

Translating information states. The role of information states in InqBQ is taken over by teams in InqBT . For a set of variables X , we let $X^\sharp$ be the set that contains x_0 along with the variables in X ‘shifted by one’:

$$X^\sharp = \{x_0\} \cup \{x_{n+1} \mid x_n \in X\}.$$

We associate to each world w of M and assignment $g : X \rightarrow D$ a corresponding assignment $g_w : X^\sharp \rightarrow \mathcal{D}^\sharp$, defined as follows:

- $g_w(x_0) = w$
- for $x_n \in X$, $g_w(x_{n+1}) = g(x_n)$

With an information state s and an X -assignment g into D , we then associate a corresponding $X^\sharp$ -team

$$T_s^g = \{g_w \mid w \in s\}.$$

Note that the information state s and the team T_s^g are in a one-to-one correspondence via the map $w \mapsto g_w$. It will thus be useful to see them as indexed over the same set $\mathcal{J}$: we will write $s_{\mathcal{J}} = \{w_j \mid j \in \mathcal{J}\}$ for an arbitrary information state, and $T_{\mathcal{J}} = \{g_j \mid j \in \mathcal{J}\}$ for the corresponding team, where $g_j := g_{w_j}$.

Translating formulas. To translate formulas of InqBQ into InqBT , we simulate an atomic formula $R(x_{i_1}, \dots, x_{i_n})$ by the formula $R'(x_0, x_{i_1+1}, \dots, x_{i_n+1})$, where the role of the variable x_n is taken over by x_{n+1} and x_0 is used as a variable for the world of evaluation. Moreover, since quantifiers in InqBQ range only over individuals, we restrict all quantifiers by means of the predicate Ind.

Definition 3.3.

For any $\varphi \in \mathcal{L}(\Sigma)$, we define $\varphi^\sharp \in \mathcal{L}(\Sigma^\sharp)$ as follows:

- $(\perp)^\sharp = \perp$
- $(R_n(x_{i_1}, \dots, x_{i_n}))^\sharp = R'_{n+1}(x_0, x_{i_1+1}, \dots, x_{i_n+1})$
- $(\psi \circ \xi)^\sharp = (\psi^\sharp \circ \xi^\sharp)$, where $\circ \in \{\wedge, \vee, \rightarrow\}$
- $(\forall x_n \psi)^\sharp = \forall x_{n+1} (\text{Ind}(x_{n+1}) \rightarrow \psi^\sharp)$
- $(\exists x_n \psi)^\sharp = \exists x_{n+1} (\text{Ind}(x_{n+1}) \wedge \psi^\sharp)$

Note that by definition, $FV(\varphi^\sharp) = FV(\varphi)^\sharp$. As a consequence, for any set X of variables we have $FV(\varphi) \subseteq X \iff FV(\varphi^\sharp) \subseteq X^\sharp$. In particular, a formula φ can be interpreted on an X -assignment g iff its translation $\varphi^\sharp$ can be interpreted on the corresponding $X^\sharp$ -team, T_s^g . We will take this fact for granted below.

The crucial property of the translation we just defined is given by the following proposition.

Proposition 3.4.

For any formula $\varphi \in \mathcal{L}(\Sigma)$, InqBQ-model M , information state s and assignment g into M with domain $X \supseteq FV(\varphi)$, we have

$$M, s \models_g \varphi \iff \mathcal{M}^\sharp \models_{T_s^g} \varphi^\sharp.$$

PROOF. We prove the following, more general claim: in the above hypotheses, for all $m \in \mathbb{N}$, for all $i_1, \dots, i_m \in \mathbb{N}$, and $d_1, \dots, d_m \in D$,

$$M, s \models_{g[x_{i_1} \mapsto d_1, \dots, x_{i_m} \mapsto d_m]} \varphi \iff \mathcal{M}^\sharp \models_{T_s^g[x_{i_1+1} \mapsto d_1, \dots, x_{i_m+1} \mapsto d_m]} \varphi^\sharp.$$

For convenience, we represent the state s as $s_{\mathcal{J}} = \{w_j \mid j \in \mathcal{J}\}$ and the corresponding team T_s^g as $T_{\mathcal{J}} = \{g_j \mid j \in \mathcal{J}\}$, and we also adopt a similar notation for their subsets: for any $\mathcal{J}' \subseteq \mathcal{J}$, we let $s_{\mathcal{J}'} = \{w_j \mid j \in \mathcal{J}'\}$ and $T_{\mathcal{J}'} = \{g_j \mid j \in \mathcal{J}'\}$. Additionally, we introduce some abbreviations:

- $g^* := g[x_{i_1} \mapsto d_1, \dots, x_{i_m} \mapsto d_m]$
- $g_j^* := g_j[x_{i_1+1} \mapsto d_1, \dots, x_{i_m+1} \mapsto d_m]$
- $T_{\mathcal{J}}^* := T_{\mathcal{J}}[x_{i_1+1} \mapsto d_1, \dots, x_{i_m+1} \mapsto d_m] = \{g_j^* \mid j \in \mathcal{J}\}$
- $T_{\mathcal{J}'}^* := T_{\mathcal{J}'}[x_{i_1+1} \mapsto d_1, \dots, x_{i_m+1} \mapsto d_m] = \{g_j^* \mid j \in \mathcal{J}'\}$

Using this notation, we can reformulate the above biconditional as

$$M, s_{\mathcal{J}} \models_{g^*} \varphi \iff \mathcal{M}^\sharp \models_{T_{\mathcal{J}}^*} \varphi^\sharp.$$

The claim is proved by induction on φ . We spell out four representative cases.

- Atomic case. Note that, for each $j \in \mathcal{J}$ and $i \in \mathbb{N}$, we have $g_j^*(x_{i+1}) = g^*(x_i)$. Using this fact, we have

$$\begin{aligned} & M, s_{\mathcal{J}} \models_{g^*} R_n(x_{k_1}, \dots, x_{k_n}) \\ \iff & \text{f.a. } j \in \mathcal{J}, \langle g^*(x_{k_1}), \dots, g^*(x_{k_n}) \rangle \in I_{w_j}(R_n) \\ \iff & \text{f.a. } j \in \mathcal{J}, \langle g_j^*(x_{k_1+1}), \dots, g_j^*(x_{k_n+1}) \rangle \in I_{w_j}(R_n) \\ \iff & \text{f.a. } j \in \mathcal{J}, \langle w_j, g_j^*(x_{k_1+1}), \dots, g_j^*(x_{k_n+1}) \rangle \in \mathcal{I}^\sharp(R'_{n+1}) \\ \iff & \text{f.a. } j \in \mathcal{J}, \langle g_j^*(x_0), g_j^*(x_{k_1+1}), \dots, g_j^*(x_{k_n+1}) \rangle \in \mathcal{I}^\sharp(R'_{n+1}) \\ \iff & \mathcal{M}^\sharp \models_{T_{\mathcal{J}}^*} R'_{n+1}(x_0, x_{k_1+1}, \dots, x_{k_n+1}) \end{aligned}$$

- Falsum. The claim follows from the fact that $T_{\mathcal{J}}^* = \emptyset$ iff $s_{\mathcal{J}} = \emptyset$.
- Inductive step for $\rightarrow$.

$$\begin{aligned}
 & M, s_{\mathcal{J}} \models_{g^*} \psi \rightarrow \xi \\
 \iff & \text{f.a. } t \subseteq s_{\mathcal{J}}, t \models_{g^*} \psi \text{ implies } t \models_{g^*} \xi \\
 \iff & \text{f.a. } \mathcal{J}' \subseteq \mathcal{J}, s_{\mathcal{J}'} \models_{g^*} \psi \text{ implies } s_{\mathcal{J}'} \models_{g^*} \xi \\
 \iff & \text{f.a. } \mathcal{J}' \subseteq \mathcal{J}, \mathcal{M}^{\sharp} \models_{T_{\mathcal{J}'}} \psi \text{ implies } \mathcal{M}^{\sharp} \models_{T_{\mathcal{J}'}} \xi \quad (\text{induction}) \\
 \iff & \text{f.a. } T \subseteq T_{\mathcal{J}}^*, \mathcal{M}^{\sharp} \models_T \psi \text{ implies } \mathcal{M}^{\sharp} \models_T \xi \\
 \iff & \mathcal{M}^{\sharp} \models_{T_{\mathcal{J}}^*} \psi \rightarrow \xi
 \end{aligned}$$

- Inductive step for $\forall$.⁵

$$\begin{aligned}
 & M, s_{\mathcal{J}} \models_{g^*} \forall x_h \psi \\
 \iff & \text{f.a. } d \in D, M, s_{\mathcal{J}} \models_{g^*[x_h \mapsto d]} \psi \\
 \iff & \text{f.a. } d \in D, \mathcal{M}^{\sharp} \models_{T_{\mathcal{J}}^*[x_{h+1} \mapsto d]} \psi^{\sharp} \quad (\text{induction}) \\
 \iff & \text{f.a. } d^{\sharp} \in \mathcal{D}^{\sharp}, d^{\sharp} \in D \text{ implies } \mathcal{M}^{\sharp} \models_{T_{\mathcal{J}}^*[x_{h+1} \mapsto d^{\sharp}]} \psi^{\sharp} \\
 \iff & \text{f.a. } d^{\sharp} \in \mathcal{D}^{\sharp}, \mathcal{M}^{\sharp} \models_{T_{\mathcal{J}}^*[x_{h+1} \mapsto d^{\sharp}]} \text{Ind}(x_{h+1}) \implies \mathcal{M}^{\sharp} \models_{T_{\mathcal{J}}^*[x_{h+1} \mapsto d^{\sharp}]} \psi^{\sharp} \\
 \stackrel{(!)}{\iff} & \text{f.a. } d^{\sharp} \in \mathcal{D}^{\sharp}, \mathcal{J}' \subseteq \mathcal{J}, \mathcal{M}^{\sharp} \models_{T_{\mathcal{J}'}^*[x_{h+1} \mapsto d^{\sharp}]} \text{Ind}(x_{h+1}) \implies \mathcal{M}^{\sharp} \models_{T_{\mathcal{J}'}^*[x_{h+1} \mapsto d^{\sharp}]} \psi^{\sharp} \\
 \iff & \text{f.a. } d^{\sharp} \in \mathcal{D}^{\sharp}, \mathcal{M}^{\sharp} \models_{T_{\mathcal{J}}^*[x_{h+1} \mapsto d^{\sharp}]} (\text{Ind}(x_{h+1}) \rightarrow \psi^{\sharp}) \\
 \iff & \mathcal{M}^{\sharp} \models_{T_{\mathcal{J}}^*} \forall x_{h+1} (\text{Ind}(x_{h+1}) \rightarrow \psi^{\sharp})
 \end{aligned}$$

(!): For the non-obvious $\implies$ direction, we reason by contraposition. Suppose there is a subset $\mathcal{J}' \subseteq \mathcal{J}$ such that the team $T_{\mathcal{J}'}^*[x_{h+1} \mapsto d^{\sharp}]$ satisfies $\text{Ind}(x_{h+1})$ but not $\psi^{\sharp}$. Then, the team $T_{\mathcal{J}}^*[x_{h+1} \mapsto d^{\sharp}]$ also satisfies $\text{Ind}(x_{h+1})$ (since the teams $T_{\mathcal{J}}^*[x_{h+1} \mapsto d^{\sharp}]$ and $T_{\mathcal{J}'}^*[x_{h+1} \mapsto d^{\sharp}]$ have the same restriction to $\text{FV}(\text{Ind}(x_{h+1})) = \{x_{h+1}\}$, namely, the singleton team $\{\epsilon[x_{h+1} \mapsto d^{\sharp}]\}$) but does not satisfy $\psi^{\sharp}$ (by persistency). $\square$

Characterizing the set of translated structures. A key property of the translation function $\sharp$ is the possibility of characterizing, with a set of InqBT formulas, the model-team pairs that can be obtained as $\sharp$ -images of some model-state-assignment triple.

Definition 3.5.

Given a signature Σ and a set of variables X , we define a set $\Gamma(\Sigma, X) \subseteq \mathcal{L}(\Sigma^{\sharp})$ as the set containing the following InqBT formulas:

- a formula expressing the fact that x_0 is always evaluated as a world:

$$\neg \text{Ind}(x_0)$$

⁵In the third step, note that we can apply the inductive hypothesis to $T_{\mathcal{J}}^*[x_i \mapsto d]$, since $T_{\mathcal{J}}^*[x_i \mapsto d] = T_{\mathcal{J}}[x_{i_1} \mapsto d_1, \dots, x_{i_m} \mapsto d_m][x_i \mapsto d] = T_{\mathcal{J}}[x_{i_1} \mapsto d_1, \dots, x_{i_m} \mapsto d_m, x_i \mapsto d]$.

12 Connecting state-based and team-based inquisitive logic

- formulas specifying that the $\sharp$ -translation of any variable of X is evaluated as an individual and in a constant way across the team; for every $x_n \in X$:

$$\text{Ind}(x_{n+1}) \quad \exists x(x_{n+1} = x)$$

- for every $R'_{n+1} \in \Sigma^\sharp$, a formula specifying that tuples in the extension of R'_{n+1} consist of one world followed by n individuals:

$$\forall x \forall y_1 \dots \forall y_n \left(R'_{n+1}(x, y_1, \dots, y_n) \rightarrow \neg \text{Ind}(x) \wedge \bigwedge_{i=1, \dots, n} \text{Ind}(y_i) \right).$$

Note that $FV(\Gamma(\Sigma, X)) = X^\sharp$, and that if both Σ and X are finite, $\Gamma(\Sigma, X)$ is a finite set. When Σ and X are clear from the context or irrelevant, we drop reference to them and simply write Γ instead of $\Gamma(\Sigma, X)$.

The following proposition says that any model–team pair that satisfies Γ is equivalent, on $\sharp$ -translations, to some model–team pair that arises as a $\sharp$ -image of a model–state–assignment triple in **InqBQ**. In fact, in this case we will prove something stronger: the model–team pairs satisfying Γ are *exactly* the $\sharp$ -images of these triples. However, we state the result in a weaker form since this formulation carries over to the general case of non-relational signatures, and gives us just enough to prove the entailment preservation result.

Proposition 3.6.

Let $\mathcal{M} = \langle \mathcal{D}, \mathcal{I} \rangle$ be a model of **InqBT** in the signature $\Sigma^\sharp$ and T a non-empty $X^\sharp$ -team over $\mathcal{M}$ such that $\mathcal{M} \models_T \Gamma(\Sigma, X)$. Then, there exists an **InqBQ** model $M := \langle D, W, I \rangle$ in the signature Σ , a state s and an assignment $g : X \rightarrow D$, such that for all formulas $\varphi \in \mathcal{L}(\Sigma, X)$:

$$\mathcal{M}^\sharp \models_{T_s^g} \varphi^\sharp \iff \mathcal{M} \models_T \varphi^\sharp.$$

PROOF. We define M , s and g as follows:

- $D := \mathcal{I}(\text{Ind})$
- $W := \mathcal{D} - \mathcal{I}(\text{Ind})$
- $I_w(R_n) := \{ \langle d_1 \dots, d_n \rangle \in D^n \mid \langle w, d_1, \dots, d_n \rangle \in \mathcal{I}(R'_{n+1}) \}$
- $s := \{g(x_0) \mid g \in T\}$
- $g(x_n) := g'(x_{n+1})$ where g' is an arbitrary element of T

The condition $\mathcal{M} \models_T \Gamma(\Sigma, X)$ ensures that these objects are well defined and satisfy the conditions for being, respectively, an **InqBQ** model, a state and an X -assignment.

Now, we show that $\mathcal{M}^\sharp = \mathcal{M}$ and $T_s^g = T$, which trivially implies the conclusion $\mathcal{M}^\sharp \models_{T_s^g} \varphi^\sharp \iff \mathcal{M} \models_T \varphi^\sharp$. To prove the first equality, we consider the two components of $\mathcal{M}^\sharp = \langle \mathcal{D}^\sharp, \mathcal{I}^\sharp \rangle$ and $\mathcal{M} = \langle \mathcal{D}, \mathcal{I} \rangle$ separately:

- $\mathcal{D}^\sharp = D \uplus W = \mathcal{I}(\text{Ind}) \uplus (\mathcal{D} - \mathcal{I}(\text{Ind})) = \mathcal{D}$
- Let $R'_{n+1} \in \Sigma^\sharp$ and $d_1, \dots, d_{n+1} \in (\mathcal{D}^\sharp)^{n+1} = \mathcal{D}^{n+1}$. Then,

$$\begin{aligned}
\langle d_1, \dots, d_{n+1} \rangle \in \mathcal{I}^\sharp(R'_{n+1}) &\iff d_1 \in W, \langle d_2, \dots, d_{n+1} \rangle \in I_{d_1}(R_n) \\
&\iff d_1 \notin \mathcal{I}(\text{Ind}), d_2, \dots, d_{n+1} \in \mathcal{I}(\text{Ind}) \text{ and} \\
&\quad \langle d_1, d_2, \dots, d_{n+1} \rangle \in \mathcal{I}(R'_{n+1}) \\
&\iff \langle d_1, \dots, d_{n+1} \rangle \in \mathcal{I}(R'_{n+1}),
\end{aligned}$$

where the last equivalence is guaranteed by the fact that $\mathcal{M} \models_T \Gamma(\Sigma, X)$. From this, we get $\mathcal{I}^\sharp(R'_{n+1}) = \mathcal{I}(R'_{n+1})$. Since R'_{n+1} was picked arbitrarily, we get $\mathcal{I}^\sharp \equiv \mathcal{I}$.

To prove $T_s^g = T$, we show that $T_s^g \subseteq T$ and $T_s^g \supseteq T$.

$$\begin{aligned}
(\subseteq): \quad g_w \in T_s^g &\implies \begin{cases} g_w(x_0) = w \text{ for some } w \in s \\ g_w(x_{n+1}) = g(x_n) \text{ for all } x_{n+1} \in X^\sharp \end{cases} \\
&\implies \begin{cases} \text{for some } g' \in T, g_w(x_0) = w = g'(x_0) \\ \text{for all } g' \in T, x_{n+1} \in X^\sharp, g_w(x_{n+1}) = g(x_n) = g'(x_{n+1}) \end{cases} \\
&\implies g_w = g' \text{ for some } g' \in T \\
(\supseteq): \quad g' \in T &\implies \begin{cases} g'(x_0) = w \text{ for some } w \in s \\ g'(x_{n+1}) = g(x_n) \text{ for all } x_{n+1} \in X^\sharp \end{cases} \\
&\implies \begin{cases} \text{for some } w \in s, g'(x_0) = w = g_w(x_0) \\ \text{for all } w \in s, x_{n+1} \in X^\sharp, g'(x_{n+1}) = g(x_n) = g_w(x_{n+1}) \end{cases} \\
&\implies g' = g_w \text{ for some } w \in s \\
&\iff g' = g_w \text{ for some } g_w \in T_s^g.
\end{aligned}$$

□

Translating entailments. We are now ready to show that entailments between certain formulas in InqBQ correspond to entailments between their translations in InqBT, given the formulas in the set Γ as auxiliary premises.

Theorem 3.7.

Let Σ be a relational signature without identity and $\Phi \cup \{\psi\} \subseteq \mathcal{L}(\Sigma)$. For any $X \supseteq FV(\Phi \cup \{\psi\})$,

$$\Phi \models_{\text{InqBQ}} \psi \iff \Gamma(\Sigma, X), \Phi^\sharp \models_{\text{InqBT}} \psi^\sharp.$$

PROOF. Suppose $\Phi \not\models_{\text{InqBQ}} \psi$. Then, there exists a model M , a state s and an assignment g such that $M, s \models_g \Phi$ and $M, s \not\models_g \psi$. By Proposition 3.4, we get

$$\begin{aligned}
M, s \models_g \Phi &\iff \mathcal{M}^\sharp \models_{T_s^g} \Phi^\sharp \\
M, s \not\models_g \psi &\iff \mathcal{M}^\sharp \not\models_{T_s^g} \psi^\sharp.
\end{aligned}$$

Since $\mathcal{M}^\sharp \models_{T_s^g} \Gamma(\Sigma, \text{Var})$ holds by construction, the same must be true for $\Gamma(\Sigma, X)$ when $X \subseteq \text{Var}$. Thus, for any $X \supseteq FV(\Phi \cup \{\psi\})$, we have a counterexample showing that $\Gamma(\Sigma, X), \Phi^\sharp \not\models_{\text{InqBT}} \psi^\sharp$.

Conversely, suppose $\Gamma(\Sigma, X), \Phi^\sharp \not\models_{\text{InqBT}} \psi^\sharp$ for some $X \supseteq FV(\Phi \cup \{\psi\})$. Then, there exists a model $\mathcal{M}$ and a non-empty $X^\sharp$ -team T such that $\mathcal{M} \models_T \Gamma(\Sigma, X)$, $\mathcal{M} \models_T \Phi^\sharp$ and $\mathcal{M} \not\models_T \psi^\sharp$. Since $\mathcal{M} \models_T \Gamma(\Sigma, X)$, we may define M , s and g as in Lemma 3.6. By Proposition 3.4 and Lemma 3.6,

we have

$$\begin{aligned}\mathcal{M} \models_T \Phi^\sharp &\iff \mathcal{M}^\sharp \models_{T_s^g} \Phi^\sharp \iff M, s \models_g \Phi \\ \mathcal{M} \not\models_T \psi^\sharp &\iff \mathcal{M}^\sharp \not\models_{T_s^g} \psi^\sharp \iff M, s \not\models_g \psi.\end{aligned}$$

Thus, we have a counterexample showing that $\Phi \not\models_{\text{InqBQ}} \psi$. $\square$

3.2 Translating from InqBT to InqBQ

Similarly to how we proceeded above, we construct a translation $(\cdot)^b$ from InqBT to InqBQ by simulating the semantics of InqBT in InqBQ. To do this, we encode an InqBT model $\mathcal{M}$ and a team T as a corresponding InqBQ model M_T^b , in such a way that the semantics of φ^b with respect to M_T^b mirrors that of φ under $\mathcal{M}$ and T .

Translating models and teams. In InqBT we have teams, where variables are interpreted differently in different assignments. To simulate this behaviour in InqBQ, where we only have a single assignment, we replace variables x_i by corresponding non-rigid constants a_i , and encode each assignment g in the team as a possible world w_g where the denotation of a_i is $g(x_i)$. The interpretation of relation symbols may simply be taken to be the same in each of these worlds.

Definition 3.8.

Let Σ be a relational signature without identity and X a set of variables that we choose to represent as $X = \{x_i \mid i \in \mathcal{K}\}$. We define a new signature $\Sigma_X^b := \Sigma \cup \{a_i \mid i \in \mathcal{K}\}$, where the a_i are non-rigid constants. Whenever possible, we omit the reference to X and simply write Σ^b .

Definition 3.9.

Let $\mathcal{M} := \langle \mathcal{D}, \mathcal{I} \rangle$ be a model of InqBT in the signature Σ and let T be a team for $\mathcal{M}$. We define the model $M_T^b = \langle D^b, W^b, I^b \rangle$ of InqBQ in the signature Σ^b as follows⁶:

- $D^b := \mathcal{D}$
- $W^b := \{w_g \mid g \in T\}$

and, for any $w_g \in W^b$,

- $I_{w_g}^b(R) := \mathcal{I}(R)$ for each $R \in \Sigma^b$
- $I_{w_g}^b(a_i) := g(x_i)$ for each $i \in \mathcal{K}$

For a subteam $T' \subseteq T$, we define $s_{T'} \subseteq W^b$ to be the information state $s_{T'} = \{w_g \mid g \in T'\}$.

Translating formulas. In accordance with our encoding of teams, we translate formulas by replacing each free occurrence of a variable with its associated constant. The clauses for the quantifiers make sure that bound occurrences of variables are not replaced.

⁶One may, like one of the reviewers, be puzzled by the asymmetry between the two translations, as an InqBT model–team pair is translated by an InqBQ model, while an InqBQ model is described as translated just by an InqBT model. In fact, the asymmetry is only apparent. InqBQ models and states are in a one to one correspondence, as each model is uniquely associated with the state given by its universe and, conversely, a state s in a model M behaves exactly like the local restriction $M|_s$. As a result, from an InqBQ model, the translation $\sharp$ determines both an InqBT model (directly) and a team (via the associated state).

Definition 3.10.

For any $\varphi \in \mathcal{L}(\Sigma, X)$, we define $\varphi^b \in \mathcal{L}(\Sigma_X^b)$ to be the result of replacing each free occurrence of a variable x_i in φ by the constant a_i :

$$\varphi(x_{i_1}, \dots, x_{i_n})^b = \varphi(a_{i_1}, \dots, a_{i_n}).$$

Note that for each φ , its translation φ^b is a sentence, and as such, its interpretation does not depend on the assignment under consideration.

The key property of the translation is expressed by the following proposition.

Proposition 3.11.

For any InqBT model $\mathcal{M}$, $X \subseteq \text{Var}$, X -team T , subteam $T' \subseteq T$ and formula $\varphi \in \mathcal{L}(\Sigma, X)$,

$$\mathcal{M} \models_{T'} \varphi \iff M_{T, S_{T'}}^b \models \varphi^b.$$

PROOF. First, let g be an arbitrary X -assignment over the model M_T^b . We prove the following more general claim: in the above hypotheses, for all $m \in \mathbb{N}$, $i_1, \dots, i_m \in \mathbb{N}$ and $d_1, \dots, d_m \in D$,

$$\mathcal{M} \models_{T'[x_{i_1} \mapsto d_1, \dots, x_{i_m} \mapsto d_m]} \varphi \iff M_{T, S_{T'}}^b \models_{g[x_{i_1} \mapsto d_1, \dots, x_{i_m} \mapsto d_m]} \varphi^b[x_{i_1}/a_{i_1}, \dots, x_{i_m}/a_{i_m}].$$

For convenience, we view the team T' and the corresponding state $s_{T'}$ (which are in a one-to-one correspondence via the map $g \mapsto w_g$) as indexed over the same set $\mathcal{J}$, and thus represent them, respectively, as $T_{\mathcal{J}} = \{g_j \mid j \in \mathcal{J}\}$ and $s_{\mathcal{J}} = \{w_j \mid j \in \mathcal{J}\}$. We also make use of the following abbreviations:

- $g^* := g[x_{i_1} \mapsto d_1, \dots, x_{i_m} \mapsto d_m]$
- $g_j^* := g_j[x_{i_1} \mapsto d_1, \dots, x_{i_m} \mapsto d_m]$
- $T_{\mathcal{J}}^* := T_{\mathcal{J}}[x_{i_1} \mapsto d_1, \dots, x_{i_m} \mapsto d_m] = \{g_j^* \mid j \in \mathcal{J}\}$
- $\sigma[\bar{x}/\bar{a}] := \sigma[x_{i_1}/a_{i_1}, \dots, x_{i_m}/a_{i_m}]$, where σ is a term or a formula

Using this notation, the above biconditional now reads

$$\mathcal{M} \models_{T_{\mathcal{J}}^*} \varphi \iff M_{T, s_{\mathcal{J}}}^b \models_{g^*} \varphi^b[\bar{x}/\bar{a}].$$

We prove the claim by induction on φ . We only spell out the atomic case, as the remaining cases are straightforward. First note that for all $j \in \mathcal{J}$ and $x_i \in X$, the value of g_j^* on the i -th variable x_i coincides with the referent under g^* and at w_j , of the corresponding constant a_i (see (2.1) in Section 2), after performing the substitution $\bar{x}/\bar{a}$

$$g_j^*(x_i) = [a_i[\bar{x}/\bar{a}]]_{g^*}^{w_j} \quad (3.1)$$

since we have that, for any $x_i \in X$,

$$g_j^*(x_i) = \begin{cases} d_i = g^*(x_i) = [x_i]_{g^*}^{w_j} = [a_i[\bar{x}/\bar{a}]]_{g^*}^{w_j} & \text{if } i \in \{i_1, \dots, i_m\} \\ I_{w_j}^b(a_i) = [a_i]_{g^*}^{w_j} = [a_i[\bar{x}/\bar{a}]]_{g^*}^{w_j} & \text{if } i \notin \{i_1, \dots, i_m\}. \end{cases}$$

Using this fact, we can prove the atomic case of our inductive proof as follows:

$$\begin{aligned}
& \mathcal{M} \models_{T^*_{\mathcal{J}}} R(x_{k_1}, \dots, x_{k_n}) \\
& \iff \text{f.a. } j \in \mathcal{J}, \langle g_j^*(x_{k_1}), \dots, g_j^*(x_{k_n}) \rangle \in I(R) \\
& \iff \text{f.a. } j \in \mathcal{J}, \langle [(a_{k_1})[\bar{x}/\bar{a}]]_{g^*}^{w_j}, \dots, [(a_{k_n})[\bar{x}/\bar{a}]]_{g^*}^{w_j} \rangle \in I(R) \quad \text{by (3.1)} \\
& \iff M_{T, s_{\mathcal{J}}}^b \models_{g^*} R(a_{k_1}[\bar{x}/\bar{a}], \dots, a_{k_n}[\bar{x}/\bar{a}]) \\
& \iff M_T^b, s_{\mathcal{J}} \models_{g^*} R(a_{k_1}, \dots, a_{k_n})[\bar{x}/\bar{a}] \\
& \iff M_T^b, s_{\mathcal{J}} \models_{g^*} (R(x_{k_1}, \dots, x_{k_n}))^b[\bar{x}/\bar{a}].
\end{aligned}$$

□

Characterizing the set of translated structures. Again, we can identify a set of **InqBQ** formulas that characterize those **InqBQ** models M that behave, with respect to $(\cdot)^b$ -translations, just like b -images of **InqBT** model–team pairs.

Definition 3.12.

Given a signature Σ , we define a set of **InqBQ** formulas $\Delta(\Sigma)$ (or simply Δ when no confusion arises) as the set containing, for each predicate $R \in \Sigma^b$, a formula specifying that R is evaluated in the same way at each world:

$$\forall x_1 \dots \forall x_n ?R(x_1, \dots, x_n).$$

Clearly, any b -image of an **InqBT** model–team pair supports Δ . The following proposition says that, conversely, any model–state pair that supports Δ is equivalent, on b -translations, to the b -image of a model–team pair.

Proposition 3.13.

Let $M = \langle D, W, I \rangle$ be a model of **InqBQ** in the signature Σ_X^b such that $M \models \Delta$. Then, there exists a model $\mathcal{M} := \langle \mathcal{D}, \mathcal{I} \rangle$ and an X -team T in the signature Σ such that for any $\varphi \in \mathcal{L}(\Sigma, X)$ and for any assignment g :

$$M \models_g \varphi^b \iff M_T^b \models_g \varphi^b.$$

PROOF. We define $\mathcal{M}$ and T as follows:

- $\mathcal{D} := D$
- $\mathcal{I}(R) := I_w(R)$, where w is an arbitrary element of W
- $g_w : X \rightarrow \mathcal{D}$ defined by $g_w(x_i) := I_w(a_i)$ f.a. $i \in \mathcal{K}$, f.a. $w \in W$
- $T := \{g_w \mid w \in W\}$

The fact that $M \models \Delta$ guarantees that the definition of $\mathcal{I}(R)$ does not depend on the particular choice of w . We leave it to the reader to verify that $M_T^b = M^e$, where M^e is the essential quotient of M (cf. Definition 2.4). By Proposition 2.5, this implies that for all X -assignments g and formulas $\psi \in \mathcal{L}(\Sigma^b, X)$, $M \models_g \psi \iff M_T^b \models_g \psi$. Our conclusion follows as a special case when $\psi = \varphi^b$. □

Translating entailments. We are now ready to state how **InqBT** entailments between certain formulas are equivalent to the **InqBQ** entailments of their translations, provided we include the formulas in Δ as premises.

Theorem 3.14.

For any relational signature without identity Σ and any set of formulas $\Phi \cup \{\psi\} \subseteq \mathcal{L}(\Sigma)$,

$$\Phi \models_{\text{InqBT}} \psi \iff \Delta(\Sigma), \Phi^b \models_{\text{InqBQ}} \psi^b.$$

The proof of this result is analogous to that of Theorem 3.7.

3.3 Generalizing

In this section we sketch how the translations can be extended to the case of signatures Σ , which may contain function symbols, as well as the identity symbol. The extension is relatively straightforward, so we only describe how to extend the definitions, omitting the proofs. A thorough presentation can be found in [11].

Translating InqBQ to InqBT, general case. Given an arbitrary signature Σ for InqBQ, we extend the definition of $\Sigma^\sharp$ with the following items:

- a ternary relation symbol $R^\equiv$, to encode the identity predicate;
- for each n -ary rigid function symbol $h_n \in \Sigma$, an n -ary function symbol h'_n ;
- for each n -ary non-rigid function symbol $f_n \in \Sigma$, an $n + 1$ -ary function symbol f'_{n+1} .

Given an InqBQ model $M = \langle D, W, I, \sim \rangle$, we define its encoding $\mathcal{M}^\sharp$ by supplementing Definition 3.2 with the following clauses:

- $\mathcal{I}^\sharp(R^\equiv) = \{ \langle w, d, d' \rangle \mid w \in W, d \sim_w d' \}$
- $\mathcal{I}^\sharp(h'_n) = I_w(h_n)$ for an arbitrary $w \in W$
- $\mathcal{I}^\sharp(f'_n) = F$ for an arbitrary function $F : (\mathcal{D}^\sharp)^{n+1} \rightarrow \mathcal{D}^\sharp$ that satisfies, for all $w \in W$ and $d_1, \dots, d_n \in D$:

$$F(w, d_1, \dots, d_n) = I_w(f)(d_1, \dots, d_n).$$

Next, we define the translation of an arbitrary term t of the signature Σ :

- $(x_i)^\sharp := x_{i+1}$
- $(h_n(t_1, \dots, t_n))^\sharp := h'_n((t_1)^\sharp, \dots, (t_n)^\sharp)$
- $(f_n(t_1, \dots, t_n))^\sharp := f'_{n+1}(x_0, (t_1)^\sharp, \dots, (t_n)^\sharp)$

Finally, we generalize the translation $(\cdot)^\sharp$ given in Definition 3.3 by giving the following translation for atomic formulas (everything else remains unchanged):

- $(R_n(t_1, \dots, t_n))^\sharp := R'_{n+1}(x_0, (t_1)^\sharp, \dots, (t_n)^\sharp)$
- $(t = t')^\sharp := R^\equiv(x_0, (t)^\sharp, (t')^\sharp)$

We can then show that with these definitions, Proposition 3.4 extends: for every model M , state s , X -assignment g and formula $\varphi \in \mathcal{L}(\Sigma, X)$, we have

$$M, s \models_g \varphi \iff \mathcal{M}^\sharp \models_{T_s^g} \varphi^\sharp. \quad (3.2)$$

To characterize the models induced by the translation, we need to supplement the definition of Γ with some extra formulas, which specify:

- that the interpretation of $R^\equiv$ must relate a world and a pair of individuals;
- that for each particular world, $R^\equiv$ induces an equivalence relation on individuals that is a congruence with respect to the symbols in the signature;

- that the value of the functions h' on tuples of individuals must be an individual;
- that the value of the functions f' on tuples consisting of a world and a sequence of individuals must be an individual.

The relevant formulas are given explicitly in Appendix A.

The generalized translation $\sharp$ verifies entailment preservation with respect to the extended set of characteristic formulas (which we also call $\Gamma(\Sigma, X)$ here).

Theorem 3.15.

Let $\Phi \cup \{\psi\} \subseteq \mathcal{L}(\Sigma, X)$. Then,

$$\Phi \models_{\text{InqBQ}} \psi \iff \Gamma(\Sigma, X), \Phi^\sharp \models_{\text{InqBT}} \psi^\sharp.$$

The proof of this result relies on a generalization of Proposition 3.6, showing that any model-team pair that satisfies $\Gamma(\Sigma, X)$ in InqBT is equivalent, on $\sharp$ -translations, to the $\sharp$ -image of a model-state-assignment triple.

Translating InqBT to InqBQ in the general case: Given $X \subseteq \text{Var}$ and a signature Σ , we extend the definition of Σ_X^b by adding

- for each function symbol f , a rigid function symbol $\mathbf{f}$

Given an InqBT model $\mathcal{M} = \langle \mathcal{D}, \mathcal{I} \rangle$, we define an InqBQ model M^b by extending Definition 3.9 with two additional clauses: for any $w_g \in W^b$,

- $I_{w_g}^b(\mathbf{f}) = \mathcal{I}(f)$ for each $\mathbf{f} \in \Sigma^b$
- $d(\sim^b)_{w_g} d' \iff d = d'$

Given a term t , its translation $(t)^b$ is obtained by replacing all occurrences in t of each variable x_i with the constant a_i . For the formulas, we integrate the clauses for atomic formulas to include terms and the equality symbol:

- $(R(t_1, \dots, t_n))^b = R((t_1)^b, \dots, (t_n)^b)$
- $(t_1 = t_2)^b = ((t_1)^b = (t_2)^b)$

These definitions validate the preservation of support: for any InqBT model $\mathcal{M}$, X -team T , subteam $T' \subseteq T$ and formula $\varphi \in \mathcal{L}(\Sigma, X)$,

$$\mathcal{M} \models_{T'} \varphi \iff M_{T', S_{T'}}^b \models \varphi^b.$$

To characterize the structures generated by b , we need to extend the set Δ with a formula specifying that identity is interpreted in the same way at every world:

$$\forall x \forall y ?(x = y).$$

With this extension of Δ , we can prove that entailment preservation holds for the generalized translation b .

Theorem 3.16.

Let $\Phi \cup \{\psi\} \subseteq \mathcal{L}(\Sigma)$. Then,

$$\Phi \models_{\text{InqBT}} \psi \iff \Delta(\Sigma), \Phi^b \models_{\text{InqBQ}} \psi^b.$$

4 Repercussions

The main pay-off of having entailment-preserving translations between two logics is that they facilitate the transfer of results from one to the other. This is also the case for InqBQ and InqBT . In this section, we demonstrate the usefulness of the translations by establishing the equivalence of major open problems about the two systems and by transferring notable results about some fragments of InqBQ to their InqBT counterparts.

4.1 Equivalence of open problems

The first open problem we consider is entailment compactness. Recall that a logical system $\mathbf{S}$ is compact with respect to its entailment relation (say, $\models_{\mathbf{S}}$) if, for any set $\Phi \cup \{\psi\}$ of formulas of $\mathbf{S}$, whenever $\Phi \models_{\mathbf{S}} \psi$ holds then there is a finite subset $\Phi_0 \subseteq \Phi$ such that $\Phi_0 \models_{\mathbf{S}} \psi$ holds. As we mentioned in the introduction, it is currently unknown whether InqBQ and InqBT are entailment compact. Using the translations, we can show that these two problems are equivalent.

Proposition 4.1.

Entailment compactness holds for InqBQ if and only if it holds for InqBT .

PROOF. Suppose entailment compactness holds for InqBQ . To prove that it also holds for InqBT , we reason as follows:

$$\begin{aligned}
 \Phi \models_{\text{InqBT}} \psi &\stackrel{(3.16)}{\iff} \Delta, \Phi^b \models_{\text{InqBQ}} \psi^b \\
 &\stackrel{(\text{hyp})}{\iff} \text{f.s. finite } \Delta_0 \subseteq \Delta \text{ and } (\Phi^b)_0 \subseteq \Phi^b : \Delta_0, (\Phi^b)_0 \models_{\text{InqBQ}} \psi^b \\
 &\implies \text{f.s. finite } (\Phi^b)_0 \subseteq \Phi^b : \Delta, (\Phi^b)_0 \models_{\text{InqBQ}} \psi^b \\
 &\iff \text{f.s. finite } \Phi_0 \subseteq \Phi : \Delta, (\Phi_0)^b \models_{\text{InqBQ}} \psi^b \\
 &\stackrel{(3.16)}{\iff} \text{f.s. finite } \Phi_0 \subseteq \Phi : \Phi_0 \models_{\text{InqBT}} \psi.
 \end{aligned}$$

The proof of the converse implication is completely analogous. $\square$

Another fundamental metatheoretic property that we are able to connect across the two systems is the recursive enumerability of the set of validities. To prove this, we first need to note two immediate corollaries of Theorems 3.15 and 3.16.

For a formula ψ , let Σ_ψ to be the signature that contains only those non-logical items that appear in ψ , and let $X_\psi = FV(\psi)$ be the set of free variables in ψ . Note that $\mathcal{L}(\Sigma_\psi, X_\psi)$ is the smallest language that includes ψ .

Corollary 4.2.

Let ψ be a formula of InqBQ . By Theorem 3.15,

$$\models_{\text{InqBQ}} \psi \iff \Gamma(\Sigma_\psi, X_\psi) \models_{\text{InqBT}} \psi^\# \iff \models_{\text{InqBT}} \bigwedge \Gamma(\Sigma_\psi, X_\psi) \rightarrow \psi^\#.$$

Note that the conjunction on the right-hand side is well defined since Σ_ψ and X_ψ are finite and, therefore, so is $\Gamma(\Sigma_\psi, X_\psi)$. An analogous result holds for the translation b (in this case, the set Δ does not depend on the free variables in ψ).

Corollary 4.3.

Let ψ be a formula of **InqBT**. By Theorem 3.16,

$$\models_{\text{InqBT}} \psi \iff \Delta(\Sigma_\psi) \models_{\text{InqBQ}} \psi^b \iff \models_{\text{InqBQ}} \bigwedge \Delta(\Sigma_\psi) \rightarrow \psi.$$

We can now use these corollaries to establish the following result.

Proposition 4.4.

The validities of **InqBQ** are recursively enumerable if and only if the validities of **InqBT** are recursively enumerable.

PROOF. We show that, given a formula in either one of the systems, there is a computable way to determine a formula of the other system, which is a validity if and only if the original formula is.

For the right-to-left direction, Corollary 4.2 implies that for any $\psi \in \mathcal{L}$, ψ is valid in **InqBQ** if and only if $\bigwedge \Gamma(\Sigma_\psi, X_\psi) \rightarrow \psi^\sharp$ is valid in **InqBT**. Note that $\bigwedge \Gamma(\Sigma_\psi, X_\psi) \rightarrow \psi^\sharp$ is computable from ψ since (i) $\psi^\sharp$ is computable from ψ , and (ii) the set $\Gamma(\Sigma_\psi, X_\psi)$ is both finite and computable from ψ .

The left-to-right direction is proved analogously, using Corollary 4.3. $\square$

The previous proposition generalizes to the case of valid entailments having recursively enumerable premises.⁷

Proposition 4.5.

The following are equivalent:

- For any recursively enumerable set of formulas Φ , the set $\{\psi \mid \Phi \models_{\text{InqBQ}} \psi\}$ is recursively enumerable.
- For any recursively enumerable set of formulas Φ , the set $\{\psi \mid \Phi \models_{\text{InqBT}} \psi\}$ is recursively enumerable.

PROOF. First, notice that, in both claims, the set of signature symbols and variables appearing in $\Phi \cup \{\psi\}$ must be recursively enumerable. Let us write $\Sigma_{\Phi, \psi}$ and $X_{\Phi, \psi}$ to denote, respectively, these two sets. Clearly, $\Phi \cup \psi \subseteq \mathcal{L}(\Sigma_{\Phi, \psi}, X_{\Phi, \psi})$.

We prove only the implication from the first to the second claim. The proof of the converse is completely analogous. Let Φ be a recursively enumerable set of formulas. We show that the set of its **InqBT** consequences $\{\psi \mid \Phi \models_{\text{InqBT}} \psi\}$ is recursively enumerable. Assume that $\Phi \models_{\text{InqBT}} \psi$. By Theorem 3.16, this holds if and only if the entailment $\Phi^b, \Delta(\Sigma_{\Phi, \psi}) \models_{\text{InqBQ}} \psi^b$ is valid. The set of premises of the new entailment is still recursively enumerable, since the procedure transforming a formula of Φ into its translation is computable, and the set $\Delta(\Sigma_{\Phi, \psi})$ contains at most one formula for each symbol of the recursively enumerable signature $\Sigma_{\Phi, \psi}$. Therefore, by hypothesis, the set $\{\xi \mid \Phi^b, \Delta(\Sigma_{\Phi, \psi}) \models_{\text{InqBQ}} \xi\}$ is recursively enumerable. To know whether the original ψ is entailed by Φ , it suffices to check whether its b -translation ψ^b appears in the enumeration of this set. $\square$

4.2 Transferring properties of fragments

We now turn our attention to some notable fragments of the two systems. The fragments of **InqBQ** have been an area of great research interest in recent years. We will focus in particular on the Rex and Clant syntactical fragments, for which an axiomatization has been determined.

⁷We thank an anonymous reviewer for suggesting this generalization.

The section's contents will be structured as follows: first we define a fragment, property or concept, then we outline known results of **InqBQ** that involve said definition and, finally, we show how these results can be transferred to **InqBT** by using the translations. We will omit proofs of known properties of the **InqBQ** fragments. We refer to [6] and [4] for a more complete treatment of the Rex fragment and of the Clant fragment, respectively.

Rex fragment. The *restricted existential* fragment (Rex for short) is the fragment of $\mathcal{L}$ where inquisitive existentials can only appear in the antecedent of an implication. More formally, if Σ is a signature (possibly including function symbols and the identity symbol), the set of rex formulas $\mathcal{L}_{\text{Rex}}(\Sigma)$ is given by the following syntax:

$$\varphi ::= p \mid \perp \mid \varphi \wedge \psi \mid \psi \rightarrow \varphi \mid \varphi \vee \psi \mid \forall x \varphi,$$

where p ranges over atomic formulas in the signature Σ and ψ ranges over arbitrary formulas of $\mathcal{L}(\Sigma)$. It will be useful to note that, by definition, an implication is a rex formula if and only if its consequent is a rex formula.

The Rex fragment of **InqBQ** has been studied in detail in [6], where a number of properties of this fragment have been established. We will now show how these results can be transferred to the Rex fragment of **InqBT**, using the translations.

We start by observing that both our translations map rex formulas to rex formulas. This can be verified by noting that the translations never add or remove an inquisitive existential quantifier, and do not move any subformula away from the antecedent of an implication.

Proposition 4.6.

For any $\varphi \in \mathcal{L}(\Sigma)$,

$$\begin{aligned} \varphi \in \mathcal{L}_{\text{Rex}}(\Sigma) &\iff \varphi^\sharp \in \mathcal{L}_{\text{Rex}}(\Sigma^\sharp), \\ \varphi \in \mathcal{L}_{\text{Rex}}(\Sigma) &\iff \varphi^\flat \in \mathcal{L}_{\text{Rex}}(\Sigma^\flat). \end{aligned}$$

We also introduce the notion of *coherence* for both **InqBQ** and **InqBT**. Coherence, first introduced in [21], and recently investigated in [6], is a highly consequential property that formulas can possess in state-/team-based logics.

Definition 4.7.

For a cardinal k and a formula $\varphi \in \mathcal{L}$, we say that φ is k -coherent in **InqBQ** if, for all models M , states s and assignments g ,

$$M, s \models_g \varphi \iff (\text{f.a. } t \subseteq s \text{ s.t. } \#t \leq k, M, t \models_g \varphi),$$

where with $\#X$ we indicate the cardinality of the set X . Similarly, we say that φ is k -coherent in **InqBT** if, for all models M and for all teams T ,

$$\mathcal{M} \models_T \varphi \iff (\text{f.a. } T' \subseteq T \text{ s.t. } \#T' \leq k, \mathcal{M} \models_{T'} \varphi).$$

In the following sense, the translations interact well with coherence:⁸

⁸A less general converse of this proposition can also be obtained: k -coherence of a formula implies k -coherence of its $\sharp$ -translation (resp. $\flat$ -translation) only in restriction to Γ -structures (resp. Δ -structures). We choose to omit it as its formulation is less natural and the result is not needed for the remainder of this paper.

Proposition 4.8.

For any formula $\varphi \in \mathcal{L}$, for any cardinal k ,

$$\varphi^\sharp \text{ is } k\text{-coherent in } \text{InqBT} \implies \varphi \text{ is } k\text{-coherent in } \text{InqBQ},$$

$$\varphi^\flat \text{ is } k\text{-coherent in } \text{InqBQ} \implies \varphi \text{ is } k\text{-coherent in } \text{InqBT}.$$

PROOF. We only provide a proof of the first claim, as the proof of the second claim is completely analogous. Suppose that $\varphi^\sharp$ is k -coherent in InqBT . Let M be a model of InqBQ and let g be an assignment over M . Then, for all states $s_{\mathcal{J}}$ in the model $\mathcal{M}^\sharp$, which we view as indexed over a set $\mathcal{J}$ as in Proposition 3.4,

$$\begin{aligned} M, s_{\mathcal{J}} \models_g \varphi &\iff \mathcal{M}^\sharp \models_{T_{\mathcal{J}}} \varphi^\sharp \\ &\stackrel{(k\text{-C.})}{\iff} \text{f.a. } T \subseteq T_{\mathcal{J}} \text{ with } \#T \leq k, \mathcal{M}^\sharp \models_T \varphi^\sharp \\ &\iff \text{f.a. } \mathcal{J}' \subseteq \mathcal{J} \text{ with } \#\mathcal{J}' \leq k, \mathcal{M}^\sharp \models_{T_{\mathcal{J}'}} \varphi^\sharp \\ &\iff \text{f.a. } \mathcal{J}' \subseteq \mathcal{J} \text{ with } \#\mathcal{J}' \leq k, M, s_{\mathcal{J}'} \models_g \varphi \\ &\iff \text{f.a. } s \subseteq s_{\mathcal{J}} \text{ with } \#s \leq k, M, s \models_g \varphi. \end{aligned}$$

where the first and fourth equivalence follow from Equation 3.2. This proves k -coherence in InqBQ for φ . $\square$

Having established some general properties of the translations, we now provide an overview of a number of known results about the Rex fragment of InqBQ , which we will later transfer to InqBT using the translations. As mentioned above, the properties of Rex in InqBQ have been studied extensively in [6], to which we refer for a complete treatment of these matters.

We start by stating that all rex formulas are finitely coherent.

Proposition 4.9. (see [6], Proposition 5.3)

For every formula $\varphi \in \mathcal{L}_{\text{Rex}}$ there is a natural number n_φ , computable from the syntax of φ , such that φ is n_φ -coherent in InqBQ . More specifically, n_φ can be defined in the following way for $\varphi \in \mathcal{L}_{\text{Rex}}$:

- $n_\varphi = 1$ for any atomic formula
- $n_{(\psi \wedge \xi)} = \max\{n_\psi, n_\xi\}$
- $n_{(\psi \vee \xi)} = n_\psi + n_\xi$
- $n_{(\psi \rightarrow \xi)} = n_\xi$
- $n_{(\forall x_i \psi)} = n_\psi$

The following proposition states a crucial consequence of finite coherence, namely, entailment compactness for finitely coherent conclusions.

Proposition 4.10. (see [6], Theorem 4.3)

If $\Phi \models_{\text{InqBQ}} \psi$ and ψ is finitely coherent, there exists a finite subset $\Phi_0 \subseteq \Phi$ such that $\Phi_0 \models_{\text{InqBQ}} \psi$.

Combining the two results above, we immediately get the following.

Corollary 4.11. (Compactness for rex conclusions, InqBQ)

Entailment compactness holds for all InqBQ entailments where the conclusion is a rex formula.

Finally, rex validities are known to be recursively enumerable.

Proposition 4.12. (see [6], Theorem 5.4)

The set of valid rex formulas, $Val_{Rex}^Q := \{\varphi \in \mathcal{L}_{Rex} \mid \varphi \text{ is valid in InqBQ}\}$, is recursively enumerable.

These results can be straightforwardly transferred to InqBT with the use of the translations. We present the proofs of the relevant properties of Rex in InqBT following the order established above. We begin with the finite coherence of rex formulas in InqBT.

Proposition 4.13. (Finite coherence, InqBT)

For every formula $\varphi \in \mathcal{L}_{Rex}$ there is a natural number n_φ , computable from the syntax of φ , such that φ is n_φ -coherent in InqBT. For all formulas, the estimate given for InqBQ in Proposition 4.9 also works for InqBT.

PROOF. Both the coherence degree and its computability can easily be proved by induction, with the help of Proposition 4.8. $\square$

Entailments with rex conclusions satisfy compactness in InqBT.

Proposition 4.14. (Compactness for rex conclusions, InqBT)

Entailment compactness holds for all InqBT entailments where the conclusion is a rex formula.

PROOF. We leave the details of this proof to the reader, as it only involves a straightforward adaptation of the proof of the left-to-right direction of Proposition 4.1. The adapted proof makes use of the closure of Rex under the translations (Proposition 4.6) and of the fact that Δ only contains rex formulas. $\square$

The recursive enumerability of rex validities also transfers smoothly to InqBT.

Proposition 4.15. (Recursive enumerability of rex validities, InqBT)

The set $Val_{Rex}^T := \{\varphi \in \mathcal{L}_{Rex} \mid \varphi \text{ is valid in InqBT}\}$ is recursively enumerable.

PROOF. The proof of this result is an adaption of the proof of the left-to-right direction of Proposition 4.4, with the added information that Rex is closed under the translation $\flat$. We only need to prove that $\bigwedge \Delta(\Sigma_\varphi) \rightarrow \varphi^\flat$ is a rex formula when φ is. Indeed, $\bigwedge \Delta(\Sigma_\varphi) \rightarrow \varphi^\flat \in \mathcal{L}_{Rex} \iff \varphi^\flat \in \mathcal{L}_{Rex} \iff \varphi \in \mathcal{L}_{Rex}$. $\square$

Clant fragment. The *classical antecedent* fragment (Clant for short) is the fragment of $\mathcal{L}$ where only classical formulas can appear as antecedents of implications. Formally, given a signature Σ (possibly including function symbols and the identity symbol), the set of clant formulas $\mathcal{L}_{Clant}$ is given by the following syntax:

$$\varphi ::= p \mid \perp \mid \varphi \wedge \varphi \mid \alpha \rightarrow \varphi \mid \varphi \vee \varphi \mid \forall x \varphi \mid \exists \varphi,$$

where p ranges over the atomic formulas of Σ and α ranges over the classical formulas of Σ .

Various properties of the Clant fragment of InqBQ have been established in [16]. We will transfer some of these result to InqBT by use of the translations.

First, we show that Clant is closed under both translations. The proof of this result relies on the following lemma, which states that the set of classical formulas is itself closed under the translations.

Lemma 4.16.

Let $\varphi \in \mathcal{L}$, then

$$\varphi \text{ is a classical formula} \iff \varphi^\sharp \text{ is a classical formula,}$$

$$\varphi \text{ is a classical formula} \iff \varphi^\flat \text{ is a classical formula.}$$

PROOF. It suffices to observe that the translations $\sharp$ and $\flat$ neither add nor remove any inquisitive symbols. Using this fact, both claims follow by induction. $\square$

Proposition 4.17.

Let Σ be a signature and let $\varphi \in \Sigma$. Then,

$$\begin{aligned}\varphi \in \mathcal{L}_{Clant}(\Sigma) &\iff \varphi^\sharp \in \mathcal{L}_{Clant}(\Sigma^\sharp), \\ \varphi \in \mathcal{L}_{Clant}(\Sigma) &\iff \varphi^\flat \in \mathcal{L}_{Clant}(\Sigma^\flat).\end{aligned}$$

PROOF. Both claims are easily proved by induction, using Lemma 4.16. The only interesting case is the translation of $\forall x_i \varphi$ into $\forall x_{i+1} (Ind(x_{i+1}) \rightarrow \varphi^\sharp)$. This operation introduces the classical antecedent $Ind(x_{i+1})$ and, therefore, turns clant formulas into other clant formulas. $\square$

We now restate two key metatheoretical results about the Clant fragment of InqBQ. These were established in [16], to which we refer for a more complete treatment of these matters. However, for specific proposition references, we instead refer to the account given in [4].

First, we know that Clant entailments in InqBQ verify compactness.

Proposition 4.18. (see [4], Theorem 5.7.9)

Entailment compactness holds for InqBQ entailments between clant formulas.

Clant entailments with finite premises in InqBQ are also recursively enumerable.

Proposition 4.19. (see [4], Theorem 5.7.10)

The following set is r.e.:

$$\{ \langle \varphi_1, \dots, \varphi_n, \psi \rangle \mid n \in \mathbb{N}, \{ \varphi_1, \dots, \varphi_n, \psi \} \subseteq \mathcal{L}_{Clant}, \varphi_1, \dots, \varphi_n \models_{\text{InqBQ}} \psi \}.$$

These results can easily be transferred to InqBT by using the translations.

Proposition 4.20. (Compactness for clant entailments, InqBT)

Entailment compactness holds for InqBT entailments between clant formulas.

PROOF. Similarly to the Rex case, this proof is an adaptation of the proof we gave for the left-to-right direction of Proposition 4.1, integrated with the closure of Clant under the translations (Proposition 4.17) and with the fact that Δ is a set of clant formulas. $\square$

Proposition 4.21. (Recursive enumerability of clant entailments, InqBT)

The following set is recursively enumerable:

$$\{ \langle \varphi_1, \dots, \varphi_n, \psi \rangle \mid n \in \mathbb{N}, \{ \varphi_1, \dots, \varphi_n, \psi \} \subseteq \mathcal{L}_{Clant}, \varphi_1, \dots, \varphi_n \models_{\text{InqBT}} \psi \}.$$

PROOF. We proceed as in the proof of Proposition 4.4. Let $\varphi_1, \dots, \varphi_n, \psi \in \mathcal{L}_{Clant}$. We compute a corresponding finite set of formulas $\Phi' \cup \{ \psi' \} \subseteq \mathcal{L}_{Clant}$ such that $\varphi_1, \dots, \varphi_n \models_{\text{InqBT}} \psi$ if and only if $\Phi' \models_{\text{InqBQ}} \psi'$. If we can compute such a set, the claim follows immediately from the recursive enumerability of clant entailments for InqBQ. Let $\Phi := \{ \varphi_1, \dots, \varphi_n \}$. Then, by Corollary 4.3 we have

$$\Phi \models_{\text{InqBT}} \psi \iff \Delta(\Sigma_\psi), \Phi^\flat \models_{\text{InqBQ}} \psi^\flat.$$

Now, $\Delta(\Sigma_\psi)$, Φ^b and ψ^b are all computable from Φ and ψ , they only include clant formulas (from Proposition 4.17 and from the definition of Δ) and they are finite (using the finiteness of Φ and that of $\Delta(\Sigma_\psi)$). Therefore, letting $\Phi' = \Delta(\Sigma_\psi) \cup \Phi^b$, the set $\Phi' \cup \{\psi'\} \subseteq \mathcal{L}_{Clant}$ satisfies the required properties. $\square$

5 Future work

To conclude, we outline some potential directions for future research. First, a natural extension of this work would be the transfer of additional results from the Rex and Clant fragments of **InqBQ** to those of **InqBT**. Using the translations, we proved entailment compactness and the recursive enumerability of validities for both fragments in **InqBT**. In **InqBQ**, however, Rex and Clant are also known to admit sound and complete proof systems. In future work, it may be interesting to consider whether our translation allows for a transfer of these axiomatizations to the corresponding fragments of **InqBT**.

Second, as we pointed out above, our translation relies on a strategy to encode states as teams and vice versa. In future work, it may be interesting to explore whether this strategy may be applied to other state-based and team-based systems beyond the special case of **InqBQ** and **InqBT**, e.g. it seems plausible that our strategy would allow us to translate various versions of inquisitive modal logic (e.g. [2, 3, 5, 9, 10]) into suitably defined modal team-based logics.

Acknowledgements

We are grateful to two anonymous reviewers for detailed feedback and precious suggestions. Funding from the European Research Council under the Horizon Europe research and innovation programme (Grant Agreement No. 101116774) is gratefully acknowledged.

References

- [1] M. Aloni. Logic and conversation: the case of free choice. *Semantics and Pragmatics*, **15**, 5, 2022.
- [2] I. Ciardelli. Dependency as question entailment. In *Dependence Logic*, S. Abramsky, J. Kontinen, J. Väänänen and H. Vollmer, eds, pp. 129–181. Springer International Publishing, 2016.
- [3] I. Ciardelli. Describing neighborhoods in inquisitive modal logic. In *Advances in Modal Logic*, D. Fernández-Duque, A. Palmigiano and S. Pinchinat, eds, vol. **14**. College Publications, 2022.
- [4] I. Ciardelli. *Inquisitive Logic: Consequence and Inference in the Realm of Questions*. Springer International Publishing, 2022.
- [5] I. Ciardelli. Global supervenience in inquisitive modal logic. *The Review of Symbolic Logic*, **18**, 589–615, 2025.
- [6] I. Ciardelli and G. Grilletti. Coherence in inquisitive first-order logic. *Annals of Pure and Applied Logic*, **173**, 103155, 2022.
- [7] I. Ciardelli, J. Groenendijk and F. Roelofsen. *Inquisitive Semantics*. Oxford University Press, 2018.
- [8] I. Ciardelli, R. Iemhoff and F. Yang. Questions and dependency in intuitionistic logic. *Notre Dame Journal of Formal Logic*, **61**, 75–115, 2020.

- [9] I. Ciardelli and M. Otto. Inquisitive bisimulation. *The Journal of Symbolic Logic*, **86**, 77–109, 2021.
- [10] I. Ciardelli and F. Roelofsens. Inquisitive dynamic epistemic logic. *Synthese*, **192**, 1643–1687, 2014.
- [11] S. Conti. *Connecting State-Based and Team-Based Inquisitive Logic*. MSc Thesis. Padua, Italy: University of Padua, 2024.
- [12] A. Durand and J. Kontinen. Hierarchies in dependence logic. *ACM Transactions on Computational Logic (TOCL)*, **13**, 1–21, 2012.
- [13] P. Galliani. Inclusion and exclusion dependencies in team semantics—on some logics of imperfect information. *Annals of Pure and Applied Logic*, **163**, 68–84, 2012.
- [14] E. Grädel and J. Väänänen. Dependence and independence. *Studia Logica*, **101**, 399–410, 2013.
- [15] G. Grilletti. *Questions & Quantification: A Study of First Order Inquisitive Logic*. PhD Thesis. Amsterdam, Netherlands: Institute for Logic, Language and Computation, 2020.
- [16] G. Grilletti. Completeness for the classical antecedent fragment of inquisitive first-order logic. *Journal of Logic, Language and Information*, **30**, 1–27, 2021.
- [17] G. Grilletti and I. Ciardelli. Games and cardinalities in inquisitive first-order logic. *The Review of Symbolic Logic*, **16**, 241–267, 2023.
- [18] P. Hawke and S. Steinert-Threlkeld. Semantic expressivism for epistemic modals. *Linguistics and Philosophy*, **44**, 475–511, 2020.
- [19] W. Hodges. Compositional semantics for a language of imperfect information. *Logic Journal of the IGPL*, **5**, 539–563, 1997.
- [20] W. Hodges. Some strange quantifiers. In *Structures in Logic and Computer Science*, J. Mycielski, G. Rozenberg and A. Salomaa, eds, pp. 51–65. Springer, Berlin Heidelberg, 1997.
- [21] J. Kontinen. Coherence and computational complexity of quantifier-free dependence logic formulas. *Studia Logica*, **101**, 267–291, 2013.
- [22] J. Kontinen and J. Väänänen. On definability in dependence logic. *Journal of Logic, Language and Information*, **18**, 317–332, 2009.
- [23] J. Kontinen and J. Väänänen. Axiomatizing first-order consequences in dependence logic. *Annals of Pure and Applied Logic*, **164**, 1101–1117, 2013.
- [24] J. Kontinen and F. Yang. Complete logics for elementary team properties. *The Journal of Symbolic Logic*, **88**, 579–619, 2023.
- [25] J. Väänänen. *Dependence Logic: A New Approach to Independence Friendly Logic*. London Mathematical Society Student Texts, Cambridge University Press, 2007.
- [26] F. Yang. *On Extensions and Variants of Dependence Logic*. PhD Thesis. Helsinki, Finland: University of Helsinki, 2014.
- [27] F. Yang. Negation and partial axiomatizations of dependence and independence logic revisited. *Annals of Pure and Applied Logic*, **170**, 1128–1149, 2019.
- [28] F. Yang and J. Väänänen. Propositional logics of dependence. *Annals of Pure and Applied Logic*, **167**, 557–589, 2016.

A Characterizing the sets of translated structures in the general case

To characterize the models induced by the translation $\sharp$ in the general case, we need to supplement Definition 3.5 with a number of new items. Given a signature Σ and a set of variables X , we add the following **InqBT** formulas to the set $\Gamma(\Sigma, X)$:

- a formula expressing the fact that tuples in the extension of $R^=$ consist of one world followed by two individuals:

$$\forall x \forall y_1 \forall y_2 (R^=(x, y_1, y_2) \rightarrow (\neg \text{Ind}(x) \wedge \text{Ind}(y_1) \wedge \text{Ind}(y_2)))$$

- formulas specifying that $R^=$ induces an equivalence relation on D

$$\forall x \forall y ((\neg \text{Ind}(x) \wedge \text{Ind}(y)) \rightarrow R^=(x, y, y))$$

$$\forall x \forall y_1 \forall y_2 (R^=(x, y_1, y_2) \rightarrow R^=(x, y_2, y_1))$$

$$\forall x \forall y_1 \forall y_2 \forall y_3 ((R^=(x, y_1, y_2) \wedge R^=(x, y_2, y_3)) \rightarrow R^=(x, y_1, y_3))$$

- formulas expressing the fact that $R^=$ induces a congruence with respect to the interpretation of predicates and function symbols

$$\forall x \forall y_1 \dots \forall y_n \forall z_1 \dots \forall z_n$$

$$(R^=(x, y_1, z_1) \wedge \dots \wedge R^=(x, y_n, z_n) \rightarrow f'(x, y_1, \dots, y_n) = f'(x, z_1, \dots, z_n))$$

$$\forall x \forall y_1 \dots \forall y_n \forall z_1 \dots \forall z_n$$

$$(R^=(x, y_1, z_1) \wedge \dots \wedge R^=(x, y_n, z_n) \rightarrow h'(y_1, \dots, y_n) = h'(z_1, \dots, z_n))$$

$$\forall x \forall y_1 \dots \forall y_n \forall z_1 \dots \forall z_n$$

$$((R^=(x, y_1, z_1) \wedge \dots \wedge R^=(x, y_n, z_n) \wedge R'(x, y_1, \dots, y_n)) \rightarrow R'(x, z_1, \dots, z_n))$$

- for every $f'_{n+1} \in \Sigma^\sharp$, a formula specifying that its interpretation maps all tuples consisting of one world and n individuals to individuals:

$$\forall x \forall y_1 \dots \forall y_n \forall z$$

$$\left(\left(f'(x, y_1, \dots, y_n) = z \wedge \neg \text{Ind}(x) \wedge \bigwedge_{i=1, \dots, n} \text{Ind}(y_i) \right) \rightarrow \text{Ind}(z) \right)$$

- an analogous formula for each $h'_n \in \Sigma^\sharp$:

$$\forall y_1 \dots \forall y_n \forall z \left(\left(h'(y_1, \dots, y_n) = z \wedge \bigwedge_{i=1, \dots, n} \text{Ind}(y_i) \right) \rightarrow \text{Ind}(z) \right)$$

Received 9 April 2025