The flattening operator in team-based logics

Arnaud Durand[1,*], Juha Kontinen[2,†], Werner Mérian[1,‡], and Jouko Väänänen[2,3,§]

[1]Université Paris Cité, CNRS, IMJ-PRG, Paris, France
[2]Department of Mathematics and Statistics, University of Helsinki, Finland
[3]Institute for Logic, Language and Computation, University of Amsterdam, The Netherlands
[*]durand@imj-prg.fr, [†]juha.kontinen@helsinki.fi, [‡]merian@imj-prg.fr,
[§]jouko.vaananen@helsinki.fi

May 19, 2025

Abstract

We propose a systematic study of the so-called *flattening operator* in team semantics. This operator was first introduced by Hodges in 1997, and has not been studied in more detail since. We begin a systematic study of the expressive power this operator adds to the most well-known team-based logics, such as *dependence logic*, *anonymity logic*, *inclusion logic* and *exclusion logic*.

1 Introduction

Team semantics is a mathematical framework for studying concepts and phenomena that arise in the presence of plurality of data. Examples of such concepts are, for example, functional dependence in database theory and conditional independence of random variables in statistics. The beginning of the area can be traced back to the introduction of dependence logic in [Vää07]. In dependence logic, formulas are interpreted by sets of assignments (called *teams*) and it extends the syntax of first-order logic by dependence atoms $=(x,y)$ expressing that the values of the variables x functionally determine the values of y in a team.

Since the introduction of dependence logic, the expressivity and complexity aspects of logics in team semantics have been extensively studied (see, *e.g.*, [Dur+22]) and interesting connections have been found to areas such as database theory [KLV13; HKV20], Bayesian networks [Cor+19], quantum foundations [APV21], and inquisitive and separation logic [CIY20; HGW22]. These works have focused on logics in the first-order, propositional and modal team semantics, and more recently also in the multi-set and probabilistic settings [APV21; Dur+18a; Dur+18b].

A defining feature of team semantics is that satisfaction of a formula is defined with reference to a *set* of assignments, not just a *single* assignment, as is the case in classical first-order logic. In this respect, logics with team semantics resemble modal logic, where a Kripke structure offers a spectrum of possible assignments. In a Kripke structure, there is the so-called *accessibility relation* which brings in the aspect that assignments, for example, develop in time. This makes it possible to talk in modal logic about *necessity* and *possibility*. Team semantics is more rigid. There is just the *set* (called *team*) of assignments without any built-in relations between its elements. However,

this makes it possible to express interesting and highly non-trivial "combinatorial" properties of variables, such as the aforementioned functional dependence but also inclusion and exclusion atoms [Gal12] and independence atoms [GV13].

We can think of the sub-team relation as an accessibility relation, and then "necessity" and "possibility" make perfect sense. Then "necessarily ϕ" means "ϕ is true in every sub-team", and "possibly ϕ" means "ϕ is true in some non-empty sub-team".

Some formulas do not seem to take advantage of the existence of a team around an assignment. One way to make this phenomenon exact is the following: a formula ϕ is called *flat*, or is said to satisfy the *flatness criterion*, if for all model $\mathfrak{M}$ and for all team X the following holds

$$\mathfrak{M} \vDash_X \phi \iff \forall s \in X \colon \mathfrak{M} \vDash_{\{s\}} \phi. \tag{1.1}$$

This criterion holds for first-order formulas ϕ, *i.e.* all first-order formulas are flat. This offers a method to show that some given formula is not logically equivalent to a first-order formula. One just shows that the formula is not flat. To take two simple examples, $=(x)$ and $=(x,y)$ are not flat, hence they cannot be logically equivalent to a first-order formula, which is intuitively obvious but still needs an argument. Flatness provides such an argument. In general, the problem of deciding whether a given dependence logic formula is equivalent to a first-order formula is highly undecidable. Therefore, the question of first-order expressibility is non-trivial and any method, even a partial one, is useful.

One can say that flat formulas are, in a sense, "local", in that their truth is determined by what holds for single assignments. Still, non-trivial properties of teams can be expressed. For example, $x = y$ is flat and it says that the variables x and y have the same value in the team. It is a trivial property of a team but still such teams are clearly distinguished from totally arbitrary teams. Flatness just means that the distinguishing property is "flatly spread out" in the team.

One might think that flatness of a formula is reflected in the formula being built from flat elements. But all *sentences* are flat because their truth in any non-empty team is determined by their truth in $\{\varnothing\}$, *i.e.* the team containing only the empty assignment. A sentence expresses something about the model that the team is based on. A sentence says nothing about any team. The flatness of sentences is, therefore, like an anomaly. But it follows that being flat is not equivalent to being, up to logical equivalence, first-order. A flat formula may very well have non-flat sub-formulas.

Example. We consider a signature with a 2-ary relation $R(\cdot,\cdot)$. Consider the following formula $\phi(x)$

$$\exists u(R(x,u) \land \forall v(R(x,v) \to \exists w(R(x,w) \land =(w,v) \land w \neq u))).$$

This is satisfied by a team X if for every assignment of the variable x in the team X, the set $R(x,\cdot)$ is infinite. It is a flat formula, but its sub-formula $=(w,v)$ is not flat. Thus, condition (1.1) only says that ϕ itself behaves in a first order way by being flat, but its sub-formulas need not do so.

The class of teams satisfying a flat formula is closed downwards and closed under unions and intersections. There is a maximal team, and the class of teams satisfying a flat formula is precisely the power-set of that team. The maximal team of a flat formula ϕ is given by

$$\left\{ s \mid \mathfrak{M} \vDash_{\{s\}} \phi \right\}.$$

Thus the dimension (in the sense of [HLV24]) of a flat formula is always 1.

We can make an arbitrary formula flat by using the following new logical operator, first introduced by Hodges in 1997 [Hod97], called the *flattening operator* and denoted here by F:

$$\mathfrak{M} \vDash_X \mathsf{F}\phi \iff \forall s \in X \colon \mathfrak{M} \vDash_{\{s\}} \phi. \tag{1.2}$$

Thus, $\mathsf{F}\phi$ is always flat, and for a flat formula ϕ, we have $\phi \equiv \mathsf{F}\phi$. The flattening operator is somewhat similar to the necessity operator $\Box\phi$ and the possibility operator $\Diamond\phi$ from Modal Logic. The "mode" of ϕ that $\mathsf{F}\phi$ brings about is not that ϕ is "necessary" or "possible", but that on the question whether the entire team satisfies ϕ or not, at least we know that ϕ is true in singleton sub-teams. We can perhaps anticipate that $\mathsf{F}\phi$ is computationally much simpler to check for satisfaction than ϕ itself, since we have to check only single assignments. However, if ϕ is a sentence, then $\mathsf{F}\phi \equiv \phi$. So $\mathsf{F}\phi$ may be computationally as complicated as ϕ itself.

The flattening operator F makes an arbitrary formula flat. This happens with no regard to what the formula looks like inside. There is another way to make a formula flat. This is based on changing the formula inside as follows: The *flattening* ϕ^{f} of a formula ϕ is obtained inductively by replacing non-first-order atomic formulas by something that is flat. If there are non-first-order logical operations, then they are similarly replaced by something flat. We can impose axioms for the flattening, because it is *a priori* not clear what it is. There may be different ways to flatten a formula, but we expect them all to satisfy some basic axioms. This will be discussed in section 3.

Although $\mathsf{F}\phi$ is always flat, it is not always equivalent to ϕ^{f}. We can use the flattening of a formula to define a kind of *negation* of a formula: we take the flattening ϕ^{f}, which is first-order, and then apply negation to the flattening.

A weaker version of flatness is the following: a formula is *downwards flat* if

$$\mathfrak{M} \vDash_X \phi \implies \forall s \in X \colon \mathfrak{M} \vDash_{\{s\}} \phi. \tag{1.3}$$

Thus, the truth of ϕ in a team is inherited by the singleton sub-teams. Of course, downwards closed formulas are downwards flat, but not necessarily flat. An example is $=(x)$. Every singleton team satisfies $=(x)$ but not every team satisfies $=(x)$.

A similar weaker version of flatness is the following: a formula is *upwards flat* if

$$(\forall s \in X \colon \mathfrak{M} \vDash_{\{s\}} \phi) \implies \mathfrak{M} \vDash_X \phi. \tag{1.4}$$

So for the truth of ϕ in a team, it is enough that ϕ "gathers" truth from little singleton pieces inside the team.

Still another variation of flatness is *n-coherence* (introduced in [Kon13], see also [MO22] and [CG22]):

$$\mathfrak{M} \vDash_X \phi \iff (\forall Y \subseteq X \text{ such that } |Y| = n, \ \mathfrak{M} \vDash_Y \phi). \tag{1.5}$$

Usually, flatness corresponds in this terminology to 1-coherence. For example, dependence atoms are not flat but they are 2-coherent. In [Kon13], it is shown that n-coherence is preserved by conjunction but not by disjunction, except if one of the disjuncts is 1-coherent. The disjunction of two dependence atoms need not be n-coherent for any n. The model-checking problem for n-coherent formulas is in $\mathsf{LOGSPACE}$. Model-checking of the disjunction of two 2-coherent formulas is in $\mathsf{NLOGSPACE}$. In [Kon13], it is asked whether there is a dependence logic formula which is not k-coherent for any k but which is "$\sqrt{n}$-coherent"?

Coherence, just like flatness, can be approached from the point of view of an operator as well. The *n-coherence operator* applied to a formula says of a team that every sub-team of size n satisfies the formula. The *co-n-coherence operator* applied to a formula says of a team of size m that every sub-team of size $m - n$ satisfies the formula. The co-n-coherence operator is a way to say that a formula is approximately satisfied by the team. In [Vää17] the approximation is more generous (proportional). We can even let an operator say that the team satisfies the formula even if any n assignments satisfying a given fixed first order formula are *added* to the team or *taken away* from the team. So, the team satisfies the formula even if it is made a little smaller or a little bigger. It is a kind of "give or take at most n" operator. Dependence logic is closed under all these operators. We will not pursue this line of argument further in this paper.

Our contributions We study the effects of extending several team-based logics by the flattening operator of Hodges. We also take a new look at the concept of a flattening of a formula that has been utilized in several works in the area. In section 2 we recall the basic concepts and definitions relevant for this work. In section 3 we show that the flattening of a formula can be characterized axiomatically but also give explicit inductive definitions of it for the most prominent team-based logics. In section 4 we introduce the flattening operator and study its interactions with the aforementioned logics. In section 5 we show that the flattening operator increases the expressive power of the unary fragments of inclusion and anonymity logics. These results show that the two different ways of defining a flattening of a formula are not in general equivalent.

2 Preliminaries

In this section, we will briefly recall the notation used in this work, the definition of team semantics, and some basic results that will be used in the rest of this work. Through all of this work, we assume that we have countable sets of individual variable symbols x_i, y_i, z_i, etc. for $i \in \mathbb{N}$, and of relation variable symbols X_i, Y_i, Z_i, etc. for $i \in \mathbb{N}$ of all arities. We will use Fraktur font to represent models, such as $\mathfrak{A}$, $\mathfrak{B}$, $\mathfrak{M}$, etc. and use Roman font to represent their corresponding domain A, B, M, etc. We will write $\vec{x}$, $\vec{y}$, $\vec{z}$ and so on to describe tuples of variable symbols; and likewise, we will write $\vec{a}$, $\vec{b}$, $\vec{m}$ and so forth to describe tuples of elements of a model. Given any set A, we will furthermore write $\mathcal{P}(A)$ for the powerset $\{B \mid B \subseteq A\}$ of A, and $\mathcal{P}^*(A)$ for $\mathcal{P}(A) \setminus \{\varnothing\}$.

2.1 Team semantics

If s is an assignment, we write $s[m/v]$ for the assignment in which we substituted the variable v by the value m. We recall from [Vää07] the following concepts: a *team* X is any set of assignments for a fixed set of variables, denoted $\mathrm{dom}(X)$. Suppose now X is a team. It is sometimes useful to think of X as a *relation* in the following sense: suppose $x_1, \ldots, x_n \in \mathrm{dom}(X)$. We obtain from X the n-ary relation $\{\langle s(x_1), \ldots, s(x_n) \rangle \mid s \in X\}$. If m is an element the domain M of our model $\mathfrak{M}$, then $X[m/v]$ is the team consisting of all $s[m/v]$, where $s \in X$. We use $X[M/v]$, called the *duplication* of X at v, to denote the set of all $s[m/v]$, where $m \in M$ and $s \in X$. Finally, if $H \colon X \to \mathcal{P}^*(M)$, then $X[H/v]$, called the *supplementation* of X at v by H, is the team consisting of all $s[m/v]$, where $s \in X$ and $m \in H(s)$.

For our purposes, it will be useful to first present team semantics for first-order logic proper, and then progressively add new atoms (dependency, non-dependency, inclusion, exclusion, etc.).

We recall the usual definition of team semantics (in its lax formulation). For the sake of simplicity, it is assumed that all expressions are in *negation normal form*, meaning that negations $\neg$ only appear in front of first-order literals.

Definition 2.1 (Team semantics for first-order logic). Let σ be a first-order signature. Let $\mathfrak{M}$ be a first-order σ-model, let φ be a first-order σ-formula, and let X be a team over $\mathfrak{M}$ such that $\mathrm{dom}(X) \supseteq \mathrm{FV}(\varphi)$. Then we define the relation $\mathfrak{M} \vDash_X \varphi$ by induction over the structure of the formula φ as follows:

TS-lit: If $\varphi \equiv \alpha$ where α is a first-order σ-literal, then $\mathfrak{M} \vDash_X \varphi$ if and only if for all assignments $s \in X$ it holds that $\mathfrak{M} \vDash_s \alpha$ (here according to Tarskian semantics).

TS-$\vee$: If $\varphi \equiv \psi_1 \vee \psi_2$ where ψ_1 and ψ_2 are first-order σ-formulas, then $\mathfrak{M} \vDash_X \varphi$ if and only if there exist some teams Y and Z (not necessarily disjoint) such that $X = Y \cup Z$, $\mathfrak{M} \vDash_Y \psi_1$ and $\mathfrak{M} \vDash_Z \psi_2$.

TS-∧: If $\varphi \equiv \psi_1 \wedge \psi_2$ where ψ_1 and ψ_2 are first-order σ-formulas, then $\mathfrak{M} \vDash_X \varphi$ if and only if $\mathfrak{M} \vDash_X \psi_1$ and $\mathfrak{M} \vDash_X \psi_2$.

TS-∃: If $\varphi \equiv \exists v \psi$ where v is a variable symbol and ψ is a first-order σ-formula, then $\mathfrak{M} \vDash_X \varphi$ if and only if there exists some $H \colon X \to \mathcal{P}^*(M)$ such that $\mathfrak{M} \vDash_{X[H/v]} \psi$.

TS-∀: If $\varphi \equiv \forall v \psi$ where v is a variable symbol and ψ is a first-order σ-formula, then $\mathfrak{M} \vDash_X \varphi$ if and only if $\mathfrak{M} \vDash_{X[M/v]} \psi$.

If it is the case that $\mathfrak{M} \vDash_X \varphi$, then we say that the model $\mathfrak{M}$ together with the team X *satisfy* φ, and if it is not the case, then we say that the model $\mathfrak{M}$ together with the team X *do not satisfy* φ and we write it $\mathfrak{M} \nvDash_X \varphi$. If φ is a sentence (*i.e.* has no free-variables), we say that φ is *true* in $\mathfrak{M}$ according to team semantics, and we write it $\mathfrak{M} \vDash \varphi$, if $\mathfrak{M} \vDash_{\{\varnothing\}} \varphi$ where $\{\varnothing\}$ is the team containing only the empty assignment. Otherwise, we say that φ is *false* in $\mathfrak{M}$ according to team semantics, and we write it $\mathfrak{M} \nvDash \varphi$.

We recall the most well-known team-based logics. We first give definitions for *dependence atoms*, introduced in [Vää07]. Let $k \in \mathbb{N}$. If $\vec{t}_1$ is a k-tuple of σ-terms and t_2 is a σ-term, then $=(\vec{t}_1, t_2)$ is called a *k-ary dependence atom*. For 0-ary atoms, called *constancy atoms*, we may simply write $=(t_2)$. We define $\mathrm{FV}(=(\vec{t}_1, t_2)) = \mathrm{Vr}(\vec{t}_1) \cup \mathrm{Vr}(t_2)$. The set of formulas of *dependence logic* over a signature σ, denoted by $\mathsf{FO}(=(\cdot,\cdot))[\sigma]$ or even more simply $\mathsf{DEP}[\sigma]$, is defined by adding dependence atoms of all arities to the definition of the set of formulas of $\mathsf{FO}[\sigma]$. By allowing only at most k-ary dependence atoms in the condition above, we obtain the fragment $\mathsf{FO}(=_k(\cdot,\cdot))[\sigma]$ or even more simply $\mathsf{DEP}_k[\sigma]$ called *k-ary dependence logic*. If $\mathfrak{M}$ is a model and X is a team over $\mathfrak{M}$ such that $\mathrm{dom}(X) \supseteq \mathrm{Vr}(\vec{t}_1 t_2)$, then we define the truth of $=(\vec{t}_1, t_2)$ in the model $\mathfrak{M}$ and the team X:

$$\mathfrak{M} \vDash_X \; =(\vec{t}_1, t_2) \quad \text{iff} \quad \text{for all } s, s' \in X, \text{ if } s(\vec{t}_1) = s'(\vec{t}_1), \text{ then } s(t_2) = s'(t_2).$$

Next, we present so-called, *anonymity atoms* also called *non-dependency atoms* ([Gal12]), or *afunctional dependence atoms* ([Par+89]). They have been studied, *e.g.*, in [Vää22; Rön18]. Let $k \in \mathbb{N}$. If $\vec{t}_1$ is a k-tuple of σ-terms and t_2 is a σ-term, then $\vec{t}_1 \; \Upsilon \; t_2$ is called a *k-ary anonymity atom*. For 0-ary atoms $\Upsilon_0 t$, called *non-constancy atoms*, we may simply write Υt. We define $\mathrm{FV}(\vec{t}_1 \; \Upsilon \; t_2) = \mathrm{Vr}(\vec{t}_1) \cup \mathrm{Vr}(t_2)$. The set of formulas of *anonymity logic* over a signature σ, denoted by $\mathsf{FO}(\Upsilon)[\sigma]$ or even more simply $\mathsf{ANON}[\sigma]$, is defined by adding anonymity atoms of all arities to the definition of the set of formulas of $\mathsf{FO}[\sigma]$. By allowing only at most k-ary anonymity atoms in the condition above, we obtain the fragment $\mathsf{FO}(\Upsilon_k)[\sigma]$ or even more simply $\mathsf{ANON}_k[\sigma]$ called *k-ary anonymity logic*. If $\mathfrak{M}$ is a model and X is a team over $\mathfrak{M}$ such that $\mathrm{dom}(X) \supseteq \mathrm{Vr}(\vec{t}_1 t_2)$, then we define the truth of $\vec{t}_1 \; \Upsilon \; t_2$ in the model $\mathfrak{M}$ and the team X:

$$\mathfrak{M} \vDash_X \vec{t}_1 \; \Upsilon \; t_2 \quad \text{iff} \quad \text{for every } s \in X, \text{ there is } s' \in X \text{ such that } s(\vec{t}_1) = s'(\vec{t}_1) \text{ but } s(t_2) \neq s'(t_2).$$

Next, we turn to *inclusion atoms*. Let $k \in \mathbb{N}$. If $\vec{t}_1, \vec{t}_2$ are k-tuples of σ-terms, then $\vec{t}_1 \subseteq \vec{t}_2$ is a *k-ary inclusion atom*. We define $\mathrm{FV}(\vec{t}_1 \subseteq \vec{t}_2) = \mathrm{Vr}(\vec{t}_1) \cup \mathrm{Vr}(\vec{t}_2)$. The set of formulas of *inclusion logic* over a signature σ, denoted by $\mathsf{FO}(\subseteq)[\sigma]$ or even more simply $\mathsf{INC}[\sigma]$, is defined by adding inclusion atoms of all arities to the definition of the set of formulas of $\mathsf{FO}[\sigma]$. By allowing only at most k-ary inclusion atoms in the condition above, we obtain the fragment $\mathsf{FO}(\subseteq_k)[\sigma]$ or even more simply $\mathsf{INC}_k[\sigma]$ called *k-ary inclusion logic*. If $\mathfrak{M}$ is a model and X is a team over $\mathfrak{M}$ such that $\mathrm{dom}(X) \supseteq \mathrm{Vr}(\vec{t}_1 \vec{t}_2)$, then we define the truth of $\vec{t}_1 \subseteq \vec{t}_2$ in the model $\mathfrak{M}$ and the team X:

$$\mathfrak{M} \vDash_X \vec{t}_1 \subseteq \vec{t}_2 \quad \text{iff} \quad \text{for all } s \in X, \text{ there exists } s' \in X \text{ such that } s(\vec{t}_1) = s'(\vec{t}_2).$$

Let us now recall *(conditional) independence atoms*. Let $k_1, k_2, k_3 \in \mathbb{N}$. If $\vec{t}_1$ is a k_1-tuple, $\vec{t}_2$ a k_2-tuple of σ-terms and $\vec{t}_3$ a k_3-tuple of σ-terms, then $\vec{t}_2 \perp_{\vec{t}_1} \vec{t}_3$ is a *(k_1, k_2, k_3)-ary (conditional)*

independence atom. We define $\mathrm{FV}(\vec{t_2} \perp_{\vec{t_1}} \vec{t_3}) = \mathrm{Vr}(\vec{t_1}) \cup \mathrm{Vr}(\vec{t_2}) \cup \mathrm{Vr}(\vec{t_3})$. The set of formulas of *(conditional) independence logic* over a signature σ, denoted by $\mathsf{FO}(\perp)[\sigma]$ or even more simply $\mathsf{IND}[\sigma]$ is defined by adding independence atoms of all arities to the definition of the set of formulas of $\mathsf{FO}[\sigma]$. By allowing only at most (k_1, k_2, k_3)-ary independence atoms in the condition above, we obtain the fragment $\mathsf{FO}(\perp_{k_1,k_2,k_3})[\sigma]$ or even more simply $\mathsf{IND}_{k_1,k_2,k_3}[\sigma]$ called (k_1, k_2, k_3)-*ary (conditional) independence logic*. If $\mathfrak{M}$ is a model and X is a team over $\mathfrak{M}$ such that $\mathrm{dom}(X) \supseteq \mathrm{Vr}(\vec{t_1}\vec{t_2}\vec{t_3})$, then we define the truth of $\vec{t_2} \perp_{\vec{t_1}} \vec{t_3}$ in the model $\mathfrak{M}$ and the team X:

$$\mathfrak{M} \vDash_X \vec{t_2} \perp_{\vec{t_1}} \vec{t_3} \quad \text{iff} \quad \text{for all } s, s' \in X, \text{ if } s(\vec{t_1}) = s'(\vec{t_1}), \text{ then there exists } s'' \in X \text{ such that}$$
$$s''(\vec{t_1}\vec{t_2}) = s(\vec{t_1}\vec{t_2}) \text{ and } s''(\vec{t_1}\vec{t_3}) = s'(\vec{t_1}\vec{t_3}).$$

Finally, we present the syntax and the semantics for *exclusion logic*. Let $k \in \mathbb{N}$. If $\vec{t_1}, \vec{t_2}$ are k-tuples of σ-terms, then $\vec{t_1} \mid \vec{t_2}$ is a k-*ary exclusion atom*. We define $\mathrm{FV}(\vec{t_1} \mid \vec{t_2}) = \mathrm{Vr}(\vec{t_1}) \cup \mathrm{Vr}(\vec{t_2})$. The set of formulas of *exclusion logic* over a signature σ, denoted by $\mathsf{FO}(\mid)[\sigma]$ or even more simply $\mathsf{EXC}[\sigma]$, is defined by adding exclusion atoms of all arities to the definition of the set of formulas of $\mathsf{FO}[\sigma]$. By allowing only at most k-ary exclusion atoms in the condition above, we obtain the fragment $\mathsf{FO}(\mid_k)[\sigma]$ or even more simply $\mathsf{EXC}_k[\sigma]$ called k-*ary exclusion logic*. If $\mathfrak{M}$ is a model and X is a team over $\mathfrak{M}$ such that $\mathrm{dom}(X) \supseteq \mathrm{Vr}(\vec{t_1}\vec{t_2})$. We define the truth of $\vec{t_1} \mid \vec{t_2}$ in the model $\mathfrak{M}$ and the team X:

$$\mathfrak{M} \vDash_X \vec{t_1} \mid \vec{t_2} \quad \text{iff} \quad \text{for all } s, s' \in X, \text{ it holds that } s(\vec{t_1}) \neq s'(\vec{t_2}).$$

More generally, we will speak of *team-based logic* to speak of one of these logics based on team semantics (*i.e.* independence logic, anonymity logic, inclusion logic, exclusion logic, or independence logic) and of *team-based formula* to speak of a formula from one of these logics (*i.e.* a formula which uses dependency atoms, anonymity atoms, inclusion atoms, exclusion atoms, or independence atoms).

2.2   Closure properties

The logics that we have presented so far can be differentiated on the basis of their distinct closure properties. Let us start by defining the closure properties.

Definition 2.2. Let σ be a signature. A team-based σ-formula φ:

- is said to have the *empty team property* if for any σ-model $\mathfrak{M}$, it holds that $\mathfrak{M} \vDash_\varnothing \varphi$ (according to team semantics).

- is said to be *downwards closed* if for all σ-models $\mathfrak{M}$ and teams X over $\mathfrak{M}$, it holds that

$$\mathfrak{M} \vDash_X \varphi \implies \forall Y \subseteq X, \mathfrak{M} \vDash_Y \varphi.$$

- is said to be *union closed* if for all σ-models $\mathfrak{M}$ and collections $\{X_i \mid i \in I\}$ of teams X_i over $\mathfrak{M}$, it holds that

$$\mathfrak{M} \vDash_{X_i} \varphi \text{ for all } i \in I \implies \mathfrak{M} \vDash_{\cup_{i \in I} X_i} \varphi.$$

- is said to be *flat* if for all σ-models $\mathfrak{M}$ and teams X over $\mathfrak{M}$, it holds that

$$\mathfrak{M} \vDash_X \varphi \iff \mathfrak{M} \vDash_{\{s\}} \varphi \text{ for all } s \in X.$$

One can remark that the flatness property is equivalent to the combination of the downwards closure and the union closure properties.

Each of these properties is extended to team-based logics: a given logic (*e.g.* DEP, ANON, INC, EXC, ...) has some of these properties if *all* formulas of the logic have those properties. We can then split some of the well-known team-based logics between the ones that are downwards closed and the ones that are union closed.

3   Flattening of a formula

We will now introduce the *flattening of a formula φ*, denoted by φ^{f}, where φ belongs to some team-based logic. The notion of "flattened formulas" was first defined in [Vää07] and then used *e.g.* in [Yan14].

The definition of flattening that was given in [Vää07] was inductive and for dependence logic only. We present here an axiomatic definition which is appropriate in a wider context. We show that in most cases these axioms determine a unique inductive definition of a flattening of a formula.

1. (entailment axiom) for all models $\mathfrak{M}$ and all teams X over $\mathfrak{M}$ such that $\mathrm{dom}(X) \supseteq \mathrm{FV}(\varphi)$, the following must hold
$$\mathfrak{M} \vDash_X \varphi \implies \mathfrak{M} \vDash_X \varphi^{\mathsf{f}}.$$

2. (flatness axiom) φ^{f} is a flat formula.

3. (distributivity axiom) The following holds: $(\star(\varphi_1, \ldots, \varphi_n))^{\mathsf{f}} = \star(\varphi_1^{\mathsf{f}}, \ldots, \varphi_n^{\mathsf{f}})$ for any logical operator $\star$ of arity n, and $(Qx\varphi)^{\mathsf{f}} = Qx\varphi^{\mathsf{f}}$ for any quantifier Q.

Now, what we would like to obtain is an explicit inductive definition of the flattening of a given formula. From these requirements, we can infer the flattening of the most well-known team-based atoms. It is easy to see that taking $\top$ for the flattening of any non first-order atom is *sufficient* to satisfy all these requirements. But the interesting question is: is it *necessary*? The goal of this section is to answer this question.

We get the following flattening for the most well-known team-based atoms

Proposition 3.1. *The following equivalences are implied by the above axioms:*

(i) *for dependency atoms:* $(=(\vec{x}, y))^{\mathsf{f}} \equiv \top$, (iii) *for inclusion atoms:* $(\vec{x} \subseteq \vec{y})^{\mathsf{f}} \equiv \top$,

(ii) *for anonymity atoms:* $(\vec{x} \Upsilon y)^{\mathsf{f}} \equiv \top$, (iv) *for independence atoms:* $(\vec{x} \perp \vec{y})^{\mathsf{f}} \equiv \top$.

Proof. The proof of each claim is straightforward.

(i) Suppose $(=(\vec{x}, y))^{\mathsf{f}} \not\equiv \top$. Then $\mathfrak{M} \nvDash_X (=(\vec{x}, y))^{\mathsf{f}}$ for some model $\mathfrak{M}$ and team X. But by flatness axiom, $(=(\vec{x}, y))^{\mathsf{f}}$ is flat, and so it means that there is $s_0 \in X$ such that $\mathfrak{M} \nvDash_{\{s_0\}} (=(\vec{x}, y))^{\mathsf{f}}$. On the other hand, as a singleton team satisfies any dependence atom, then in particular $\mathfrak{M} \vDash_{\{s_0\}} =(\vec{x}, y)$, and so by the entailment axiom, $\mathfrak{M} \vDash_{\{s_0\}} (=(\vec{x}, y))^{\mathsf{f}}$. Contradiction.

(ii) Suppose that $(\vec{x} \Upsilon y)^{\mathsf{f}} \not\equiv \top$. Then $\mathfrak{M} \nvDash_X (\vec{x} \Upsilon y)^{\mathsf{f}}$ for some model $\mathfrak{M}$ and team X. But by flatness axiom, $(\vec{x} \Upsilon y)^{\mathsf{f}}$ is flat, and so it means that there is $s_0 \in X$ such that $\mathfrak{M} \nvDash_{\{s_0\}} (\vec{x} \Upsilon y)^{\mathsf{f}}$. On the other hand, we construct from scratch a new team Y as follows. We write $\vec{a} = s_0(\vec{x})$ and $b = s_0(y)$. Let $b' \in M \setminus \{b\}$ (non-empty by assumption in this article). Let s_0' be the assignment $\{\vec{x} \mapsto \vec{a}, y \mapsto b'\}$. Then consider the team $Y = \{s_0, s_0'\}$, *i.e.* Y is the following team

$$
\begin{array}{c|c}
 & \vec{x} \quad y \\
\hline
s_0\colon & \vec{a} \quad b \\
s_0'\colon & \vec{a} \quad b'
\end{array}
$$

By construction, $\mathfrak{M} \vDash_Y \vec{x} \,\Upsilon\, y$, so by the entailment axiom, we get $\mathfrak{M} \vDash_Y (\vec{x} \,\Upsilon\, y)^{\mathsf{f}}$. Now by the flatness axiom, $(\vec{x} \,\Upsilon\, y)^{\mathsf{f}}$ is flat, and so for all $s \in Y$ we get $\mathfrak{M} \vDash_{\{s\}} (\vec{x} \,\Upsilon\, y)^{\mathsf{f}}$, in particular for $s_0 \in Y$ we get $\mathfrak{M} \vDash_{\{s_0\}} (\vec{x} \,\Upsilon\, y)^{\mathsf{f}}$. Contradiction.

(*iii*) Suppose $(\vec{x} \subseteq \vec{y})^{\mathsf{f}} \not\equiv \top$. Then $\mathfrak{M} \nvDash_X (\vec{x} \subseteq \vec{y})^{\mathsf{f}}$ for some model $\mathfrak{M}$ and team X. But by flatness axiom, $(\vec{x} \subseteq \vec{y})^{\mathsf{f}}$ is flat, and so it means that there is $s_0 \in X$ such that $\mathfrak{M} \nvDash_{\{s_0\}} (\vec{x} \subseteq \vec{y})^{\mathsf{f}}$. On the other hand, we construct from scratch a new team Y as follows. We write $\vec{a} = s_0(\vec{x})$ and $\vec{b} = s_0(\vec{y})$. Let s_0' be the assignment $\{\vec{x} \mapsto \vec{a}, \vec{y} \mapsto \vec{a}\}$. Then consider the team $Y = \{s_0, s_0'\}$, *i.e.* Y is the following team

$$
\begin{array}{c|c}
 & \vec{x} \quad \vec{y} \\
\hline
s_0\colon & \vec{a} \quad \vec{b} \\
s_0'\colon & \vec{a} \quad \vec{a}
\end{array}
$$

By construction, $\mathfrak{M} \vDash_Y \vec{x} \subseteq \vec{y}$, so by the entailment axiom, we get $\mathfrak{M} \vDash_Y (\vec{x} \subseteq \vec{y})^{\mathsf{f}}$. Now by the flatness axiom, $(\vec{x} \subseteq \vec{y})^{\mathsf{f}}$ is flat, and so for all $s \in Y$ we get $\mathfrak{M} \vDash_{\{s\}} (\vec{x} \subseteq \vec{y})^{\mathsf{f}}$, in particular for $s_0 \in Y$ we get $\mathfrak{M} \vDash_{\{s_0\}} (\vec{x} \subseteq \vec{y})^{\mathsf{f}}$. Contradiction.

(*iv*) Suppose $(\vec{x} \perp \vec{y})^{\mathsf{f}} \not\equiv \top$. Then $\mathfrak{M} \nvDash_X (\vec{x} \perp \vec{y})^{\mathsf{f}}$ for some model $\mathfrak{M}$ and team X. But by flatness axiom, $(\vec{x} \perp \vec{y})^{\mathsf{f}}$ is flat, and so it means that there is $s_0 \in X$ such that $\mathfrak{M} \nvDash_{\{s_0\}} (\vec{x} \perp \vec{y})^{\mathsf{f}}$. On the other hand, as a singleton team satisfy any pure independence atom, then in particular $\mathfrak{M} \vDash_{\{s_0\}} \vec{x} \perp \vec{y}$, and so by the entailment axiom, $\mathfrak{M} \vDash_{\{s_0\}} (\vec{x} \perp \vec{y})^{\mathsf{f}}$. Contradiction.

$\square$

Remark 3.2. Curiously, we do not obtain a unique solution in the case of the exclusion atom. Indeed, for the exclusion atom $(\vec{x} \mid \vec{y})^{\mathsf{f}}$, there are (at least) two choices satisfying the flattening axioms: we can let $(\vec{x} \mid \vec{y})^{\mathsf{f}}$ be $\top$, or can let $(\vec{x} \mid \vec{y})^{\mathsf{f}}$ be $\vec{x} \neq \vec{y}$. Both choices satisfy the flattening axioms.

In order to remain as symmetrical as possible, we make the choice in this thesis to take $\top$ as the flattening of the exclusion atom.

Note that in this proposition 3.1, we found the flattening only for atoms built of variables, not terms. In fact, this is not an issue as we can eliminate composite terms inside team-based atoms by existentially quantifying over new variables. Hence, we now have an explici expression for all team-based atoms built of terms. Also, it is immediate to infer that the flattening of a first-order literal in actually itself. One can then infer the flattening of any team-based formula using the distributivity axiom: substitute all team-based atoms by $\top$, and keep all other (first-order) literals as they are.

4 Flattening operator

We move now to investigating the flattening operator which just boldly asserts that the formula following the operator has to be treated in a flat way, whether the formula looks flat or not. This operator was first introduced in [Hod97] and was at that time denoted by "↓". In this article, we will denote it by "F".

4.1 Definition

First we give the syntax and the semantics of the flattening operator F.

Definition 4.1 (Syntax of the flattening operator). Let φ be a team-based σ-formula, then $\mathsf{F}\varphi$ is a team-based σ-formula. The flattening operator does not bind variables, thus $\mathrm{FV}(\mathsf{F}\varphi) = \mathrm{FV}(\varphi)$.

It should be noted that the flattening operator F may be applied to any kind of team-based formula. Let $\mathcal{D} = \{d_1, \ldots, d_n\}$ be a set of team-based constructors (e.g. dependency, anonymity, independency, inclusion, exclusion, etc.). We will refer to the extension of $\mathsf{FO}(\mathcal{D})$ by the flattening operator F by $\mathsf{FO}(\mathcal{D}, \mathsf{F})$.

Definition 4.2 (Semantics of the flattening operator). Let φ be a team-based σ-formula, $\mathfrak{M}$ a σ-model and X a team over $\mathfrak{M}$ such that $\mathrm{dom}(X) \supseteq \mathrm{FV}(\varphi)$. We define the truth of the formula $\mathsf{F}\varphi$ in the model $\mathfrak{M}$ and the team X as:

$$\mathfrak{M} \vDash_X \mathsf{F}\varphi \iff \forall s \in X, \mathfrak{M} \vDash_{\{s\}} \varphi$$

Therefore, one can rewrite the flatness definition as follows: let φ be a team-based formula, $\mathfrak{M}$ a model and X a team over $\mathfrak{M}$ such that $\mathrm{dom}(X) \supseteq \mathrm{FV}(\varphi)$ (also equal to $\mathrm{FV}(\mathsf{F}\varphi)$), then φ is flat if and only if it holds that

$$\mathfrak{M} \vDash_X \varphi \iff \mathfrak{M} \vDash_X \mathsf{F}\varphi.$$

It is precisely because of this characterization of flatness that the flattening operator was introduced, and that it bears the name that it does. One can easily show the following immediate properties about the flattening operator.

Property 4.3. *Let φ be any team-based formula. Then the following properties hold:*

(i) *the formula $\mathsf{F}\varphi$ is always flat.*

(ii) *φ is flat if and only if $\mathsf{F}\varphi \equiv \varphi$.*

(iii) *the operator F is idempotent, that is, $\mathsf{FF}\varphi \equiv \mathsf{F}\varphi$.*

Proof.

(i) Let $\mathfrak{M}$ be a model, and let X be a team over $\mathfrak{M}$ such that $\mathrm{dom}(X) \supseteq \mathrm{FV}(\mathsf{F}\varphi)$. According to definition 4.1, we get that $\mathrm{FV}(\mathsf{F}\varphi) = \mathrm{FV}(\varphi)$. Then

$$\begin{aligned}
&\mathfrak{M} \vDash_X \mathsf{F}\varphi \\
\iff & \forall s \in X, \mathfrak{M} \vDash_{\{s\}} \varphi && \text{(semantics of } \mathsf{F}) \\
\iff & \forall s \in X, \left(\forall s' \in \{s\}, \mathfrak{M} \vDash_{\{s'\}} \varphi\right) \\
\iff & \forall s \in X, \mathfrak{M} \vDash_{\{s\}} \mathsf{F}\varphi
\end{aligned}$$

(ii) Let $\mathfrak{M}$ be a model, and let X be a team over $\mathfrak{M}$ such that $\mathrm{dom}(X) \supseteq \mathrm{FV}(\varphi)$. Suppose that φ is flat, then $\mathfrak{M} \vDash_X \mathsf{F}\varphi$ iff for all $s \in X$, $\mathfrak{M} \vDash_{\{s\}} \varphi$ iff $\mathfrak{M} \vDash_X \varphi$. Conversely, suppose that $\mathsf{F}\varphi \equiv \varphi$, then $\mathfrak{M} \vDash_X \varphi$ iff $\mathfrak{M} \vDash_X \mathsf{F}\varphi$ iff for all $s \in X$, $\mathfrak{M} \vDash_{\{s\}} \varphi$.

(iii) This is a direct consequence of the two previous items.

$\square$

Property 4.4. *Let φ be a team-based formula, $\mathfrak{M}$ a model and X a team over $\mathfrak{M}$ such that* dom $(X) \supseteq$ FV (φ). *If φ is a flat formula, then*

$$\mathfrak{M} \vDash_X \mathsf{F}\varphi \implies \mathfrak{M} \vDash_X \varphi^{\mathsf{f}}.$$

Proof. Suppose that φ is flat, then $\mathfrak{M} \vDash_X \mathsf{F}\varphi$ implies by property 4.3.(*ii*) that $\mathfrak{M} \vDash_X \varphi$, which implies by the entailment axiom that $\mathfrak{M} \vDash_X \varphi^{\mathsf{f}}$. $\qquad\square$

Remark 4.5. Caution: the reciprocal of property 4.4 is false. Consider the ANON-sentence $\varphi := \forall x \exists y \, (x \Upsilon y)$ and the model $\mathfrak{M}$ with domain $M = \{a\}$ consisting in only one element a. Then $\varphi^{\mathsf{f}} \equiv \exists y. \top$, and so $\mathfrak{M} \vDash \varphi^{\mathsf{f}}$. But $\mathfrak{M} \nvDash \varphi$, and as φ is a sentence, then $\mathsf{F}\varphi \equiv \varphi$, and so $\mathfrak{M} \nvDash \mathsf{F}\varphi$. Hence $\varphi^{\mathsf{f}} \nvDash \mathsf{F}\varphi$.

As a consequence, $\mathsf{F}\varphi$ needs not be logically equivalent to φ^{f}, even for φ that are already flat.

4.2   Basic properties of the flattening operator

We start by stating some simple facts about the application of the flattening operator to the most well-known atoms of dependency. The proofs are straightforward and are thus omitted.

Proposition 4.6.

(*i*) *for first-order literals α: $\mathsf{F}\alpha \equiv \alpha$,*   (*iv*) *for inclusion atoms: $\mathsf{F}(\vec{t_1} \subseteq \vec{t_2}) \equiv \vec{t_1} = \vec{t_2}$,*

(*ii*) *for dependency atoms: $\mathsf{F}(=\!(\vec{t_1}, t_2)) \equiv \top$,*   (*v*) *for exclusion atoms: $\mathsf{F}(\vec{t_1} \mid \vec{t_2}) \equiv \vec{t_1} \neq \vec{t_2}$,*

(*iii*) *for anonymity atoms: $\mathsf{F}(\vec{t_1} \Upsilon t_2) \equiv \bot$,*   (*vi*) *for independency atoms: $\mathsf{F}(\vec{t_2} \perp_{\vec{t_1}} \vec{t_3}) \equiv \top$.*

We now state simple properties about the distributivity of the flattening operator with respect to the Boolean connectives. The proof of the next proposition is a routine application of the semantics.

Proposition 4.7. *For any team-based formulas φ, ψ it holds that*

$$\mathsf{F}(\varphi \wedge \psi) \equiv \mathsf{F}\varphi \wedge \mathsf{F}\psi.$$

If, φ, ψ are both satisfied by the empty team, then additionally

$$\mathsf{F}(\varphi \vee \psi) \equiv \mathsf{F}\varphi \vee \mathsf{F}\psi.$$

When the logic does not have the empty team property, this proposition does not hold. A counter-example is provided by the formula $\mathsf{NE} \vee \mathsf{NE}$, where NE is satisfied by a team if the team is non-empty.

We continue by studying commutativity of the flattening operator with the quantifiers.

Definition 4.8. A formula φ is called *downwards flat* (DF) if

$$\mathfrak{M} \vDash_X \varphi \implies \forall s \in X, \mathfrak{M} \vDash_{\{s\}} \varphi$$

for all models $\mathfrak{M}$ and for all teams X over $\mathfrak{M}$ such that $\text{dom}(X) \supseteq \text{FV}(\varphi)$. Similarly, φ is called *upwards flat* (UF) if

$$(\forall s \in X, \mathfrak{M} \vDash_{\{s\}} \varphi) \implies \mathfrak{M} \vDash_X \varphi$$

for all models $\mathfrak{M}$ and for all teams X over $\mathfrak{M}$ such that $\text{dom}(X) \supseteq \text{FV}(\varphi)$.
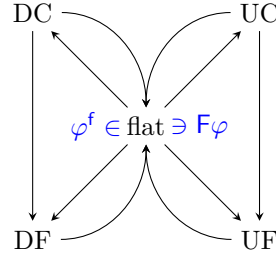
Figure 4.1: Relation between all flatness notions

We sum-up the relations between the different flatness notions we introduced so far in the figure 4.1.

One can easily remark that the property of downwards closure is strictly stronger assumption than downwards flatness. However, it will be a sufficient assumption for the following propositions.

Proposition 4.9. (*i*) *If* $\exists x\varphi$ *is downwards flat, then* $\mathsf{F}(\exists x\varphi) \vDash \exists x\mathsf{F}\varphi$.

(*ii*) *If* $\exists x\varphi$ *is upwards flat, then* $\exists x\mathsf{F}\varphi \vDash \mathsf{F}(\exists x\varphi)$.

Proof. We consider the proof of the first claim. Let $\mathfrak{M}$ be a model, and let X be a team over $\mathfrak{M}$ such that $\mathrm{dom}(X) \supseteq \mathrm{FV}(\varphi) \cup \mathrm{FV}(\psi)$. Then

$$
\begin{aligned}
&\mathfrak{M} \vDash_X \mathsf{F}(\exists x\varphi) \\
\implies{}& \forall s \in X, \mathfrak{M} \vDash_{\{s\}} \exists x\varphi \\
\implies{}& \forall s \in X, \exists H : \{s\} \to \mathcal{P}^*(M), \mathfrak{M} \vDash_{\{s\}[H/x]} \varphi && \text{(semantics of } \exists\text{)} \\
\implies{}& \exists H' : X \to \mathcal{P}^*(M), \forall s \in X, \mathfrak{M} \vDash_{\{s\}[H'/x]} \varphi \\
\implies{}& \exists H' : X \to \mathcal{P}^*(M), \forall s \in X, \forall s' \in \{s\}[H'/x], \mathfrak{M} \vDash_{\{s'\}} \varphi && \text{(downwards flatness)} \\
\implies{}& \exists H' : X \to \mathcal{P}^*(M), \forall s \in X[H'/x], \mathfrak{M} \vDash_{\{s\}} \varphi \\
\implies{}& \exists H' : X \to \mathcal{P}^*(M), \mathfrak{M} \vDash_{X[H'/x]} \mathsf{F}\varphi \\
\implies{}& \mathfrak{M} \vDash_X \exists x(\mathsf{F}\varphi)
\end{aligned}
$$

$\square$

Proposition 4.10. (*i*) *if* $\forall x\varphi$ *is downwards flat, then* $\mathsf{F}(\forall x\varphi) \vDash \forall x(\mathsf{F}\varphi)$,

(*ii*) *if* $\forall x\varphi$ *is upwards flat, then* $\forall x(\mathsf{F}\varphi) \vDash \mathsf{F}(\forall x\varphi)$.

Proof. Again we consider the proof of the first claim only as the proof is similar in the second case. Let $\mathfrak{M}$ be a model, and let X be a team over $\mathfrak{M}$ such that $\mathrm{dom}(X) \supseteq \mathrm{FV}(\varphi) \cup \mathrm{FV}(\psi)$. Then

$$
\begin{aligned}
&\mathfrak{M} \vDash_X \mathsf{F}(\forall x\varphi) \\
\implies{}& \forall s \in X, \mathfrak{M} \vDash_{\{s\}} \forall x\varphi \\
\implies{}& \forall s \in X, \mathfrak{M} \vDash_{\{s\}[M/x]} \varphi \\
\implies{}& \forall s \in X, \forall s' \in \{s\}[M/x], \mathfrak{M} \vDash_{\{s'\}} \varphi && \text{(downwards flatness)} \\
\implies{}& \forall s \in X[M/x], \mathfrak{M} \vDash_{\{s\}} \varphi \\
\implies{}& \mathfrak{M} \vDash_{X[M/x]} \mathsf{F}\varphi \\
\implies{}& \mathfrak{M} \vDash_X \forall x(\mathsf{F}\varphi)
\end{aligned}
$$

□

4.3   Preservation of closure properties

The proof of the next lemma is a routine application of the semantics.

Lemma 4.11.   (*i*) *If φ is downwards closed (resp. union closed), then $\mathsf{F}\varphi$ is also downwards closed (resp. union closed).*

(*ii*) *If φ is downwards flat (resp. upwards flat), then $\mathsf{F}\varphi$ is also downwards flat (resp. upwards flat).*

Corollary 4.12. *Let $\mathcal{L}$ be team-based logic extending FO by some dependencies and operators.*

(*i*) *If $\mathcal{L}$ is downwards closed (resp. union closed, resp. downwards flat, resp. upwards flat), then so is $\mathcal{L}(\mathsf{F})$.*

(*ii*) *If $\mathcal{L}$ is flat, then $\mathcal{L} \equiv \mathcal{L}(\mathsf{F})$.*

5   The expressive power of the flattening operator

We ask now, does the flattening operator, when added to a logic, increase the expressive power of the logic? In [Gal18] the flattening operator F was said to be *safe* for a team-based logic $\mathcal{L}$ if $\mathcal{L} \equiv \mathcal{L}(\mathsf{F})$. The idea is that the operator F can be "safely" added to the logic, as it does not increase its expressive power. In this section we systematically study the effect of F to the expressive power of various team-based logics.

5.1   Adding F to inclusion and anonymity logic

The goal of this subsection is to show that inclusion logic together with the flattening operator $\mathsf{FO}\,(\subseteq, \mathsf{F})$ is equivalent to the positive fragment of greatest fixed-point logic GFP^+. From this it follows that $\mathsf{FO}\,(\subseteq, \mathsf{F})$ has exactly the same expressive power as $\mathsf{FO}(\subseteq)$ by the results of [GH13].

Theorem 5.1. *For every $\mathsf{FO}\,(\subseteq, \mathsf{F})$-formula $\varphi(\vec{x})$ with free variables in $\vec{x} = (x_1, \ldots, x_n)$ there is a GFP^+-formula $\varphi^* = \varphi^*(R, \vec{x})$ such that $\mathrm{ar}(R) = n$, R occurs only positively in φ^*, and the condition*

$$\mathfrak{M} \vDash_X \varphi(\vec{x}) \quad \textit{iff} \quad \langle \mathfrak{M}, R \coloneqq X(\vec{x}) \rangle \vDash_s \varphi^*(R, \vec{x}) \textit{ for all } s \in X$$

holds for all models $\mathfrak{M}$ and all teams X over $\mathfrak{M}$ with $\mathrm{dom}(X) \supseteq \{x_1, \ldots, x_n\}$.

Proof. The proof is by structural induction on φ. The cases for first-order literals, inclusion atoms, connectives, and quantifiers are treated as in [GH13, Theorem 15]. We show the inductive case corresponding to F.

If $\varphi(\vec{x})$ is of the form $\mathsf{F}\psi(\vec{x})$, then we define

$$\varphi^*(R, \vec{x}) = \forall \vec{y}(\neg R\vec{y} \vee \theta(\vec{x}, \vec{y}))$$

where $\theta(\vec{x}, \vec{y}) = \psi^*(S, \vec{x})\left[(\vec{y} = \vec{t})/S(\vec{t})\right]$, that is, θ is the formula $\psi^*(S, \vec{x})$ in which we replaced each atom of the form $S(\vec{t})$ by the atom $\vec{y} = \vec{t}$. Let $\mathfrak{M}$ be a model and X be a team over $\mathfrak{M}$ with $\mathrm{dom}(X) \supseteq \{x_1, \ldots, x_n\}$. One can remark that

$$\begin{aligned}
\mathfrak{M} \vDash_X \mathsf{F}\psi(\vec{x}) &\iff \forall s \in X. & \mathfrak{M} \vDash_{\{s\}} \psi(\vec{x}) \\
&\iff \forall s \in X.\forall s' \in \{s\}. & \langle \mathfrak{M}, S := \{s\}\,(\vec{x})\rangle \vDash_{s'} \psi^*(S, \vec{x}) & \quad \text{(by IH on } \psi(\vec{x})) \\
&\iff \forall s \in X. & \langle \mathfrak{M}, S := \{s\}\,(\vec{x})\rangle \vDash_s \psi^*(S, \vec{x}) \\
&\iff \forall s \in X. & \langle \mathfrak{M}, S := \{s(\vec{x})\}\rangle \vDash_s \psi^*(S, \vec{x}) \\
&\iff \forall s \in X. & \mathfrak{M} \vDash_s \psi^*(S, \vec{x})\left[(s(\vec{x}) = \vec{t})/S(\vec{t})\right] & \quad (5.1)
\end{aligned}$$

The last equivalence is justified by the following observation: as S is assigned to a relation with a unique element $s(\vec{x})$, having an atom on the form $S(\vec{t})$ is ψ^* is equivalent to have the atom $s(\vec{x}) = \vec{t}$ instead (we do the test "by hand"), so we perform this substitution in ψ^* while preserving equivalence.

Now, suppose $\mathfrak{M} \vDash_X \mathsf{F}\psi(\vec{x})$, then condition 5.1 holds. We want to show

$$\langle \mathfrak{M}, R := X(\vec{x})\rangle \vDash_s \forall \vec{y}(R\vec{y} \to \psi^*(S, \vec{x})\left[(\vec{y} = \vec{t})/S(\vec{t})\right]) \quad \text{for all } s \in X$$

To do so, let $\vec{b} \in M$ such that $\vec{b} \in X(\vec{x})$. Then there is an assignment $s \in X$ such that $\vec{b} = s(\vec{x})$, we fix such an assignment. And by condition 5.1 we obtain

$$\mathfrak{M} \vDash_s \psi^*(S, \vec{x})\left[(\vec{b} = \vec{t})\Big/S(\vec{t})\right]$$

Conversely, suppose that

$$\langle \mathfrak{M}, R := X(\vec{x})\rangle \vDash_s \forall \vec{y}(R\vec{y} \to \psi^*(S, \vec{x})\left[(\vec{y} = \vec{t})/S(\vec{t})\right]) \quad \text{for all } s \in X$$

So by taking $\vec{y} := s(x)$, as $s(\vec{x}) \in \{s(x) \mid s \in X\} \overset{\text{def}}{=} X(\vec{x})$, then the condition "$R\vec{y}$" is satisfied, so we get

$$\mathfrak{M} \vDash_s \psi^*(S, \vec{x})\left[(s(\vec{x}) = \vec{t})/S(\vec{t})\right] \quad \text{for all } s \in X$$

and the result follows from condition 5.1. □

Corollary 5.2.

(*i*) *The logics* $\mathsf{FO}\,(\subseteq, \mathsf{F})$ *and* $\mathsf{FO}(\subseteq)$ *are equivalent for sentences.*

(*ii*) *The logics* $\mathsf{FO}\,(\Upsilon, \mathsf{F})$ *and* $\mathsf{FO}(\Upsilon)$ *are equivalent for sentences.*

Proof. The first claim follows by theorem 5.1.

The second claim follows from the fact inclusion atoms can be expressed using anonymity atoms and vice versa. □

5.2   Adding F to unary inclusion logic

Galliani [Gal18] proved that constancy atoms increase the expressive power of unary inclusion logic. More precisely, he proved that non-connectedness of graphs can be expressed using constancy atoms and unary inclusion atoms, but not in terms of unary inclusion atoms alone. The next proposition shows that the same effect can be obtained using F and inclusion atoms.

Notation 5.3. Let ϕ is a first-order formula, and let ψ be a team-based formula. We will then denote $\varphi \hookrightarrow \psi$ as an abbreviation for $\neg\varphi \vee (\varphi \wedge \psi)$.

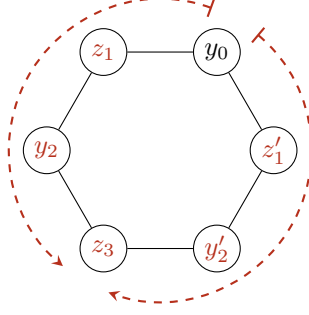We thank Pietro Galliani for suggesting the following proposition.

Figure 5.1: Illustration of the anonymity atom mechanism to fill a cycle

Proposition 5.4. *The following* $\mathsf{FO}\left(\subseteq_1, \mathsf{F}\right)$ *sentence*

$$\exists y(\mathsf{F}(\exists x(x \neq y \wedge \forall z(E(x,z) \hookrightarrow z \subseteq x))))$$

is true in a graph $\mathfrak{G} = \langle V, E \rangle$ *if and only if the graph* $\mathfrak{G}$ *is disconnected.*

Proof. It is straightforward to check that the formula above is logically equivalent with the formula used in [Gal18] to show the same result.   □

Corollary 5.5. *The flattening operator* F *increases the expressive power of* $\mathsf{FO}(\subseteq_1)$.

5.3   Adding F to unary anonymity logic

5.3.1   Expressivity of unary anonymity atoms and the flattening operator

For any $n \in \mathbb{N}$, we denote by $\mathfrak{A}_n$ a graph consisting of two cycles each of length 2^{n+1} and $\mathfrak{B}_n$ a graph consisting of a single cycle of length 2^{n+2}.

Proposition 5.6. *The following* $\mathsf{FO}(\Upsilon_1, \mathsf{F})$ *sentence*

$$\exists x \mathsf{F}(\exists y(y \neq x \wedge \forall z(E(y,z) \hookrightarrow (z \neq x \wedge y \Upsilon z \wedge z \Upsilon y))))$$

is true in the models $\mathfrak{A}_n$ *and is false in* $\mathfrak{B}_n$.

Proof. The intuition behind this $\mathsf{FO}\left(\Upsilon_1, \mathsf{F}\right)$-sentence is as follows: consider a cycle of even length. We would like to fill it completely with variable assignments, in order to be able to detect if there exists another node in the graph that is not part of the considered cycle. To do so, we are going to use the mechanism of the anonymity atom to fill this cycle. More precisely, we will partition the cycle into two sets of nodes, that will be assigned to y and z respectively, in such a way that there will be an alternation between the values taken by y and z along the cycle. This is illustrated in the figure 5.1.

This filling process is done via the sub-formula $E(y,z) \hookrightarrow (y \Upsilon z \wedge z \Upsilon y)$. Here, the black node y_0 in the figure 5.1 corresponds to a starting value for the variable y. Then $E(y,z) \hookrightarrow y \Upsilon z$ forces the variable z to take two different values z_1 and z_1' that are in fact the two adjacent vertices of y_0. Then $E(y,z) \hookrightarrow z \Upsilon y$ firstly forces the variable y to take two different values y_2 and y_0 that are in fact the two adjacent vertices of z_1, and secondly forces the variable y to take two different values y_2' and y_0 that are in fact the two adjacent vertices of z_1'. And so on. At the end, as the considered cycle is of even length, then the last two values assigned to the variable z actually coincide. Such filled components then satisfy the clause $\forall z(E(y,z) \hookrightarrow y \Upsilon z \wedge z \Upsilon y)$.

The whole objective will be to succeed in distinguishing the graphs $\mathfrak{A}_n$ from the graphs $\mathfrak{B}_n$. In fact, the graphs $\mathfrak{A}_n$ and $\mathfrak{B}_n$ are all composed of one or two cycles of even length.

At the end of the filling process, we obtained a cycle filled with an alternation of values for the variables z and y. In order to distinguish $\mathfrak{A}_n$ from $\mathfrak{B}_n$, we need to find an x such that it does not belong to the alternation of y and z. Hence, we need to check both that $y \neq x$ and $z \neq x$, which gives

$$y \neq x \wedge \forall z(E(y,z) \hookrightarrow (z \neq x \wedge y \Upsilon z \wedge z \Upsilon y))$$

The end of the construction of the desired formula is identical to the one we made for the unary inclusion logic together with the flattening operator F, and at the end we finally get

$$\exists x \mathsf{F}(\exists y(y \neq x \wedge \forall z(E(y,z) \hookrightarrow (z \neq x \wedge y \Upsilon z \wedge z \Upsilon y))))$$

By construction, it is clear that this formula is true for the graphs $\mathfrak{A}_n$ but is false for the graphs $\mathfrak{B}_n$. $\qquad\square$

5.3.2   Unary anonymity logic does not express non-connectedness

The next technical lemma will be used to show that the structures $\mathfrak{A}_n$ and $\mathfrak{B}_n$ from the previous section cannot be separated by formulas of unary anonymity logic.

Let $\mathfrak{M}$ be a model, let X be a team over $\mathfrak{M}$, and let $\mathcal{F} \subseteq \mathrm{Aut}\,(\mathfrak{M})$. Then we define the *team closure of X under $\mathcal{F}$*, written $\mathrm{Cl}_{\mathcal{F}}\,(X)$, as the set of all assignments obtained by applying all automorphisms from $\mathcal{F}$ to all assignments of X, that is

$$\mathrm{Cl}_{\mathcal{F}}\,(X) \overset{\mathrm{def}}{=} \{f\,(s) \mid s \in X, f \in \mathcal{F}\}$$

For $\mathcal{F} = \mathrm{Aut}\,(\mathfrak{M})$, we simply write $\mathrm{Cl}\,(X)$ instead of $\mathrm{Cl}_{\mathrm{Aut}(\mathfrak{M})}\,(X)$, that is

$$\mathrm{Cl}\,(X) \overset{\mathrm{def}}{=} \{f\,(s) \mid s \in X, f \in \mathrm{Aut}\,(\mathfrak{M})\}$$

Recall that a *magma* is a set M equipped with a binary operation $\star$ that sends any two elements from M to another element in M, and in this case we call $\star$ an *internal binary operation*. We call a *unitary magma* any magma whose internal binary operation admits a neutral element (which is then unique), that is, if there is $e \in M$ such that for all $m \in M$ we get $e \star m = m \star e = m$.

Lemma 5.7. *Let $\mathfrak{M}$ be a model. Let $\mathcal{F} \subseteq \mathrm{Aut}(\mathfrak{M})$ be a unitary magma such that for any two points $m_1, m_2 \in M$ there exists an automorphism $f \in \mathcal{F}$ such that $f(m_1) = m_1$ and $f(m_2) \neq m_2$. Then for all teams X over $\mathfrak{M}$ such that $\mathrm{Cl}_{\mathcal{F}}(X) = X$ and all formulas $\varphi \in \mathsf{FO}(\Upsilon_1)$ with free variables in $\mathrm{dom}(X)$ we have that*

$$\mathfrak{M} \vDash_X \varphi \quad \textit{iff} \quad \mathfrak{M} \vDash_X \varphi^{\mathsf{f}}$$

Proof. It suffices to show the right to left direction that is proved via structural induction. The proof for the connectives and quantifiers are similar to that of Lemma 21 in [Gal18].

Assume $\varphi = x_1 \Upsilon x_2$. As $(x_1 \Upsilon x_2)^{\mathsf{f}} = \top$ we need to prove that $\mathfrak{M} \vDash_X x_1 \Upsilon x_2$ only from the hypothesis of the lemma, that is, whenever X is a team whose domain contains the variables x_1 and x_2 and such that $X = \mathrm{Cl}_{\mathcal{F}}(X)$. But this is the case. Indeed, suppose that $s(x_1) = m_1$ and $s(x_2) = m_2$. Then by assumption of the lemma, there is an automorphism $f \in \mathcal{F}$ such that $f(m_1) = m_1$ and $f(m_2) \neq m_2$, and since $X = \mathrm{Cl}_{\mathcal{F}}(X)$ there exists some assignment $s' \in X$ such that $s' = f(s)$, that is, $s'(x) = f(s(x))$ for all $x \in \mathrm{dom}(s)$. This implies in particular that

$$\begin{cases} s'(x_1) = f(s(x_1)) = f(m_1) = m_1 = s(x_1) \\ s'(x_2) = f(s(x_2)) = f(m_2) \neq m_2 = s(x_2) \end{cases}$$

and thus, for any assignment $s \in X$ there exists some assignment $s' \in X$ such that $s'(x_1) = s(x_1)$ but $s'(x_2) \neq s(x_2)$. This shows that $\mathfrak{M} \vDash_X x_1 \Upsilon x_2$, as required. $\qquad \square$

Corollary 5.8. *The flattening operator increases the expressive power of unary anonymity logic.*

Proof. By Proposition 5.6, the formula Θ separates the models $\mathfrak{A}_n$ and $\mathfrak{B}_n$. Assume for a contradiction that Θ is equivalent to some φ of unary anonymity logic. By Lemma 5.7, φ is equivalent to φ^{f} in the models $\mathfrak{A}_n$ and $\mathfrak{B}_n$, that is

$$\mathfrak{A}_n \vDash \varphi^{\mathsf{f}} \iff \mathfrak{B}_n \vDash \varphi^{\mathsf{f}},$$

where φ^{f} is a first-order formula. But by an Ehrenfeucht-Fraïssé game argument, this is impossible. $\qquad \square$

5.4   Adding F to dependence, exclusion and independence logic

In this section we show that $\mathsf{FO}(=(\cdot,\cdot),\mathsf{F})$ has exactly the same expressive power as $\mathsf{FO}(=(\cdot,\cdot))$. We will use the same strategy as in subsection 5.1. More precisely, we will show that $\mathsf{FO}(=(\cdot,\cdot),\mathsf{F}) \subseteqq \mathsf{ESO}$, and as it is well known that $\mathsf{ESO} \equiv \mathsf{FO}(=(\cdot,\cdot))$, we will obtain the result. The same idea works also for exclusion and independence logic.

Exactly as for inclusion logic and GFP^+ (theorem 5.1), we have to adapt the equivalence between $\mathsf{FO}(=(\cdot,\cdot),\mathsf{F})$ and ESO because they do not use the same semantics.

Recall that a sentence $\phi(R)$ from second-order logic is called *downwards closed* if $(\mathfrak{M}, A) \vDash \phi(R)$ and $A' \subseteq A$ imply $(\mathfrak{M}, A') \models \phi(R)$.

Lemma 5.9. *For every $\mathsf{FO}(=(\cdot,\cdot),\mathsf{F})$-formula $\varphi(\vec{x})$ with the free variables in $\vec{x} = x_1, \ldots, x_n$, there is an ESO-formula $\varphi^* \equiv \varphi^*(R)$ which is downwards closed and such that $\operatorname{ar}(R) = n$, and that satisfies the following condition:*

$$\mathfrak{M} \vDash_X \varphi(\vec{x}) \quad \text{iff} \quad \langle \mathfrak{M}, R \coloneqq X(\vec{x}) \rangle \vDash \varphi^*(R) \vee \forall \vec{x} \neg R(\vec{x}) \tag{5.2}$$

holds for all models $\mathfrak{M}$ and all teams X over $\mathfrak{M}$ with $\operatorname{dom}(X) \supseteq \{\vec{x}\}$.

Proof. We consider the inductive case corresponding to $\varphi = \mathsf{F}\psi$ as the other cases can be dealt with analogously to [Vää07, Theorem 6.2]. It turns out that the formula φ^* that we need to take is exactly the same as for the translation from $\mathsf{FO}(\subseteq,\mathsf{F})$ to GFP^+ which is clearly in ESO assuming ψ^* is. $\qquad \square$

Corollary 5.10. *The operator F does not increase the expressive power of dependence and exclusion logic.*

Proof. This follows by lemma 5.9 as any downwards closed ESO-formula $\varphi(R)$ can be translated into a dependence logic formula satisfying (5.2), see [KV09; KV11]. The claim for exclusion logic follows from the fact that exclusion atoms can be expressed using dependence atoms and vice versa. $\qquad \square$

Corollary 5.11. *The operator F does not increase the expressive power of independence logic.*

Proof. The proof is analogous to the case of dependence logic using the analogue of lemma 5.9 (without the assumption of downward closure) ans its converse shown in [Gal12]. $\qquad \square$

5.5   Adding F to fragments of dependence logic

We begin by showing that the flattening operator reduces to the flattening of a formula for existential dependence logic formulas.

Proposition 5.12. *If $\varphi \in \mathsf{FO}(=(\cdot,\cdot))$ is existential then $\mathsf{F}\varphi \equiv \varphi^{\mathsf{f}}$.*

Proof. Recall that $\mathfrak{M} \vDash_X \mathsf{F}\varphi$ iff $\mathfrak{M} \vDash_{\{s\}} \varphi$, for all $s \in X$. Now it is easy to show using induction on φ using downwards closure (and singleton-valued supplements functions) that $\mathfrak{M} \vDash_{\{s\}} \varphi$ iff $\mathfrak{M} \vDash_{\{s\}} \varphi^f$ for any singleton team $\{s\}$, from which the claim follows.   □

Next we turn to the so-called constancy logic the formulas of which has the following normal form.

Proposition 5.13 ([Gal12]). *Let $\varphi(\vec{x}) \in \mathsf{FO}(=(\cdot))$. Then φ is logically equivalent to a constancy logic formula of the form*

$$\exists \vec{y} \left(\bigwedge_i =(y_i) \wedge \alpha(\vec{x}, \vec{y}) \right)$$

for some first-order formula α.

Proposition 5.14. *The operator F does not increase the expressive power of constancy logic, that is $\mathsf{FO}(=(\cdot), \mathsf{F}) \equiv \mathsf{FO}(=(\cdot))$.*

Proof. We prove the claim using induction on $\varphi \in \mathsf{FO}(=(\cdot), \mathsf{F})$. We consider only the case where $\varphi = \mathsf{F}\psi$. By the induction hypothesis and the previous lemma, we get that ψ is equivalent with some formula in the normal form:

$$\exists \vec{y} \left(\bigwedge_i =(y_i) \wedge \alpha(\vec{x}, \vec{y}) \right)$$

for some first-order formula α. It now follows from Proposition 5.12 that $\varphi \equiv \exists \vec{y}\alpha$.   □

5.6   Adding F to FO with the Boolean negation

In this section we consider the effects of adding F to $\mathsf{FO}(\sim)$ (also denoted by FON in the following).

Definition 5.15. Let φ be a σ-formula, $\mathfrak{M}$ a σ-model and X a team over $\mathfrak{M}$ such that $\mathrm{dom}(X) \supseteq \mathrm{FV}(\varphi)$. We define the truth of the formula $\sim\varphi$ (the Boolean negation of φ) in the model $\mathfrak{M}$ and the team X as:

$$\mathfrak{M} \vDash_X \sim\varphi \iff \mathfrak{M} \nvDash_X \varphi$$

We will use the following notations from [Lüc20]:

- $\mathsf{E}\varphi \overset{\text{def}}{=} \sim\neg\varphi$, meaning that at least one assignment in the considered team satisfies φ;

- $\varphi \oslash \psi \overset{\text{def}}{=} \sim(\sim\varphi \wedge \sim\psi)$, so that $\mathfrak{M} \vDash_X \varphi \oslash \psi$ iff $\mathfrak{M} \vDash_X \varphi$ or $\mathfrak{M} \vDash_X \psi$, which corresponds to the Boolean disjunction (a.k.a. intuitionistic disjunction).

We will utilize the following normal form theorem.

Proposition 5.16 ([**Lüc20**]). *Let $\varphi(\vec{x}) \in \mathsf{FO}(\sim)$. Then φ is logically equivalent to a formula of the form*

$$\bigvee_{i \in I} \left(\alpha_i \wedge \bigwedge_{j \in J_i} \mathsf{E}\beta_{i,j} \right)$$

for some finite sets I and J_i and for first-order formulas α_i and $\beta_{i,j}$ for all $i \in I$ and $j \in J_i$.

Proposition 5.17. $\mathsf{FO}(\sim, \mathsf{F}) \equiv \mathsf{FO}(\sim)$.

Proof. To prove the non-trivial direction ($\Rightarrow$), we need to show that for all $\varphi(\vec{x}) \in \mathsf{FO}(\sim, \mathsf{F})$, there is $\varphi^*(\vec{x}) \in \mathsf{FO}(\sim)$ such that for all models $\mathfrak{M}$ and for all teams X over $\mathfrak{M}$ such that $\mathrm{dom}(X) \supseteq \{\vec{x}\}$, the following holds

$$\mathfrak{M} \vDash_X \varphi(\vec{x}) \iff \mathfrak{M} \vDash_x \varphi^*(\vec{x})$$

We prove it by structural induction on φ. The only interesting case is when $\varphi(\vec{x}) = \mathsf{F}\psi(\vec{x})$.

According to proposition 5.16, we get that

$$\psi(\vec{x}) \equiv \bigvee_{i \in I} \left(\alpha_i \wedge \bigwedge_{j \in J_i} \mathsf{E}\beta_{i,j} \right)$$

for some finite sets I and J_i and for first-order formulas α_i and $\beta_{i,j}$ for all $i \in I$ and $j \in J_i$. It is now a routine application of the semantics to show that φ is equivalent with the first-order formula

$$\bigvee_{i \in I} \left(\alpha_i \wedge \bigwedge_{j \in J_i} \beta_{i,j} \right),$$

using the facts that in singleton teams $\mathsf{E}\beta$ is equivalent with β and that for all $s \in X$

$$\mathfrak{M} \vDash_{\{s\}} \bigvee_i \varphi_i \quad \text{iff} \quad \mathfrak{M} \vDash_X \bigvee_i \varphi_i,$$

for first-order formulas φ_i. $\qquad\square$

6   Summing up expressive powers

We sum up all the previous results in the following figure 6.1. It shows when a fragment is comparable to another in terms of expressive power, and when it is the case, it indicates which fragment is more expressive than another. The red edges are the comparisons that are not known at the moment, but which we believe to be true. These questions have some echoes in finite model theory where beyond the seminal and old results about the monadic case (i.e. arity 1) of existential second order logic, very few separation results are known [Ajt83; AFS00; DLS98]. Considering the tight connections between arity based fragments of existential second order logic and of some team logics [DK12], such a situation is all but surprising.

Conclusion

We have explored the flattening operator F within the framework of team semantics, a powerful extension of classical first-order logic that allows for a more powerful treatment of dependencies and independencies in logical formulas. The flattening operator was originally introduced by Wilfrid
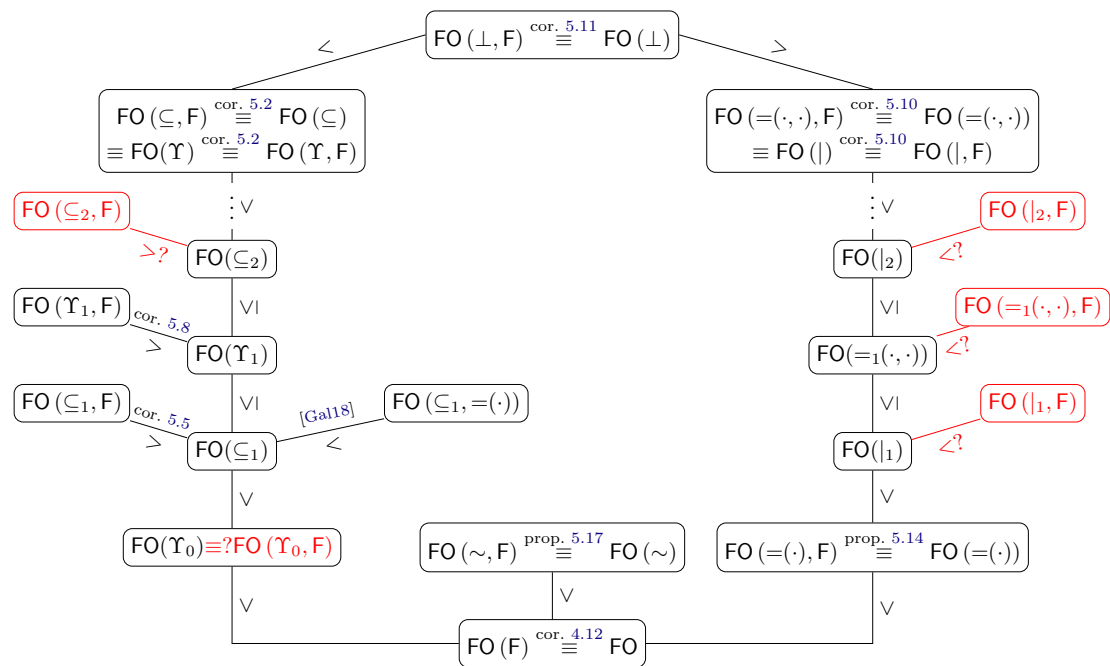
Figure 6.1: Expressive power of logics on the level of formulas

Hodges in 1997 [Hod97], but its potential and implications have remained largely unexplored until now.

Through our investigation, we demonstrated that the introduction of the flattening operator F does not add expressive power to logics such as dependence logic, inclusion logic, constancy logic, inconstancy logic and $\mathsf{FO}(\sim)$. It turns out that this is not the case with the fixed arity fragments of these logics. For example, we have seen that unary inclusion logic with the flattening operator $\mathsf{FO}(\subseteq_1, \mathsf{F})$ is strictly more expressive than unary inclusion logic $\mathsf{FO}(\subseteq_1)$. We conjecture that the same is true of all the other fragments of fixed arity, too.

The work presented here opens several avenues for future research. One possible direction is the further exploration of the computational complexity associated with the translation of some logic fragments augmented with the flattening operator into SAT problems, along the lines of [DKV22].

Acknowledgments

References

[APV21]   Samson Abramsky, Joni Puljujärvi, and Jouko Väänänen. *Team Semantics and Independence Notions in Quantum Physics*. 2021. arXiv: 2107.10817 [math.LO] (cit. on p. 1).

[Ajt83]   Miklós Ajtai. "Σ_1^1-Formulae on finite structures". In: *Annals of Pure and Applied Logic* 24.1 (1983), pp. 1–48. DOI: 10.1016/0168-0072(83)90038-6 (cit. on p. 18).

[AFS00]   Miklós Ajtai, Ronald Fagin, and Larry Stockmeyer. "The Closure of Monadic NP". In: *Journal of Computer and System Sciences* 60.3 (2000), pp. 660–716. DOI: 10.1006/JCSS.1999.1691 (cit. on p. 18).

[CG22]   Ivano Ciardelli and Gianluca Grilletti. "Coherence in inquisitive first-order logic". In: *Annals of Pure and Applied Logic* 173.9 (2022), pp. 103–155. DOI: 10.1016/j.apal.2022.103155 (cit. on p. 3).

[CIY20]   Ivano Ciardelli, Rosalie Iemhoff, and Fan Yang. "Questions and Dependency in Intuitionistic Logic". In: *Notre Dame Journal of Formal Logic* 61.1 (2020), pp. 75–115. DOI: 10.1215/00294527-2019-0033 (cit. on p. 1).

[Cor+19]   Jukka Corander, Antti Hyttinen, Juha Kontinen, Johan Pensar, and Jouko Väänänen. "A logical approach to context-specific independence". In: *Annals of Pure and Applied Logic* 170.9 (2019), pp. 975–992. DOI: 10.1016/j.apal.2019.04.004 (cit. on p. 1).

[Dur+18a]   Arnaud Durand, Miika Hannula, Juha Kontinen, Arne Meier, and Jonni Virtema. "Approximation and dependence via multiteam semantics". In: *Annals of Mathematics and Artificial Intelligence* 83.3-4 (2018), pp. 297–320. DOI: 10.1007/s10472-017-9568-4 (cit. on p. 1).

[Dur+18b]   Arnaud Durand, Miika Hannula, Juha Kontinen, Arne Meier, and Jonni Virtema. "Probabilistic Team Semantics". In: *Foundations of Information and Knowledge Systems.* Lecture Notes in Computer Science. Springer International Publishing, 2018, pp. 186–206. DOI: 10.1007/978-3-319-90050-6_11 (cit. on p. 1).

[DK12]   Arnaud Durand and Juha Kontinen. "Hierarchies in Dependence Logic". In: *ACM Transactions on Computational Logic* 13.4 (2012), 31:1–31:21. DOI: 10.1145/2362355.2362359 (cit. on p. 18).

[Dur+22]   Arnaud Durand, Juha Kontinen, Nicolas de Rugy-Altherre, and Jouko Väänänen. "Tractability Frontier of Data Complexity in Team Semantics". In: *ACM Transactions on Computational Logic* 23.1 (2022), 3:1–3:21. DOI: 10.1145/3471618 (cit. on p. 1).

[DKV22]   Arnaud Durand, Juha Kontinen, and Jouko Väänänen. *Modular SAT-based techniques for reasoning tasks in team semantics.* 2022. arXiv: 2204.00576 [cs.LO] (cit. on p. 19).

[DLS98]   Arnaud Durand, Clemens Lautemann, and Thomas Schwentick. "Subclasses of Binary NP". In: *Journal of Logic and Computation* 8.2 (1998), pp. 189–207. DOI: 10.1093/logcom/8.2.189 (cit. on p. 18).

[Gal12]   Pietro Galliani. "Inclusion and exclusion dependencies in team semantics – On some logics of imperfect information". In: *Annals of Pure and Applied Logic* 163.1 (2012), pp. 68–84. DOI: 10.1016/j.apal.2011.08.005 (cit. on pp. 2, 5, 16, 17).

[Gal18]   Pietro Galliani. "Safe Dependency Atoms and Possibility Operators in Team Semantics". In: *Electronic Proceedings in Theoretical Computer Science* 277 (Sept. 2018), pp. 58–72. DOI: 10.4204/eptcs.277.5 (cit. on pp. 12–15, 19).

[GH13]   Pietro Galliani and Lauri Hella. "Inclusion logic and fixed point logic". In: *Computer Science Logic 2013 (CSL 2013).* Vol. 23. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2013, pp. 281–295. DOI: 10.4230/LIPIcs.CSL.2013.281 (cit. on p. 12).

[GV13]   Erich Grädel and Jouko Väänänen. "Dependence and Independence". In: *Studia Logica* 101.2 (2013), pp. 339–410. DOI: 10.1007/s11225-013-9479-2 (cit. on p. 2).

[HGW22]   Darion Haase, Erich Grädel, and Richard Wilke. "Separation logic and logics with team semantics". In: *Annals of Pure and Applied Logic* 173.10 (2022). DOI: 10.1016/j.apal.2021.103063 (cit. on p. 1).

[HKV20]   Miika Hannula, Juha Kontinen, and Jonni Virtema. "Polyteam semantics". In: *Journal of Logic and Computation* 30.8 (2020), pp. 1541–1566. DOI: 10.1093/logcom/exaa048 (cit. on p. 1).

[HLV24]   Lauri Hella, Kerkko Luosto, and Jouko Väänänen. "Dimension in team semantics". In: *Mathematical Structures in Computer Science* 34.5 (2024), pp. 410–454. DOI: 10.1017/S0960129524000021 (cit. on p. 2).

[Hod97]   Wilfrid Hodges. "Some Strange Quantifiers". In: *Structures in Logic and Computer Science: A Selection of Essays in Honor of A. Ehrenfeucht.* Berlin, Heidelberg: Springer Berlin Heidelberg, 1997, pp. 51–65. DOI: 10.1007/3-540-63246-8_4 (cit. on pp. 2, 8, 19).

[Kon13]   Jarmo Kontinen. "Coherence and computational complexity of quantifier-free dependence logic formulas". In: *Studia Logica* 101.2 (2013), pp. 267–291. DOI: 10.1007/s11225-013-9481-8 (cit. on p. 3).

[KLV13]   Juha Kontinen, Sebastian Link, and Jouko Väänänen. "Independence in Database Relations". In: *Logic, Language, Information, and Computation.* Vol. 8071. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2013, pp. 179–193. DOI: 10.1007/978-3-642-39992-3_17 (cit. on p. 1).

[KV11]   Juha Kontinen and Jouko Väänänen. "Erratum to: On definability in dependence logic". In: *Journal of Logic, Language and Information* 20.1 (2011), pp. 133–134. DOI: 10.1007/s10849-010-9125-6 (cit. on p. 16).

[KV09]   Juha Kontinen and Jouko Väänänen. "On definability in dependence logic". In: *Journal of Logic, Language and Information* 18.3 (2009), pp. 317–332. DOI: 10.1007/s10849-009-9082-0 (cit. on p. 16).

[Lüc20]   Martin Lück. "Team logic: axioms, expressiveness, complexity". PhD thesis. 2020. DOI: 10.15488/9376 (cit. on pp. 17, 18).

[MO22]   Silke Meissner and Martin Otto. "A first-order framework for inquisitive modal logic". In: *The Review of Symbolic Logic* 15.2 (2022), pp. 311–333. DOI: 10.1017/S175502032100037X (cit. on p. 3).

[Par+89]   Jan Paredaens, Paul De Bra, Marc Gyssens, and Dirk Van Gucht. *The structure of the relational database model.* Vol. 17. EATCS Monographs on Theoretical Computer Science. Springer-Verlag, Berlin, 1989, pp. x+231. DOI: 10.1007/978-3-642-69956-6 (cit. on p. 5).

[Rön18]   Raine Rönnholm. "Arity Fragments of Logics with Team Semantics". PhD thesis. 2018. URL: https://urn.fi/URN:ISBN:978-952-03-0912-1 (cit. on p. 5).

[Vää22]   Jouko Väänänen. "An atom's worth of anonymity". In: *Logic Journal of the IGPL* 31.6 (2022), pp. 1078–1083. DOI: 10.1093/jigpal/jzac074 (cit. on p. 5).

[Vää07]   Jouko Väänänen. *Dependence logic: a new approach to independence friendly logic.* London Mathematical Society Student Texts. Cambridge University Press, 2007. DOI: 10.1017/CBO9780511611193 (cit. on pp. 1, 4, 5, 7, 16).

[Vää17]   Jouko Väänänen. "The logic of approximate dependence". In: *Rohit Parikh on Logic, Language and Society*. Vol. 11. Outstanding Contributions to Logic. Springer, 2017, pp. 227–234. DOI: 10.1007/978-3-319-47843-2_12 (cit. on p. 3).

[Yan14]   Fan Yang. "On Extensions and Variants of Dependence Logic". PhD thesis. 2014. URL: http://hdl.handle.net/10138/43011 (cit. on p. 7).