Transition semantics: the dynamics of dependence logic

P. Galliani

Abstract We examine the relationship between dependence logic and game logics. A variant of dynamic game logic, called *Transition Logic*, is developed, and we show that its relationship with dependence logic is comparable to the one between first-order logic and dynamic game logic discussed by van Benthem. This suggests a new perspective on the interpretation of dependence logic formulas, in terms of assertions about *reachability* in games of imperfect information against Nature. We then capitalize on this intuition by developing expressively equivalent variants of dependence logic in which this interpretation is taken to the foreground.

1 Introduction

1.1 Dependence logic

Dependence logic (Väänänen 2007a) is an extension of first-order logic which adds *dependence atoms* of the form $=(t_1, \ldots, t_n)$ to it, with the intended interpretation of "the value of the term t_n is a function of the values of the terms $t_1 \ldots t_{n-1}$."

The introduction of such atoms is roughly equivalent to the introduction of nonlinear patterns of dependence and independence between variables of branching quantifier logic (Henkin 1961) or independence friendly logic (Hintikka and Sandu 1989; Hintikka 1996; Mann et al. 2011): for example, both the branching quantifier logic sentence

P. Galliani (✉)
Department of Mathematics and Statistics, University of Helsinki, Helsinki, Finland
e-mail: pgallian@gmail.com

$$\begin{pmatrix} \forall x \ \exists y \\ \forall z \ \exists w \end{pmatrix} R(x, y, z, w)$$

and the independence friendly logic sentence

$$\forall x \exists y \forall z (\exists w/x, y) R(x, y, z, w)$$

correspond in dependence logic to

$$\forall x \exists y \forall z \exists w (=(z, w) \wedge R(x, y, z, w)),$$

in the sense that all of these expressions are equivalent to the Skolem formula

$$\exists f \exists g \forall x \forall z R(x, f(x), z, g(z)).$$

As this example illustrates, the main peculiarity of dependence logic compared to the others above-mentioned logics lies in the fact that, in dependence logic, the notion of *dependence and independence between variables* is explicitly separated from the notion of quantification. This makes it an eminently suitable formalism for the formal analysis of the properties of *dependence itself* in a first-order setting, and some recent papers (Grädel and Väänänen 2013; Engström 2012; Galliani 2012) explore the effects of replace dependence atoms with other similar primitives such as *independence atoms* (Grädel and Väänänen 2013), *multivalued dependence atoms* (Engström 2012), or *inclusion* or *exclusion* atoms (Galliani 2011, 2012).

Branching quantifier logic, independence friendly logic and dependence logic, as well as their variants, are called *logics of imperfect information*: indeed, the truth conditions of their sentences can be obtained by defining, for every model M and sentence ϕ, an imperfect-information *semantic game* $G^M(\phi)$ between a *Verifier* (also called Eloise) and a *Falsifier* (also called Abelard), and then asserting that ϕ is true in M if and only if the Verifier has a winning strategy in $G^M(\phi)$. As an alternative of this (non-compositional) *Game-Theoretic Semantics*, which is an imperfect-information variant of Hintikka's game-theoretic semantics for first order logic (Hintikka 1968), Hodges introduced in Hodges (1997) *Team Semantics* (also called *Trump Semantics*), a compositional semantics for logics of imperfect information which is equivalent to Game-Theoretic Semantics over sentences and in which formulas are satisfied or not satisfied not by single assignments, but by *sets* of assignments (called *Teams*).

In this work, we will be mostly concerned with team semantics and some of its variants. We refer the reader to the relevant literature (for example to Väänänen 2007a; Mann et al. 2011) for further information regarding these logics: in the rest of this section, we will content ourselves with recalling the definitions and results which will be useful for the rest of this work.

Definition 1.1 (*Assignments and substitutions*) Let M be a first order model and let V be a finite set of variables. Then an *assignment* over M with *domain* V is a function s from V to the set $\mathrm{dom}(M)$ of all elements of M.

Furthermore, for any assignment s over M with domain V, any element $m \in$ dom(M) and any variable v (not necessarily in V), we write $s[m/v]$ for the assignment with domain $V \cup \{v\}$ such that

$$s[m/v](w) = \begin{cases} m & \text{if } w = v; \\ s(w) & \text{if } w \in V \backslash \{v\} \end{cases}$$

for all $w \in V \cup \{v\}$.

Definition 1.2 (*Team*) Let M be a first-order model and let V be a finite set of variables. A *team X* over M with *domain* dom$(X) = V$ is a set of assignments from V to M.

Definition 1.3 (*Relations corresponding to teams*) Let X be a team over M, and let V be a finite set of variables. and let $\vec{v}$ be a finite tuple of variables in its domain. Then $X(\vec{v})$ is the relation $\{s(\vec{v}) : s \in X\}$. Furthermore, we write $\texttt{Rel}(X)$ for $X(\texttt{Dom}(X))$.

As is often the case for Dependence Logic, we will assume that all our formulas are in negation normal form:

Definition 1.4 (*Dependence logic, syntax*) Let Σ be a first-order signature. Then the set of all dependence logic formula with signature Σ is given by

$$\phi ::= R\vec{t} \mid \neg R\vec{t} \mid \, = (t_1, \ldots, t_n) \mid \phi \vee \phi \mid \phi \wedge \phi \mid \exists v \phi \mid \forall v \phi$$

where R ranges over all relation symbols, $\vec{t}$ ranges over all tuples of terms of the appropriate arities, $t_1 \ldots t_n$ range over all terms and v ranges over the set $\texttt{Var}$ of all variables.

The set $\texttt{Free}(\phi)$ of all *free variables* of a formula ϕ is defined precisely as in First Order Logic, with the additional condition that all variables occurring in a dependence atom are free with respect to it.

Definition 1.5 (*Dependence logic, semantics*) Let M be a first-order model, let X be a team over it, and let ϕ be a dependence logic formula with the same signature of M and with free variables in $\texttt{Dom}(X)$. Then we say that X *satisfies ϕ* in M, and we write $M \models_X \phi$, if and only if

> **TS-lit:** ϕ is a first-order literal and $M \models_s \phi$ for all $s \in X$;
> **TS-dep:** ϕ is a dependence atom $= (t_1, \ldots, t_n)$ and any two assignments $s, s' \in X$ which assign the same values to $t_1 \ldots t_{n-1}$ also assign the same value to t_n;
> **TS-$\vee$:** ϕ is of the form $\psi_1 \vee \psi_2$ and there exist two teams Y_1 and Y_2 such that $X = Y_1 \cup Y_2$, $M \models_{Y_1} \psi_1$ and $M \models_{Y_2} \psi_2$;
> **TS-$\wedge$:** ϕ is of the form $\psi_1 \wedge \psi_2$, $M \models_X \psi_1$ and $M \models_X \psi_2$;
> **TS-$\exists$:** ϕ is of the form $\exists v \psi$ and there exists a function $F : X \to \texttt{Dom}(M)$ such that $M \models_{X[F/v]} \psi$, where

$$X[F/v] = \{s[F(s)/v] : s \in X\}$$

TS-$\forall$: ϕ is of the form $\forall v \psi$ and $M \models_{X[M/v]} \psi$, where

$$X[M/v] = \{s[m/v] : s \in X, m \in \mathrm{Dom}(M)\}.$$

The disjunction of dependence logic does not behave like the classical disjunction: for example, it is easy to see that $= (x) \vee = (x)$ is not equivalent to $= (x)$, as the former holds for the team $X = \{\{(x, 0)\}, \{(x, 1)\}\}$ and the latter does not. However, it is possible to define the classical disjunction in terms of the other connectives:

Definition 1.6 (*Classical disjunction*) Let ψ_1 and ψ_2 be two dependence logic formulas, and let u_1 and u_2 be two variables not occurring in them. Then we write $\psi_1 \sqcup \psi_2$ as a shorthand for

$$\exists u_1 \exists u_2 (=(u_1) \wedge \, =(u_2) \wedge ((u_1 = u_2 \wedge \psi_1) \vee (u_1 \neq u_2 \wedge \psi_2))).$$

Proposition 1.7 *For all formulas ψ_1 and ψ_2, all models M with at least two elements*[1] *whose signature contains that of ψ_1 and ψ_2 and all teams X whose domain contains the free variables of ψ_1 and ψ_2*

$$M \models_X \psi_1 \sqcup \psi_2 \Leftrightarrow M \models_X \psi_1 \text{ or } M \models_X \psi_2.$$

The following four proportions are from Väänänen (2007a):

Proposition 1.8 *For all models M and dependence logic formulas ϕ, $M \models_\emptyset \phi$.*

Proposition 1.9 (Downwards closure) *If $M \models_X \phi$ and $Y \subseteq X$ then $M \models_Y \psi$.*

Proposition 1.10 (Locality) *If $M \models_X \phi$ and $X(\mathrm{Free}(\phi)) = Y(\mathrm{Free}(\phi))$ then $M \models_Y \phi$.*

Proposition 1.11 (From dependence logic to Σ^1_1) *Let $\phi(\vec{v})$ be a dependence logic formula with free variables in $\vec{v}$. Then there exists a Σ^1_1 sentence $\Phi(R)$ such that*

$$M \models_X \phi \Leftrightarrow M \models \Phi(X(\vec{v}))$$

for all suitable models M and for all nonempty teams X. Furthermore, in $\Phi(R)$ the symbol R occurs only negatively.

As proved in Kontinen and Väänänen (2009), there is also a converse for the last proposition:

Theorem 1.12 (From Σ^1_1 to dependence logic) *Let $\Phi(R)$ be a Σ^1_1 sentence in which R occurs only negatively. Then there exists a dependence logic formula $\phi(\vec{v})$, where $|\vec{v}|$ is the arity of R, such that*

$$M \models_X \phi \Leftrightarrow M \models \Phi(X(\vec{v}))$$

for all suitable models M and for all nonempty teams X whose domain contains $\vec{v}$.

---

[1] In general, we will assume through this whole work that all first-order models which we are considering have at least two elements. As one-element models are trivial, this is not a very onerous restriction.

Because of this correspondence between dependence logic and existential second order logic, it is easy to see that dependence logic is closed under existential quantification: for all dependence logic formulas $\phi(\vec{v}, P)$ over the signature $\Sigma \cup \{P\}$ there exists a dependence logic formula $\exists P \phi(\vec{v}, P)$ over the signature Σ such that

$$M \models_X \exists P \phi(\vec{v}, P) \Leftrightarrow \exists P \text{ s.t. } M \models_X \phi(\vec{v}, P)$$

for all models M with domain Σ and for all teams X over the free variables of ϕ. Therefore, in the rest of this work we will use $\exists P \phi(\vec{v}, P)$ as a shorthand for the corresponding dependence logic expression.

1.2 Dynamic game logic

Game logics are logical formalisms for reasoning about games and their properties in a very general setting. Whereas the game theoretic semantics approach attempts to use game-theoretic techniques to *interpret* logical systems, game logics attempt to put logic to the service of game theory, by providing a high-level language for the study of games.

They generally contain two different kinds of expressions:

1. *Game terms*, which are descriptions of games in terms of compositions of certain primitive *atomic games*, whose interpretation is presumed fixed for any given game model;
2. *Formulas*, which, in general, correspond to assertions about the abilities of players in games.

In this subsection, we are going to summarize the definition of a variant of dynamic game logic (Parikh 1985).[2] Then, in the next subsection, we will discuss a remarkable connection between first-order logic and dynamic game logic discovered by van Benthem (2003).

One of the fundamental semantic concepts of dynamic game logic is the notion of *forcing relation:*

Definition 1.13 (*Forcing relation*) Let S be a nonempty set of *states*. A *forcing relation* over S is a set $\rho \subseteq S \times \texttt{Parts}(S)$, where $\texttt{Parts}(S)$ is the powerset of S.

In brief, a forcing relation specifies the abilities of a player in a perfect-information game: $(s, X) \in \rho$ if and only if the player has a strategy that guarantees that, whenever the initial position of the game is s, the terminal position of the game will be in X.

A (two-player) *game* is then defined as a pair of forcing relations satisfying some axioms:

Definition 1.14 (*Game*) Let S be a nonempty set of states. A *game* over S is a pair (ρ^E, ρ^A) of forcing relations over S satisfying the following conditions for all $i \in \{E, A\}$, all $s \in S$ and all $X, Y \subseteq S$:

---

[2] The main difference between this version and the one of Parikh's original paper lies in the absence of the iteration operator γ^* from our formalism. In this, we follow van Benthem (2003) and van Benthem et al. (2008).

Monotonicity: If $(s, X) \in \rho^i$ and $X \subseteq Y$ then $(s, Y) \in \rho^i$;
Consistency: If $(s, X) \in \rho^E$ and $(s, Y) \in \rho^A$ then $X \cap Y \neq \emptyset$;
Non-triviality: $(s, \emptyset) \notin \rho^i$.
Determinacy: If $(s, X) \notin \rho^i$ then $(s, S \backslash X) \in \rho^j$, where $j \in \{E, A\} \backslash \{i\}$.[3]

Definition 1.15 (*Game Model*) Let S be a nonempty set of states, let Φ be a nonempty set of *atomic propositions* and let Γ be a nonempty set of *atomic game symbols*. Then a *game model* over S, Φ and Γ is a triple $(S, \{(\rho_g^E, \rho_g^A) : g \in \Gamma\}, V)$, where (ρ_g^E, ρ_g^A) is a game over S for all $g \in \Gamma$ and where V is a valuation function associating each $p \in \Phi$ to a subset $V(p) \subseteq S$.

The language of dynamic game logic, as we already mentioned, consists of *game terms*, built up from atomic games, and of *formulas*, built up from atomic proposition. The connection between these two parts of the language is given by the *test* operation $\phi?$, which turns any formula ϕ into a test game, and the *diamond* operation, which combines a game term γ and a formula ϕ into a new formula $\langle \gamma, i \rangle \phi$ which asserts that agent i can guarantee that the game γ will end in a state satisfying ϕ.

Definition 1.16 (*Dynamic game logic—Syntax*) Let Φ be a nonempty set of *atomic propositions* and let Γ be a nonempty set of *atomic game formulas*. Then the sets of all game terms γ and formulas ϕ are defined as

$$\gamma ::= g \mid \phi? \mid \gamma; \gamma \mid \gamma \cup \gamma \mid \gamma^d$$
$$\phi ::= \bot \mid p \mid \neg\phi \mid \phi \vee \phi \mid \langle \gamma, i \rangle \phi$$

for p ranging over Φ, g ranging over Γ, and i ranging over $\{E, A\}$.

We already mentioned the intended interpretations of the test connective $\phi?$ and of the diamond connective $\langle \gamma, i \rangle \phi$. The interpretations of the other game connectives should be clear: γ^d is obtained by swapping the roles of the players in γ, $\gamma_1 \cup \gamma_2$ is a game in which the existential player E chooses whether to play γ_1 or γ_2, and $\gamma_1; \gamma_2$ is the *concatenation* of the two games corresponding to γ_1 and γ_2 respectively.

Definition 1.17 (*Dynamic game logic—semantics*) Let $G = (S, \{(\rho_g^E, \rho_g^A) : g \in \Gamma\}, V)$ be a game model over S, Γ and Φ. Then for all game terms γ and all formulas ϕ of dynamic game logic over Γ and Φ we define a game $\|\gamma\|_G$ and a set $\|\phi\|_G \subseteq S$ as follows:
DGL-atomic-game: For all $g \in \Gamma$, $\|g\|_G = (\rho_g^E, \rho_g^A)$;
DGL-test: For all formulas ϕ, $\|\phi?\|_G = (\rho^E, \rho^A)$, where

– $s\rho^E X$ iff $s \in \|\phi\|_G$ and $s \in X$;

– $s\rho^A X$ iff $s \notin \|\phi\|_G$ or $s \in X$

---

[3] This requirement is nothing but a formal version of Zermelo's Theorem: if one of the players cannot force the outcome of the game to belong to a set of "winning outcomes" X, this implies that the other player can force it to belong to the complement of X.

for all $s \in S$ and all X with $\emptyset \neq X \subseteq S$;

DGL-concat: For all game terms γ_1 and γ_2, $\|\gamma_1; \gamma_2\|_G = (\rho^E, \rho^A)$, where, for all $i \in \{E, A\}$ and for $\|\gamma_1\|_G = (\rho_1^E, \rho_1^A)$, $\|\gamma_2\|_G = (\rho_2^E, \rho_2^A)$,

- $s\rho^i X$ if and only if there exists a Z such that $s\rho_1^i Z$ and for each $z \in Z$ there exists a set X_z satisfying $z\rho_2^i X_z$ such that

$$X = \bigcup_{z \in Z} X_z;$$

DGL-$\cup$: For all game terms γ_1 and γ_2, $\|\gamma_1 \cup \gamma_2\|_G = (\rho^E, \rho^A)$, where

- $s\rho^E X$ if and only if $s\rho_1^E X$ or $s\rho_2^E X$, and
- $s\rho^A X$ if and only if $s\rho_1^A X$ and $s\rho_2^A X$

where, as before, $\|\gamma_1\|_G = (\rho_1^E, \rho_1^A)$ and $\|\gamma_2\|_G = (\rho_2^E, \rho_2^A)$;[4]

   DGL-dual: If $\|\gamma\|_G = (\rho^E, \rho^A)$ then $\|\gamma^d\|_G = (\rho^A, \rho^E)$;

   DGL-$\perp$: $\|\perp\|_G = \emptyset$;

   DGL-atomic-pr: $\|p\|_G = V(p)$;

   DGL-$\neg$: $\|\neg\phi\|_G = S \setminus \|\phi\|_G$;

   DGL-$\vee$: $\|\phi_1 \vee \phi_2\|_G = \|\phi_1\|_G \cup \|\phi_2\|_G$;

   DGL-$\diamond$: If $\|\gamma\|_G = (\rho^E, \rho^A)$ then for all ϕ,

$$\|\langle\gamma, i\rangle\phi\|_G = \{s \in S : \exists X_s \subseteq \|\phi\|_G \text{ s.t. } s\rho^i X_s\}.$$

If $s \in \|\phi\|_G$, we say that ϕ is *satisfied* by s in G and we write $M \models_s \phi$.

We will not discuss here the properties of this logic, or the vast amount of variants and extensions of it which have been developed and studied. It is worth pointing out, however, that van Benthem et al. (2008) introduced a *Concurrent Dynamic Game Logic* that can be considered one of the main sources of inspiration for the Transition Logic that we will develop in Subsect. 3.2.

1.3 Dynamic game logic and first order logic

In this subsection, we will briefly recall a remarkable result from van Benthem (2003) which establishes a connection between dynamic game logic and first-order logic.

In brief, as the following two theorems demonstrate, either of these logics can be seen as a special case of the other, in the sense that models and formulas of the one can be uniformly translated into models of the other in a way which preserves satisfiability and truth:

---

[4] van Benthem et al. (2008) gives the following alternative condition for the powers of the universal player:

- $s\rho^A X$ if and only if $X = Z_1 \cup Z_2$ for two Z_1 and Z_2 such that $s\rho_1^A Z_1$ and $s\rho_2^A Z_2$.

It is trivial to see that, if our games satisfy the monotonicity condition, this rules is equivalent to the one we presented.

Theorem 1.18 *Let $G = (S, \{(\rho_g^E, \rho_g^A) : g \in \Gamma\}, V)$ be any game model, let ϕ be any game formula for the same language, and let $s \in S$. Then it is possible to uniformly construct a first-order model G^{FO}, a first-order formula ϕ^{FO} and an assignment s^{FO} of G^{FO} such that*

$$G \models_s \phi \Leftrightarrow G^{FO} \models_{s^{FO}} \phi^{FO}.$$

Theorem 1.19 *Let M be any first order model, let ϕ be any first-order formula for the signature of M, and let s be an assignment of M. Then it is possible to uniformly construct a game model G^{DGL}, a game formula ϕ^{DGL} and a state s^{DGL} such that*

$$M \models_s \phi \Leftrightarrow G^{DGL} \models_{s^{DGL}} \phi^{DGL}.$$

We will not discuss here the proofs of these two results. Their *significance*, however, is something about which is necessary to spend a few words. In brief, what this back-and-forth representation between first order logic and dynamic game logic tells us is that it is possible to understand first order logic as a *logic for reasoning about determined games*!

In the next sections, we will attempt to develop a similar result for the case of dependence logic.

2 Transition logic

2.1 A logic for imperfect information games against nature

We will now define a variant of dynamic game logic, which we will call *Transition Logic*. It deviates from the basic framework of dynamic game logic in two fundamental ways:

1. It considers *one-player* games against Nature, instead of *two-player games* as is usual in dynamic game logic;
2. It allows for *uncertainty* about the initial position of the game.

Hence, transition logic can be seen as a *decision-theoretic logic*, rather than a *game-theoretic* one: transition logic formulas, as we will see, correspond to assertions about the abilities of a single agent acting under uncertainty, instead of assertions about the abilities of agents interacting with each other.

In principle, it is certainly possible to generalize the approach discussed here to multiple agents acting in situations of imperfect information, and doing so might cause interesting phenomena to surface; but for the time being, we will content ourselves with developing this formalism and discussing its connection with Dependence Logic.

Our first definition is a fairly straightforward generalization of the concept of forcing relation:

Definition 2.1 (*Transition system*) Let S be a nonempty set of *states*. A *transition system* over S is a nonempty relation $\theta \subseteq \texttt{Parts}(S) \times \texttt{Parts}(S)$ satisfying the following requirements:

Downwards closure: If $(X, Y) \in \theta$ and $X' \subseteq X$ then $(X', Y) \in \theta$;
Monotonicity: If $(X, Y) \in \theta$ and $Y \subseteq Y'$ then $(X, Y') \in \theta$;
Non-creation: $(\emptyset, Y) \in \theta$ for all $Y \subseteq S$;
Non-triviality: If $X \neq \emptyset$ then $(X, \emptyset) \notin \theta$.

Informally speaking, a transition system specifies the abilities of an agent: for all $X, Y \subseteq S$ such that $(X, Y) \in \theta$, the agent has a strategy which guarantees that the output of the transition will be in Y whenever the input of the transition is in X.

The four axioms which we gave capture precisely this intended meaning, as we will see:

Definition 2.2 (*Decision game*) A *decision game* is a triple $\Gamma = (S, E, O)$, where S is a nonempty set of *states*, E is a nonempty set of possible *decisions* for our agent and O is an *outcome function* from $S \times E$ to $\mathtt{Parts}(S)$.

If $s' \in O(s, e)$, we say that s' is a *possible outcome* of s under e; if $O(s, e) = \emptyset$, we say that *e fails on input s*.

Definition 2.3 (*Abilities in a decision game*) Let $\Gamma = (S, E, O)$ be a decision game, and let $X, Y \subseteq S$. Then we say that Γ *allows* the transition $X \to Y$, and we write $\Gamma : X \to Y$, if and only if there exists a $e \in E$ such that $\emptyset \neq O(s, e) \subseteq Y$ for all $s \in X$ (that is, if and only if our agent can make a decision which guarantees that the outcome will be in Y whenever the input is in X).

Theorem 2.4 (*Transition systems and abilities*) *A set $\theta \subseteq \mathtt{Parts}(S) \times \mathtt{Parts}(S)$ is a transition system if and only if there exists a decision game $\Gamma = (S, E, O)$ such that*

$$(X, Y) \in \theta \Leftrightarrow \Gamma : X \to Y.$$

Proof Let $\theta \subseteq \mathtt{Parts}(S) \times \mathtt{Parts}(S)$ be any transition system, let us enumerate its elements $\{(X_i, Y_i) : i \in I\}$, and let us consider the game $\Gamma = (S, I, O)$, where

$$O(s, i) = \begin{cases} Y_i & \text{if } s \in X_i; \\ \emptyset & \text{otherwise.} \end{cases}$$

Suppose that $(X, Y) \in \theta$. If $X = \emptyset$, then $\Gamma : X \to Y$ follows at once by definition. If instead $X \neq \emptyset$, by *non-triviality* we have that Y is nonempty too, and furthermore $(X, Y) = (X_i, Y_i)$ for some $i \in I$. Then $O(s, i) = Y_i \neq \emptyset$ for all $s \in X_i$, as required.

Now suppose that $\Gamma : X \to Y$. Then there exists a $i \in I$ such that $\emptyset \neq O(s, i) \subseteq Y$ for all $s \in X$. If $X \neq \emptyset$, this implies that $X \subseteq X_i$ and $Y_i \subseteq Y$. Hence, by *monotonicity* and *downwards closure*, $(X, Y) \in \theta$, as required. If instead $X = \emptyset$, then by *non-creation* we have again that $(X, Y) \in \theta$.

Conversely, consider a decision game $\Gamma = (S, E, O)$. Then the set of its abilities satisfies our four axioms:

Downwards closure: Suppose that $\Gamma : X \to Y$ and that $X' \subseteq X$. By definition, there exists a $e \in E$ such that $\emptyset \neq O(s, e) \subseteq Y$ for all $s \in X$. But then the same holds for all $s \in X'$, and hence $\Gamma : X' \to Y$.

Monotonicity: Suppose that $\Gamma : X \to Y$ and that $Y \subseteq Y'$. By definition, there exists a $e \in E$ such that $\emptyset \neq O(s, e) \subseteq Y$ for all $s \in X$. But then, for all such s, $O(s, e) \subseteq Y'$ too, and hence $\Gamma : X \to Y'$.

Non-creation: Let $Y \subseteq S$ and let $e \in E$ be any possible decision. Then trivially $\emptyset \neq O(s, e) \subseteq Y$ for all $s \in \emptyset$, and hence $\Gamma : \emptyset \to Y$.

Non-triviality: Let $s_0 \in X$, and suppose that $\Gamma : X \to Y$. Then there exists a e such that $\emptyset \neq O(s, e) \subseteq Y$ for all $s \in X$, and hence in particular $\emptyset \neq O(s_0, e) \subseteq Y$. Therefore, Y is nonempty.

$\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \square$

What this theorem tells us is that our notion of transition system is the correct one: it captures precisely the abilities of an agent making choices under imperfect information and attempting to guarantee that, if the initial state is in a set X, the outcome will be in a set Y.

Definition 2.5 (*Trump*) Let S be a nonempty set of states. A *trump* over S is a nonempty, downwards closed family of subsets of S.

Whereas a transition system describes the abilities of an agent to transition from a set of possible initial states to a set of possible terminal states, a trump describes the agent's abilities to reach *some* terminal state from a set of possible initial states:[5]

Proposition 2.6 *Let θ be a transition system and let $Y \subseteq S \neq \emptyset$. Then* $\mathrm{reach}(\theta, Y) = \{X \mid (X, Y) \in \theta\}$ *forms a trump. Conversely, for any trump $\mathcal{X}$ over S there exists a transition system θ such that $\mathcal{X} = \mathrm{reach}(\theta, Y)$ for any nonempty $Y \subseteq S$.*

Proof Let θ be a transition system. Then if $(X, Y) \in \theta$ and $X' \subseteq X$, by downwards closure we have at once that $(X', Y) \in \theta$. Furthermore, $(\emptyset, Y) \in \theta$ for any Y. Hence, $\mathrm{reach}(\theta, Y)$ is a trump, as required.

Conversely, let $\mathcal{X} \subseteq \mathrm{Parts}(\mathrm{Parts}(S))$ be a trump, and let us enumerate its elements as $\{X_i : i \in I\}$. Then define θ as

$$\theta = \{(A, B) : \emptyset \neq B \subseteq S, \exists i \in I \text{ s.t. } A \subseteq X_i\} \cup \{(\emptyset, \emptyset)\}$$

It is easy to see that θ is a transition system; and by construction, for $Y \neq \emptyset$ we have that $(A, Y) \in \theta \Leftrightarrow \exists i$ s.t. $A \subseteq X_i \Leftrightarrow A \in \mathcal{X}$, where we used the fact that $\mathcal{X}$ is downwards closed. $\qquad\square$

We can now define the syntax and semantics of transition logic:

Definition 2.7 (*Transition model*) Let Φ be a set of *atomic propositional symbols* and let Θ be a set of *atomic transition symbols*. Then a *transition model* is a tuple $T = (S, \{\theta_t : t \in \Theta\}, V)$, where S is a nonempty set of states, θ_t is a transition system over S for any $t \in \Theta$, and V is a function sending each $p \in \Phi$ into a trump of S.

---

[5] The term "trump" is taken from Hodges (1997), where it is used to describe the set of all teams which satisfy a given formula.

Definition 2.8 (*Transition logic—syntax*) Let Φ be a set of atomic propositions and let Θ be a set of atomic transitions. Then the *transition terms* and *formulas* of our language are defined respectively as

$$\tau ::= t \mid \phi? \mid \tau \otimes \tau \mid \tau \cap \tau \mid \tau; \tau$$
$$\phi ::= \top \mid p \mid \phi \vee \phi \mid \phi \wedge \phi \mid \langle \tau \rangle \phi$$

where t ranges over Θ and p ranges over Φ.

Definition 2.9 (*Transition logic—semantics*) Let $T = (S, \{\theta_t : t \in \Theta\}, V)$ be a transition model, let τ be a transition term, and let $X, Y \subseteq S$. Then we say that τ *allows* the transition from X to Y, and we write $T \models_{X \to Y} \tau$, if and only if

TL-atomic-tr: $\tau = t$ for some $t \in \Theta$ and $(X, Y) \in \theta_t$;

TL-test: $\tau = \phi?$ for some transition formula ϕ such that $T \models_X \phi$ in the sense described later in this definition, and $X \subseteq Y$;

TL-$\otimes$: $\tau = \tau_1 \otimes \tau_2$, and $X = X_1 \cup X_2$ for two X_1 and X_2 such that $T \models_{X_1 \to Y} \tau_1$ and $T \models_{X_2 \to Y} \tau_2$;

TL-$\cap$: $\tau = \tau_1 \cap \tau_2$, $T \models_{X \to Y} \tau_1$ and $T \models_{X \to Y} \tau_2$;

TL-concat: $\tau = \tau_1; \tau_2$ and there exists a $Z \subseteq S$ such that $T \models_{X \to Z} \tau_1$ and $T \models_{Z \to Y} \tau_2$.

Analogously, let ϕ be a transition formula, and let $X \subseteq S$. Then we say that X *satisfies* ϕ, and we write $T \models_X \phi$, if and only if

TL-$\top$: $\phi = \top$;

TL-atomic-pr: $\phi = p$ for some $p \in \Phi$ and $X \in V(p)$;

TL-$\vee$: $\phi = \psi_1 \vee \psi_2$ and $T \models_X \psi_1$ or $T \models_X \psi_2$;

TL-$\wedge$: $\phi = \psi_1 \wedge \psi_2$, $T \models_X \psi_1$ and $T \models_X \psi_2$;

TL-$\diamond$: $\phi = \langle \tau \rangle \psi$ and there exists a Y such that $T \models_{X \to Y} \tau$ and $T \models_Y \psi$.

Proposition 2.10 *For any transition model T, transition term τ and transition formula ϕ, the set*

$$\|\tau\|_T = \{(X, Y) : T \models_{X \to Y} \tau\}$$

is a transition system and the set

$$\|\phi\|_T = \{X : T \models_X \phi\}$$

is a trump.

Proof By induction.   □

We end this subsection with a few simple observations about this logic.

First of all, we did not take the negation as one of the primitive connectives. Indeed, transition logic, much like dependence logic, has an intrinsically *existential* character: it can be used to reason about which sets of possible states an agent *may* reach, but not

to reason about which ones such an agent *must* reach. There is of course no reason, in principle, why a negation could not be added to the language, just as there is no reason why a negation cannot be added to Dependence Logic, thus obtaining the far more powerful *Team Logic* (Väänänen 2007b; Kontinen and Nurmi 2009): however, this possible extension will not be studied in this work.

The connectives of transition logic are, for the most part, very similar to those of dynamic game logic, and their interpretation should pose no difficulties. The exception is the *tensor operator* $\tau_1 \otimes \tau_2$, which substitutes the game union operator $\gamma_1 \cup \gamma_2$ and which, while sharing roughly the same informal meaning, behaves in a very different way from the semantic point of view (for example, it is not in general idempotent!).

The decision game corresponding to $\tau_1 \otimes \tau_2$ can be described as follows: first the agent chooses an index $i \in \{1, 2\}$, then he or she picks a strategy for τ_i and plays accordingly. However, the choice of i may be a function of the initial state: hence, the agent can guarantee that the output state will be in Y whenever the input state is in X only if he or she can split X into two subsets X_1 and X_2 and guarantee that the state in Y will be reached from any state in X_1 when τ_1 is played, and from any state in X_2 when τ_2 is played.

It is also of course possible to introduce a "true" choice operator $\tau_1 \cup \tau_2$, with semantical condition

TL-$\cup$: $T \models_{X \to Y} \tau_1 \cup \tau_2$ iff $T \models_{X \to Y} \tau_1$ or $T \models_{X \to Y} \tau_2$;

but we will not explore this possibility any further in this work, nor we will consider any other possible connectives such as, for example, the iteration operator

TL-$*$: $T \models_{X \to Y} \tau^*$ iff there exist $n \in \mathbb{N}$ and $Z_0 \ldots Z_n$ such that $Z_0 = X$, $Z_n = Y$ and $T \models_{Z_i \to Z_{i+1}} \tau$ for all $i \in 1 \ldots n - 1$.

2.2 Transition logic and dependence logic

This subsection contains the central result of this work, that is, the analogues of Theorems 1.18 and 1.19 for dependence logic and transition logic.

Representing dependence logic models and formulas in transition logic is fairly simple:

Definition 2.11 (M^{TL}) Let M be a first-order model. Then M^{TL} is the transition model (S, Θ, V) such that

- S is the set of all teams over M;
- The set of all atomic transition symbols is $\{\exists v, \forall v : v \in \mathtt{Var}\}$, and hence Θ is $\{\theta_{\exists v}, \theta_{\forall v} : v \in \mathtt{Var}\}$;
- For any variable v, $\theta_{\exists v} = \{(X, Y) : \exists F \text{ s.t. } X[F/v] \subseteq Y\}$ and $\theta_{\forall v} = \{(X, Y) : X[M/v] \subseteq Y\}$;
- For any first-order literal or dependence atom α, $V(\alpha) = \{X : M \models_X \phi\}$.

Definition 2.12 (ϕ^{TL}) Let ϕ be a dependence logic formula. Then ϕ^{TL} is the transition term defined as follows:

1. If ϕ is a literal or a dependence atom, $\phi^{TL} = \phi?$;
2. If $\phi = \psi_1 \vee \psi_2$, $\phi^{TL} = (\psi_1)^{TL} \otimes (\psi_2)^{TL}$;
3. If $\phi = \psi_1 \wedge \psi_2$, $\phi^{TL} = (\psi_1)^{TL} \wedge (\psi_2)^{TL}$;
4. If $\phi = \exists v \psi$, $\phi^{TL} = \exists v; (\psi)^{TL}$;
5. If $\phi = \forall v \psi$, $\phi^{TL} = \forall v; (\psi)^{TL}$.

Theorem 2.13 *For all first-order models M, teams X and formulas ϕ, the following are equivalent:*

- $M \models_X \phi$;
- $\exists Y$ *s.t.* $M^{TL} \models_{X \to Y} \phi^{TL}$;
- $M^{TL} \models_X \langle \phi^{TL} \rangle \top$;
- $M^{TL} \models_{X \to S} \phi^{TL}$.

Proof We show, by structural induction on ϕ, that the first condition is equivalent to the last one. The equivalences between the last one and the second and third ones are then trivial.

1. If ϕ is a literal or a dependence atom, $M^{TL} \models_{X \to S} \phi?$ if and only if $X \in V(\phi)$, that is, if and only if $M \models_X \phi$;
2. $M^{TL} \models_{X \to S} (\psi_1)^{TL} \otimes (\psi_2)^{TL}$ if and only if $X = X_1 \cup X_2$ for two $X_1, X_2 \subseteq S$ such that $M^{TL} \models_{X_1 \to S} (\psi_1)^{TL}$ and $M^{TL} \models_{X_2 \to S} (\psi_2)^{TL}$. By induction hypothesis, this can be the case if and only if $M \models_{X_1} \psi_1$ and $M \models_{X_2} \psi_2$, that is, if and only if $M \models_X \psi_1 \vee \psi_2$.
3. $M^{TL} \models_{X \to S} (\psi_1)^{TL} \wedge (\psi_2)^{TL}$ if and only if $M^{TL} \models_{X \to S} (\psi_1)^{TL}$ and $M^{TL} \models_{X \to S} (\psi_2)^{TL}$, that is, by induction hypothesis, if and only if $M \models_X \psi_1 \wedge \psi_2$.
4. $M^{TL} \models_{X \to S} \exists v; (\psi)^{TL}$ if and only if there exists a Y such that $Y \supseteq X[F/v]$ for some F and $M^{TL} \models_{Y \to S} \psi$. By induction hypothesis and downwards closure, this can be the case if and only if $M \models_{X[F/v]} \psi$ for some F, that is, if and only if $M \models_X \exists v \psi$;
5. $M^{TL} \models_{X \to S} \forall v; (\psi)^{TL}$ if and only if $M^{TL} \models_{Y \to S} (\psi)^{TL}$ for some $Y \supseteq X[M/v]$, that is, if and only if $M \models_{X[M/v]} \psi$, that is, if and only if $M \models_X \forall v \psi$. $\square$

One interesting aspect of this representation result is that dependence logic *formulas* correspond to transition logic *transitions*, not to transition logic *formulas*. This can be thought of as one first hint of the fact that Dependence Logic can be thought of as a logic of transitions: and in the later sections, we will explore this idea more in depth.

Representing transition models, game terms and formulas in dependence logic is somewhat more complex:

Definition 2.14 (T^{DL}) Let $T = (S, (\theta_t : t \in \Theta), V)$ be a transition model. Furthermore, for any $t \in \Theta$, let $\theta_t = \{(X_i, Y_i) : i \in I_t\}$, and, for any $p \in \Phi$, let $V(p) = \{X_j : j \in J_p\}$. Then T^{DL} is the first-order model with domain[6] $S \uplus \biguplus \{I_t : t \in \Theta\} \uplus \biguplus \{J_p : p \in \Phi\}$ whose signature contains

- For every $t \in \Theta$, a ternary relation R_t whose interpretation is $\{(i, x, y) : i \in I_t, x \in X_i, y \in Y_i\}$;

---

[6] Here we write $A \uplus B$ for the *disjoint union* of the sets A and B.

– For every $p \in \Phi$, a binary relation V_p whose interpretation is $\{(j, x) : j \in J_p, x \in X_j\}$.

Definition 2.15 (ϕ_x^{DL} *and* τ_x^{DL}) For any transition formula ϕ and variable x, the dependence logic formula ϕ_x^{DL} is defined as

1. $\top_x^{DL}$ is $\top$;
2. For all $p \in \Phi$, p_x^{DL} is $\exists j (=(j) \land V_p(j, x))$;
3. $(\psi_1 \lor \psi_2)_x^{DL}$ is $(\psi_1)_x^{DL} \sqcup (\psi_2)^{DL}$, where $\sqcup$ is the classical disjunction introduced in Definition 1.6;
4. $(\psi_1 \land \psi_2)_x^{DL}$ is $(\psi_1)_x^{DL} \land (\psi_2)_x^{DL}$;
5. $(\langle \tau \rangle \psi)_x^{DL}$ is $\exists P ((\tau)_x^{DL}(P) \land \forall y (\neg Py \lor (\psi)_y^{DL}))$,

where for any transition term τ, variable x and unary relation symbol P, $\tau_x^{DL}(P)$ is defined as

6. For all $t \in \Theta$, $t_x^{DL}(P)$ is $\exists i (=(i) \land \exists y (R_t(i, x, y)) \land \forall y (\neg R_t(i, x, y) \lor Py))$;
7. For all formulas ϕ, $(\phi?)_x^{DL}(P)$ is $\phi_x^{DL} \land Px$;
8. $(\tau_1 \otimes \tau_2)_x^{DL}(P) = (\tau_1)_x^{DL}(P) \lor (\tau_2)_x^{DL}(P)$;
9. $(\tau_1 \cap \tau_2)_x^{DL}(P) = (\tau_1)_x^{DL}(P) \land (\tau_2)_x^{DL}(P)$;
10. $(\tau_1; \tau_2)_x^{DL}(P) = \exists Q ((\tau_1)_x^{DL}(Q) \land \forall y (\neg Qy \lor (\tau_2)_y^{DL}(P)))$ for a new and unused variable y.

Theorem 2.16 *For all transition models $T = (S, (\theta_t : t \in \Theta), V)$, transition terms τ, transition formulas ϕ, variables x, sets $P \subseteq S$ and teams X over T^{DL} with $X(x) \subseteq S$,*[7]

$$T^{DL} \models_X \phi_x^{DL} \Leftrightarrow T \models_{X(x)} \phi$$

and

$$T^{DL} \models_X \tau_x^{DL}(P) \Leftrightarrow T \models_{X(x) \to P} \tau.$$

Proof The proof is by structural induction on terms and formulas.

Let us first consider the cases corresponding to formulas:

1. For all teams X, $T^{DL} \models_X \top$ and $T \models_{X(x)} \top$, as required;
2. Suppose that $T^{DL} \models_X \exists j (=(j) \land V_p(j, x))$. Then there exists a $m \in \text{Dom}(T^{DL})$ such that $T^{DL} \models_{X[m/j]} V_p(j, x)$. Hence, we have that $X(x) \subseteq X_m \in V(p)$; and, by downwards closure, this implies that $X(x) \in V(p)$, and hence that $T \models_{X(x)} p$ as required.
   Conversely, suppose that $T \models_{X(x)} p$. Then $X(x) \in V(p)$, and hence $X(x) = X_m$ for some $m \in J_p$. Then we have by definition that $T^{DL} \models_{X[m/j]} V_p(j, x)$, and finally that $T^{DL} \models_X T_x(p)$.
3. By Proposition 1.7, $T^{DL} \models_X (\psi_1 \lor \psi_2)_x^{DL}$ if and only if $T^{DL} \models_X (\psi_1)_x^{DL}$ or $T^{DL} \models_X (\psi_2)_x^{DL}$. By induction hypothesis, this is the case if and only if $T \models_{X(x)} \psi_1$ or $T \models_{X(x)} \psi_2$, that is, if and only if $T \models_{X(x)} \psi_1 \lor \psi_2$.

---

[7] That is, such that $X(x)$ is a set of states of the transition model.

4. $T^{DL} \models_X (\psi_1 \wedge \psi_2)^{DL}_x$ if and only if $T^{DL} \models_X (\psi_1)^{DL}_x$ and $T^{DL} \models_x (\psi_2)^{DL}_x$, that is, by induction hypothesis, if and only if $T \models_X \psi_1 \wedge \psi_2$.

5. $T^{DL} \models_X (\langle \tau \rangle \psi)^{DL}_x$ if and only if there exists a P such that $T^{DL} \models_X (\tau)^{DL}_x(P)$ and $T^{DL} \models_{X[T^{DL}/y]} \neg Py \vee (\psi)^{DL}_y$. By induction hypothesis, the first condition holds if and only if $T \models_{X(x) \to P} \tau$. As for the second one, it holds if and only if $X[T^{DL}/y] = Y_1 \cup Y_2$ for two Y_1, Y_2 such that $T^{DL} \models_{Y_1} \neg Py$ and $T^{DL} \models_{Y_2} \tau_y(\psi)$. But then we must have that $T \models_{Y_2(y)} \psi$ and that $P \subseteq Y_2(y)$; therefore, by downwards closure, $T \models_P \psi$ and finally $T \models_{X(x)} \langle \tau \rangle \psi$.

   Conversely, suppose that there exists a P such that $T \models_{X(x) \to P} \tau$ and $T \models_P \psi$; then by induction hypothesis we have that $T^{DL} \models_X (\tau)^{DL}_x(P)$ and that $T^{DL} \models_{X[T^{DL}/y]} \neg Py \vee (\psi)^{DL}_x$, and hence $T^{DL} \models_X (\langle \tau \rangle \psi)^{DL}_x$.

Now let us consider the cases corresponding to transition terms:

6. Suppose that $T^{DL} \models_X \exists i (= (i) \wedge \exists y (R_t(i, x, y)) \wedge \forall y (\neg R_t(i, x, y) \vee Py))$. If $X = \emptyset$ then $X(x) = \emptyset$, and hence by *non-creation* we have that $(X(x), P) = (\emptyset, P) \in \theta_t$, as required.

   Let us assume instead that $X \neq \emptyset$. Then, by hypothesis, there exists a $m \in \mathrm{Dom}(T^{DL})$ such that
   - There exists a F such that $T^{DL} \models_{X[m/i][F/y]} R_t(i, x, y)$;
   - $T^{DL} \models_{X[m/i][T^{DL}/y]} \neg R_t(i, x, y) \vee Py$.

   From the first condition it follows that for every $p \in X(x)$ there exists a q such that $R_t(m, p, q)$: therefore, by the definition of R_t, every such p must be in X_m. From the second condition it follows that whenever $R_t(m, p, q)$ and $p \in X(x) \subseteq X_m$, $q \in P$; and, since $X(x) \neq \emptyset$, this implies that $Y_m \subseteq P$ by the definition of R_t.

   Hence, by *monotonicity* and *downwards closure*, we have that $(X(x), P) \in \theta_t$ and that $T \models_{X(x) \to P} t$, as required.

   Conversely, suppose that $(X(x), P) = (X_m, Y_m) \in \theta_t$ for some $m \in I_t$. If $X(x) = \emptyset$ then $X = \emptyset$, and hence by Proposition 1.8 we have that $T^{DL} \models_X t^{DL}_x(P)$, as required. Otherwise, by *non-triviality*, $P = Y_m \neq \emptyset$. Let now $p \in P$ be any of its elements and let $F(s) = p$ for all $p \in X[m/i]$: then $M \models_{X[m/i][F/y]} R_t(i, x, y)$, as any assignment of this team sends x to some element of X_m and y to $p \in Y_m$. Furthermore, let $s \in X(x) = X_m$, and let q be such that $R_t(m, s(x), q)$: then $q \in Y_m = P$, and hence $M \models_{X[m/i][T^{DL}/y]} \neg R_t(i, x, y) \vee Py$. So, in conclusion, $M \models_X t^{DL}_x(P)$, as required.

7. $T^{DL} \models_X \phi^{DL}_x \wedge Px$ if and only if $T \models_{X(x)} \phi$ and $X(x) \subseteq P$, that is, if and only if $T \models_{X(x) \to P} \phi?$.

8. $T^{DL} \models_X (\tau_1)^{DL}_x(P) \vee (\tau_2)^{DL}_x(P)$ if and only if $X = X_1 \cup X_2$ for two X_1, X_2 such that
   - $X = X_1 \cup X_2$, and therefore $X(x) = X_1(x) \cup X_2(x)$;
   - $T^{DL} \models_{X_1} (\tau_1)^{DL}_x(P)$, that is, by induction hypothesis, $T \models_{X_1(x) \to P} \tau_1$;
   - $T^{DL} \models_{X_2} (\tau_2)^{DL}_x(P)$, that is, by induction hypothesis, $T \models_{X_2(x) \to P} \tau_2$;

   Hence, if $T^{DL} \models_X (\tau_1 \otimes \tau_2)^{DL}_x(P)$ then $T \models_{X(x) \to P} \tau_1 \otimes \tau_2$.

Conversely, if $X(x) = A \cup B$ for two A, B such that $T \models_{A \to P} \tau_1$ and $T \models_{B \to P} \tau_2$, let

$$X_1 = \{s \in X : s(x) \in A\}$$
$$X_2 = \{s \in X : s(x) \in B\}.$$

Clearly $X = X_1 \cup X_2$, and furthermore by induction hypothesis $T^{DL} \models_{X_1} (\tau_1)_x^{DL}(P)$ and $T^{DL} \models_{X_2} (\tau_2)_x^{DL}(P)$. Hence, $T^{DL} \models_X (\tau_1 \otimes \tau_2)_x^{DL}(P)$, as required.

9. $T^{DL} \models_X (\tau_1 \cap \tau_2)_x^{DL}(P)$ if and only if $T^{DL} \models_X (\tau_1)_x^{DL}(P)$ and $T^{DL} \models_X (\tau_2)_x^{DL}(P)$, that is, by induction hypothesis, if and only if $T \models_{X(x) \to P} \tau_1 \cap \tau_2$.

10. $T^{DL} \models_X \exists Q((\tau_1)_x^{DL}(Q) \wedge \forall y(\neg Qy \vee (\tau_2)_x^{DL}(P)))$ if and only if there exists a Q such that $T \models_{X(x) \to Q} \tau_1$ and there exists a $Q' \supseteq Q$ such that $T \models_{Q' \to P} \tau_2$. By downwards closure, if this is the case then $T \models_{Q \to P} \tau_2$ too, and hence $T \models_{X(x) \to P} \tau_1; \tau_2$, as required.

Conversely, suppose that there exists a Q such that $T \models_{X(x) \to Q} \tau_1$ and $T \models_{Q \to P} \tau_2$. Then, by induction hypothesis $T^{DL} \models_X (\tau_1)_x^{DL}(Q)$; and furthermore, $X[T^{DL}/y]$ can be split into

$$Z_1 = \{s \in X[T^{DL}/y] : s(y) \notin Q\}$$

and

$$Z_2 = \{s \in X[T^{DL}/y] : s(y) \in Q\}$$

It is trivial to see that $T^{DL} \models_{Z_1} \neg Qy$; and furthermore, since $Z_2(y) = Q$ and $T \models_{Q \to P} \tau_2$, by induction hypothesis we have that $T^{DL} \models_{Z_2} (\tau_2)_y^{DL}$. Thus $T^{DL} \models_{X[T^{DL}/y]} \forall y(\neg Qy \vee (\tau_2)_y^{DL}(P))$ and finally $T^{DL} \models_X (\tau_1; \tau_2)_x^{DL}(P)$, and this concludes the proof. $\square$

Hence, the relationship between transition logic and dependence logic is analogous to the one between dynamic game logic and first-order logic. In the next sections, we will develop variants of dependence logic which are syntactically closer to transition logic, while still being first-order: as we will see, the resulting frameworks are expressively equivalent to dependence logic on the level of satisfiability, but can be used to represent finer-grained phenomena of *transitions* between sets of assignments.

3 Dynamic variants of dependence logic

3.1 Dependence logic and transitions between teams

Now that we have established a connection between dependence logic and a variant of dynamic game logic, it is time to explore what this might imply for the further development of logics of imperfect information. If, as Theorems 2.13 and 2.16 suggest, dependence logic can be thought of as a logic of imperfect-information decision

problems, perhaps it could be possible to develop variants of dependence logic in which expressions can be interpreted directly as transition systems?

In what follows, we will do exactly that, first with *Transition Dependence Logic* – a variant of dependence logic, expressively equivalent to it, which is also a quantified version of transition logic—and then with *Dynamic Dependence Logic*, in which *all* expressions are interpreted as transitions!

But why would we interested in such variants of dependence logic? One possible answer, which we will discuss in this subsection, is that transitions between teams are *already* a central object of study in the field of dependence logic, albeit in a non-explicit manner: after all, the semantics of dependence logic interprets quantifiers in terms of transformations of teams, and disjunctions in terms of decompositions of teams into subteams. This intuition is central to the study of issues of interdefinability in Dependence Logic and its variants, like for example the ones discussed in Galliani (2012). As a simple example, let us recall Definition 1.6:

$$\psi_1 \sqcup \psi_2 := \exists u_1 \exists u_2 (=(u_1) \wedge =(u_2) \wedge ((u_1 = u_2 \wedge \psi_1) \vee (u_1 \neq u_2 \wedge \psi_2))),$$

where u_1 and u_2 are new variables.

As we said in Proposition 1.7, $M \models_X \psi_1 \sqcup \psi_2$ if and only if $M \models_X \psi_1$ or $M \models_X \psi_2$. We will now sketch the proof of this result, and – as we will see – this proof will hinge on the fact that the above expression can be read as a specification of the following algorithm:

1. Choose an element $a \in \mathrm{dom}(M)$ and extend the team X by assigning a as the value of u_1 for all assignments;
2. Choose an element $b \in \mathrm{dom}(M)$ and further extend the team by assigning b as the value of u_2 for all assignments;
3. Split the resulting team into two subteams Y_1 and Y_2 such that
   a) ψ_1 holds in Y_1, and the values of u_1 and u_2 coincide for all assignments in it;
   b) ψ_2 holds in Y_2, and the values of u_1 and u_2 differ for all assignments in it.

Since the values of u_1 and u_2 are chosen to always be respectively a and b, one of Y_1 and Y_2 is empty and the other is of the form $X[ab/u_1u_2]$, and since u_1 and u_2 do not occur in ψ_1 or ψ_2 the above algorithm can succeed (for some choice of a and b) only if $M \models_X \psi_1$ or $M \models_X \psi_2$.

As another, slightly more complicated example, let us consider the following problem. Given four variables x_1, x_2, y_1 and y_2, let $x_1x_2 \mid y_1y_2$ be an *exclusion atom* holding in a team X if and only if for all $s, s' \in X$, $s(x_1x_2) \neq s'(y_1y_2)$ – that is, if and only if the sets of the values taken by x_1x_2 and by y_1y_2 in X are disjoint.

By Theorem 1.12, we can tell at once that there exists some dependence logic formula $\phi(x_1, x_2, y_1, y_2)$ such that for all suitable M and X, $M \models_X \phi(x_1, x_2, y_1, y_2)$ if and only if $M \models_X x_1x_2 \mid y_1y_2$; but what about the converse? For example, can we find an expression $\psi(x, y)$, in the language of first order logic augmented with these exclusion atoms (but with no dependence atoms), such that for all suitable M and X $M \models_X \psi(x, y)$ if and only if $M \models_X =(x, y)$?
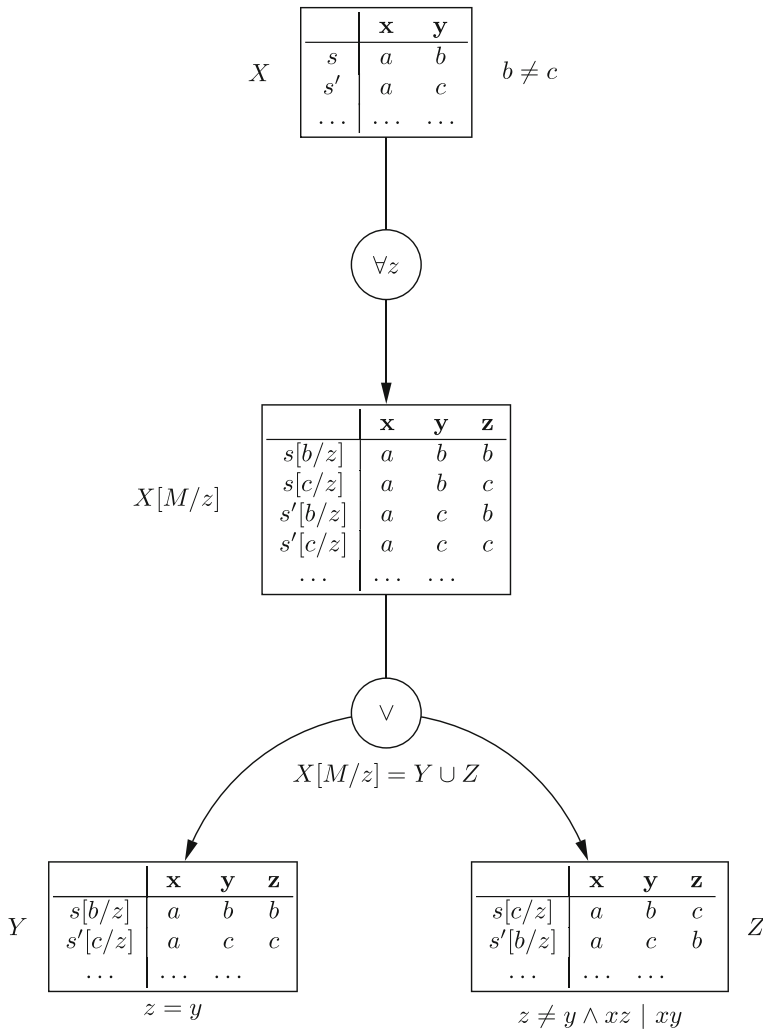
Fig. 1 Checking $=(x, y)$ by evaluating $\forall z(z = y \vee (z \neq y \wedge xz \mid xy))$. If $b \neq c$, then Z does not satisfy the $xz \mid xy$

As discussed in Galliani (2012) in a more general setting, the answer is positive, and one such $\psi(x, y)$ is $\forall z(z = y \vee (z \neq y \wedge xz \mid xy))$, where z is some variable other than x and y.[8] Why is this the case?

Well, let us consider any team X with domain containing x and y, and let us evaluate $\psi(x, y)$ over it. As shown graphically in Fig. 1, the transitions between teams occurring during the evaluation of the formula correspond to the following algorithm:

---

[8] A moment's thought shows that, by downwards closure, the condition $z \neq y$ in the second disjunct can be removed, but for simplicity we will keep it.

1. First, assign all possible values to the variable z for all assignments in x, thus obtaining $X[M/z] = \{s[m/z] : s \in X, m \in \text{dom}(M)\}$;
2. Then, remove from $X[M/z]$ all assignments s for which $s(z) = s(x)$, keeping only the ones for which $s(z) \neq s(y)$;
3. Then, verify that for any possible fixed value of x, the possible values of y and z are disjoint.

This algorithm succeeds only if y is a function of x. Indeed, suppose that instead there are two assignments $s, s' \in X$ such that $s(x) = s'(x) = a$, $s(y) = b$ and $s'(y) = c$ for three $a, b, c \in \text{dom}(M)$ with $b \neq c$. Now we have that $\{s[b/z], s[c/z], s'[b/z], s'[c/z]\} \subseteq X[M/z]$: and since $b \neq c$, we have that the assignments $s[c/z]$ and $s'[b/z]$ are not removed from the team in the second step of the proof. But then $s[c/z](xz) = ac = s'[b/z](xy)$, and therefore it is not true that $xy \mid xz$. And, conversely, if in the team X the value of y is a function of the value of x then by splitting $X[M/z]$ into the two subteams $Y = \{s[m/z] : s \in X, s(y) = s(z)\}$ and $Z = \{s[m/z] : s(y) \neq s(z)\}$ we have that $M \models_Y y = z$, $M \models_Z y \neq z$ and $M \models_Z xz \mid xy$ (since for all $s, s' \in Z$, $s(x) = s'(x) \Rightarrow s(y) = s'(y) \Rightarrow s(z) \neq s(y) = s'(z)$).

On the other hand, one dependence logic expression corresponding to $x_1 x_2 \mid y_1 y_2$ is

$$\forall w_1 w_2 \exists u_1 u_2 (=(w_1, w_2, u_1) \wedge =(w_1, w_2, u_2) \wedge$$
$$((u_1 = u_2 \wedge (w_1 \neq x_1 \vee w_2 \neq z_2)) \vee$$
$$(u_1 \neq u_2 \wedge (w_1 \neq y_1 \vee w_2 \neq y_2))))$$

where w_1, w_2, u_1 and u_2 are new variable.

We encourage the interested reader to verify that this is the case by examining the transitions between teams corresponding to the formula: in brief, the intuition is that first we extend our team by picking all possible pairs of values for w_1 and w_2, then for any such pair we flag – through our choice of u_1 and u_2 – whether $w_1 w_2$ is different from $x_1 x_2$ or from $y_1 y_2$. This implies that no such pair is equal to both $x_1 x_2$ and $y_1 y_2$, or, in other words, that $x_1 x_2$ and $y_1 y_2$ have no value in common.

More and more complex examples of definability results of this kind can be found in Galliani (2012); but what we want to emphasize here is that all these examples, like the one we discussed in depth here, have a natural interpretation in terms of algorithms which transform teams and apply simple tests to them, as the above one. Hence, we hope that the development of variants of Dependence Logic in which these transitions are made explicit might prove itself useful for the further study of this interesting class of problems.

3.2 Transition dependence logic

As stated, we will now define a variant of dependence logic which can also be seen as a quantified variant of transition logic. We will then prove that the resulting transition dependence logic is expressively equivalent to dependence logic, in the sense that any dependence logic formula is equivalent to some transition dependence logic formula and vice versa.

Definition 3.1 (*Transition dependence logic—syntax*) Let Σ be a first-order signature. Then the sets of all *transition terms* and of all *formulas* of dependence transition logic are given by the rules

$$\tau ::= \exists v \mid \forall v \mid \phi? \mid \tau \otimes \tau \mid \tau \cap \tau \mid \tau ; \tau$$
$$\phi ::= R\vec{t} \mid \neg R\vec{t} \mid \; =(t_1, \ldots, t_n) \mid \phi \vee \phi \mid \phi \wedge \phi \mid \langle \tau \rangle \phi.$$

where v ranges over all variables in Var, R ranges over all relation symbols of the signature, $\vec{t}$ ranges over all tuples of terms of the required arities, n ranges over $\mathbb{N}$ and $t_1 \ldots t_n$ range over the terms of our signature.

Definition 3.2 (*Transition dependence logic—semantics*) Let M be a first-order model, let τ be a first-order transition term of the same signature, and let X and Y be teams over M. Then we say that the transition $X \rightarrow Y$ is *allowed* by τ in M, and we write $M \models_{X \rightarrow Y} \tau$, if and only if

    TDL-∃: τ is of the form $\exists v$ for some $v \in \mathrm{Var}$ and there exists a F such that $X[F/v] \subseteq Y$;
    TDL-∀: τ is of the form $\forall v$ for some $v \in \mathrm{Var}$ and $X[M/v] \subseteq Y$;
    TDL-test: τ is of the form $\phi?$, $M \models_X \phi$ in the sense given later in this definition, and $X \subseteq Y$;
    TDL-⊗: τ is of the form $\tau_1 \otimes \tau_2$ and $X = X_1 \cup X_2$ for some X_1 and X_2 such that $M \models_{X_1 \rightarrow Y} \tau_1$ and $M \models_{X_2 \rightarrow Y} \tau_2$;
    TDL-∩: τ is of the form $\tau_1 \cap \tau_2$, $M \models_{X \rightarrow Y} \tau_1$ and $M \models_{X \rightarrow Y} \tau_2$;
    TDL-concat: τ is of the form $\tau_1 ; \tau_2$ and there exists a team Z such that $M \models_{X \rightarrow Z} \tau_1$ and $M \models_{Z \rightarrow Y} \tau_2$.

Similarly, if ϕ is a formula and X is a team with domain Var. Then we say that X *satisfies* ϕ in M, and we write $M \models_X \phi$, if and only if

    TDL-lit: ϕ is a first-order literal and $M \models_s \phi$ in the usual first-order sense for all $s \in X$;
    TDL-dep: ϕ is a dependence atom $=(t_1, \ldots, t_n)$ and any two $s, s' \in X$ which assign the same values to $t_1 \ldots t_{n-1}$ also assign the same value to t_n;
    TDL-∨: ϕ is of the form $\phi_1 \vee \phi_2$ and $M \models_X \phi_1$ or $M \models_X \phi_2$;
    TDL-∧: ϕ is of the form $\phi_1 \wedge \phi_2$, $M \models_X \phi_1$ and $M \models_X \phi_2$;
    TDL-◇: ϕ is of the form $\langle \tau \rangle \psi$ and there exists a Y such that $M \models_{X \rightarrow Y} \tau$ and $M \models_Y \psi$.

As the next theorem shows, in this semantics formulas and transitions are interpreted in terms of trumps and transition systems:

Theorem 3.3 *For all transition dependence logic formulas ϕ, all models M and all teams X and Y, we have that*

    Downwards closure: *If $M \models_X \phi$ and $Y \subseteq X$ then $M \models_Y \phi$;*
    Empty team property: *$M \models_\emptyset \phi$.*

Furthermore, for all transition dependence logic transition terms τ, all models M and all teams X, Y and Z,

> **Downwards closure:** If $M \models_{X \to Y} \tau$ and $Z \subseteq X$ then $M \models_{Z \to Y} \tau$;
> **Monotonicity:** If $M \models_{X \to Y} \tau$ and $Y \subseteq Z$ then $M \models_{X \to Z} \tau$;
> **Non-creation:** For all Y, $M \models_{\emptyset \to Y} \tau$;
> **Non-triviality:** If $X \neq \emptyset$ then $M \not\models_{X \to \emptyset} \tau$.

Proof The proof is by structural induction over ϕ and τ, and presents no difficulties whatsoever. □

Also, it is not difficult to see, on the basis of the results of the previous section, that this new variant of dependence logic is equivalent to the usual one:

Theorem 3.4 *For every dependence logic formula ϕ there exists a transition dependence logic transition term τ_ϕ such that*

$$M \models_X \phi \Leftrightarrow \exists Y \ s.t. \ M \models_{X \to Y} \tau_\phi \Leftrightarrow M \models_X \langle \tau_\phi \rangle \top$$

for all first-order models M and teams X.

Proof τ_ϕ is defined by structural induction on ϕ, as follows:

1. If ϕ is a first-order literal or a dependence atom then $\tau_\phi = \phi?$;
2. If ϕ is $\phi_1 \vee \phi_2$ then $\tau_\phi = \tau_{\phi_1} \otimes \tau_{\phi_2}$;
3. If ϕ is $\phi_1 \wedge \phi_2$ then $\tau_\phi = \tau_{\phi_1} \cap \tau_{\phi_2}$;
4. If ϕ is $\exists v \psi$ then $\tau_\phi = \exists v; \tau_\psi$;
5. If ϕ is $\forall v \psi$ then $\tau_\phi = \forall v; \tau_\psi$.

It is then trivial to verify, again by induction on ϕ, that $M \models_X \phi$ if and only if $M \models_X \langle \tau_\phi \rangle \top$, as required. □

This representation result associates dependence logic *formulas* to transition dependence logic *transition terms*. This fact highlights the dynamical nature of Dependence Logic operators, which we discussed in the previous subsection: in this framework, quantifiers describe *transformations* of teams, the dependence logic connectives are operations over games, and the literals are interpreted as tests. In fact, one might wonder what is the purpose of transition dependence logic formulas: could we do away with them altogether, and develop a variant of transition dependence logic in which *all* formulas are transitions?

Later, we will explore this idea further; but first, let us verify that Transition Dependence Logic is no more expressive than dependence logic.

Theorem 3.5 *For every transition dependence logic formula ϕ there exists a dependence logic formula $T(\phi)$ such that*

$$M \models_X \phi \Leftrightarrow M \models_X T(\phi)$$

for all first-order models M and teams X. Furthermore, for every transition dependence logic transition term τ and dependence logic formula θ there is a dependence logic formula $U(\tau, \psi)$ such that

$$M \models_X U(\tau, \theta) \Leftrightarrow \exists Y \ s.t. \ M \models_{X \to Y} \tau \ and \ M \models_Y \theta,$$

again for all first-order models M and teams X.

Proof We prove the two claims together, by structural induction over ϕ and τ.

First, let us consider the cases corresponding to formulas:

1. If ϕ is a first order literal or a dependence atom, let $T(\phi)$ be ϕ itself. As the interpretation of these expressions is the same in dependence logic and in transition dependence logic, there is nothing to prove.
2. If ϕ is of the form $\psi_1 \vee \psi_2$, let $T(\phi)$ be $T(\psi_1) \sqcup T(\psi_2)$.
   This expression holds in a team if and only if $T(\psi_1)$ or $T(\psi_2)$ hold, that is, by induction hypothesis, if and only if ψ_1 or ψ_2 do.
3. If ϕ is of the form $\psi_1 \wedge \psi_2$, let $T(\phi)$ be $T(\psi_1) \wedge T(\psi_2)$. Then $T(\phi)$ holds if and only if ψ_1 and ψ_2 do, that is, if and only if ϕ does.
4. If ϕ is of the form $\langle \tau \rangle \psi$, let $\vec{v}$ be the tuple of all variables occurring in $T(\psi)$, let R be a new $|\vec{v}|$-ary relation, and let $T(\phi)$ be $\exists R(U(\tau, R\vec{v}) \wedge \forall \vec{v}(\neg R\vec{v} \vee T(\psi)))$. Indeed, suppose that $M \models_X T(\phi)$: then for some relation R, there exists a Y such that $M \models_{X \to Y} \tau$ and $M \models_Y R\vec{v}$. Furthermore, $M \models \forall \vec{v}(\neg R\vec{v} \vee T(\psi))$, and therefore for the set $Y' = \{s : \text{dom}(s) = \vec{v}, M \models_s R\vec{v}\}$ we have that $M \models_{Y'} T(\psi)$. But then, by downwards closure and locality, $M \models_Y T(\psi)$, and therefore $M \models_X \langle \tau \rangle \psi$. Conversely, suppose that $M \models_X \langle \tau \rangle \psi$: then there exists a Y such that $M \models_{X \to Y} \tau$ and $M \models_Y \psi$. Now let R be $\{s(\vec{v}) : s \in Y\}$: clearly $M \models_X U(\tau, R\vec{v})$, since $M \models_Y R\vec{v}$, and furthermore $M \models \forall \vec{v}(\neg R\vec{v} \vee T(\psi))$, by locality and by the fact that (by induction hypothesis) $M \models_Y T(\psi)$.

Now let us consider the cases corresponding to transitions:

5. If τ is of the form $\exists v$ for some variable v, let $U(\tau, \theta)$ be $\exists v\theta$. Indeed, suppose that $M \models_X \exists v\theta$: then $M \models_{X[F/v]} \theta$ for some F, and by choosing $Y = X[F/v]$ we have that $M \models_{X \to Y} \exists v$ and $M \models_Y \theta$, as required. Conversely, suppose that for some Y, $M \models_{X \to Y} \exists v$ and $M \models_Y \theta$: then for some F, $X[F/v] \subseteq Y$, and by downwards closure we have that $M \models_{X[F/v]} \theta$.
6. If τ is of the form $\forall v$ for some variable v, let $U(\tau, \theta)$ be $\forall v\theta$. Indeed, suppose that $M \models_X \forall v\theta$: then $M \models_{X[M/v]} \theta$, and if we choose $Y = X[M/v]$ we have at once that $M \models_{X \to Y} \forall v$ and $M \models_Y \theta$. Conversely, if for some Y $M \models_{X \to Y} \forall v$ and $M \models_Y \theta$ then $X[M/v] \subseteq Y$ and, by downwards closure, $M \models_{X[M/v]} \theta$.
7. If τ is of the form $\phi?$, let $U(\tau, \theta)$ be $T(\phi) \wedge \theta$. Indeed, suppose that $M \models_X T(\phi) \wedge \theta$: then by induction hypothesis $M \models_X \phi$, and, for $Y = X$, we have that $M \models_{X \to Y} \phi?$. Furthermore, $M \models_Y \theta$, as required. Conversely, suppose that for some Y, $M \models_{X \to Y} \phi?$ and $M \models_Y \theta$. Then $M \models_X \phi$, and therefore $M \models_X T(\phi)$; and furthermore $X \subseteq Y$, and hence by downwards closure $M \models_X \theta$. Hence, $M \models_X T(\phi) \wedge \theta$.
8. If τ is of the form $\tau_1 \otimes \tau_2$ and $\vec{v}$ is the tuple of all free variables of θ then let $U(\tau, \theta)$ be $\exists R((U(\tau_1, R\vec{v}) \vee U(\tau_2, R\vec{v})) \wedge \forall \vec{v}(\neg R\vec{v} \vee \theta))$, where R is a new $|\vec{r}|$-ary relation symbol. Indeed, suppose that $M \models_X U(\tau, \theta)$: then there exists a relation R and two subteams X_1 and X_2 of X such that $X = X_1 \cup X_2$, $M \models_{X_1} U(\tau_1, R\vec{v})$ and $M \models_{X_2} U(\tau_2, R\vec{v})$. Hence, there are two teams Y_1 and Y_2 such that $M \models_{X_1 \to Y_1} \tau_1$, $M \models_{X_2 \to Y_2} \tau_2$, $M \models_{Y_1} R\vec{v}$ and $M \models_{Y_2} R\vec{v}$. Now, let Y be $Y_1 \cup Y_2$: by monotonicity, we have that $M \models_{X_1 \to Y} \tau_1$ and $M \models_{X_2 \to Y} \tau_2$, and furthermore $M \models_Y R\vec{v}$ too (that is, for all $s \in Y$, $s(\vec{v})$ is in R). Since $M \models \forall \vec{v}(\neg R\vec{v} \vee \theta)$, this implies that $M \models_Y \theta$, by locality and downwards closure.

Conversely, suppose that there is a Y such that $M \models_{X \to Y} \tau_1 \otimes \tau_2$ and $M \models_Y \theta$. Then let R be $\{s(\vec{v}) : s \in Y\}$. Now $X = X_1 \cup X_2$ for two X_1 and X_2 such that $M \models_{X_1 \to Y} \tau_1$ and $M \models_{X_2 \to Y} \tau_2$, and by induction hypothesis we have that $M \models_{X_1} U(\tau_1; R\vec{v})$ and $M \models_{X_2} U(\tau_2; R\vec{v})$. But then $M \models_X U(\tau_1; R\vec{v}) \vee U(\tau_2; R\vec{v})$; and furthermore, by locality we have that $M \models \forall \vec{v}(\neg R\vec{v} \vee \theta)$. Hence, $M \models_X U(\tau_1 \otimes \tau_2, \theta)$, as required.

9. If τ is of the form $\tau_1 \cap \tau_2$ and $\vec{v}$ is the tuple of all variables of θ then let $U(\tau, \theta)$ be $\exists \vec{R}(U(\tau_1, R\vec{v}) \wedge U(\tau_2, R\vec{v}) \wedge \forall \vec{v}(\neg R\vec{v} \vee \theta))$. Indeed, suppose that $M \models_X U(\tau, \theta)$: then for some relation R, by induction hypothesis, there exist teams Y_1 and Y_2 such that $M \models_{X \to Y_1} \tau_1$, $M \models_{X \to Y_2} \tau_2$, $M \models_{Y_1} R\vec{v}$ and $M \models_{Y_2} R\vec{v}$. Now let Y be $Y_1 \cup Y_2$: as before, by monotonicity we have that $M \models_{X \to Y} \tau_1$ and $M \models_{X \to Y} \tau_2$, and hence $M \models_{X \to Y} \tau_1 \cap \tau_2$. Finally, since $M \models \forall \vec{v}(\neg R\vec{v} \vee \theta)$ we have that $M \models_Y \theta$, as required.

Conversely, suppose that there is a Y such that $M \models_{X \to Y} \tau_1 \cap \tau_2$ and $M \models_Y \theta$. Since $M \models_{X \to Y} \tau_1 \cap \tau_2$, $M \models_{X \to Y} \tau_1$ and $M \models_{X \to Y} \tau_2$. Now let R be $\{s(\vec{v}) : s \in Y\}$. By induction hypothesis, $M \models_X U(\tau_1, R\vec{t})$ and $M \models_X U(\tau_2, R\vec{t})$; and furthermore, since $M \models_Y \theta$ we have that $M \models \forall \vec{v}(\neg R\vec{v} \vee \theta)$.

10. If τ is of the form $\tau_1; \tau_2$ let $U(\tau, \theta)$ be $U(\tau_1, U(\tau_2, \theta))$. Indeed, $M \models_X U(\tau_1, U(\tau_2, \theta))$ if and only if there is a Y such that $M \models_{X \to Y}$ and $M \models_Y U(\tau_2, \theta)$, that is, if and only if there are a Y and a Z such that $M \models_{X \to Y} \tau_1$, $M \models_{Y \to Z} \tau_2$ and $M \models_Z \theta$.   □

However, in a sense, transition dependence logic allows one to consider subtler distinctions than dependence logic does. The formula $\forall x \exists y(=(y, f(x)) \wedge Pxy)$, for example, could be translated as any of

- $\langle \forall x; \exists y \rangle(=(y, f(x)) \wedge Pxy)$;
- $\langle \forall x; \exists y \rangle \langle =(y, f(x))? \rangle Pxy$;
- $\langle \forall x; \exists y \rangle \langle Pxy? \rangle =(y, f(x))$;
- $\langle \forall x; \exists y \rangle \langle (Pxy?) \cap (=(y, f(x))?) \rangle \top$.

The intended interpretations of these formulas are rather different, even though they happen to be satisfied by the same teams: and for this reason, transition dependence logic may be thought of as a proper refinement of dependence logic even though it has exactly the same expressive power.

3.3 Dynamic predicate logic

Dynamic Semantics is the name given to a family of semantical frameworks which subscribe to the following principle (Groenendijk and Stokhof 1991):

> *The meaning of a sentence does not lie in its truth conditions, but rather in the way it changes (the representation of) the information of the interpreter.*

In various forms, this intuition can be found prefigured in some of the later work of Ludwig Wittgenstein, as well as in the research of philosophers of language such as Austin, Grice, Searle, Strawson and others (Dekker 2008); but its formal development

can be traced back to the work of Groenendijk and Stokhof about the proper treatment of pronouns in formal linguistics (Groenendijk and Stokhof 1991).

We refer to Dekker (2008) for a comprehensive analysis of the linguistic issues which caused such a development, as well as for a description of the ways in which this framework was adapted in order to model presuppositions, questions/answers and other phenomena; here we will only present a formulation of *dynamic predicate semantics*, the alternative semantics for first-order logic which was developed in the above mentioned paper by Groenendijk and Stokhof.

Definition 3.6 (*Dynamic semantics for first-order logic*) Let ϕ be a first-order formula, let M be a suitable first-order model and let s and s' be two assignments. Then we say that the transition from s to s' is *allowed* by ϕ in M, and we write $M \models_{s \to s'} \phi$, if and only if

DPL-atom: ϕ is an atomic formula, $s = s'$ and $M \models_s \phi$ in the usual sense;
DPL-$\neg$: ϕ is of the form $\neg\psi$, $s = s'$ and for all assignments h, $M \not\models_{s \to h} \psi$;
DPL-$\wedge$: ϕ is of the form $\psi_1 \wedge \psi_2$ and there exists an h such that $M \models_{s \to h} \psi_1$ and $M \models_{h \to s'} \psi_2$;
DPL-$\vee$: ϕ is of the form $\psi_1 \vee \psi_2$, $s = s'$ and there exists an h such that $M \models_{s \to h} \psi_1$ or $M \models_{s \to h} \psi_2$;
DPL-$\to$: ϕ is of the form $\psi_1 \to \psi_2$, $s = s'$ and for all h it holds that

$$M \models_{s \to h} \psi_1 \Rightarrow \exists h' \text{ s.t. } M \models_{h \to h'} \psi_2;$$

DPL-$\exists$: ϕ is of the form $\exists x \psi$ and there exists an element $m \in \mathrm{Dom}(M)$ such that $M \models_{s[m/x] \to s'} \psi$;
DPL-$\forall$: ϕ is of the form $\forall x \psi$, $s = s'$ and for all elements $m \in \mathrm{Dom}(M)$ there exists an h such that $M \models_{s[m/x] \to h} \psi$.

A formula ϕ is *satisfied* by an assignment s if and only if there exists an assignment s' such that $M \models_{s \to s'} \phi$; in this case, we will write $M \models_s \phi$.

We will discuss neither the formal properties of this formalism nor its linguistic applications here. All that is relevant for our purposes is that, according to it, formulas are interpreted as *transitions* from assignments to assignments, and furthermore that the rule for conjunction allows us to bind occurrences of a variable of the second conjunct to quantifiers occurring in the first one.[9]

The similarity between this semantics and our semantics for transition terms should be evident. Hence, it seems natural to ask whether we can adopt, for a suitable variant of dependence logic, the following variant of Groenendijk and Stokhof's motto:

The meaning of a formula does not lie in its satisfaction conditions, but rather in the team transitions it allows.

---

[9] For example, consider the formula $(\exists x\, Px) \wedge Qx$: by the rules given, it is easy to see that $M \models_s (\exists x\, Px) \wedge Qx$ if and only if $P^M \cap Q^M \neq \emptyset$, that is, if and only if $M \models_s \exists x(Px \wedge Qx)$, differently from the case of Tarski's semantics.

From this point of view, *transition terms* are the fundamental objects of our syntax, and formulas can be removed altogether from the language—although, of course, the tests corresponding to literals and dependence formulas should still be available. As in Groenendijk and Stokhof's logic, satisfaction becomes then a derived concept: in brief, a team X can be said to satisfy a term τ if and only if there exists a Y such that τ allows the transition from X to Y, or, in other words, if and only if *some* set of non-losing outcomes can be reached from the set X of initial positions in the game corresponding to τ.

In the next section, we will make use of these intuitions to develop another, terser version of dependence logic; and finally, we will discuss some implications of this new version for the further developments and for the possible applications of this interesting logical formalism.

3.4 Dynamic dependence logic

We will now develop a formula-free variant of transition dependence logic, along the lines of Groenendijk and Stokhof's dynamic predicate logic.

Definition 3.7 (*Dynamic dependence logic—syntax*) Let Σ be a first-order signature. The set of all formulas of dynamic dependence logic over Σ is given by the rules

$$\tau ::= R\vec{t} \mid \neg R\vec{t} \mid\, = (t_1, \ldots, t_n) \mid \exists v \mid \forall v \mid \tau \otimes \tau \mid \tau \cap \tau \mid \tau; \tau$$

where, as usual, R ranges over all relation symbols of our signature, $\vec{t}$ ranges over all tuples of terms of the required lengths, n ranges over $\mathbb{N}$, $t_1 \ldots t_n$ range over all terms, and v ranges over $\mathtt{Var}$.

The semantical rules associated to this language are precisely as one would expect:

Definition 3.8 (*Dynamic dependence logic—semantics*) Let M be a first-order model, let τ be a dynamic dependence logic formula over the signature of M, and let X and Y be two teams over M with domain $\mathtt{Var}$. Then we say that τ *allows* the transition $X \to Y$ in M, and we write $M \models_{X \to Y} \tau$, if and only if

DDL-lit: τ is a first-order literal, $M \models_s \tau$ in the usual first-order sense for all $s \in X$, and $X \subseteq Y$;
DDL-dep: τ is a dependence atom $= (t_1, \ldots, t_n)$, $X \subseteq Y$, and any two assignments $s, s' \in X$ which coincide over $t_1 \ldots t_{n-1}$ also coincide over t_n;
DDL-∃: τ is of the form $\exists v$ for some $v \in \mathtt{Var}$, and $X[F/v] \subseteq Y$ for some $F : X \to \mathrm{Dom}(M)$;
DDL-∀: τ is of the form $\forall v$ for some $v \in \mathtt{Var}$, and $X[M/v] \subseteq Y$;
DDL-⊗: τ is of the form $\tau_1 \otimes \tau_2$ and $X = X_1 \cup X_2$ for two teams X_1 and X_2 such that $M \models_{X_1 \to Y} \tau_1$ and $M \models_{X_2 \to Y} \tau_2$;
DDL-∩: τ is of the form $\tau_1 \cap \tau_2$, $M \models_{X \to Y} \tau_1$ and $M \models_{X \to Y} \tau_2$;
DDL-concat: τ is of the form $\tau_1; \tau_2$, and there exists a Z such that $M \models_{X \to Z} \tau_1$ and $M \models_{Z \to Y} \tau_2$.

A formula τ is said to be *satisfied* by a team X in a model M if and only if there exists a Y such that $M \models_{X \to Y} \tau$; and if this is the case, we will write $M \models_X \tau$.

It is not difficult to see that Ddynamic dependence logic is equivalent to transition dependence logic (and, therefore, to dependence logic).

Proposition 3.9 *Let ϕ be a dependence logic formula. Then there exists a dynamic dependence logic formula ϕ' which is equivalent to it, in the sense that*

$$M \models_X \phi \Leftrightarrow M \models_X \phi' \Leftrightarrow \exists Y \text{ s.t. } M \models_{X \to Y} \phi'$$

for all suitable teams X and models M

Proof We build ϕ' by structural induction:

1. If ϕ is a literal or a dependence atom then $\phi' = \phi$;
2. If ϕ is $\psi_1 \vee \psi_2$ then $\phi' = \psi_1' \otimes \psi_2'$;
3. If ϕ is $\psi_1 \wedge \psi_2$ then $\phi' = \psi_1' \cap \psi_2'$;
4. If ϕ is $\exists x \psi$ then $\phi' = \exists x; \psi'$;
5. If ϕ is $\forall x \psi$ then $\phi' = \forall x; \psi'$. $\square$

Proposition 3.10 *Let τ be a dynamic dependence logic formula. Then there exists a transition dependence logic transition term τ' such that*

$$M \models_{X \to Y} \tau \Leftrightarrow M \models_{X \to Y} \tau'$$

for all suitable X, Y and M, and such that hence

$$M \models_X \tau \Leftrightarrow M \models_X \langle \tau' \rangle \top.$$

Proof Build τ' by structural induction:

1. If τ is a literal or dependence atom then $\tau' = \tau?$;
2. If τ is of the form $\exists v$ or $\forall v$ then $\tau' = \tau$;
3. If τ is of the form $\tau_1 \otimes \tau_2$ then $\tau' = \tau_1' \otimes \tau_2'$;
4. If τ is of the form $\tau_1 \cap \tau_2$ then $\tau' = \tau_1' \cap \tau_2'$;
5. If τ is of the form $\tau_1; \tau_2$ then $\tau' = \tau_1'; \tau_2'$. $\square$

Corollary 3.11 *Dynamic dependence logic is equivalent to transition dependence logic and to dependence logic*

Proof Follows from the two previous results and from the equivalence between dependence logic and transition dependence logic. $\square$

4 Further work

In this work, we established a connection between a variant of dynamic game logic and dependence logic, and we used it as the basis for the development of variants of

dependence logic in which it is possible to talk directly about transitions from teams to teams. This suggests a new perspective on dependence logic and team semantics, one which allow us to study them as a special kind of *algebras of nondeterministic transitions between relations*. One of the main problems that is now open is whether it is possible to axiomatize these algebras, in the same sense in which, in Mann (2009), Allen Mann offers an axiomatization of the algebra of trumps corresponding to IF logic (or, equivalently, to dependence logic).

Furthermore, we might want to consider different choices of connectives, like for example ones related to the theory of database transactions. The investigation of the relationships between the resulting formalisms is a natural continuation of the currently ongoing work on the study of the relationship between various extensions of dependence logic, and promises of being of great utility for the further development of this fascinating line of research.

References

Dekker, P. (2008). A guide to dynamic semantics. ILLC Prepublication Series (PP-2008-42).

Engström, F. (2012). Generalized quantifiers in dependence logic. *Journal of Logic, Language and Information*, *21*(3), 299–324.

Galliani, P. (2011). Multivalued dependence logic and independence logic. In: *Non-classical modal and predicate logics*.

Galliani, P. (2012). Inclusion and exclusion dependencies in team semantics: On some logics of imperfect information. *Annals of Pure and Applied Logic*, *163*(1), 68–84.

Grädel, E., & Väänänen, J. (2013). Dependence and independence. *Studia Logica*, *101*(2), 399–410.

Groenendijk, J., & Stokhof, M. (1991). Dynamic predicate logic. *Linguistics and Philosophy*, *14*(1), 39–100.

Henkin, L. (1961). Some remarks on infinitely long formulas. In: *Infinitistic Methods. Proc. Symposium on Foundations of Mathematics* (pp. 167–183). Oxford: Pergamon Press.

Hintikka, J. (1968). Language-games for quantifiers. In: *American philosophical quarterly monograph series 2: Studies in logical theory* (pp. 46–72). Oxford: Basil Blackwell.

Hintikka, J. (1996). *The principles of mathematics revisited*. Cambridge, MA: Cambridge University Press.

Hintikka, J., & Sandu, G. (1989). Informational independence as a semantic phenomenon. In J. Fenstad, I. Frolov, & R. Hilpinen (Eds.), *Logic, methodology and philosophy of science*. Amsterdam: Elsevier.

Hodges, W. (1997). Compositional semantics for a language of imperfect information. *Journal of the Interest Group in Pure and Applied Logics*, *5*(4), 539–563.

Kontinen, J., & Nurmi, V. (2009). Team logic and second-order logic. In: H. Ono, M. Kanazawa, & R. de Queiroz (Eds.), *Logic, language, information and computation*, Vol. 5514 of Lecture Notes in Computer Science (pp. 230–241). Berlin: Springer.

Kontinen, J., & Väänänen, J. (2009). On definability in dependence logic. *Journal of Logic, Language and Information*, *3*(18), 317–332.

Mann, A. L. (2009). Independence-friendly cylindric set algebras. *Logic Journal of IGPL*, *17*(6), 719–754.

Mann, A. L., Sandu, G., & Sevenster, M. (2011). *Independence-friendly logic: A game–theoretic approach*. Cambridge, UK: Cambridge University Press.

Parikh, R. (1985). The logic of games and its applications. In: *Selected Papers of the International Conference on "Foundations of Computation Theory" on Topics in the Theory of Computation* (pp. 111–139). New York, NY, USA.

Väänänen, J. (2007a). *Dependence logic*. Cambridge, UK: Cambridge University Press.

Väänänen, J. (2007b). Team logic. In J. van Benthem, D. Gabbay, & B. Löwe (Eds.), *Interactive logic. Selected papers from the 7th Augustus de Morgan workshop* (pp. 281–302). Amsterdam: Amsterdam University Press.

van Benthem, J. (2003). Logic games are complete for game logics. *Studia Logica*, *75*, 183–203.

van Benthem, J., Ghosh, S., & Liu, F. (2008). Modelling simultaneous games in dynamic logic. *Synthese*, *165*, 247–268.