The dynamics of imperfect information

Galliani, P.

Citation for published version (APA):

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: https://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.
Chapter 6

Transition Dynamics

In this chapter, we will extend the mutual embedding relation between Dynamic Game Logic and First Order Logic proved by van Benthem (and presented here in Section 6.1) to a relation between Dependence Logic and a suitable imperfect-information, player-versus-Nature variant of Dynamic Game Logic, which we will call Transition Logic (Section 6.2).

This will allow us to reinterpret Dependence Logic as a logic for modeling decision problems under imperfect information; and in Section 6.3, we will exploit this intuition to develop a dynamic version of Dependence Logic in which formulas are interpreted in terms of transitions from information states (teams) to information states.

6.1 On Dynamic Game Logic and First Order Logic

6.1.1 Dynamic Game Logic

*Game logics* are logical formalisms which contain two different kinds of expressions:

1. *Game terms*, which are descriptions of games in terms of compositions of *atomic games*;

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 \cite{59}. Then, in the next subsection, we will discuss a remarkable connection between First-Order Logic and Dynamic Game Logic discovered by Johan van Benthem in \cite{69}.

One of the fundamental semantic concepts of Dynamic Game Logic is the notion of forcing relation:

**Definition 6.1.1.** Let $S$ be a nonempty set of states. A forcing relation over $S$ is a set $\rho \subseteq S \times \text{Parts}(S)$.

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

- **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:** $(s,\emptyset) \not\in \rho^i$.
- **Determinacy:** If $(s,X) \not\in \rho^i$ then $(s,S \setminus X) \in \rho^j$, where $j \in \{E,A\} \setminus \{i\}$.

**Definition 6.1.3.** Let $S$ be a nonempty set of states, let $\Phi$ be a nonempty set of atomic propositions and let $\Gamma$ be a nonempty set of atomic game symbols. Then a game model over $S$, $\Phi$ and $\Gamma$ is a triple $(S, \{(\rho^E_g, \rho^A_g) : g \in \Gamma\}, V)$, where $(\rho^E_g, \rho^A_g)$ 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 $\phi$ into a test game, and the diamond operation, which combines a game term $\gamma$ and a formula $\phi$ into a new

\footnote{The main difference between this version and the one of Parikh’s original paper lies in the absence of the iteration operator $\gamma^\ast$ from our formalism. In this, we follow \cite{69, 71}.}
6.1. On Dynamic Game Logic and First Order Logic

formula $\langle \gamma, i \rangle \phi$ which asserts that agent $i$ can guarantee that the game $\gamma$ will end in a state satisfying $\phi$.

**Definition 6.1.4.** Let $\Phi$ be a nonempty set of atomic propositions and let $\Gamma$ be a nonempty set of atomic game formulas. Then the sets of all game terms $\gamma$ and formulas $\phi$ are defined as

$$
\begin{align*}
\gamma & ::= g \mid \phi \mid \gamma : \gamma \mid \gamma^d \\
\phi & ::= \bot \mid p \mid \neg \phi \mid \phi \lor \phi \mid \langle \gamma, i \rangle \phi
\end{align*}
$$

for $p$ ranging over $\Phi$, $g$ ranging over $\Gamma$, 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: $\gamma^d$ is obtained by swapping the roles of the players in $\gamma$, $\gamma_1 \cup \gamma_2$ is a game in which the existential player $E$ chooses whether to play $\gamma_1$ or $\gamma_2$, and $\gamma_1 ; \gamma_2$ is the concatenation of the two games corresponding to $\gamma_1$ and $\gamma_2$ respectively.

**Definition 6.1.5.** Let $G = (S, \{(\rho^E_g, \rho^A_g) : g \in \Gamma, V\})$ be a game model over $S$, $\Gamma$ and $\Phi$. Then for all game terms $\gamma$ and all formulas $\phi$ of Dynamic Game Logic over $\Gamma$ and $\Phi$ we define a game $\parallel \gamma \parallel_G$ and a set $\parallel \phi \parallel_G \subseteq S$ as follows:

**DGL-atomic-game:** For all $g \in G$, $\parallel g \parallel_G = (\rho^E_g, \rho^A_g)$;

**DGL-test:** For all formulas $\phi$, $\parallel \phi ? \parallel_G = (\rho^E, \rho^A)$, where

- $s \rho^E X$ if $s \in \parallel \phi \parallel_G$ and $s \in X$;
- $s \rho^A X$ if $s \not\in \parallel \phi \parallel_G$ or $s \in X$

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

**DGL-concat:** For all game terms $\gamma_1$ and $\gamma_2$, $\parallel \gamma_1 ; \gamma_2 \parallel_G = (\rho^E, \rho^A)$, where, for all $i \in \{E, A\}$ and for $\parallel \gamma_1 \parallel_G = (\rho^E_i, \rho^A_i)$, $\parallel \gamma_2 \parallel_G = (\rho^E_i, \rho^A_i)$,

- $s \rho^E X$ if and only if there exists a $Z$ such that $s \rho^A_1 Z$ and for each $z \in Z$ there exists a set $X_z$ satisfying $z \rho^A_2 X_1$ such that $X = \bigcup_{z \in Z} X_z$;

**DGL-∪:** For all game terms $\gamma_1$ and $\gamma_2$, $\parallel \gamma_1 \cup \gamma_2 \parallel_G = (\rho^E, \rho^A)$, where
Chapter 6. Transition Dynamics

- \( s \rho^E X \) if and only if \( s \rho^E_1 X \) or \( s \rho^E_2 X \), and
- \( s \rho^A X \) if and only if \( s \rho^A_1 X \) and \( s \rho^A_2 X \)

where, as before, \( \| \gamma_1 \|_G = (\rho^E_1, \rho^A_1) \) and \( \| \gamma_2 \|_G = (\rho^E_2, \rho^A_2); \)\(^2\)

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-\( \lor \): \( \| \phi_1 \lor \phi_2 \|_G = \| \phi_1 \|_G \cup \| \phi_2 \|_G \);

DGL-\( \bullet \): If \( \| \gamma \|_G = (\rho^E, \rho^A) \) then for all \( \phi \),

\[
\|(\gamma, i)\phi\|_G = \{ s \in S : \exists X_s \subseteq \| \phi \|_G \text{ s.t. } s \rho^A X_s \}\.
\]

If \( s \in \| \phi \|_G \), we say that \( \phi \) 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 [71] 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 Subsection 6.2.3.

### 6.1.2 The Representation Theorem

In this subsection, we will briefly recall a remarkable result from [69] 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 the other in a way which preserves satisfiability and truth:

**Theorem 6.1.6.** Let \( G = (S, \{ (\rho^E_g, \rho^A_g) : g \in \Gamma \}, V) \) be any game model, let \( \phi \) be any game formula for the same language, and let \( s \in S \). Then it is possible

\(^2\)[71] 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 \( s \rho^A_1 Z_1 \) and \( s \rho^A_2 Z_2 \).

It is trivial to see that, if our games satisfy the monotonicity condition, this rules is equivalent to the one we presented.
to uniformly construct a first-order model $G^{FO}$, a first-order formula $\phi^{FO}$ and an assignment $s^{FO}$ of $G^{FO}$ such that

$$G \models_{s} \phi \iff G^{FO} \models_{s^{FO}} \phi^{FO}.$$ 

**Theorem 6.1.7.** Let $M$ be any first order model, let $\phi$ 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 $\phi^{DGL}$ and a state $s^{DGL}$ such that

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

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.

### 6.2 Transition Logic

#### 6.2.1 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 which 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 proving the analogue of van Benthem’s above mentioned results.

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

**Definition 6.2.1**. Let $S$ be a nonempty set of *states*. A *transition system* over $S$ is a nonempty relation $\theta \subseteq \text{Parts}(S) \times \text{Parts}(S)$ 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 $(X,\emptyset) \not\in \theta$.

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 6.2.2**. A *decision game* is a triple $\Gamma = (S,E,O)$, where $S$ is a nonempty set of *states*, $E$ is a nonempty set of *strategies* and $O$ is an *outcome function* from $S \times E$ to $\text{Parts}(S)$.

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 6.2.3**. Let $\Gamma = (S,E,O)$ be a decision game, and let $X,Y \subseteq S$. Then we say that $\Gamma$ *allows* the transition $X \rightarrow Y$, and we write $\Gamma : X \rightarrow Y$, if and only if there exists a $e \in E$ such that $\emptyset \neq O(s,e) \subseteq Y$ for all $s \in X$.$^3$

**Theorem 6.2.4** (Transition Systems and Abilities). A set $\theta \subseteq \text{Parts}(S) \times \text{Parts}(S)$ is a transition system if and only if there exists a decision game $\Gamma = (S,E,O)$ such that

$$(X,Y) \in \theta \iff \Gamma : X \rightarrow Y.$$ 

$^3$That is, if and only if our agent has a strategy which guarantees that the outcome will be in $Y$ whenever the input is in $X$. 
6.2. Transition Logic

Proof. Let \( \theta = \{(X_i, Y_i) : i \in I\} \) be a transition system, and let \( \Gamma = (S, I, O) \) for

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

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 an \(i \in I\) 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 strategy. 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 an \(e\) 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.

\[\square\]

**Definition 6.2.5.** 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:\(^4\)

\(^4\)The term “trump” is taken from [42], where it is used to describe the set of all teams which satisfy a given formula.
136  Chapter 6. Transition Dynamics

**Proposition 6.2.6.** Let $\theta$ be a transition system and let $Y \subseteq S \neq \emptyset$. Then $\text{reach}(\theta, Y) = \{X \mid (X, Y) \in \theta\}$ forms a trump. Conversely, for any trump $X$ over $S$ there exists a transition system $\theta$ such that $X = \text{reach}(\theta, Y)$ for any nonempty $Y \subseteq S$.

**Proof.** Let $\theta$ 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, $\text{reach}(\theta, Y)$ is a trump, as required.

Conversely, let $X = \{X_i : i \in I\}$ be a trump. Then define $\theta$ 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 $\theta$ is a transition system; and by construction, for $Y \neq \emptyset$ we have that $(A, Y) \in \theta \iff \exists i \text{ s.t. } A \subseteq X_i \iff A \in X$, where we used the fact that $X$ is downwards closed.

We can now define the syntax and semantics of Transition Logic:

**Definition 6.2.7.** Let $\Phi$ be a set of atomic propositional symbols and let $\Theta$ 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, $\theta_t$ 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 $S$.

**Definition 6.2.8.** Let $\Phi$ be a set of atomic propositions and let $\Theta$ be a set of atomic transitions. Then the transition terms and formulas of our language are defined respectively as

$$
\tau ::= t | \phi? | \tau \otimes \tau | \tau \cap \tau | \tau; \tau
$$

$$
\phi ::= \top | p | \phi \lor \phi | \phi \land \phi | (\tau)\phi
$$

where $t$ ranges over $\Theta$ and $p$ ranges over $\Phi$.

**Definition 6.2.9.** Let $T = (S, \{\theta_t : t \in \Theta\}, V)$ be a transition model, let $\tau$ be a transition term, and let $X, Y \subseteq S$. Then we say that $\tau$ allows the transition from $X$ to $Y$, and we write $T \models_{X \rightarrow Y} \tau$, 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 $\phi$ such that $T \models_X \phi$, and $X \subseteq Y$;
- **TL-⊗:** $\tau = \tau_1 \otimes \tau_2$, and $X = X_1 \cup X_2$ for two $X_1$ and $X_2$ such that $T \models_{X_1 \rightarrow Y} \tau_1$ and $T \models_{X_2 \rightarrow Y} \tau_2$.
6.2. Transition Logic

\[ TL-\cap: \quad \tau_1 \cap \tau_2, \ T \models_{X \to Y} \tau_1 \ \text{and} \ T \models_{X \to Y} \tau_2; \]

\[ TL-\text{concat:} \quad \tau = \tau_1 \cdot \tau_2 \ \text{and there exists a} \ Z \subseteq S \ \text{such that} \ T \models_{X \to Z} \tau_1 \ \text{and} \ T \models_{Z \to Y} \tau_2. \]

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

\[ TL-\top: \quad \phi = \top; \]

\[ TL-\text{atomic-pr:} \quad \phi = p \ \text{for some} \ p \in \Phi \ \text{and} \ X \in V(p); \]

\[ TL-\lor: \quad \phi = \psi_1 \lor \psi_2 \ \text{and} \ T \models_X \psi_1 \ \text{or} \ T \models_X \psi_2; \]

\[ TL-\land: \quad \phi = \psi_1 \land \psi_2, \ T \models_X \psi_1 \ \text{and} \ T \models_X \psi_2; \]

\[ TL-\Diamond: \quad \phi = \langle \tau \rangle \psi \ \text{and there exists a} \ Y \ \text{such that} \ T \models_{X \to Y} \tau \ \text{and} \ T \models_Y \psi. \]

**Proposition 6.2.10.** For any transition model \( T, \) transition term \( \tau, \) and transition formula \( \phi, \) 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. \( \square \)

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 [66, 49]; 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 replaces 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 \( \tau_i \) 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 \( \tau_1 \) is played, and from any state in \( X_2 \) when \( \tau_2 \) is played.

It is also of course possible to introduce a “true” choice operator \( \tau_1 \cup \tau_2 \), with semantical condition

\[
\text{TL-}\cup: \quad T \models_{X \rightarrow Y} \tau_1 \cup \tau_2 \text{ iff } T \models_{X \rightarrow Y} \tau_1 \text{ or } T \models_{X \rightarrow Y} \tau_2;
\]

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

\[
\text{TL-}\star: \quad T \models_{X \rightarrow Y} \tau^\star \text{ iff there exist } n \in \mathbb{N} \text{ and } Z_0 \ldots Z_n \text{ such that } Z_0 = X, Z_n = Y \text{ and } T \models_{Z_i \rightarrow Z_{i+1}} \tau \text{ for all } i \in 1 \ldots n - 1.
\]

### 6.2.2 A Representation Theorem for Dependence Logic

This subsection contains the central result of this chapter, that is, an analogue of van Benthem’s results (Theorems 6.1.6 and 6.1.7 here) for Dependence Logic and Transition Logic.

Representing Dependence Logic models and formulas in Transition Logic is fairly simple:

**Definition 6.2.11.** Let \( M \) be a first-order model. Then \( M^{TL} = (S, \{\theta_3^v, \theta_\forall^v : v \in \text{Var}\}, V) \) is the transition model such that

- \( S \) is the set of all teams over \( M \);
- For any variable \( v \), \( \theta_3^v = \{(X,Y) : \exists F \text{ s.t. } X[F/v] \subseteq Y\} \) and \( \theta_\forall^v = \{(X,Y) : X[M/v] \subseteq Y\} \);
- For any first-order literal or dependence atom \( \alpha \), \( V(\alpha) = \{X : M \models_X \alpha\} \).

**Definition 6.2.12.** Let \( \phi \) be a Dependence Logic formula. Then \( \phi^{TL} \) is the transition term defined as follows:

1. If \( \phi \) is a literal or a dependence atom, \( \phi^{TL} = \phi \);
6.2. Transition Logic

2. If $\phi = \psi_1 \lor \psi_2$, $\phi^{TL} = (\psi_1)^{TL} \otimes (\psi_2)^{TL}$;

3. If $\phi = \psi_1 \land \psi_2$, $\phi^{TL} = (\psi_1)^{TL} \land (\psi_2)^{TL}$;

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 6.2.13. For all first-order models $M$, teams $X$ and formulas $\phi$, the following are equivalent:

- $M \models_X \phi$;
- $\exists Y$ s.t. $M^{TL} \models_{X \rightarrow Y} \phi^{TL}$;
- $M^{TL} \models_X (\phi^{TL}) \top$;
- $M^{TL} \models_{X \rightarrow S} \phi^{TL}$.

Proof. We show, by structural induction on $\phi$, 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 $\phi$ is a literal or a dependence atom, $M^{TL} \models_{X \rightarrow S} \phi$ if and only if $X \in V(\phi)$, that is, if and only if $M \models_X \phi$;

2. $M^{TL} \models_{X \rightarrow S} (\psi_1)^{TL} \otimes (\psi_2)^{TL}$ if and only if $X = X_1 \cup X_2$ for two $X_1, X_2 \subseteq S$ such that $M^{TL} \models_{X_1 \rightarrow S} (\psi_1)^{TL}$ and $M^{TL} \models_{X_2 \rightarrow S} (\psi_2)^{TL}$. 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 $M \models_X \psi_1 \lor \psi_2$.

3. $M^{TL} \models_{X \rightarrow S} (\psi_1)^{TL} \land (\psi_2)^{TL}$ if and only if $M^{TL} \models_{X \rightarrow S} (\psi_1)^{TL}$ and $M^{TL} \models_{X \rightarrow S} (\psi_2)^{TL}$, that is, by induction hypothesis, if and only if $M \models_X \psi_1 \land \psi_2$.

4. $M^{TL} \models_{X \rightarrow S} \exists v (\psi)^{TL}$ if and only if there exists a $Y$ such that $Y \supseteq X [F/v]$ for some $F$ and $M^{TL} \models_{Y \rightarrow S} \psi$. 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. $M^{TL} \models_{X \rightarrow S} \forall v (\psi)^{TL}$ if and only if $M^{TL} \models_{Y \rightarrow S} (\psi)^{TL}$ 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 $M \models_X \forall v \psi$.

$\blacksquare$
Chapter 6. Transition Dynamics

Representing Transition Models, game terms and formulas in Dependence Logic is somewhat more complex:

**Definition 6.2.14.** 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\(^5\) \( S \uplus \biguplus \{I_t : t \in \Theta\} \uplus \biguplus \{J_p : p \in \Phi\} \) whose signature contains

- For every \( t \in \Theta \), a ternary relation \( R_t \) whose interpretation is \( \{(i, x, y) : i \in I_t, x \in X_i, y \in Y_i\} \);
- For every \( p \in \Phi \), a binary relation \( V_p \) whose interpretation is \( \{(j, x) : j \in J_p, x \in X_j\} \).

**Definition 6.2.15.** For any formula \( \phi \), transition term \( \tau \), variable \( x \) and unary relation symbol \( P \) the Dependence Logic formulas \( \phi^{DL}_x \) and \( \tau^{DL}_x(P) \) are defined as follows:

1. \( \top^{DL}_x \) is \( \top \);
2. For all \( p \in \Phi \), \( p^{DL}_x \) is \( \exists j(=j \land V_p(j, x)) \);
3. \( (\psi_1 \lor \psi_2)^{DL}_x \) is \( (\psi_1)^{DL}_x \sqcup (\psi_2)^{DL}_x \), where \( \sqcup \) is the classical disjunction introduced in Definition 2.2.4;
4. \( (\psi_1 \land \psi_2)^{DL}_x \) is \( (\psi_1)^{DL}_x \land (\psi_2)^{DL}_x \);
5. \( (\langle \tau \rangle^y)^{DL}_x \) is \( \exists P((\tau)^{DL}_x(P) \land \forall y(\neg P y \lor (\psi)^{DL}_y)) \), where the second-order existential quantifier is a shorthand for the construction described in Definition 2.2.16 and \( y \) is a new and unused variable;

- For all \( t \in \Theta \), \( t^{DL}_x(P) \) is \( \exists i(=i \land \exists y(R_i(i, x, y)) \land \forall y(\neg R_i(i, x, y) \lor P y)) \);
- For all formulas \( \phi \), \( (\phi?)^{DL}_x(P) \) is \( \phi^{DL}_x \land Px \);
3. \( (\tau_1 \otimes \tau_2)^{DL}_x(P) = (\tau_1)^{DL}_x(P) \lor (\tau_2)^{DL}_x(P) \);
4. \( (\tau_1 \land \tau_2)^{DL}_x(P) = (\tau_1)^{DL}_x(P) \land (\tau_2)^{DL}_x(P) \);
5. \( (\tau_1; \tau_2)^{DL}_x(P) = \exists Q((\tau_1)^{DL}_x(Q) \land \forall y(\neg Q y \lor (\tau_2)^{DL}_y(P))) \), where \( y \) a new and unused variable.

\(^5\)Here we write \( A \uplus B \) for the disjoint union of the sets \( A \) and \( B \).
Theorem 6.2.16. For all transition models $T = (S, (\theta_t : t \in \Theta), V)$, transition terms $\tau$, transition formulas $\phi$, variables $x$, sets $P \subseteq S$ and teams $X$ over $T^{DL}$,

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

and

$$T^{DL} \models_X \tau^x_w(P) \iff T \models_{X(x) \rightarrow P} \tau.$$

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 $T^{DL} \models_X \exists j (\neg (j) \land V_p(j, x))$. 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 2.2.5, $T^{DL} \models_X (\psi_1 \lor \psi_2)^x$ if and only if $T^{DL} \models_X (\psi_1)^x$ or $T^{DL} \models_X (\psi_2)^x$. 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 $T \models_{X(x)} \psi_1 \lor \psi_2$.

4. $T^{DL} \models_X (\psi_1 \land \psi_2)^x$ if and only if $T^{DL} \models_X (\psi_1)^x$ and $T^{DL} \models_X (\psi_2)^x$, that is, by induction hypothesis, if and only if $T \models_{X(x)} \psi_1 \land \psi_2$.

5. $T^{DL} \models_X ((\tau)\psi)^x$ if and only if there exists a $P$ such that $T^{DL} \models_X (\tau)^x_w(P)$ and $T^{DL} \models_{X[T^{DL}/y]} \neg Py \lor (\psi)^y_w$. By induction hypothesis, the first condition holds if and only if $T \models_{X(x)} \neg \tau$. 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 $T^{DL} \models_{Y_2} \tau_0(\psi)$. 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 $T \models_{X(x)} (\tau)\psi$.

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

Now let us consider the cases corresponding to transition terms:
1. Suppose that $T^{DL} \models X \exists i (=i) \land \exists y (R_t(i, x, y)) \land \forall y (\neg R_t(i, x, y) \lor P y))$.

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 $m \in \text{Dom}(T^{DL})$ such that

- There exists a $F$ such that $T^{DL} \models X[m/i][F/y] R_t(i, x, y)$;
- $T^{DL} \models X[m/i][T^{DL}/y] \neg R_t(i, x, y) \lor Py$.

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, $(X(x), P) \in \theta_t$, and $T \models X(x) \rightarrow_p t$, as required.

Conversely, suppose that $(X(x), P) = (X_m, Y_m) \in \theta_t$ for some $m \in I$. If $X(x) = \emptyset$ then $X = \emptyset$, and hence by Proposition 2.2.6 we have that $T^{DL} \models X i^t_x (P)$, 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 $M \models X[m/i][T^{DL}/y] \neg R_t(i, x, y) \lor Py$. So, in conclusion, $M \models X t^x_x (P)$, as required.

2. $T^{DL} \models X \phi_x^{DL} \land Px$ if and only if $T \models X(x) \phi$ and $X(x) \subseteq P$, that is, if and only if $T \models X(x) \rightarrow_p \phi$.

3. $T^{DL} \models X (\tau_1)_x^{DL}(P) \lor (\tau_2)_x^{DL}(P)$ 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 $X(x) = X_1(x) \cup X_2(x)$;
- $T^{DL} \models X_1 (\tau_1)_x^{DL}(P)$, that is, by induction hypothesis, $T \models X_1(x) \rightarrow_p \tau_1$;
- $T^{DL} \models X_2 (\tau_2)_x^{DL}(P)$, that is, by induction hypothesis, $T \models X_2(x) \rightarrow_p \tau_2$. 

Hence, if $T_{DL}^D \models X (\tau_1 \otimes \tau_2)^{DL}_x(P)$ then $T \models X(x) \rightarrow_P \tau_1 \otimes \tau_2$.

Conversely, if $X(x) = A \cup B$ for two $A$, $B$ such that $T \models A \rightarrow_P \tau_1$ and $T \models B \rightarrow_P \tau_2$, let

\begin{align*}
X_1 &= \{ s \in X : s(x) \in A \} \\
X_2 &= \{ s \in X : s(x) \in B \}.
\end{align*}

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

4. $T_{DL}^D \models X (\tau_1 \cap \tau_2)^{DL}_x(P)$ if and only if $T_{DL}^D \models X (\tau_1)^{DL}_x(P)$ and $T_{DL}^D \models X (\tau_2)^{DL}_x(P)$, that is, by induction hypothesis, if and only if $T \models X(x) \rightarrow_P \tau_1 \cap \tau_2$.

5. $T_{DL}^D \models X \exists Q((\tau_1)^{DL}_x(Q) \land \forall y(\neg Q \land \forall x (\tau_2)^{DL}_x(P)))$ if and only if there exists a $Q$ such that $T \models X(x) \rightarrow_Q \tau_1$ and there exists a $Q' \geq Q$ such that $T \models Q' \rightarrow_P \tau_2$. By downwards closure, if this is the case then $T \models Q \rightarrow_P \tau_2$ too, and hence $T \models X(x) \rightarrow_P \tau_1 \cap \tau_2$, as required.

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

\begin{align*}
Z_1 &= \{ s \in X[T_{DL}^D/y] : s(y) \notin Q \} \\
Z_2 &= \{ s \in X[T_{DL}^D/y] : s(y) \in Q \}
\end{align*}

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

The significance of the results of this subsection is comparable to that of the corresponding ones about First Order Logic which we recalled in Subsection 6.1.2. In brief, what Theorems 6.2.13 and 6.2.16 tell us is that it is possible to understand Dependence Logic as a language for reasoning about imperfect information decision problems!

In the rest of this chapter, we will examine how this insight may be used in order to further the study of Dependence Logic and its variants.
6.2.3 Transition Dependence Logic

Just as van Benthem’s theorems, which we recalled in Subsection 6.1.2, allows one to reinterpret First Order Logic as a logic of perfect information two-player games, Theorems 6.2.13 and 6.2.16 of Subsection 6.2.2 permit us to understand Dependence Logic as a logic of imperfect-information decision problems.

However, the language of Dependence Logic, in itself, does very little to support this interpretation. We may certainly associate a Dependence Logic sentence such as, for example, $\forall x \exists y (= (y, f(x)) \land P xy)$, to a certain game of imperfect information, and then establish a correspondence between the truth of the sentence and certain abilities of an agent in this game; but this interpretation - legitimate though it may be from a semantic perspective - does not appear to arise entirely naturally from the syntactical structure of the sentence.

But by exploiting of the representation of Dependence Logic inside of Transition Logic of Definitions 6.2.11 and 6.2.12, it is not difficult to define a variant of Dependence Logic in which this interpretation is manifested at the syntactical level itself:

**Definition 6.2.17.** Let $\Sigma$ 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

$$\tau ::= \exists v \mid \forall v \mid \phi? \mid \tau \otimes \tau \mid \tau \cap \tau \mid \tau; \tau$$

$$\phi ::= R\vec{t} \mid \neg R\vec{t} \mid (= (t_1 \ldots t_n) \mid \phi \lor \phi \mid \phi \land \phi \mid \langle \tau \rangle \phi).$$

where $v$ ranges over all variables in $\text{Var}$, $R$ 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 6.2.18.** Let $M$ be a first-order model, let $\tau$ 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 $\tau$ in $M$, and we write $M \models_{X \rightarrow Y} \tau$, if and only if

- **TDL-$\exists$:** $\tau$ is of the form $\exists v$ for some $v \in \text{Var}$ and there exists a $F$ such that $X[F/v] \subseteq Y$;
- **TDL-$\forall$:** $\tau$ is of the form $\forall v$ for some $v \in \text{Var}$ and $X[M/v] \subseteq Y$;
- **TDL-test:** $\tau$ is of the form $\phi?$, $M \models_X \phi$, and $X \subseteq Y$;
- **TDL-$\otimes$:** $\tau$ 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 $M \models_{X_2 \rightarrow Y} \tau_2$;
6.2. Transition Logic

TDL-$\cap$: $\tau$ is of the form $\tau_1 \cap \tau_2$, $M \models X \to \tau_1$ and $M \models X \to \tau_2$;

TDL-concat: $\tau$ is of the form $\tau_1; \tau_2$ and there exists a team $Z$ such that
$M \models X \to Z \tau_1$ and $M \models Z \to Y \tau_2$.

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

TDL-lit: $\phi$ is a first-order literal and $M \models_s \phi$ in the usual first-order sense for all $s \in X$;

TDL-dep: $\phi$ is a dependence atom $= \langle t_1 \ldots t_n \rangle$ 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-$\lor$: $\phi$ is of the form $\phi_1 \lor \phi_2$ and $M \models X \phi_1$ or $M \models X \phi_2$;

TDL-$\land$: $\phi$ is of the form $\phi_1 \land \phi_2$, $M \models X \phi_1$ and $M \models X \phi_2$;

TDL-$\diamond$: $\phi$ is of the form $\langle \tau \rangle \psi$ and there exists a $Y$ such that $M \models X \to Y \tau$ and $M \models Y \psi$.

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 6.2.19.** For every Dependence Logic formula $\phi$ there exists a Transition Dependence Logic transition term $\tau_{\phi}$ such that

$$M \models X \phi \iff \exists Y \text{ s.t. } M \models X \to \tau_{\phi} \iff M \models_X \langle \tau_{\phi} \rangle \top$$

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

**Proof.** $\tau_{\phi}$ is defined by structural induction on $\phi$, as follows:

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

It is then trivial to verify, again by induction on $\phi$, that $M \models_X \phi$ if and only if $M \models_X \langle \tau_{\phi} \rangle \top$, as required. \qed
Theorem 6.2.20. For every Transition Dependence Logic formula $\phi$ there exists a Dependence Logic formula $\phi'$ such that

$$M \models_X \phi \iff M \models_X \phi'$$

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

Proof. (Sketch) Translate $\phi$ into $\Sigma_1$, and then apply Theorem 2.2.14.

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

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

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 refinement of Dependence Logic proper, even though it has exactly the same expressive power.

6.3 Dynamic Semantics

6.3.1 Dynamic Predicate Logic

Dynamic Semantics is an approach to the formal semantics of natural language which can be summarized by the following motto, from [34]:

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.

Whereas truth-theoretic semantics takes as its primary object of investigation the conditions under which a hypothetical listener would be willing to accept a statement as truthful, dynamic semantics takes as its fundamental semantic objects the informational changes that the utterance of a sentence has on the contexts under which further statements will be interpreted, as well as on the states of mind of hypothetical listeners.
This second approach has some advantages in formal linguistics, for example with respect to the interpretation of anaphora and "Donkey Sentences"; but as this work is not concerned with linguistic applications, and – more importantly – because the author is no linguist, we will not discuss the contribution of dynamic semantics to formal linguistics any further.

One thing worth pointing out, however, is that the dynamic approach and the truth-theoretic approach to semantics are not in competition. Rather, they are complementary: given a dynamic semantics, it is possible to recover the truth conditions by examining under which circumstances the interpretation of the formula leads to a state which accepts it, and, conversely, given a truth-theoretic semantics one can recover the dynamics hidden in it by comparing the truth conditions of an expression with the ones of its components.

We refer to van Benthem’s book [68], to Dekker’s paper [14] and to van Eijk’s summary [72] for a more thorough introduction to this interesting approach to semantics. Here we will just present, as an example of a dynamic semantics, the Dynamic Predicate Logic introduced in [34]:

**Definition 6.3.1.** Let $M$ be a first order model, let $\phi$ be a first order formula over its signature, and let $s$ and $s'$ be two assignments. Then we say that the transition from $s$ to $s'$ is *allowed* by $\phi$ in $M$, and we write $M \models_s s' \phi$, if and only if

- **DPL-atom:** $\phi$ is an atomic formula, $s = s'$ and $M \models_s \phi$ in the usual sense;
- **DPL-$\neg$:** $\phi$ is of the form $\neg \psi$, $s = s'$ and for all assignments $h$, $M \not\models_{s\rightarrow h} \psi$;
- **DPL-$\land$:** $\phi$ is of the form $\psi_1 \land \psi_2$ and there exists a $h$ such that $M \models_{s\rightarrow h} \psi_1$ and $M \models_{h\rightarrow s'} \psi_2$;
- **DPL-$\lor$:** $\phi$ is of the form $\psi_1 \lor \psi_2$, $s = s'$ and there exists a $h$ such that $M \models_{s\rightarrow h} \psi_1$ or $M \models_{s\rightarrow h} \psi_2$;
- **DPL-$\rightarrow$:** $\phi$ is of the form $\psi_1 \rightarrow \psi_2$, $s = s'$ and for all $h$ it holds that $M \models_{s\rightarrow h} \psi_1 \Rightarrow \exists h' \text{ s.t. } M \models_{h\rightarrow h'} \psi_2$;
- **DPL-$\exists$:** $\phi$ is of the form $\exists x \psi$ and there exists an element $m \in \text{Dom}(M)$ such that $M \models_{s[m/x]} \psi$;
- **DPL-$\forall$:** $\phi$ is of the form $\forall x \psi$, $s = s'$ and for all elements $m \in \text{Dom}(M)$ there exists a $h$ such that $M \models_{s[m/x]\rightarrow h} \psi$. \


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

We will not examine in any detail this semantics or its applications, as this of little relevance for our purposes here. However, what is worth pointing out is that according to it, formulas are interpreted not as sets of assignments, as in the case of Tarski’s semantics for First Order Logic, but rather as sets of transitions from assignments to assignments. For example, an atomic formula $P\vec{t}$ is interpreted as a test, which allows a transition $(s, s)$ if and only if the assignment $s$ satisfies $P\vec{t}$; and instead, an existential quantification $\exists x \psi$ corresponds to a transition in which first we change the value of the variable $x$, and then we execute $\psi$. Of special interest is the conjunction $\psi_1 \land \psi_2$, which is interpreted as a concatenation of transitions: this, combined with the semantics for existential quantification, makes it so that $(\exists x \psi) \land \theta$ is logically equivalent to $\exists x (\psi \land \theta)$, differently from the case of standard First Order Logic.

Furthermore, satisfaction in this semantics is a derived property, to be understood in terms of reachability: an assignment $s$ satisfies a formula $\phi$ if and only if there exists some $s'$ such that $\phi$ allows the transition from $s$ to $s'$, that is, if and only if $s$ is not a “dead end” for $\phi$.

Even more interestingly, it is not difficult to see that in this approach to semantics, it is possible to interpret the existential quantifier as an atomic formula! Indeed, if we define

$$\text{DPL-}\exists\text{-atom } M \models_{s \rightarrow s'} \exists x \text{ if and only if there exists a } m \in \text{Dom}(M) \text{ such that } s' = s[m/x]$$

then it is easy to verify that $\exists x \phi$ is equivalent to $\exists x \land \phi$. In other words, we can isolate the semantic contribution of the existential quantifier itself, much as we did for the case of Transition Dependence Logic!

The same cannot be said, however, for the universal quantifier: it is easy to see that there exists no semantics for $\forall x$ in this framework which makes $\forall x \psi$ equivalent to $\forall x \land \psi$. The problem, of course, is that in order to verify the truth of $\forall x \psi$, we need to examine $\psi$ with respect to multiple assignments, while, according to the above rules, a conjunction $\forall x \land \psi$ allows a transition.

As an aside, the fact that in Dynamic Predicate Logic existential quantifiers can have an effect even beyond their syntactic scope was one of the main reasons why this semantics can be used to interpret natural language statements in which pronouns refer to nouns which lie beyond their apparent scopes, as in the famous example

$$(A \text{ man})_1 \text{ walks in the park. (He)}_1 \text{ whistles.}$$

We refer to [34] for further details.
from \( s \) to \( s' \) if and only if there exists at least one \( s'' \) such that \( \forall x \) allows the transition from \( s \) to \( s'' \) and \( \psi \) allows the transition from \( s'' \) to \( s' \).

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:

\[
\text{The meaning of a formula does not lie in its satisfaction conditions,}
\text{but rather in the team transitions it allows.}
\]

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 \( \tau \) if and only if there exists a \( Y \) such that \( \tau \) 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 set the initial positions \( X \) in the game corresponding to \( \tau \).

In the next section, we will make use of these intuitions to develop another, terser version of Dependence Logic; and finally, in Subsection 6.3.3 we will come full circle by showing how the semantics of this logic can be interpreted in terms of reachability conditions in a suitable variant of the Game Theoretic Semantics for standard Dependence Logic.

### 6.3.2 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.

Apart from Dynamic Predicate Logic, our treatment will be also inspired by Abramsky’s Game Semantics for multigent logics of imperfect information [1, 2]; this will be particularly evident in the next subsection, in which we will develop a Game Theoretic Semantics for our logic.

**Definition 6.3.2.** Let \( \Sigma \) be a first-order signature. The set of all transition formulas of Dynamic Dependence Logic over \( \Sigma \) 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 \( \text{Var} \).
The semantical rules associated to this language are precisely as one would expect:

**Definition 6.3.3.** Let $M$ be a first-order model, let $\tau$ be a transition formula of Dynamic Dependence Logic over the signature of $M$, and let $X$ and $Y$ be two teams over $M$ with domain $\text{Var}$. Then we say that $\tau$ allows the transition $X \rightarrow Y$ in $M$, and we write $M \models_{X \rightarrow Y} \tau$, if and only if

- **DDL-lit:** $\tau$ 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:** $\tau$ 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-$\exists$:** $\tau$ is of the form $\exists v$ for some $v \in \text{Var}$, and $X[F/v] \subseteq Y$ for some $F : X \rightarrow \text{Dom}(M)$;
- **DDL-$\forall$:** $\tau$ is of the form $\forall v$ for some $v \in \text{Var}$, and $X[M/v] \subseteq Y$;
- **DDL-$\otimes$:** $\tau$ 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 $M \models_{X_1 \rightarrow Y} \tau_1$ and $M \models_{X_2 \rightarrow Y} \tau_2$;
- **DDL-$\cap$:** $\tau$ is of the form $\tau_1 \cap \tau_2$, and there exists a $Z$ such that $M \models_{X \rightarrow Z} \tau_1$ and $M \models_{Z \rightarrow Y} \tau_2$;
- **DDL-concat:** $\tau$ is of the form $\tau_1 ; \tau_2$, and there exists a $Z$ such that $M \models_{X \rightarrow Z} \tau_1$ and $M \models_{X \rightarrow Y} \tau_2$.

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

It is not difficult to see that Dynamic Dependence Logic is equivalent to Transition Dependence Logic (and, therefore, to Dependence Logic).

**Proposition 6.3.4.** Let $\phi$ be a Dependence Logic formula. Then there exists a term $\phi'$ of Dynamic Dependence Logic which is equivalent to it, in the sense that $M \models_X \phi \Leftrightarrow M \models_X \phi' \Leftrightarrow \exists Y$ s.t. $M \models_{X \rightarrow Y} \phi'$ for all suitable teams $X$ and models $M$.

**Proof.** We build $\phi'$ by structural induction:

1. If $\phi$ is a literal or a dependence atom then $\phi' = \phi$;
2. If $\phi$ is $\psi_1 \lor \psi_2$ then $\phi' = \psi'_1 \otimes \psi'_2$;
6.3. Dynamic Semantics

3. If $\phi$ is $\psi_1 \land \psi_2$ then $\phi' = \psi_1' \land \psi_2'$;
4. If $\phi$ is $\exists x \psi$ then $\phi' = \exists x; \psi'$;
5. If $\phi$ is $\forall x \psi$ then $\phi' = \forall x; \psi'$.

**Proposition 6.3.5.** Let $\tau$ be a Dynamic Dependence Logic term. Then there exists a Transition Dependence Logic term $\tau'$ such that

$$M \models_{X \rightarrow Y} \tau \iff M \models_{X \rightarrow Y} \tau'$$

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

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

**Proof.** Build $\tau'$ by structural induction:

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

**Corollary 6.3.6.** 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.

6.3.3 Game Theoretic Semantics for Dynamic Dependence Logic

In this subsection, we will adapt the Game Theoretic Semantics of Subsection 2.2.3 to the case of Dynamic Dependence Logic.

**Definition 6.3.7.** Let $\tau$ be any Dynamic Dependence Logic formula. Then $\text{Player}(\tau) \in \{E, A\}$ is defined as follows:
1. If \( \tau \) is a first-order literal or a dependence atom, \( \text{Player}(\tau) = E \);
2. If \( \tau \) is of the form \( \tau_1 \otimes \tau_2 \) or \( \exists v \) then \( \text{Player}(\tau) = E \);
3. If \( \tau \) is of the form \( \tau_1 \cap \tau_2 \) or \( \forall v \) then \( \text{Player}(\tau) = A \);
4. If \( \tau \) is of the form \( \tau_1 ; \tau_2 \) then \( \text{Player}(\tau) = \text{Player}(\tau_1) \).

Positions of our game will be pairs \((\tau, s)\), where \( \tau \) is a transition term and \( s \) is an assignment. The successors of a given position are defined as follows:

Definition 6.3.8. Let \( M \) be a first order model, let \( \tau \) be a transition term and let \( s \) be an assignment over \( M \). Then the set \( \text{Succ}_M(\tau, s) \) of the successors of the position \((\tau, s)\) is defined as follows:

1. If \( \tau \) is a first order literal \( \phi \) then
   \[
   \text{Succ}_M(\tau, s) = \begin{cases} 
   \{(\lambda, s)\} & \text{if } M \models \alpha \text{ in First Order Logic;} \\
   \emptyset & \text{otherwise}
   \end{cases}
   \]
   where \( \lambda \) stands for the empty string;
2. If \( \tau \) is a dependence atom then \( \text{Succ}_M(\tau, s) = \{(\lambda, s)\} \);
3. If \( \tau \) is of the form \( \exists v \) or \( \forall v \) then \( \text{Succ}(\tau, s) = \{(\lambda, s[m/v]) : m \in \text{Dom}(M)\} \);
4. If \( \tau \) is of the form \( \tau_1 \otimes \tau_2 \) or \( \tau_1 \cap \tau_2 \) then \( \text{Succ}_M(\tau, s) = \{(\tau_1, s), (\tau_2, s)\} \);
5. If \( \tau \) is of the form \( \tau_1 ; \tau_2 \) then
   \[
   \text{Succ}_M(\tau, s) = \{(\tau', \tau_2, s') : (\tau', s') \in \text{Succ}_M(\tau_1, s)\}
   \]
   where, with an abuse of notation, we assume that \( \lambda; \tau_2 \) is equal to \( \tau_2 \).

We can now define formally the semantic games associated to Dynamic Dependence Logic formulas:

Definition 6.3.9. Let \( M \) be a first-order model, let \( \tau \) be a Dynamic Dependence Logic formula, and let \( X \) and \( Y \) be teams. Then the game \( G^M_{X \rightarrow Y}(\tau) \) is defined as follows:

- The set \( I \) of the initial positions of the game is \( \{(\tau, s) : s \in X\} \);
- The set \( W \) of the winning positions of the game is \( \{(\lambda, s) : s \in Y\} \);
- For any position \((\tau', s')\), the active player is \( \text{Player}(\tau') \) and the set of successors is \( \text{Succ}_M(\tau', s') \).
6.3. Dynamic Semantics

Definition 6.3.10. Let $G^M_{X \rightarrow Y}(\tau)$ be as in the above definition. Then a play of this game is a finite sequence $\vec{p} = p_1 \ldots p_n$ of positions of the game such that

1. $p_1 \in I$ is an initial position of the game;
2. For every $i \in 1 \ldots n - 1$, $p_{i+1} \in \text{Succ}_M(p_i)$.

If furthermore $\text{Succ}_M(p_n) = \emptyset$, we say that $\vec{p}$ is complete; and if $p_n \in W$ is a winning position, we say that $\vec{p}$ is winning.

So far, we did not deal with the satisfaction conditions of dependence atoms at all. Similarly to the case of Definition 2.2.21, such conditions are made to correspond as uniformity conditions over sets of plays:

Definition 6.3.11. Let $G^M_{X \rightarrow Y}(\tau)$ be a game, and let $P$ be a set of plays in it. Then $P$ is uniform if and only if for all $\vec{p}, \vec{q} \in P$ and for all $i,j$, such that $p_i$ is of the form $((= (t_1 \ldots t_n); \tau_1); \ldots \tau_k), s)$ and $q_j$ is of the form $(((t_1 \ldots t_n); \tau_1); \ldots \tau_k), s')$ for the same instance of the dependence atom $= (t_1 \ldots t_n)$ it holds that

$$(t_1 \ldots t_{n-1}) \langle s \rangle = (t_1 \ldots t_{n-1}) \langle s' \rangle \Rightarrow t_n \langle s \rangle = t_n \langle s' \rangle$$

where, with an abuse of notation, we identify $= (t_1 \ldots t_n); \lambda$ with $= (t_1 \ldots t_n)$.

As always, we will only consider positional strategies, that is, strategies that depend only on the current position.

Definition 6.3.12. Let $G^M_{X \rightarrow Y}(\tau)$ be as above, and let $\tau'$ be any expression such that $(\tau', s')$ is a possible position of the game for some $s'$. Then a local strategy for $\tau'$ is a function $f_{\tau'}$ sending each $s'$ into a $((\tau'', s''), s)$ in $\text{Succ}_M(\tau', s')$.

Definition 6.3.13. Let $G^M_{X \rightarrow Y}(\tau)$ be as above, let $\vec{p} = p_1 \ldots p_n$ be a play in it, and let $f_{\tau'}$ be a local strategy for some $\tau'$. Then $\vec{p}$ is said to follow $f_{\tau'}$ if and only if for all $i \in 1 \ldots n - 1$ and all $s'$,

$$p_i = (\tau', s') \Rightarrow p_{i+1} = f_{\tau'}(s').$$

Definition 6.3.14. Let $G^M_{X \rightarrow Y}(\tau)$ be as above. Then a global strategy (for $E$) in this game is a function $f$ associating to each expression $\tau'$ occurring in some nonterminal position of the game and such that $\text{Player}(\tau') = E$ with some local strategy $f_{\tau'}$ for $\tau'$.

\footnote{As a limit case for $k = 0$, this condition also applies to $p_i = (= (t_1 \ldots t_k), s)$ and $q_j = (= (t_1 \ldots t_k), s')$.}
Definition 6.3.15. A play $\vec{p}$ of a game $G^M_X \rightarrow Y(\tau)$ is said to follow a global strategy $f$ if and only if it follows $f_{\tau'}$ for all $\tau'$.

Definition 6.3.16. A global strategy $f$ for a game $G^M_X \rightarrow Y(\tau)$ is said to be winning if and only if all complete plays which follow $f$ are winning.

Definition 6.3.17. A global strategy $f$ for a game $G^M_X \rightarrow Y(\tau)$ is said to be uniform if and only if the set of all complete plays which follow $f$ respects the uniformity condition of Definition 6.3.11.

The following result then connects the Game Theoretic Semantics we just defined and the Team Transition Semantics for Dynamic Dependence Logic:

Theorem 6.3.18. Let $M$ be a first-order model, let $X$ and $Y$ be teams, and let $\tau$ be any Dynamic Dependence Logic transition term. Then $M \models X \rightarrow Y \tau$ if and only if the existential player $E$ has a uniform winning strategy for $G^M_X \rightarrow Y(\tau)$.

Proof. The proof is by structural induction on $\tau$:

1. If $\tau$ is a first-order literal and $M \models X \rightarrow Y \tau$, then $X \subseteq Y$ and $M \models s \tau$ in the usual first-order sense for all $s \in X$. Then there exists only one strategy $f$ for $E$ in $G^M_X \rightarrow Y(\tau)$, and for this strategy we have that $f_{\tau}(s) = (\lambda, s)$ for all $s \in X$. Since $X \subseteq Y$, this strategy is winning; and furthermore, it is trivially uniform. Hence, $E$ has a uniform winning strategy in $G^M_X \rightarrow Y(\tau)$.

Conversely, suppose that $E$ has an uniform winning strategy $f$ in $G^M_X \rightarrow Y(\tau)$. If $M \not\models s \tau$ for some $s \in X$, then the position $(\tau, s)$ is terminal in this game and it is not winning, which contradicts our hypothesis. Hence $M \models s \tau$ for all $s \in X$, and furthermore - since $f_{\tau}(s) = (\lambda, s)$ for all $s \in X$ - we have that $X \subseteq Y$. Thus, $M \models X \rightarrow Y \tau$, as required.

2. If $\tau$ is a dependence atom $= (t_1, \ldots, t_n)$ and $M \models X \rightarrow Y \tau$, then $X \subseteq Y$ and $X$ satisfies the dependency condition associated with the atom. But then the only strategy $f$ available to player $E$ is winning, as the set of its terminal position is $\{(\lambda, s) : s \in X\}$ and $X \subseteq Y$, and it is also uniform, since the set of all possible plays of the game is $\{(= (t_1, \ldots, t_n), s) : s \in X\}$.

Conversely, suppose that the only strategy $f$ available to $E$ in $G^M_X \rightarrow Y(\tau)$ is uniform and winning. Since it is uniform, it follows at once that $X$ satisfies the dependency condition; and since it is winning and the set of all terminal positions is $\{(\lambda, s) : s \in X\}$, we have that $X \subseteq Y$, and hence that $M \models X \rightarrow Y \tau$. 


3. If \( \tau \) is \( \exists v \) for some variable \( v \) and \( M \models X \rightarrow Y \), then \( X[F/v] \subseteq Y \) for some \( F : X \rightarrow \text{Dom}(M) \). Now let \( f \) be the strategy for \( E \) in \( G^M_{X \rightarrow Y}(\tau) \) such that \( f_\tau(s) = s[F(s)/v] \): this strategy is uniform, and the set of its terminal positions is \( \{(\lambda, s[F(s)/v]) : s \in X\} = \{(\lambda, s') : s' \in X[F/v]\} \). Hence, \( f_\tau \) is also winning, as required.

Conversely, let \( f \) be any uniform winning strategy for \( E \) in \( G^M_{X \rightarrow Y}(\tau) \), and define \( F : X \rightarrow \text{Dom}(M) \) so that

\[
f_\tau(s) = (\lambda, s[F(s)/v])
\]

for all \( s \in X \).

Since \( f \) is winning, \( f_\tau(s) \) is a winning position for all \( s \in X \), and hence \( X[F/v] = \{s[F(s)/v] : s \in X\} \subseteq Y \). Hence, \( M \models X \rightarrow Y \), as required.

4. If \( \tau \) is \( \forall v \) for some variable \( v \) and \( M \models X \rightarrow Y \), then \( X[M/v] \subseteq Y \). There exists only one (trivial, and trivially uniform) strategy for \( E \) in the game \( G^M_{X \rightarrow Y}(\tau) \), as the universal player \( A \) moves in all non-terminal positions; and the set of all possible outcomes of the game for all the initial positions is \( \{(\lambda, s[m/v]) : s \in X, m \in \text{Dom}(M)\} = \{(\lambda, s') : s' \in X[M/v]\} \). Hence this strategy is winning, as required.

Conversely, suppose that the unique strategy for \( E \) in \( G^M_{X \rightarrow Y}(\tau) \) is winning for this game. Then, as the set of all possible outcomes is \( \{(\lambda, s') : s' \in X[M/v]\} \), we have that \( X[M/v] \subseteq Y \), and hence that \( M \models X \rightarrow Y \).

5. If \( \tau \) is \( \tau_1 \oplus \tau_2 \) for two transition terms \( \tau_1 \) and \( \tau_2 \) and \( M \models X \rightarrow Y \), then \( X = X_1 \cup X_2 \) for two \( X_1 \) and \( X_2 \) such that \( M \models X_1 \rightarrow Y \tau_1 \) and \( M \models X_2 \rightarrow Y \tau_2 \). By induction hypothesis, this implies that there exist two strategies \( f_1 \) and \( f_2 \) for \( E \) which are uniform and winning in \( G^M_{X_1 \rightarrow Y}(\tau_1) \) and \( G^M_{X_2 \rightarrow Y}(\tau_2) \) respectively. Now define a strategy \( f \) for \( E \) in \( G^M_{X \rightarrow Y}(\tau_1 \oplus \tau_2) \) as follows:

- If \( \tau' \) is part of \( \tau_1 \) then \( f_\tau'(s) = (f_1)_\tau'; \)
- If \( \tau' \) is part of \( \tau_2 \) then \( f_\tau'(s) = (f_2)_\tau'; \)
- If \( \tau' \) is \( \tau_1 \oplus \tau_2 \) then \( f_\tau'(s) = \begin{cases} (\tau_1, s) & \text{if } s \in X_1; \\ (\tau_2, s) & \text{if } s \in X_2 \setminus X_1. \end{cases} \)

This strategy is clearly uniform, as \( f_1 \) and \( f_2 \) are uniform. Furthermore, it is winning: indeed, any play of \( G^M_{X \rightarrow Y}(\tau_1 \oplus \tau_2) \) in which \( E \) follows it strictly contains a play of \( G^M_{X_1 \rightarrow Y}(\tau_1) \) in which \( E \) follows \( f_1 \) or a play of \( G^M_{X_2 \rightarrow Y}(\tau_2) \) in which \( E \) follows \( f_2 \), and in either case the game ends in a winning position in \( Y \).
Conversely, suppose that $f$ is a uniform winning strategy for $E$ in $G_{X_1 \rightarrow Y}(\tau)$. Now let $X_1 = \{ s \in X : f_1(s) = (\tau_1, s) \}$, let $X_2 = \{ s \in X : f_2(s) = (\tau_2, s) \}$, and let $f_1$ and $f_2$ be the restrictions of $f$ to the subgames corresponding to $\tau_1$ and $\tau_2$ respectively. Then $f_1$ and $f_2$ are uniform and winning for $G_{X_1 \rightarrow Y}(\tau_1)$ and $G_{X_2 \rightarrow Y}(\tau_2)$ respectively, and hence by induction hypothesis $M \models X_1 \rightarrow Y \tau_1$ and $M \models X_2 \rightarrow Y \tau_2$. But $X = X_1 \cup X_2$, and hence this implies that $M \models X \rightarrow Y \tau$.

6. If $\tau$ is $\tau_1 \cap \tau_2$ for some $\tau_1$ and $\tau_2$ and $M \models X \rightarrow Y \tau_1 \cap \tau_2$, then $M \models X \rightarrow Y \tau_1$ and $M \models X \rightarrow Y \tau_2$. By induction hypothesis, this implies that $E$ has two uniform winning strategies $f_1$ and $f_2$ for $G_{X_1 \rightarrow Y}(\tau_1)$ and $G_{X_2 \rightarrow Y}(\tau_2)$ respectively. Now let $f$ be the strategy for $G_{X_1 \rightarrow Y}(\tau_1 \cap \tau_2)$ which behaves like $f_1$ over the subgame corresponding to $\tau_1$ and like $f_2$ over the subgame corresponding to $\tau_2$ (it is not up to $E$ to choose the successors of the initial positions $(\tau_1 \cap \tau_2, s)$, so she needs not specify a strategy for those). This strategy is winning and uniform, as required, because $\tau_1$ and $\tau_2$ are so.

Conversely, suppose that $E$ has a uniform winning strategy $f$ for $G_{X_1 \rightarrow Y}(\tau_1 \cap \tau_2)$. Since the opponent $A$ chooses the successor of the initial positions $(\tau_1 \cap \tau_2, s) : s \in X$, any element of $\{(\tau_1, s) : s \in X\}$ and of $\{(\tau_2, s) : s \in X\}$ can occur as part of a play in which $E$ follows $f$. Now, let $f_1$ and $f_2$ be the restrictions of $f$ to the subgames corresponding to $\tau_1$ and $\tau_2$ respectively: then $f_1$ and $f_2$ are uniform, because $f$ is so, and they are winning for $G_{X_1 \rightarrow Y}(\tau_1)$ and $G_{X_2 \rightarrow Y}(\tau_2)$ respectively, because every play of these games in which $E$ follows $f_1$ (resp $f_2$) starting from a position $(\tau_1, s)$ (resp. $(\tau_2, s)$) for $s \in X$ can be transformed into a play of $G_{X_1 \rightarrow Y}(\tau_1 \cap \tau_2)$ in which $E$ follows $f$ simply by appending the initial position $(\tau_1 \cap \tau_2, s)$ at the beginning.

7. If $\tau$ is $\tau_1 \cap \tau_2$ for some $\tau_1$ and $\tau_2$ and $M \models X \rightarrow Y \tau_1 \cap \tau_2$, then there exists a $Z$ such that $M \models X \rightarrow Z \tau_1$ and $M \models Z \rightarrow Y \tau_2$. By induction hypothesis, this implies that there exist two strategies $f_1$ and $f_2$ which are winning for $E$ in $G_{X_1 \rightarrow Z}(\tau_1)$ and in $G_{Z_2 \rightarrow Y}(\tau_2)$ respectively. Now define a strategy $f$ for $E$ in $G_{X_1 \rightarrow Y}(\tau_1 \cap \tau_2)$ as

- If $(f_1)_\tau(s') = (\tau'', s'')$ then $f_1 ; \tau_2(s') = (\tau'' ; \tau_2, s'')$;
- If $\tau'$ is part of $\tau_2$ then $f_\tau_2 = (f_2)_\tau_2$.

We need to prove that this $f$ is uniform and winning. Now, let us consider the set of all plays in which $E$ follows $f$: it is easy to see that they will be played exactly as a game of $G_{X_1 \rightarrow Z}(\tau_1)$ until a position of the form
(τ₂, s') is reached for some s' ∈ Z, and then they will be played exactly as a game of G⁵MUX→Y(τ₁; τ₂) until a position of the form (λ, s'') is reached for some s'' ∈ Y. Hence, the strategy is winning, as it will always end in a winning position for Y, and it is uniform, because any violation of uniformity would also be a violation for f₁ or f₂.

Conversely, let f be a uniform winning strategy for E in G⁵MUX→Y(τ₁; τ₂), and let Z be the set of all assignments s such that the position (τ₂, s) occurs as part of some play of G⁵MUX→Y(τ₁; τ₂) in which E follows f. Furthermore, let the two strategies f₁ and f₂ for E in G⁵MX→Z(τ₁) and G⁵MZ→Y(τ₂) respectively be defined as

- If fₜ₁;ₜ₂(s') = (τ''; τ₂, s'') then (f₁)ₜ₁(s) = (τ'', s'');
- If τ' is part of τ₂ then (f₂)ₜ₂ = fₜ₂.

By construction and definition, it follows at once that τ₁ and τ₂ are uniform winning strategies for G⁵MX→Z(τ₁) and G⁵MZ→Y(τ₂) respectively. By induction hypothesis, this implies that M |= X→Z τ₁ and that M |= Z→Y τ₂, and finally that M |= X→Y τ₁; τ₂.

Theorem 6.3.18 shows that Dynamic Dependence Logic can be interpreted in terms of reachability: M |= X→Y τ if and only if, in the game corresponding to τ, the existential player can guarantee that the final assignment will be in Y whenever the initial assignment is in X. This corresponds exactly to the intuitions behind the notion of transition system which we introduced in Subsection 6.2.1, and further confirms that Dependence Logic and its variants are suitable frameworks for exploring decision-theoretic reasoning under imperfect information in a first-order setting.