Research Report 307

Directed Simulations

Natasha Kurtonina and Maarten de Rijke

RR307

Although negation-free languages are widely used in logic and computer science, relatively little is known about their expressive power. To address this issue we consider a kind of non-symmetric bisimulations called directed simulations, and use these to analyze the expressive power and model theory of negation-free modal and temporal languages. We first use them to obtain preservation, safety and definability results for a simple negation-free modal language. We then obtain analogous results for stronger negation-free languages. Finally, we extend our methods to deal with languages with non-boolean negation.

Department of Computer Science
University of Warwick
Coventry CV4 7AL
United Kingdom

June 1996

Directed Simulations

Natasha Kurtonina[1] and Maarten de Rijke[2]

[1] Center for Logic, Ghent University
Rozier 44, 9000 Ghent, Belgium
Natasha.Kurtonina@rug.ac.be

[2] Dept. of Computer Science, University of Warwick
Coventry CV4 7AL, England
mdr@dcs.warwick.ac.uk

Abstract

Although negation-free languages are widely used in logic and computer science, relatively little is known about their expressive power. To address this issue we consider a kind of non-symmetric bisimulations called directed simulations, and use these to analyze the expressive power and model theory of negation-free modal and temporal languages. We first use them to obtain preservation, safety and definability results for a simple negation-free modal language. We then obtain analogous results for stronger negation-free languages. Finally, we extend our methods to deal with languages with non-boolean negation.

1 Introduction

In many areas of computer science one finds logical formalisms that lack some or all of the standard boolean connectives 'and', 'or' and 'not.' In particular, negation-free logics are widely used in areas as diverse as semantics of programming and knowledge representation. In some applications boolean negation is unnatural [21]. Excluding boolean negation may improve the complexity of the satisfiability problem [7], and it may restore monotonicity of the semantic interpretation function [19].

Despite their wide applicability negation-free languages haven't been studied as extensively as languages with full boolean expressivity. We want to fill this gap by studying the expressive power of negation-free *modal* languages. Recently, these have attracted considerable attention, both at an applied and at a theoretical level; cf. [8, 11, 14]. Now, for modal languages with a full boolean repertoire *bisimulations* have proved to be an important tool in understanding their expressive power (cf. [9, 4, 1, 17, 15]). In this paper we develop analogous tools for negation-free modal languages. We a kind of non-symmetric simulations called *directed simulations* between transition systems that allow us to study the expressive power and develop the model theory of negation-free languages. As far as we know, this is the first paper to do so in a systematic way.

Our point of the departure is a simple negation-free modal language with boolean conjunction and disjunction, and $\Diamond$ and $\Box$; for this language we introduce directed simulations, and use these to arrive at results on expressiveness and definability. We then extend our ideas and techniques so as to cope with other negation-free description languages, including terminological logics, negation-free fragments of Since/Until logic, and feature logics. After that we adapt our methods to cope with languages containing non-boolean negation. We conclude with a summary and suggestions for further work.

2 Definitions

Negation-free $\Diamond$, $\Box$*-formulas* are built up from propositional variables p, q, $\ldots$, and the constants $\top$ and $\bot$, using boolean conjunction $\wedge$ and disjunction $\vee$, and the unary modal operators $\Diamond$ (diamond) and $\Box$ (box). We use $\mathcal{L}_{\Diamond,\Box}$ to denote this language, and $\mathcal{L}^{\neg}_{\Diamond,\Box}$ to denote $\mathcal{L}_{\Diamond,\Box}$ with boolean negation.

A *transition system* (or *model*) for $\mathcal{L}_{\Diamond,\Box}$ is a triple $M = (W, R, V)$, where W is a non-empty set of *states*, R is a binary relation on W, and V is a *valuation* on M, that is: a function assigning a subset of W to every proposition letter. We sometimes write $|M|$ to denote the domain of M.

The *satisfaction relation* is defined in the familiar way for the atomic case and for the boolean connectives $\wedge$ and $\vee$; observe that we can always interpret boolean negation on our models, even when it is not present in our language, For the modal connectives we put $M, w \models \Diamond\phi$ iff there exists w' such that Rww' and $M, w' \models \phi$; and $M, w \models \Box\phi$ iff for all w' such that Rww', $M, w' \models \phi$.

The *(negation-free) modal theory* of a state w is the set *nf-tp*$(w) = \{\phi \in \mathcal{L}_{\Diamond,\Box} \mid w \models \phi\}$. If we want to emphasize the transition system M in which w lives, we write *nf-tp*$_M(w)$.

Modal logic is just one of many possible description languages for specifying and constraining transition systems. We will encounter several languages in this paper, and we relate them all to first-order logic. To be precise, let $\mathcal{L}_1$ be the first-order language with unary predicate symbols corresponding to the proposition letters in $\mathcal{L}_{\Diamond,\Box}$, and with one binary relation symbol R. $\mathcal{L}_1$ is called the *correspondence language* for $\mathcal{L}_{\Diamond,\Box}$. $\mathcal{L}_1(x)$ denotes the set of all $\mathcal{L}_1$-formulas having one free variable x.

To view transition systems as $\mathcal{L}_1$-structures in the usual first-order sense, we use $V(p)$ to interpret the unary predicate symbol P that corresponds to p. The *standard translation* takes modal formulas ϕ to equivalent formulas

$ST_x(\phi)$ of $\mathcal{L}_1$. It maps proposition letters p onto unary predicate symbols Px, it commutes with the booleans, and the modal cases are

$$ST_x(\Diamond\phi) = \exists y\,(Rxy \wedge ST_y(\phi)) \text{ and } ST_x(\Box\phi) = \forall y\,(Rxy \to ST_y(\phi)).$$

For all transition systems M and states w we have $M, w \models \phi$ iff $M \models ST_x(\phi)[w]$, where the latter denotes first-order satisfaction of $ST_x(\phi)$ under the assignment of w to the free variable x of $ST_x(\phi)$. A modal formula ϕ is said to *correspond* to a first-order formula $\alpha(x)$ if $\models ST_x(\phi) \leftrightarrow \alpha(x)$.

3 Simulations for $\mathcal{L}_{\Diamond,\Box}$

In this section we adapt the notion of bisimulation to the setting of negation-free modal formulas. The resulting notion of directed simulations is then used to analyze the expressive power of negation-free formulas in three different ways: in terms of preservation, safety, and definability.

Definition 3.1 (Directed Modal Simulations) Let Z be a non-empty binary relation between two transition systems M and N, that is, $Z \subseteq |M| \times |N|$. Then Z is called a *directed (modal) simulation between M and N* if it satisfies the following clauses:

1. If wZv and p is a proposition letter such that $M, w \models p$, then $N, v \models p$.
2. If wZv and Rvv', then there exists w' in M such that Rww' and $w'Zv'$ (back).
3. If wZv and Rww', then there exists v' in N such that Rvv' and $w'Zv'$ (forth).

We write $Z : M \rightrightarrows N$ ($Z : M, w \rightrightarrows N, v$) to indicate that Z is a directed simulation between M and N (that links w to v).

A *(strong) bisimulation* is a directed simulation for which clause 1 above is an equivalence: if wZv then $M, w \models p$ iff $N, v \models p$. We write $Z : M \leftrightarrows N$ to indicate that Z is a bisimulation between M and N.

The back-and-forth conditions in clauses 2 and 3 of the definition of directed simulation allow us to transfer true box and diamond formulas from one transition system to another. Unlike the atomic clause in ordinary bisimulations, our atomic clause 1 does not display this back-and-forth behavior. As Theorem 3.5 below shows, this is exactly what is needed to characterize negation-free modal formulas.

Hennessy and Milner [9, Section 2.2] introduce a notion of simulation where even more of the back-and-forth conditions from ordinary bisimulations are missing: it lacks clause 3 of Definition 3.1. In the conclusion of the paper we point how our results carry over to that setting. As far as we know, Definition 3.1 is new.

§3.1. **Preservation.** Our first perspective on the expressive power of negation-free modal formulas is in terms of preservation.

Proposition 3.2 *For all negation-free modal formulas ϕ, and all transition systems M and N, and all states $w \in |M|$ and $v \in |N|$, if there exists a directed simulation $Z : M, w \rightrightarrows N, v$, then $M, w \models \phi$ implies $N, v \models \phi$.*

Proof. Use induction on formulas in $\mathcal{L}_{\Diamond,\Box}$. The back and forth clauses in Definition 3.1 were introduced especially to deal with the two modal cases.

Here's a proof for the $\Box$ case (the $\Diamond$ case is similar). Assume $w \models \Box\phi$, $Z : M, w \rightrightarrows N, v$, and Rvv'. By clause 2 of Definition 3.1 there exists w' such that Rww' and $Z : M, w' \rightrightarrows N, v'$. As Rww', we get $w' \models \phi$, and as $Z : w' \rightrightarrows v'$, we get $v' \models \phi$. Since v' was arbitrary, it follows that $v \models \Box\phi$, as required. $\dashv$

Thus, the existence of a directed simulation between M, w and N, v guarantees that $nf\text{-}tp_M(w) \subseteq nf\text{-}tp_N(v)$. Clearly, if in addition, there is a directed simulation going in the opposite direction, from N, v to M, w, then $nf\text{-}tp_M(w) = nf\text{-}tp_N(v)$. The obvious question, then, is: does $M, w \rightrightarrows N, v$ and $N, v \rightrightarrows M, w$ imply that $M, w \leftrightarrows N, v$? The answer is negative, and the following example shows that directed similarity in two directions is, in general, weaker than strong bisimilarity.

Example 3.3 Consider the models M_1 and M_2 as in Figure 1. That is, $M_1 = (\{a, a_0, a_1, a_2\}, R_1, V_1)$ and $M_2 = (\{b, b_1, b_2\}, R_2, V_2)$, where R_1 is $\{(a, a_i) \mid 0 \leq i \leq n\}$ and $R_2 = \{(b, b_1), (b, b_2)\}$; the valuations V_1 and V_2 are defined by $V_1(p) = \{a_0, a_1, a_2\}$, $V_1(q) = \{a_0, a_1\}$, and $V_1(r) = \{a_1\}$; $V_2(p) = \{b_1, b_2\}$, $V_2(q) = V_2(r) = \{b_1\}$.

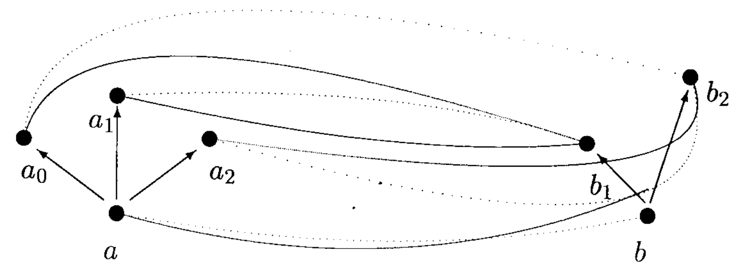

Figure 1: Two directed simulations.

Define $Z_0 \subseteq |M_1| \times |M_2|$ and $Z_1 \subseteq |M_2| \times |M_1|$ by

$$\begin{aligned} Z_0 &= \{(a, b), (a_0, b_1), (a_1, b_1), (a_2, b_2)\} \\ Z_1 &= \{(b, a), (b_1, a_1), (b_2, a_2), (b_2, a_0)\}. \end{aligned}$$

Then Z_0 is a directed simulation between M_1 and M_2 that links a to b, and Z_1 is a directed simulation that links b to a. However, there is no ordinary bisimulation linking a to b — there is no state in M_2 to which a_0 can be linked in a bisimulation.

To formulate a converse to Proposition 3.2 we define a transition system M to be *image-finite* if for every state $w \in |M|$ the set of its successors $\{v \in |M| \mid Rwv\}$ is finite.

Proposition 3.4 *Let M and N be image-finite models with $w \in |M|$, $v \in |N|$. Then the following are equivalent:*

1. $\textit{nf-tp}_M(w) \subseteq \textit{nf-tp}_N(v)$
2. $M, w \rightrightarrows N, v$.

A first-order formula $\alpha(x)$ in $\mathcal{L}_1$ is *preserved under directed simulations* if for all transition systems M, N, all states $w \in |M|$ and $v \in |N|$ and all directed simulations $Z : M, w \rightrightarrows N, v$ we have that $M \models \alpha[w]$ implies $N \models \alpha[v]$.

Theorem 3.5 (Preservation Theorem) *Let $\alpha(x)$ be an $\mathcal{L}_1(x)$-formula. Then α is equivalent to the standard translation of a (negation-free) modal formula iff it is preserved under directed simulations.*

Proof. The proof uses some basic first-order model theory; we refer the reader to Hodges [10] for background material. The right-to-left implication is immediate from Proposition 3.2. For the other direction, assume that $\alpha(x)$ is preserved under directed simulations. By a simple compactness argument it suffices to show that the negation-free consequences of α imply α:

(1) $\quad \textit{NF-Mod-Cons}(\alpha) := \{ST_x(\phi) \mid \alpha \models ST_x(\phi) \text{ and } \phi \in \mathcal{L}_{\Diamond,\Box}\} \models \alpha.$

To prove (1), assume that $M \models \textit{NF-Mod-Cons}(\alpha)[w]$; we have to show that $M \models \alpha[w]$. Consider the following set of $\mathcal{L}^{\neg}_{\Diamond,\Box}$-formulas:

$$\neg tp(w) := \{\neg\phi \mid \phi \in \mathcal{L}_{\Diamond,\Box} \text{ and } M, w \not\models \phi\}.$$

That is: $\neg tp(w)$ consists of negations of negation-free modal formulas that are refuted at x.

Claim 1. *The set $\{\alpha\} \cup \{ST_x(\neg\phi) \mid \neg\phi \in \neg tp(w)\}$ is satisfiable.*

Proof. Assume that it is not. Then there exist formulas $\neg\phi_1, \ldots, \neg\phi_n \in \neg tp(w)$ such that

$$\alpha \models \neg(ST_x(\neg\phi_1) \wedge \ldots \wedge ST_x(\neg\phi_n)),$$

or

$$\alpha \models ST_x(\phi_1) \vee \ldots \vee ST_x(\phi_n).$$

By definition, $\phi_1, \ldots, \phi_n$ are negation-free, so $M, w \models \phi_1 \vee \ldots \vee \phi_n$, and hence $M, w \models \phi_i$ for some i with $0 \leq i \leq n$. But then $\neg\phi \notin \neg tp(w)$ — a contradiction. This proves Claim 1. $\dashv$

Let N, v be such that $N \models \alpha[v]$ and $N, v \models \neg\phi$ for every formula $\neg\phi \in \neg tp(w)$.

Claim 2. $\textit{nf-tp}_N(v) \subseteq \textit{nf-tp}_M(w)$.

Proof. Suppose $\phi \in \textit{nf-tp}_N(v)$, but $M, w \not\models \phi$. Then $\neg\phi \in \neg tp(w)$, and hence $N, v \models \neg\phi$, by the definition of N, v — a contradiction. This proves Claim 2. $\dashv$

Now, to 'lift' α from N, v to M, w we make a detour via two other transition systems as follows. Take ω-saturated elementary extensions M^*, w and N^*, v of M, w and N, v, respectively. And define a relation $Z \subseteq |N^*| \times |M^*|$ by putting

$$uZt \text{ iff } \textit{nf-tp}_{N^*}(u) \subseteq \textit{nf-tp}_{M^*}(t).$$

Note first that Z is non-empty: by Claim 2 we have $\textit{nf-tp}_N(v) \subseteq \textit{nf-tp}_M(w)$, and so $\textit{nf-tp}_{N^*}(v) \subseteq \textit{nf-tp}_{M^*}(w)$, as M^*, w and N^*, v are elementary extensions of M, w and N, v, respectively.

Next, clause 1 of Definition 3.1 is trivially fulfilled. To see that clause 2 is satisfied, suppose that uZt and Rtt'; we need to find a u' such that Ruu' and $u'Zt'$. Put

$$-\textit{nf-tp}(t') = \{\phi \in \mathcal{L}_{\Diamond,\Box} \mid N^*, t' \not\models \phi\}.$$

We will show that any finite subset of $-\textit{nf-tp}(t')$ is *refutable* in an R-successor of u. Let $\phi_1, \ldots, \phi_n \in -\textit{nf-tp}(t')$. Then $N^*, t \not\models \Box(\phi_1 \vee \ldots \vee \phi_n)$, so, as uZt, $M^*, u \not\models \Box(\phi_1 \vee \ldots \vee \phi_n)$. This implies that for some $u' \in |M^*|$ both Ruu' and $u' \not\models \phi_1 \vee \ldots \vee \phi_n$ hold. Now, by ω-saturation of M^*, all of $-\textit{nf-tp}(t')$ can be refuted at an R-successor u' of u. For this u' we have Ruu' and $\textit{nf-tp}(u') \subseteq \textit{nf-tp}(t')$, that is: $u'Zt'$, as required.

For clause 3 we argue as follows. Suppose that uZt and Ruu'. We need to find a t' such that Rtt' and $u'Zt'$; we achieve this by showing that every finite subset of $\textit{nf-tp}(u')$ is satisfiable in a successor of t. Let $\phi_1, \ldots, \phi_n \in \textit{nf-tp}(u')$. Then $u \models \Diamond(\phi_1 \wedge \ldots \wedge \phi_n)$, and hence $t \models \Diamond(\phi_1 \wedge \ldots \wedge \phi_n)$. So there exists a t' in N^* with $N^*, t' \models \phi_1 \wedge \ldots \wedge \phi_n$ and Rtt'. By ω-saturation *all* of $\textit{nf-tp}(u')$ can be satisfied in a successor t' of t. For this t' we have Rtt' and $\textit{nf-tp}(u') \subseteq \textit{nf-tp}(t')$, that is: $u'Zt'$, as required.

Putting things together, we find that $N \models \alpha[v]$ implies $N^* \models \alpha[v]$ by elementary extension. As $Z : N^*, v \rightrightarrows M^*, w$ it follows that $M^* \models \alpha[w]$, and hence $M \models \alpha[w]$ by elementary submodel, we're done. $\dashv$

Example 3.6 Let $\alpha(x)$ be a first-order that is preserved under strong bisimulations. Then $\alpha(x)$ is equivalent to a modal formula ϕ in $\mathcal{L}^{\neg}_{\Diamond,\Box}$ that may include boolean negation. By testing whether $\alpha(x)$ is preserved under directed simulations we can find out whether ϕ is in fact equivalent to a negation-free modal formula.

An easy example is the first-order formula $\alpha(x) = \exists y\,(Rxy \wedge \neg Py)$. This formula is the first-order translation of $\Diamond\neg p$, and it is certainly preserved under strong bisimulation — but here's an example showing that it is not preserved under directed simulations: take $M_1 = (\{a_1, a_2\}, \{(a_1, a_2\}, V_1)$, and $M_2 = (\{b_1, b_2\}, \{(b_1, b_2\}, V_2)$, where V_1 and V_2 are such that all a_1, b_1, b_2 verify all proposition letters, and such that all proposition letter but p are true in a_2. Clearly, there exists a directed simulation linking a_1 and b_1, but $M_1 \models \alpha[a_1]$, whereas $M_2 \not\models \alpha[b_1]$.

By Theorem 3.5 directed simulations uniquely identify a certain fragment of first-order logic, namely the 'negation-free modal fragment.' By identifying and comparing fragments of first-order logic that correspond to modal languages in this manner, we have a method for comparing the expressive power of (negation-free) modal languages.

We proceed with three corollaries to Theorem 3.5 and its proof. The first of these concerns a 'dual' to preservation under directed simulations: a first-order formula $\alpha(x)$ is said to be *anti-preserved under directed simulations* if for all transition systems M, N, all states $w \in |M|$, $v \in |N|$ and all directed simulations $Z : M, w \rightrightarrows N, v$, we have that $N \models \alpha[v]$ implies $M \models \alpha[w]$. Next, call a modal formula in $\mathcal{L}^{\neg}_{\Diamond,\Box}$ *negation-rich* if it is built up from constants and negated atoms $\neg p$, using only $\vee$, $\wedge$, $\Diamond$ and $\Box$.

Corollary 3.7 *Let $\alpha(x)$ be an $\mathcal{L}_1(x)$-formula. Then α is equivalent to the standard translation of a negation-rich modal formula iff it is anti-preserved under directed simulations.*

Proof. Use Theorem 3.5 and the fact that the negation of a negation-free formula is equivalent to a negation-rich formula. $\dashv$

The following corollary characterizes the relation '*nf-tp*$(v) \subseteq$ *nf-tp*(w)' between states w, v in terms of directed simulations. We refer the reader to Hodges [10] for the notion of an ultrapower.

Corollary 3.8 *Let M and N be two transition systems, and let $w \in |M|$ and $v \in |N|$. Then nf-tp$_N(v) \subseteq$ nf-tp$_M(w)$ iff for some ultrapowers M^*, w of M, w, and N^*, v of N, v, we have that $N^*, v \rightrightarrows M^*, w$.*

Proof. The right-to-left implication is easy. For the converse, consider the proof of Theorem 3.5 again. For the ω-saturated elementary extensions M^*, w and N^*, v of M, w and N, v, respectively, we showed that $N^*, v \rightrightarrows$

M^*, w, starting from the assumption that $nf\text{-}tp_N(v) \subseteq nf\text{-}tp_M(w)$. By a result in first-order model theory, these ω-saturated extensions may be obtained as suitable ultrapowers of the original models M, w and N, v; see Chang and Keisler [6, Theorem 6.1.1] for details. $\dashv$

Corollary 3.9 *A modal formula in $\mathcal{L}^{\neg}_{\Diamond,\Box}$ is equivalent to a negation-free modal formula iff it is preserved under directed simulations.*

Proof. The left-to-right implication is Proposition 3.2. For the right-to-left implication, use Theorem 3.5 plus the fact that modal formulas are equivalent to their first-order translations under ST_x. $\dashv$

Example 3.10 The formula $\Diamond\neg p$ whose first-order translation was considered in Example 3.6 provides an example of a formula that is not preserved under directed simulations, and hence not equivalent to a negation-free modal formula. The formula $\Diamond(\top \vee \neg p)$, on the other hand, is preserved under directed simulations — and hence equivalent to a negation-free modal formula, namely $\Diamond\top$.

To conclude this subsection we present an alternative semantic characterization of the (modal) formulas preserved under directed simulations in terms of their monotonicity behavior. We call a modal formula ϕ *upward monotone in a proposition letter* p if for all models M and states w we have that if $M, w \models \phi$ and M' is obtained from M be extending the interpretation of p (and leaving the rest unchanged), then $M', w \models \phi$; the notion of *downward monotonicity* is defined dually.

By [17] a modal formula is ϕ is upward monotone in p iff p occurs only positively in ϕ, meaning that all occurrences of p should be in the scope of an even number of negation signs. More generally, a modal formula is called *positive* iff it can be built up from $\top$ and proposition letters, using only $\vee$, $\wedge$ and $\Diamond$ (see [17, Section 7] for the general picture). Although every positive formula is (equivalent to) a negation-free one, not every negation-free formula is (equivalent to) a positive one: $\Box\bot$ is an example. Therefore, the semantic characterization of positive modal formulas in terms of preservation under surjective homomorphisms given in [17, Theorem 7.15] doesn't apply to negation-free modal formulas. What we do have is the following extension of Corollary 3.9.

Theorem 3.11 *Let ϕ be a modal formula. The following are equivalent:*

1. *ϕ is equivalent to a formula in which all proposition letters occur only positively.*
2. *ϕ is equivalent to a negation-free formula.*
3. *ϕ is preserved under directed simulations.*

4. ϕ is upward monotone in all its proposition letters.

Proof. The implication 1 $\Rightarrow$ 2 is easy; the implication 2 $\Rightarrow$ 3 is Proposition 3.2, and the implication 3 $\Rightarrow$ 4 is immediate from the fact that if M' is transition system obtained from a transition system M by extending the interpretation of a proposition letter (and leaving the rest unaltered), then the identity relation is a directed simulation from M to M'. Finally, the implication 4 $\Rightarrow$ 1 is [17, Theorem 7.15]. $\dashv$

§3.2. Safety. In this subsection we take a different perspective on the expressive power of negation-free modal languages by considering the notion of *safety* recently introduced by van Benthem [3].

Let $\alpha(x,y)$ denote a first-order formula with at most two free variables. Then $\alpha(x,y)$ is called *safe for bisimulation* if whenever $Z : M \leftrightarrows N$ with wZv and $M \models \alpha[ww']$, then there exists a v' such that $w'Zv'$ and $N \models \alpha[vv']$. The formula $\alpha(x,y)$ is best thought of as an operation on the binary relations living inside the transition systems M and N, and the question for safety can be understood as asking whether the back-and-forth-conditions of Definition 3.1 hold for α whenever they hold for the relation symbols in α. The definition of safety depends in an essential way on the symmetric character of bisimulations: if the operation expressed by α is performed in M, then it can be matched by an α step in N, and vice versa.

What is the appropriate notion of safety for directed simulations? Their non-symmetric character causes a split in the notion of safety, depending on whether the operation α is performed on the left-hand side or on the right-hand side of a pair of directedly similar transition systems M and N. To be precise, a first-order formula $\alpha(x,y)$ is *left safe* for directed simulations if whenever $Z : M \rightrightarrows N$ with wZv and $M \models \alpha[ww']$ then there exists a v' such that $w'Zv'$ and $N \models \alpha[vv']$. A first-order formula $\alpha(x,y)$ is *right safe* for directed simulations if whenever $Z : M \rightrightarrows N$ with wZv and $N \models \alpha[vv']$ then there exists a w' such that $w'Zv'$ and $M \models \alpha[ww']$.

For example, atomic tests $P?$, whose semantics are given by $(x = y) \wedge Px$, are left safe, but not right safe. On the other hand, tests on negated atoms $\neg p$ are right safe, but not left safe. More generally, all negation-rich formulas are right safe.

Recall that the composition $R\,;S$ of two relations R and S is given by $R\,;S = \{(x,y) \mid \exists z\,(Rxz \wedge Szy)\}$. The dual operation of composition is denoted by $\div$, and defined by $R \div S = \{(x,y) \mid \forall z\,(Rxz \vee Szy)\}$. Then, we have the following characterizations of left and right safety.

Theorem 3.12 (Safety) *Let $\alpha(x,y)$ be a first-order formula in $\mathcal{L}_1(x,y)$.*

1. *Then $\alpha(x,y)$ is left safe for directed simulations iff it can be defined from the atomic relation R and tests on negation-free modal formulas using only ; and $\cup$.*

2. *Further, $\alpha(x, y)$ is right safe for directed simulations iff it can be defined from the negated atomic relation R and tests on negation-rich modal formulas using only $\div$ and $\cap$.*

Our proofs of the above results are tailored after similar results in [3]; they require a careful analysis of so-called continuous negation-free formulas, which we have included in an appendix. Here are the relevant definition and lemma.

Definition 3.13 A modal formula $\phi(p)$ is *continuous in p* if the following holds for every transition system (W, R, V):

> for each family of subsets $\{X_i\}_{i \in I}$ such that $V(p) = \bigcup_i X_i$: $(W, R, V), w \models \phi$ iff, for some i, $(W, R, V_i), w \models \phi$, where $V_i(p) = X_i$ and $V_i(q) = V(q)$ for $q \neq p$.

Example 3.14 The formula $\Box p$ is not continuous in p, but $\Diamond p$ is. And in fact the latter format typical for safety, as is shown by the following lemma.

Lemma 3.15 *A negation-free formula is continuous in p iff it is equivalent to a disjunction of formulas of the form $\phi_0 \wedge \Diamond(\phi_1 \wedge \cdots \wedge \Diamond(\phi_n \wedge p) \cdots)$, where each of the formulas ϕ_i is negation-free and p-free in the sense that they don't contain occurrences of p.*

A proof of the above lemma may be found in Appendix A below.

Proof of Theorem 3.12. We first prove part 1 of Theorem 3.12. To see that the constructions mentioned are indeed left safe, argue as follows. It is clear that the atomic relation and tests on negation-free formulas are left safe. To see that composition is left safe, assume that $Z : M, w \rightrightarrows N, v$, and that $wS_1 ; S_2w'$, where the back-and-forth conditions of Definition 3.1 hold for S_1 and S_2 in M. Then, there exists w'' with $wS_1w''S_2$. As Z is assumed to be a directed simulation for S_1, there exists a v'' in N with vS_1v'' and $w''Zv''$, and, likewise, there exists a $v' \in |N|$ with $v''S_2v'$ and $w'Zv'$. The latter is the required $S_1 ; S_2$-successor in N. Showing that choice ($\cup$) is left safe is left to the reader.

Now, to prove the more complex left-to-right half of part 1 of Theorem 3.12, let $\alpha(x, y)$ be a first-order operation that is left safe, and choose a *new* proposition letter p. Our first observation is that $\exists y\,(\alpha(x, y) \wedge ST_y(p))$ is preserved under directed simulations — this is immediate from the fact that $\alpha(x, y)$ is left safe. As a corollary we have that, by our Preservation Theorem 3.5, $\exists y\,(\alpha(x, y) \wedge ST_y(p))$ is equivalent to a negation-free modal formula ϕ. In addition, because of the special syntactic form of $\exists y\,(\alpha(x, y) \wedge ST_y(p))$, this formula ϕ is continuous in p. Therefore, by Lemma 3.15 we may assume that it is a disjunction of formulas of the form

$$\phi_0 \wedge \Diamond(\phi_1 \wedge \cdots \wedge \Diamond(\phi_n \wedge p) \cdots),$$

where each of the formulas ϕ_i is negation-free and p-free. To complete the proof we need one more observation, viz. that $\alpha(x,y)$ is definable as a union of relations of the form

$$(2) \qquad (\phi_0?)\,;R\,;(\phi_1?)\,;\cdots;R\,;(\phi_n?),$$

where, again, each of the formulas ϕ_i is negation-free. But this is exactly the syntactic form specified in the theorem, and, hence, this proves part 1.

We now turn to part 2 of Theorem 3.12. Observe first that all the operations listed in part 2 of the theorem are indeed right safe. For the converse, we argue as follows. If $\alpha(x,y)$ is a right safe first-order formula, then $\neg\alpha(x,y)$ is a left safe formula, hence, by part 1, it is equivalent to a union of formulas of the form specified in (2). But then $\alpha(x,y)$ must be equivalent to an intersection of formulas of the dual form

$$(\psi_0?) \div \neg R \div (\psi_1?) \div \cdots \div \neg R \div (\psi_n?),$$

where each of the ψ_i is a negation-rich formula. As this is the required syntactic form, this proves part 2 of the theorem. $\dashv$

There is a natural follow-up to Theorem 3.12: what are the first-order operations $\alpha(x,y)$ that are *doubly safe* for directed simulations, i.e., formulas that are both left and right safe.

Theorem 3.16 *Let $\alpha(x,y)$ be a formula in $\mathcal{L}_1(x,y)$. Then $\alpha(x,y)$ is doubly safe for directed simulations iff it can be defined from the atomic relation R and tests on negation-free modal formulas without occurrences of proposition letters using only ; and $\cup$.*

Proof. The right-to-left implication is easily verified. For the converse, assume that $\alpha(x,y)$ is doubly safe. By Theorem 3.12 $\alpha(x,y)$ is equivalent to a formula

$$\beta := \bigvee_i ((\phi_{i0}?)\,;R\,;\cdots;R\,;(\phi_{in_i}?)),$$

where each ϕ_{ik} is negation-free; $\alpha(x,y)$ is also equivalent to a formula

$$\gamma := \bigwedge_j ((\psi_{j0}?) \div \neg R \div \cdots \div \neg R \div (\psi_{jn_j}?)),$$

where each ψ_{jk} is negation-rich.

Let us write $[\top/\vec{p}]\delta$ to denote the result of substituting $\top$ for all occurrences of all proposition letters in χ. We will show that $\models \alpha \leftrightarrow [\top/\vec{p}]\beta$, and we will use the fact that formulas in which all (translations of) proposition letters occur only positively (negatively) are upward (downward) monotone. If M is any transition system, then we write M^+ to denote the transition system that is just like M except that it assigns $|M|$ to every proposition letter.

Observing that all proposition letters in β occur only positively in β, and that all proposition letters in γ occur only negatively in γ, we have, for any transition system M,

$$\begin{aligned} M \models \alpha[wv] &\Rightarrow M \models \beta[wv] \\ &\Rightarrow M^+ \models \beta[wv] \\ &\Rightarrow M \models [\top/\vec{p}]\beta[wv], \end{aligned}$$

and

$$\begin{aligned} M \not\models \alpha[wv] &\Rightarrow M \not\models \gamma[wv] \\ &\Rightarrow M^+ \not\models \gamma[wv] \\ &\Rightarrow M^+ \not\models \beta[wv] \\ &\Rightarrow M \not\models [\top/\vec{p}]\beta[wv]. \end{aligned}$$

This proves $\models \alpha \leftrightarrow [\top/\vec{p}]\beta$, and the latter is of the required form. $\dashv$

§3.3. Definability. In this subsection we offer a third and final perspective on the expressive power of negation-free modal languages by analyzing which properties of transition systems are definable by a negation-free modal formula. Our analysis is in terms of definable classes of transition systems, and to smoothen the results and the presentation we will work with so-called *pointed transition systems*; these are structures of the form (M, w), where M is a transition system as defined in Section 2 and $w \in |M|$ is the *distinguished state* of (M, w). $(M, w) \models \phi$ will mean the same thing as $M, w \models \phi$. Bisimulations between pointed transition systems are required to link the distinguished states.

A class of pointed transition systems K is *negation-free definable by a set of formulas* if there exists a set of negation-free formulas Δ such that $\mathsf{K} = \{(M, x) \mid (M, x) \models \phi \text{ for all } \phi \in \Delta\}$. K is called *negation-free definable by a single formula* if it is negation-free definable by means of a singleton set.

If K is a class of pointed transition systems, we write $\overline{\mathsf{K}}$ to denote the class of pointed transition systems that are not in K. We say that K is *closed under* ultraproducts (ultrapowers) if any ultraproduct (ultrapower) of transitions systems in K is itself in K. Likewise, K is *closed under* directed simulations if $(M, w) \in \mathsf{K}$ and $Z : (M, w) \rightrightarrows (N, v)$ implies $(N, v) \in \mathsf{K}$.

Theorem 3.17 (Definability) *Let* K *be a class of pointed transition systems. Then*

1. K *is negation-free definable by a set of formulas iff* K *is closed under directed simulations and ultraproducts, while* $\overline{\mathsf{K}}$ *is closed ultrapowers.*

2. K *is negation-free definable by a single formula iff* K *is closed under directed simulations and ultraproducts, while* $\overline{\mathsf{K}}$ *is closed under ultraproducts.*

Proof. The left-to-right implications are left to the reader. For the right-to-left implication of item 1, argue as follows. If K and $\overline{\mathsf{K}}$ satisfy the stated closure conditions, then both K and $\overline{\mathsf{K}}$ are also closed under bisimulations, and hence, by [17, Theorem 6.3] they are definable by a set of modal formulas Δ. Now, as K is closed under directed simulations, each formula in Δ must be preserved under directed simulations, and hence equivalent to a negation-free modal formula by Corollary 3.9. This shows that K is negation-free definable.

Next, for the right-to-left implication of item 2 we use a similar argument. If K and $\overline{\mathsf{K}}$ satisfy the stated closure conditions, then they are both closed under bisimulations and ultraproducts. By [17, Theorem 6.3], again, this implies that K is definable by a single modal formula ϕ. As K is closed under directed simulations, ϕ must be preserved under directed simulations, and hence it must be equivalent to a negation-free modal formula by Corollary 3.9. $\dashv$

The characterization of definability given in Theorem 3.17 is hard to use in practice as ultraproducts are rather abstract objects. The following gives a more manageable Fraïssé-type characterization.

Let M, w and N, v be pointed transition systems. We define *directed similarity up to n* between M, w and N, v ($n \in \mathbb{N}$) by requiring that there exists a sequence of binary relations $Z_0, \ldots, Z_n \subseteq |M| \times |N|$ such that

1. $Z_n \subseteq \cdots \subseteq Z_0$ and wZ_0v
2. for each $i \leq n$, if uZ_it and $u \models p$, then $t \models p$
3. for $i+1 \leq n$ the back-and-forth properties of Definition 3.1 are satisfied relative to the indices:
   (a) if $uZ_{i+1}t$ and Rtt' in N, then there exists $u' \in |M|$ such that Ruu' and $u'Z_it'$
   (b) if $uZ_{i+1}t$ and Ruu' in N, then there exists $t' \in |N|$ such that Rtt' and $u'Z_it'$.

We write $M, w \rightrightarrows^n N, v$ to denote that there exists a a directed simulation up to n.

Recall that the *degree* $\deg(\phi)$ of a modal formula ϕ is the largest number of nested modal operators occurring in it.

Theorem 3.18 *Assume that $\mathcal{L}_{\Diamond,\Box}$ is finite (i.e., contains only finitely many proposition letters), and let* K *be a class of pointed transition systems. Then*

K *is negation-free definable by a single formula iff, for some* $n \in \mathbb{N}$, K *is closed under directed simulations up to* n.

Proof. Clearly, if K is negation-free definable by a single formula of degree n, then it is closed under directed simulations up to n. To prove the converse, let $(M, w) \in$ K, and define $\phi^n_{M,w}$ to be the conjunction of all formulas in $nf\text{-}tp_M(w)$ of degree at most n — as we are working in a finite language, we can assume that there are only finitely many non-equivalent negation-free formulas of any given degree, hence we may assume $\phi^n_{M,w}$ to be a (finitary) formula in $\mathcal{L}_{\Diamond,\Box}$.

Using the finite character of the language again, we find that there are only finitely many non-equivalent formulas $\phi^n_{M,w}$ for $(M, w) \in$ K. Let Φ^n be their disjunction. Then Φ^n defines K. For, assume that $(N, v) \models \Phi^n$; we need to show that $(N, v) \in$ K. First, from $(N, v) \models \Phi^n$ it follows that for some $(M, w) \in$ K, (N, v) agrees with (M, w) on all negation-free formulas of degree at most n. Second, the latter fact implies that $M, w \rightrightarrows^n N, v$. To see this, define relations $Z_i \subseteq |M| \times |N|$ for $1 \leq i \leq n$ by putting uZ_it iff every negation-free modal formula of degree at most i that is true at u, is also true at t. Then $Z_0, \ldots, Z_n$ is a directed simulation up to n that links M, w to N, v. As K is closed under directed simulations up to n, this implies $(N, v) \in$ K, and we are done. $\dashv$

Example 3.19 The class of pointed models defined by the first-order formula Px is closed under directed simulations up to 0, and hence definable by a modal formula of degree 0. The class of pointed models defined by the first-order formula $\forall y\,(Rxy \to \exists z\,(Ryz))$ is not closed under directed simulations up to 0 or 1 (and hence not definable by a modal formula of degree less than 2), but it is closed under directed simulations up to 2, and it is therefore definable by a modal formula of degree at most 2.

4 Extensions

The main idea that underlies our work in Section 3 is a very simple one: replace the 'symmetric' atomic condition in the definition of a bisimulation by a non-symmetric or directed one. In this section we apply the same strategy to study further negation-free languages arising in terminological logic, Since/Until logic, and feature logic. Our presentation will be somewhat impressionistic, aimed at indicating the applicability of the main ideas rather than giving full details.

§4.1. Terminological Logics. Terminological logics are description logics stemming from semantic networks and designed for representing structured concepts. The system KL-ONE is a well-known knowledge representation system based on terminological logics. In a terminological logic the structure of a concept (or set) is described using some or all of the booleans, and various forms of quantification over the attributes of a concept (in terms of binary relations). One of the main concerns in the area is the computational complexity of reasoning problems in terminological logics; although this is closely related to matters of expressive power, until recently the latter has never been studied in a systematic way (see Baader [2]). However, there is a close connection between modal and terminological logics which can be exploited to improve on this.

Constructor name	Syntax	Semantics	Modal
concept name	A	$A^{\mathcal{I}} \subseteq W$	p
top	$\top$	W	$\top$
bottom	$\bot$	$\emptyset$	$\bot$
conjunction	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$	$\phi \wedge \psi$
disjunction ($\mathcal{U}$)	$C \sqcup D$	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$	$\phi \vee \psi$
negation ($\mathcal{C}$)	$\neg C$	$W \setminus C^{\mathcal{I}}$	$\neg\phi$
univ. quantification	$\forall R.C$	$\{d_1 \mid \forall d_2\, (d_1, d_2) \in R^{\mathcal{I}} \rightarrow d_2 \in C^{\mathcal{I}}\}$	$\Box\phi$
exist. quantification ($\mathcal{E}$)	$\exists R.C$	$\{d_1 \mid \exists d_2\, (d_1, d_2) \in R^{\mathcal{I}} \wedge d_2 \in C^{\mathcal{I}}\}$	$\Diamond\phi$

Table 1: Syntax and semantics of concept-forming constructors.

In their survey paper, Donini et al. [7] study several hierarchies of terminological languages. By way of example, we consider the hierarchy of $\mathcal{FL}^-$-languages specified in Table 1. Here, we use A, B to denote atomic concept names ('proposition letters'), and C, D to denote complex concepts ('modal formulas'); and we use R to denote roles ('binary relations'). Terminological expressions are interpreted using on interpretation function $(\cdot)^{\mathcal{I}}$ on transition systems (W, R).

The various languages differ in the constructions they admit; $\mathcal{FL}^-$ denotes the language with universal quantification, conjunction and unqualified existential quantification $\exists R.\top$. Superlanguages of $\mathcal{FL}^-$ are identified by strings of the form $\mathcal{FL}[\mathcal{E}][\mathcal{U}][\mathcal{C}]^-$. We will assume that $\mathcal{FL}^-$ contains $\top$ and $\bot$.

Clearly, $\mathcal{FLEU}^-$ coincides with (a multi-modal version) of our negation-free modal language $\mathcal{L}_{\Diamond,\Box}$, and hence Theorems 3.5, 3.12, 3.17, and 3.18 all carry over without effort to $\mathcal{FLEU}^-$. Likewise, the analogous results on expressivity for the standard modal language $\mathcal{L}^{\neg}_{\Diamond,\Box}$ carry over to the corresponding terminological language $\mathcal{FLEUC}^-$. (Further details on the latter connection may be found in [20, 11].) Thus, two of the languages in the $\mathcal{FL}^-$ hierarchy have been equipped with model-theoretic tools for analyzing their expressive power. The remaining terminological languages in Table 1 call for further non-standard notions of (bi-)simulation; coming up with such notions and using them to arrive at a model-theoretic analysis of the remaining languages in Table 1 is part of our ongoing work.

§4.2. Since and Until. Our next example concerns directed simulations for a negation-free fragment of Since/Until logic. To simplify matters we restrict ourselves to the forward looking fragment of the language that only contains the Until operator U. Recall its truth definition on transition systems:

$$M, w \models U(\phi, \psi) \text{ iff there exists } w' \text{ with } M, w' \models \phi \text{ and for all } w'', \text{ if } wRw''Rw' \text{ then } M, w'' \models \psi.$$

Recently, a notion of bisimulations for Since and Until has been introduced that allows for a complete development of the model theory of the full Since/Until language (see [15]). Building on this, we define the following simulations for the negation-free forward looking fragment $\mathcal{L}_U$. A *directed U-simulation* from M to N is a pair (Z_0, Z_1), where $Z_0 \subseteq |M| \times |N|$ and $Z_1 \subseteq |M^2| \times |N^2|$ such that

1. wZ_0v and $w \models p$ implies $v \models p$
2. if wZ_0v and Rww' then there exists v' such that Rvv', $w'Z_0v'$ and $(w, w')Z_1(v, v')$
3. if $(w, w')Z_1(v, v')$ and $vRv''Rv'$ then there exists w'' with $wRw''Rw'$ and $w''Z_0v''$.

The first of the above clauses is the same as before; the second records transitions in simulating pairs of states, and the third clause makes sure that if two pairs of states simulate each other, than they 'agree' on intermediate states.

Remark 4.1 In the definition of directed simulation for $\mathcal{L}_{\Diamond,\Box}$ we had back-and-forth clauses to be able to transfer true formulas involving the diamond operator and its dual the box operator from one model to another. To simplify matters, we have left out a dual for the U-operator from our negation-free fragment of the Since/Until language. As a consequence we can make do with clauses 2 and 3 above; in the presence of a dual of U, we would have to add clauses $2'$ and $3'$ going in the opposite directions.

Here's an example of a directed simulation: consider the models $M_1 = (\{a, a_0, a_1, a_2\}, R_1, V_1)$ and $M_2 = (\{b, b_1, b_2\}, R_2, V_2)$, where R_1 is $\{(a, a)\} \cup \{(a, a_i) \mid 0 \leq i \leq n\}$ and $R_2 = \{(b, b)\} \cup \{(b, b_1), (b, b_2)\}$; the valuations V_1 and V_2 are defined by $V_1(p) = \{a_0, a_1, a_2\}$, $V_1(q) = \{a_0, a_1\}$, $V_1(r) = \{a_1\}$, and $V_1(s) = \{a\}$; $V_2(p) = \{b_1, b_2\}$, $V_2(q) = V_2(r) = \{b_1\}$, and b verifies all proposition letters (see Figure 2). Define $Z_0 \subseteq |M_1| \times |M_2|$ by

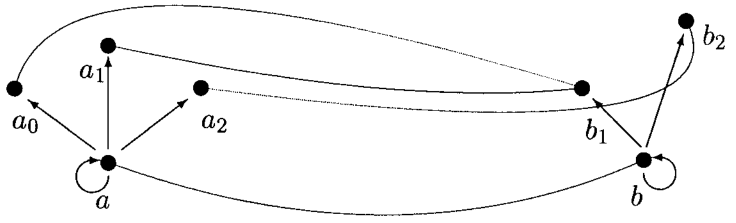

Figure 2: Directed U-simulation.

$$Z_0 = \{(a,b),(a_0,b_1),(a_1,b_1),(a_2,b_2)\}.$$

To define $Z_1 \subseteq |M_1|^2 \times |M_2|^2$, define a pair of states (x,y) with Rxy to be minimal if there are no states in between x and y. Put

$$\begin{aligned} Z_1 &= \{((a,a),(b,b))\} \cup \\ &\quad \{((w,w'),(v,w')) \mid R_1ww',\ R_2vv' \text{ and } (w,w'),\ (v,v') \text{ minimal}\}. \end{aligned}$$

We leave it to the reader to check that (Z_0, Z_1) is a directed U-simulation between M_1 and M_2 that links a to b. It follows that every negation-free Until-formula true in a is also true in b — but the converse obviously fails.

It is precisely the fact that directed U-simulations link points to points and pairs of points to pairs of pairs of points that allows to prove analogs of Theorems 3.5, 3.12, 3.17 and 3.18 for $\mathcal{L}_U$ by combining the techniques and results in Section 3 with those in [15].

§4.3. Feature Logics. We conclude our list of extensions of the basic results in Section 3 with directed simulations for feature logics. Feature logics are description languages for a special kind of data structures called *feature structures*. These are related to the record structures of computer science and the frames of artificial intelligence. In computational linguistics they are labeled graphs carrying syntactic, semantic, morphological and phonetic information, and the purpose of a linguistic theory is to describe admissible graphs of this kind that underlie text and speech. Feature structures have also been used to characterize partially defined concrete data types in programming languages.

Formally, let L be a set of *feature names*, and A a set of *sort names*. The pair (L, A) is called a *feature signature*. A *feature system* of signature (L, A) is a tuple $M = (D, \{R_l\}_{l\in L}, \{D_a\}_{p\in A})$, where for each feature name l, f_l is a partial function on D, and for each sort name p, D_p is a subset of D. (Feature systems are simply labeled transition systems of a special kind.)

Various logical systems have been proposed to constrain feature structures. Each takes a slightly different view of its models, but often they are non-boolean fragments of modal logics. In this paper we consider a single example of a feature logic; see Rounds [19] for a survey of feature languages.

The logic we consider is called *Kasper-Rounds logic*. Assuming L and A as above, the atomic expressions of $\mathcal{L}(KR)$ are the following: proposition letters p ($p \in A$), and so-called *path equations* $\pi \doteq \rho$, for π, $\rho \in L^*$. Complex formulas are built up using conjunction, disjunction and modalities $\langle l \rangle$ and $[l]$, for $l \in L$.[1] The only novel aspect in the interpretation of $\mathcal{L}(KR)$ is the interpretation of the path equations and of the indexed modal operators $\langle l \rangle$. Path equations are meant to express that two sequences of transitions lead to the same state; for convenience, we will assume that every finite sequence of feature names π comes with its own transition relation R_π.

- $M, w \models \pi \doteq \rho$ if there exists w' with $(w, w') \in R_\pi \cap R_\rho$. (If it exists, this w' will be unique.)
- $M, w \models \langle l \rangle \phi$ if there exists w with $R_l ww'$ and $M, w' \models \phi$. (Again, if it exists, this w' will be unique.)

Our next aim is to state an analog of Theorem 3.5 for $\mathcal{L}(KR)$. First we fix a first-order language $\mathcal{L}_1^{KR}$ into which we translate Kasper-Rounds formulas. $\mathcal{L}_1^{KR}$ has binary relation symbols R_l for $l \in L$, and unary predicate symbols P for all sort names $p \in A$. The novel clauses for the standard translation ST_x are the following:

$$ST_x(\pi \doteq \rho) = \exists y\, (xR_{\pi_1}\, ; \cdots ; R_{\pi_n} y \wedge xR_{\rho_1}\, ; \cdots ; R_{\rho_m} y),$$

where $\pi = \pi_1 \cdots \pi_n$ and $\rho = \rho_1 \cdots \rho_m$, and all the π_i and ρ_i are 'atomic' feature names in L; and $ST_x(\langle l \rangle \phi) = \exists y\, (R_l xy \wedge ST_y(\phi))$, and similarly for $[l]\phi$.

What kind of simulations are we to use to identify $\mathcal{L}(KR)$ as a fragment of the first-order language $\mathcal{L}_1^{KR}$? Note that path equations $\pi \doteq \rho$ are essentially intersections of compositions of 'atomic' transition relations R_l. Their intersective character calls for a special kind of simulations in which we ensure that intersecting paths are preserved. The definition below achieves this by relating states to states and pairs of states to pairs of states; it is based on [5] and [13].

We write $w \stackrel{*}{\longrightarrow} w'$ for the reflexive, transitive closure of $\bigcup_{l \in L} R_l$. A *directed KR-simulation* from M to N is a triple (Z_0, Z_1, Z_2) where $Z_0 \subseteq |M| \times |N|$, $Z_1 \subseteq |M^2| \times |N^2|$, and $Z_2 \subseteq |N^2| \times |M^2|$ such that

1. wZ_0v and $w \models p$ implies $v \models p$
2. (a) $(w, w')Z_1(v, v')$ and $R_l ww'$ implies $R_l vv'$.
   (b) $(v, v')Z_2(w, w')$ and $R_l vv'$ implies $R_l ww'$.
3. (a) if wZ_0v and $R_l ww'$ then there exists v' with $(w, w')Z_1(v, v')$

[1] Our syntax deviates from the one presented in e.g., [19], but the differences are inessential.

(b) if wZ_0v and R_lvv' then there exists w' with $(v, v')Z_2(w, w')$

4. $(w, w')Z_1(v, v')$ implies wZ_0v and $w'Z_0v'$; similarly for $(v, v')Z_2(w, w')$

5. (a) if $(w, w')Z_1(v, v')$ and $w \xrightarrow{*} w'' \xrightarrow{*} w'$, then there exists v'' such that $(w, w'')Z_1(v, v'')$ and $(w'', w')Z_1(v'', v')$

   (b) if $(v, v')Z_2(w, w')$ and $v \xrightarrow{*} v'' \xrightarrow{*} v'$, then there exists w'' such that $(v, v'')Z_2(w, w'')$ and $(v'', v')Z_1(w'', w')$

Clause 1 is familiar. The back-and-forth conditions in clauses 2 and 3 ensure that transitions are recorded in simulating pairs of states; together with clause 5 they allow us to simulate intersecting paths in one transition system with intersecting paths in the other. Finally, clause 4 is a bookkeeping clause that relates the behavior of (Z_0, Z_1, Z_2) on pairs of states to its behavior on single states.

With this notion of directed KR-simulation one can proceed to prove analogs of Theorems 3.5, 3.12, 3.17 and 3.18 for $\mathcal{L}(KR)$ by combining the techniques and results of Section 3 and [13]. The details would take us to far astray from the main points of the present paper to be included here; instead, we refer the reader to [16].

5 Non-Classical Negation

Although in many application areas boolean negation is unwanted, some form of negation is often called for. This motivates the introduction of non-classical negations. The first example that comes to mind is probably *intuitionistic negation.* In this section we show how our directed simulations have to be amended for the results of Section 3 to carry over to intuitionistic logic.

Recall that a transition system $M = (W, \leq, V)$ is called an *intuitionistic model* if $\leq$ is a partial order, and V is a valuation that assigns $\leq$-closed subsets of W to proposition letters.

We assume that the language of intuitionistic logic has $\bot$, $\wedge$, $\vee$, and $\Rightarrow$. Conjunction and disjunction are interpreted in the boolean manner, while $\bot$ is false at all states, and $w \models \phi \Rightarrow \psi$ if for all w', $w \leq w'$ and $w' \models \phi$ implies $w' \models \psi$. As usual, negation is introduced as an abbreviation for $\phi \Rightarrow \bot$.

Let M, N be two intuitionistic models. A *directed intuitionistic bisimulation* is a pair (Z_0, Z_1) with $Z_0 \subseteq |M| \times |N|$ and $Z_1 \subseteq |N| \times |M|$ such that

1. (a) if wZ_0v and $w \models p$, then $v \models p$

   (b) if vZ_1w and $v \models p$, then $w \models p$

2. (a) if wZ_0v and $v \leq v'$, then there exists w' such that $w \leq w'$, $w'Z_0v'$ and $v'Z_1w'$

(b) if vZ_1w and $w \leq w'$, then there exists v' such that $v \leq v'$, $v'Z_1w'$ and $w'Z_0v'$

We use $Z : M \leftrightarroweq_i N$ to denote that Z is a directed intuitionistic bisimulations between M and N.

The intuition behind the above definition is the following. As we have seen in Section 3, in the absence of negation we can make do with directed simulations, and as is known from the literature, in the presence of full boolean negation we need bisimulations with full back-and-forth clauses. In intuitionistic logic, we are somewhere in between. The intuitionistic implication introduces negative occurrences (formulas occurring on its left-hand side). To account for this we need to increase the interaction between simulating. We do this by having two relations going in opposite directions.

Observe that directed intuitionistic bisimulations are not as strong as strong bisimulations (as this would be appropriate for boolean negation only). However, if (Z_0, Z_1) is a directed intuitionistic bisimulation between M and N, then $Z_0 \cap Z_1{}^{\smallsmile}$ is a strong bisimulation between M and N.

Proposition 5.1 *Intuitionistic formulas are preserved under directed intuitionistic bisimulations: if $(Z_0, Z_1) : M \leftrightarroweq_i N$, wZ_0v and $w \models \phi$, then $v \models \phi$ (and likewise, if vZ_1w and $v \models \phi$, then $w \models \phi$).*

Proof. Use induction on intuitionistic formulas. The atomic case is clear, and so are the inductive cases for $\wedge$ and $\vee$. The case for implication shows why directed intuitionistic bisimulations are defined the way they are: assume that $w \in |M|$, $v \in |N|$, wZ_0v and $w \models \phi \Rightarrow \psi$; we need to show $v \models \phi \Rightarrow \psi$. Take any v' such that $v \leq v'$ in N, and assume $v' \models \phi$. We need to show $v' \models \psi$. Now, by clause 2 (a) of the above definition, there exists a w' with (i) $w \leq w'$, (ii) $v'Z_1w'$ and (iii) $w'Z_0v'$. Use (i) to conclude that (iv) if $w' \models \phi$ then $w' \models \psi$; use (ii) and $v' \models \phi$ and the induction hypothesis to conclude that $w' \models \phi$. Then, by (iv), $w' \models \psi$ and, by (iii) and the induction hypothesis again, $v' \models \psi$ — and we're done. $\dashv$

Using the notion of directed intuitionistic bisimulation, one can establish counterparts of Theorems 3.5, 3.12, 3.17 and 3.18. To prove a preservation result along the lines of Theorem 3.5, we need to define a translation of intuitionistic formulas in to first-order formulas. The *intuitionistic standard translation* $IST_x(\cdot)$ takes intuitionistic formulas to $\mathcal{L}_1$-formulas as follows: $IST_x(p) = Px$; IST_x commutes with $\wedge$ and $\vee$; and

$$IST_x(\phi \Rightarrow \psi) = \forall y\, (Rxy \to (IST_y(\phi) \to IST_y(\psi))).$$

Theorem 5.2 (Preservation Theorem) *Let $\alpha(x)$ be an $\mathcal{L}_1(x)$-formula. Then $\alpha(x)$ is equivalent (on intuitionistic models) to the translation of an intuitionistic formula iff it is preserved under directed intuitionistic bisimulations.*

Proof. The left-to-right implication is Proposition 5.1. The converse is proved along the lines of Theorem 3.5. There are a few things to take into account:

- we need infinitely many axioms to express the $\leq$-closedness of the interpretation of proposition letters (unary predicates)
- we need axioms to express that intuitionistic models are partial orders.

As these axioms are all first-order axioms, we can use the techniques of Theorem 3.5 as before. Hence, we only sketch the main steps here.

By a compactness argument it suffices to show that show that α is itself a consequence of the set of its intuitionistic consequences

$$\mathit{Int\text{-}Cons}(\alpha(x)) := \{IST_x(\phi) \mid \alpha \models IST_x(\phi), \phi \text{ intuitionistic}\}.$$

So, consider a model $M \models \mathit{Int\text{-}Cons}(\alpha(x))[w]$; we have to show that $M \models \alpha[w]$. We achieve this by showing that there exists a model (N, v) for

$$\{\alpha\} \cup \mathit{itp}(w) \cup \neg\mathit{itp}(w),$$

where $\mathit{itp}(w)$ is the set of (translations of) intuitionistic formulas satisfied by w, and $\neg\mathit{itp}(w)$ is the set of boolean negations of (translations of) intuitionistic formulas false at w. Using this, we move to two ω-saturated elementary extensions of M, w and N, v and show that there must be a directed intuitionistic bisimulation relating w and v between those two models. The latter allows us to conclude that $M \models \alpha[w]$. $\dashv$

Corollary 5.3 *A modal formula in $\mathcal{L}^{\neg}_{\Diamond,\Box}$ is equivalent to an intuitionistic formula iff it is preserved under directed intuitionistic bisimulations.*

Example 5.4 The modal formulas $\Box\Diamond p$ and $\Box\Diamond\neg p$ are preserved under directed intuitionistic bisimulations (between intuitionistic models), and, so, on intuitionistic models they are equivalent to intuitionistic formulas. We leave it to the reader to show that, more generally, every modal formula which contains negation only in the scope of modal operators is preserved under directed intuitionistic bisimulations, and therefore equivalent to an intuitionistic formula.

Our next goal is to state and prove a safety theorem for intuitionistic logic along the lines of Theorem 3.12. We call a first-order formula $\alpha(x, y)$ in $\mathcal{L}_1(x, y)$ *safe for directed intuitionistic bisimulations* if whenever $(Z_0, Z_1) : M \leftrightarroweq_i N$ with wZ_0v and $M \models \alpha[ww']$, then there exists a v' with $w'Z_0v'$ and $M \models \alpha[vv']$; and vice versa.

Theorem 5.5 *Let $\alpha(x,y)$ be a first-order formula in $\mathcal{L}_1(x,y)$. Then $\alpha(x,y)$ is safe for directed intuitionistic bisimulations iff it can be defined from the the atomic relation Rxy and tests on atomic formula p, using only composition ;, choice $\cup$ and the counter-domain operation $\sim$ defined by $\sim S := \{(x,y) \mid x = y \wedge \neg\exists z\, Sxz\}$.*

Proof. As an example we first show that composition ; is indeed safe for directed intuitionistic bisimulations. So, suppose that $(Z_0, Z_1) : M, w \rightrightarrows_i N, v$, with wZ_0v and $wS_1 ; S_2w'$. We have to show that there exists v' with $vS_1 ; S_2v'$ and $w'Z_0v'$. First, there exists a w'' with $wS_1w''S_2w'$. From wZ_0v and S_1ww'' we get a v'' with S_1zw and $w''Z_0v''$. Together with $S_2w''w'$ this implies that there exists v' with $w'Z_0v'$ and $S_2v''v'$ — which is what we need.

To show that any operation that is safe for directed intuitionistic bisimulations can be defined as described, requires a small trick. First, recall the *Gödel translation* g that takes intuitionistic formulas to equivalent modal formulas:

$$
\begin{aligned}
g(\bot) &= \bot \\
g(p) &= \Box p \\
g(\phi \wedge \psi) &= g(\phi) \wedge g(\psi) \\
g(\phi \vee \psi) &= g(\phi) \vee g(\psi) \\
g(\phi \Rightarrow \psi) &= \Box(g(\phi) \rightarrow g(\psi)).
\end{aligned}
$$

Now, assume that $\alpha(x,y)$ is safe for directed intuitionistic bisimulations. Then $\exists y\, \alpha(xy)$ is preserved under such bisimulations, and, hence, by Theorem 5.2 it is equivalent to am intuitionistic formula ϕ.

Observe that ϕ is equivalent (in intuitionistic logic) to $\phi \wedge (\bot \Rightarrow p)$, where p is a new proposition letter. And the latter, in turn, is equivalent to its Gödel translation $g(\phi) \wedge \Box(\bot \rightarrow \Box p)$. Observe that this formula is continuous in p; hence by [3, Chapter 5] it is equivalent to a disjunction of formulas of the form

$$\phi_0 \wedge \Diamond(\phi_1 \wedge \ldots \wedge \Diamond(\phi_n \wedge p) \cdots),$$

where each of the ϕ_i's is a p-free modal formula (compare Lemma 3.15). Following the proof of the Safety Theorem in [3], one can then deduce that $\alpha(x,y)$ must be equivalent to the union of formulas of the form

$$(\phi_0?) ; R ; \cdots ; R ; (\phi_n?).$$

Using equivalences such as $(\phi \wedge \psi)? \leftrightarrow (\phi? ; \psi?)$, $(\neg\phi)? \leftrightarrow \sim(\phi?)$, and $(\Diamond\phi)? \leftrightarrow \sim\sim(R ; (\phi?))$, the tests can be pushed inside, so as to produce a formula of the required form. $\dashv$

As a corollary to the above result and van Benthem's original safety theorem, we have that a first-order formula is safe for directed intuitionistic bisimulations iff it is safe for ordinary bisimulations. There is room for alternative approaches to intuitionistic safety: instead of characterizing the safe first-order definable operations, one can try to characterize the safe *intuitionistically definable* operations. We conjecture that, in contrast with the classical case, the set of intuitionistic formulas that are safe for intuitionistic directed bisimulations does *not* coincide with the set of intuitionistic formulas that are safe for ordinary bisimulations.

To conclude this section we turn to definability.

Theorem 5.6 *Let* K *be a class of pointed intuitionistic models. Then*

1. K *is definable by a set of intuitionistic formulas iff* K *is closed under directed intuitionistic bisimulations and ultraproducts, while* $\overline{\mathsf{K}}$ *is closed ultrapowers.*

2. K *is definable by a single intuitionistic formula iff* K *is closed under directed intuitionistic bisimulations and ultraproducts, while* $\overline{\mathsf{K}}$ *is closed under ultraproducts.*

Proof. Similar to the proof of Theorem 3.17, using Corollary 5.3. See also Rodenburg [18] for related results. ⊣

We leave it to the reader to introduce the notion of a *directed intuitionistic bisimulation up to n*, and to formulate an intuitionistic analogue of Theorem 3.18.

6 Conclusion

In this paper we have introduced the notion of a directed simulation to analyze the expressive power of a number of negation-free description languages for transition systems. Our results concerned preservation, safety and definability aspects of negation-free modal logic and some extensions, and we established similar results for intuitionistic logic. Moreover, our results can also be applied to full modal languages with boolean negation. For example, if a first-order formula is preserved under strong bisimulations, but not under directed simulations, then we know that its modal equivalent must contain negation in an essential way.

To conclude we mention some possibilities for building on the work reported here. The paper is part of a general enterprise that aims to give model-theoretic characterizations of logic-based description formalisms. A lot of work remains to be done, even on arbitrary sub-boolean fragments of first-order logic. More concretely, as mentioned in Section 4 there are several hierarchies of terminological languages waiting to be analyzed using the tools of this paper, A second example concerns the study of expressiveness of feature logics touched upon in §3.2; this theme is developed in a separate paper [16]. Third, one can build on recent work on general fragments of first-order logic, including finite-variable fragments (see [1]), and develop the theory of their negation-free fragments. And a fourth line concerns negation-free substructural logics; there is a close relation there between notions of directed simulation and generative capacities of various formal languages (see [14]), and we plan to report on this in a future paper.

A Characterizing Continuity

This appendix is devoted to a proof of the following result from §3.2.

Lemma 3.15 *A negation-free formula is continuous in p iff it is equivalent to a disjunction of formulas of the form*

$$\phi_0 \wedge \Diamond(\phi_1 \wedge \cdots \wedge \Diamond(\phi_n \wedge p) \cdots), \tag{3}$$

where each of the formulas ϕ_i is negation-free and p-free in the sense that they don't contain occurrences of p.

We need two technical lemmas.

Lemma A.1 *Every transition system M, w is bisimilar to an intransitive tree-like transition system M', w whose root is w.*

Proof. See, for example, [17, Proposition 4.5]. ⊣

To state the second technical lemma we need some notation. Fix a proposition letter p. We write $\underline{\rightarrow}^-$ to denote the existence of a directed simulation for the language without the proposition letter p (exactly which proposition letter is meant will be clear in the applications of the lemma).

Lemma A.2 *Assume $Z : M, w_0 \underline{\rightarrow}^- N, v_0$, where M, N are intransitive tree-like transition systems with $w_0 R \cdots R w_n$ (in M), $v_0 R \cdots R v_n$ (in N) and $w_i Z v_i$ ($1 \leq i \leq n$). Then there are extensions (M^+, w) of (M, w) and (N^+, v) of (N, v) (i.e., $|M^+| \supseteq |M|$ and $|N^+| \supseteq |N|$) such that*

$$\begin{array}{ccc} (M, w) & Z : \underline{\rightarrow}^- & (N, v) \\ \underline{\leftrightarrow} \Big| & & \Big| \underline{\leftrightarrow} \\ (M^*, w) & Z' : \underline{\rightarrow}^- & (N^*, v). \end{array}$$

where Z' is a bijective function such that $w_i Z' v_i$ ($1 \leq i \leq n$).

Proof. See [3, Chapter 5] or [17, Sections 4 and 7] for similar results. ⊣

Proof of Lemma 3.15. We only prove the hard direction. Assume that ϕ is continuous in p. Define

$$\Delta := \bigvee\{\psi \mid \psi \text{ is of the form (3) and } \psi \models \phi\}.$$

We will show that $\phi \models \Delta$; then, by compactness ϕ is equivalent to a finite disjunction of formulas of the form specified in (3), and this proves the lemma.

So, assume that $M, w_0 \models \phi$; we need to show $M, w_0 \models \Delta$. That is, it suffices to find a formula ψ of the form specified in (3) such that $M, w_0 \models \psi$ and $\psi \models \phi$. Here we go. By Lemma A.1 we may assume that M is an intransitive, tree-like transition system with root w_0. As ϕ is continuous in p, we may also assume that $V(p)$ is just a singleton w_n:

$$w_0 \xrightarrow{R} w_1 \xrightarrow{R} \cdots \xrightarrow{R} w_n \models p.$$

Consider the following negation-free description of the above path leading up to w_n:

$$\begin{aligned} \Phi \;=\; & \{ST_{x_i}(\psi) \mid \psi \in \textit{nf-tp}^-(w_i) \text{ and } 0 \leq i \leq n\} \\ & \cup \{Rx_ix_{i+1} \mid 0 \leq i \leq n-1\} \cup \{Px_n\}, \end{aligned}$$

where we use the superscript $-$ in *nf-tp*$^-$ to indicate that the formulas considered are p free. The remainder of the proof is devoted to a proof that $\Phi \models ST_{x_0}(\phi)$, and this will do to prove the lemma. For if $\Phi \models ST_{x_0}(\phi)$, then, for some finite part $\Phi_0 \subseteq \Phi$ we have $\Phi_0 \models ST_{x_0}(\phi)$, by compactness. This is a disjunct in Δ, and, hence, in every model ϕ implies a finite part of Δ, and so ϕ implies in Δ.

To show that $\Phi \models ST_{x_0}(\phi)$ we proceed as follows. Take a transition system N with $N \models \Phi[v_0v_1 \ldots v_n]$; we need to show that $N \models ST(\phi)[v_0]$. Then

(4) $$\textit{nf-tp}^-(w_0) \subseteq \textit{nf-tp}^-(v_0).$$

We may assume that N is an intransitive tree with root v. Take ω-saturated elementary extensions $M^\dagger, w_0$ and $N^\dagger, v_0$ of M, w_0 and N, v_0, respectively. $M^\dagger, w_0$ and $N^\dagger, v_0$ may again be assumed to be intransitive trees with roots w_0 and v_0, respectively.

From (4) we obtain a directed simulation Z such that $Z : M^\dagger, w_i \rightrightarrows^- N^\dagger, v_0$ $(0 \leq i \leq n)$ as in the proof of Theorem 3.5. By Lemma A.2 we can move to bisimilar extensions $M^{\dagger *}$ and $N^{\dagger *}$ of $M^\dagger$ and $N^\dagger$, respectively, and find a *functional* directed simulation Z' linking w_i to v_i (for $1 \leq i \leq n$):

$$\begin{array}{ccc} (M^\dagger, w_0) & Z : \rightrightarrows^- & (N^\dagger, v_0) \\ \leftrightarrow\big| & & \big|\leftrightarrow \\ (M^{\dagger *}, w_0) & Z' : \rightrightarrows^- & (N^{\dagger *}, v_0). \end{array}$$

We will amend the transition systems $M^{\dagger *}$ and $N^{\dagger *}$ as follows. We shrink the interpretation of the proposition letter p so that it only holds at w_i and v_i. This allows us to extend Z' to a full directed simulation Z'' for the whole language:

$$
\begin{array}{rcccl}
 & (M^{\dagger}, w_0) & Z : \rightrightarrows^{-} & (N^{\dagger}, v_0) & \\
 & \underline{\leftrightarrow} \Big| & & \Big| \underline{\leftrightarrow} & \\
 & (M^{\dagger *}, w_0) & Z' : \rightrightarrows^{-} & (N^{\dagger *}, v_0) & \\
\text{Shrink } V(p) & \Big| & & \Big| & \text{Expand } V(p) \\
 & (M^{\dagger **}, w_0) & Z'' : \rightrightarrows & (N^{\dagger **}, v_0). &
\end{array}
$$

By the following chain of steps, we can lift ϕ from M, w_0 to N, v_0:

$$
\begin{array}{rcl}
M, w_0 \models \phi & \Rightarrow & M^{\dagger}, w_0 \models \phi, \text{ by elementary extension} \\
 & \Rightarrow & M^{\dagger *}, w_0 \models \phi \\
 & \Rightarrow & M^{\dagger **}, w_0 \models \phi \text{ by downward monotonicity} \\
 & \Rightarrow & N^{\dagger **}, v_0 \models \phi \text{ by directed similarity} \\
 & \Rightarrow & N^{\dagger *}, v_0 \models \phi \text{ by upward monotonicity} \\
 & \Rightarrow & N^{\dagger}, v_0 \models \phi \\
 & \Rightarrow & N, v_0 \models \phi \text{ by elementary extension.}
\end{array}
$$

This proves the lemma. $\dashv$

References

[1] H. Andréka, J. van Benthem, and I. Németi. Back and Forth between Modal Logic and Classical Logic. *Journal of the IGPL*, 3:685–720, 1995.

[2] F. Baader. A Formal Definition for the Expressive Power of Terminological Knowledge Representation Languages. *Journal of Logic and Computation*, 6:33–54, 1996.

[3] J. van Benthem. *Exploring Logical Dynamics*. Studies in Logic, Language and Information. CSLI Publications, Stanford, 1996. To appear.

[4] J. van Benthem and J. Bergstra. Logic of Transition Systems. *Journal of Logic, Language and Information*, 3:247–283, 1994.

[5] J. van Benthem, J. van Eijck, and V. Stebletsova. Modal Logic, Transition Systems and Processes. *Journal of Logic and Computation*, 4:811–855, 1994.

[6] C.C. Chang and H.J. Keisler. *Model Theory*. North-Holland, Amsterdam, 1973.

[7] F.M. Donini, M. Lenzerini, D. Nardi, and A. Schaerf. Reasoning in Description Logics. In G. Brewka, editor, *Principles of Knowledge Representation*. Studies in Logic, Language and Information. CSLI Publications, Stanford, 1996. To appear.

[8] J.M. Dunn. Positive Modal Logic. *Studia Logica*, 55:301–317, 1995.

[9] M. Hennessy and R. Milner. Algebraic Laws for Indeterminism and Concurrency. *Journal of the ACM*, 32:137–162, 1985.

[10] W. Hodges. *Model Theory*. Cambridge University Press, 1993.

[11] W. van der Hoek and M. de Rijke. Counting Objects. *Journal of Logic and Computation*, 5:325–345, 1995.

[12] M. Hollenberg. Hennessy-Milner Classes and Process Algebra. In A. Ponse, M. de Rijke, and Y. Venema, editors, *Modal Logic and Process Algebra*, pages 187–216. CSLI Publications, 1995.

[13] M. Hollenberg. Bisimulation for Intersection. Manuscript, Department of Philosophy, Utrecht University, 1996.

[14] N. Kurtonina. *Frames and Labels*. PhD Thesis, OTS, Utrecht/ILLC, University of Amsterdam, 1995.

[15] N. Kurtonina and M. de Rijke. Bisimulations for Temporal Logic. Research report RR304, Department of Computer Science, University of Warwick, 1996.

[16] N. Kurtonina and M. de Rijke. Analyzing Feature Logic. Manuscript, Department of Computer Science, University of Warwick, 1996.

[17] M. de Rijke. Modal Model Theory. *Annals of Pure and Applied Logic*, to appear.

[18] P. Rodenburg. *Intuitionistic Correspondence Theory*. PhD Thesis, Department of Mathematics, University of Amsterdam, 1986.

[19] W.C. Rounds. Feature Logics. In J. van Benthem and A. ter Meulen, editors, *Handbook of Logic and Language*, Elsevier Science, Amsterdam, 1996. To appear.

[20] K. Schild. A Correspondence Theory for Terminological Logics. In *Proceedings 12th IJCAI*, pages 466–471, Sydney, 1991.

[21] S. Vickers. *Topology via Logic*. Tracts in Theoretical Computer Science, Vol. 5. Cambridge University Press, Cambridge, 1989.