Journal of Applied Non–Classical Logics
Vol. 2 – n°2, pp.157 – 180, Hermes, Paris (1992)

EXPRESSIVE COMPLETENESS OF TEMPORAL LOGIC OF TREES

BERND–HOLGER SCHLINGLOFF
Institut für Informatik der Technischen Universität München
Orleansstr. 34, D—81667 München, Germany

Many temporal and modal logic languages can be regarded as subsets of first order logic, i.e. the semantics of a temporal logic formula is given as a first order condition on points of the underlying models (Kripke structures). Often the set of possible models is restricted to models which are trees. A temporal logic language is (first order) expressively complete, if for every first order condition for a node of a tree there exists an equivalent temporal formula which expresses the same condition. In this paper expressive completeness of the temporal logic language with the set of operators $\mathcal{U}$ (until), $\mathcal{S}$ (since), and $\mathcal{X}_k$ (k–next) is proved, and the result is extended to various other tree–like structures.

Keywords: Temporal logic, modal logic, expressive completeness, branching time logic, trees

0 Introduction

During the past decade temporal logic has turned out to be an adequate tool for expressing properties which depend on the flow of time. The variable–free operator formalism mostly is more convenient than the usual first or second order logic notation as a means of formalization. However, depending on the structure of time, not all first order logic statements may be expressible in temporal logic. Kamp[1] proved that for continuous linear flow of time every first order formula with exactly one free variable can be translated into the temporal logic language with the operators $\mathcal{U}$ (until) and $\mathcal{S}$ (since) . Gabbay[2] showed that for arbitrary branching time no finite set of operators exists to express every first order property. Kozen and Immerman[3] gave a semantical proof that for b–bounded branching trees (b finite) there must exist a set of $(b+1)$–dimensional expressively complete operators. Hafer and Thomas[4] showed that for binary trees ($b = 2$) every variable free second order formula (with second order quantification restricted to path quantifiers) can be translated into the temporal logic CTL*. In this paper we combine and extend the above used methods and results to show that for b–bounded branching time first order logic is expressively equivalent to temporal logic with $\mathcal{S}$,$\mathcal{U}$ and special "nexttime"–operators $\mathcal{X}_1, \ldots, \mathcal{X}_b$. These operators allow to count the number of different successors of a node with the same label. This result can be extended to ordered trees and to trees with b distinguished successor relations. The proof proceeds via a so–called two dimensional temporal logic with the operators $\mathcal{S}$, $\mathcal{U}$, $\mathcal{X}_k$ and $\mathcal{U}!$. Though two dimensional formulae can be syntactically transformed into the one dimensional logic with $\mathcal{S}$, $\mathcal{U}$ and $\mathcal{X}_k$, the two dimensional logic seems to be of some interest of its own, since it allows the convenient specification of "interval properties".

This paper is organized as follows: In section 1, basic definitions and lemmas are given. In section 2, expressive completeness of the two dimensional logic is proved. In section 3, a syntactical transformation of two dimensional to one dimensional formulae is given. In section 4, extensions and limitations of our methods are discussed.

1 Definitions

Definition 1.1 Let $b < \omega$ be a finite number. A b–ary tree $\langle \mathcal{N}, S \rangle$ is a set of nodes $\mathcal{N}$ together with a successor relation $S \subset \mathcal{N} \times \mathcal{N}$ such that

- for every node x there are at most b successor nodes, i.e. nodes y with xSy
- for every node x there is at most one predecessor node, i.e. a node y with ySx
- there is a root node r without predecessor, such that every node can be reached in finitely many S–steps from r.

Let S^+ be the transitive and S^* be the reflexive and transitive closure of S.

Definition 1.2 Let $\Im = \{x_0, \ldots, x_k\}$ be a set of individual variables and $\mathcal{P} = \{p_1, \ldots, p_m\}$ be a set of (monadic) predicate symbols. The language $\mathbf{PL}^n(\mathcal{P}, \Im)$ of first order predicate logic wich uses at most the predicate symbols $\mathcal{P}$, free variables $\Im$ and has at most quantifier depth n is defined as follows:

- $xS^*y \in \mathbf{PL}^n(\mathcal{P}, \Im \cup \{x, y\})$ for every n, $\mathcal{P}$, $\Im$.
- $p(x) \in \mathbf{PL}^n(\mathcal{P} \cup \{p\}, \Im \cup \{x\})$ for every n, $\mathcal{P}$, $\Im$.
- $\bot \in \mathbf{PL}^n(\mathcal{P}, \Im)$ for every n, $\mathcal{P}$, $\Im$.
- If $A \in \mathbf{PL}^{n_1}(\mathcal{P}_1, \Im_1)$, $B \in \mathbf{PL}^{n_2}(\mathcal{P}_2, \Im_2)$, then $(A \rightarrow B) \in \mathbf{PL}^{\max(n_1,n_2)}(\mathcal{P}_1 \cup \mathcal{P}_2, \Im_1 \cup \Im_2)$
- If $A \in \mathbf{PL}^n(\mathcal{P}, \Im)$ and $x \in \Im$, then $\exists x(A) \in \mathbf{PL}^{n+1}(\mathcal{P}, \Im \setminus \{x\})$

Let $\mathbf{PL}(\mathcal{P}, \Im) = \bigcup_{n\in\omega} \mathbf{PL}^n(\mathcal{P}, \Im)$.
We write $\mathbf{PL}^n(\mathcal{P}, x_0, x_1, ...)$ for $\mathbf{PL}^n(\mathcal{P}, \{x_0, x_1, ...\})$. The free variables $x_0, x_1, \ldots$ of a formula are also called its parameters.
Additional junctors $\top, \neg, \wedge, \vee, \leftrightarrow, \bigwedge, \bigvee, \forall$ are introduced as abbreviations as usual. Superfluous brackets are usually omitted.

Definition 1.3 Let $A \in \mathbf{PL}(\mathcal{P}, \Im)$. A model (also called (Kripke–)structure) $\$ = \langle \mathbf{B}, \eta, \xi \rangle$ for A consists of a tree $\mathbf{B} = \langle \mathcal{N}, S \rangle$, an interpretation $\eta : \mathcal{P} \rightarrow 2^{\mathcal{N}}$ for the predicate symbols and an interpretation $\xi : \Im \rightarrow \mathcal{N}$ for the free individual variables. The forcing relation $\models$ between models and formulae is defined as usual such that the relation symbol S^* is interpreted as the reflexive and transitive closure of the successor relation S.

Additional relations $=, \neq, S^+, S$ are introduced as abbreviations via $x = y$ if $xS^*y \wedge yS^*x$, $x \neq y$ if $\neg x = y$, xS^+y if $xS^*y \wedge x \neq y$, xSy if $xS^+y \wedge \neg\exists z(xS^+z \wedge zS^+y)$.
We write $\langle \mathbf{B}, \eta, a_0, a_1, ... \rangle \models \varphi(x_0, x_1, ...)$, if $\Im = \{x_0, x_1, ...\}$ and $\xi(x_0) = a_0$, $\xi(x_1) = a_1, \ldots$. Often we name nodes with the same letters x, y,... as variables and let ξ be the identity function.

Languages on a finite signature $(\mathcal{P}, \Im)$ with finite quantifier depth are essentially finite:

Lemma 1.4 For every n, $\mathcal{P}$, $\Im$ there is a *finite* set $\theta \subset \mathbf{PL}^n(\mathcal{P}, \Im)$ such that every formula from $\mathbf{PL}^n(\mathcal{P}, \Im)$ is equivalent to a formula from θ.

The proof of this lemma is standard and can e.g. be found in [3].

Definition 1.5 Let $\mathbf{O} = \{\mathcal{O}_1^{(i_1)}, \ldots, \mathcal{O}_n^{(i_n)}\}$ be a set of operators $\mathcal{O}_j$ with arities i_j and $\mathcal{P} = \{p_1, \ldots, p_m\}$ be a set of propositional variables. The language $\mathbf{TL}(\mathbf{O}, \mathcal{P})$ of temporal logic is defined by

- If $p \in \mathcal{P}$, then $p \in \mathbf{TL}(\mathbf{O}, \mathcal{P})$
- $\bot \in \mathbf{TL}(\mathbf{O}, \mathcal{P})$
- If $A, B \in \mathbf{TL}(\mathbf{O}, \mathcal{P})$, then $(A \rightarrow B) \in \mathbf{TL}(\mathbf{O}, \mathcal{P})$
- If $\mathcal{O}_j^{(i)} \in \mathbf{O}$ and $A_1, \ldots, A_i \in \mathbf{TL}(\mathbf{O}, \mathcal{P})$, then $\mathcal{O}_j(A_1, \ldots, A_i) \in \mathbf{TL}(\mathbf{O}, \mathcal{P})$.

The semantics of $\mathbf{TL}(\mathbf{O}, \mathcal{P})$ is given by the semantics of the operators:

Definition 1.6 Let for every i–ary Operator $\mathcal{O} \in \mathbf{O}$ a formula $\varphi_{\mathcal{O}} \in \mathbf{PL}(\mathcal{P}, x_0)$ (its table) be given. Then a translation $\tau : \mathbf{TL}(\mathbf{O}, \mathcal{P}) \rightarrow \mathbf{PL}(\mathcal{P}, x_0)$ can be defined by

- $(p_j)^\tau = p_j(x_0)$
- $(\bot)^\tau = \bot$
- $(A \rightarrow B)^\tau = (A)^\tau \rightarrow (B)^\tau$
- $(\mathcal{O}_j(A_1, \ldots, A_i))^\tau = \varphi_{\mathcal{O}}\left({}^{p_1(y)}/_{(A_1)^\tau(^{x_0}/_y)}, \ldots, {}^{p_i(y)}/_{(A_i)^\tau(^{x_0}/_y)}\right)$

Here ${}^{p_k(y)}/_{(A_k)^\tau(^{x_0}/_y)}$ means that every occurrence of p_k with parameter y is replaced by the formula $(A_k)^\tau$, where the parameter y of p_k is substituted for the free variable x_0 of $(A_k)^\tau$. When substituting inside the scope of a quantification bound variables may have to be renamed.
Note that $\bot(^{x_0}/_y) \equiv \bot$.

Example 1.7 Let the operators $\mathcal{U}$ (until), $\mathcal{S}$ (since), and $\mathcal{X}_k$ (k–next) be defined by the following tables:

$$\varphi_{\mathcal{U}} = \exists y\big[x_0 S^+ y \wedge p_1(y) \wedge \forall z(x_0 S^+ z \wedge z S^+ y \rightarrow p_2(z))\big]$$
$$\varphi_{\mathcal{S}} = \exists y\big[y S^+ x_0 \wedge p_1(y) \wedge \forall z(y S^+ z \wedge z S^+ x_0 \rightarrow p_2(z))\big]$$
$$\varphi_{\mathcal{X}_k} = \exists y_1, \ldots, y_k \bigwedge_i \big[x_0 S y_i \wedge p_1(y_i) \wedge \bigwedge_{j \neq i}(y_i \neq y_j)\big]$$

The table of $\mathcal{X}_k$ defines an operator for every k between 1 and b; whenever we write $\mathbf{TL}(..., \mathcal{X}_k)$ we mean that all operators $\mathcal{X}_1, \ldots, \mathcal{X}_b$ are present.

The $\mathcal{X}_k$–operators allow to count the number of different successors of x_0 with the same label; e.g.

$$\begin{aligned}
\big(p \wedge \mathcal{S}(\mathcal{X}_2 p, \bot)\big)^\tau &\equiv p(x_0) \wedge \exists y\Big[y S^+ x_0 \wedge (\mathcal{X}_2 p)^\tau(^{x_0}/_y) \wedge \forall z\big(y S^+ z \wedge z S^+ x_0 \rightarrow \bot(^{x_0}/_z)\big)\Big] \\
&\equiv p(x_0) \wedge \exists y\big[y S x_0 \wedge \exists y_1, y_2(y S y_1 \wedge y S y_2 \wedge y_1 \neq y_2 \wedge p(y_1) \wedge p(y_2))\big] \\
&\leftrightarrow p(x_0) \wedge \exists y, y_1\big[y S x_0 \wedge y S y_1 \wedge y_1 \neq x_0 \wedge p(y_1)\big]
\end{aligned}$$

means that besides x_0 there is another successor of x_0's predecessor which satifies p. Similarly

$$\mathcal{S}(\mathcal{X}_1 p, \bot) \wedge \big(p \rightarrow \mathcal{S}(\mathcal{X}_2 p, \bot)\big)$$

means that there is another successor of x_0's predecessor which satisfies p.
Note that $\mathcal{X}_1(A)$ can be defined as $\mathcal{U}(A, \bot)$. Similarly we write $\mathcal{Y}(A)$ for $\mathcal{S}(A, \bot)$.

Let $F \in \mathbf{TL}(\mathbf{O}, \mathcal{P})$ and $\$ = \langle \mathbf{B}, \eta, a \rangle$. Validity of F in $\$$ is defined by $\$ \models F$ if $\$ \models F^\tau$. $F \in \mathbf{TL}(\mathbf{O}, \mathcal{P})$ is equivalent to $\varphi \in \mathbf{PL}(\mathcal{P}, x_0)$ if for every $\$ = \langle \mathbf{B}, \eta, a \rangle$ it holds that $\$ \models F$ iff $\$ \models \varphi$.
So by definition, for every formula $F \in \mathbf{TL}(\mathbf{O}, \mathcal{P})$ there exists an equivalent formula $\varphi_F \in \mathbf{PL}(\mathcal{P}, x_0)$. Expressive completeness means the existence of a translation in the opposite direction:

Definition 1.8 A set of operators $\mathbf{O}$ is expressively (or functional) complete, if for every formula $\varphi \in \mathbf{PL}(\mathcal{P}, x_0)$ there exists a formula $F_\varphi \in \mathbf{TL}(\mathbf{O}, \mathcal{P})$ equivalent to φ.

A famous result in this context is Kamps Theorem[1]:

Theorem 1.9 If $b = 1$, then $\{\mathcal{U}, \mathcal{S}\}$ is expressively complete.

Gabbay[2] sharpened this result by proving:

Theorem 1.10 If $b = 1$, then for every formula of $\mathbf{TL}(\mathcal{U}, \mathcal{S}, \mathcal{P})$ there exists an equivalent boolean combination of formulae from $\mathbf{TL}(\mathcal{U}, \mathcal{P})$ and $\mathbf{TL}(\mathcal{S}, \mathcal{P})$.
Hence for $b = 1$, $\{\mathcal{U}\}$ is expressively complete, if we restrict all quantifiers $\exists y$ to nodes y with $x_0 S^* y$.

The above definition of τ throws every temporal formula onto a predicate logic formula with *monadic* predicate symbols $\mathcal{P}$ and *one* free variable x_0. A two dimensional temporal logic is defined by operator tables using $\mathcal{P}$ as *dyadic* predicate symbols and *two* free variables x_0, x_1. The appropriate translation function for two dimensional operators is defined by

$$(\mathcal{O}_j(A_1, \ldots, A_i))^\tau = \varphi_{\mathcal{O}}\big({}^{p_1(y_0, y_1)}/_{(A_1)^\tau(x_0, x_1/_{y_0, y_1})}, \ldots, {}^{p_i(y_0, y_1)}/_{(A_i)^\tau(x_0, x_1/_{y_0, y_1})}\big).$$

In the following example a two dimensional redefinition of the operators $\mathcal{U}, \mathcal{S}$ and $\mathcal{X}_k$ is given and a new operator $\mathcal{U}!$ is defined:

Example 1.11

$$\begin{aligned}
\varphi_{\mathcal{U}!} &= \exists y\big[x_0 S^+ y \wedge (x_1 S^* y \vee y S^+ x_1) \wedge p_1(y, x_1) \wedge \forall z(x_0 S^+ z \wedge z S^+ y \rightarrow p_2(z, x_1))\big] \\
\varphi_{\mathcal{U}} &= \exists y\big[x_0 S^+ y \wedge p_1(y, y) \wedge \forall z(x_0 S^+ z \wedge z S^+ y \rightarrow p_2(z, y))\big] \\
\varphi_{\mathcal{S}} &= \exists y\big[y S^+ x_0 \wedge p_1(y, x_1) \wedge \forall z(y S^+ z \wedge z S^+ x_0 \rightarrow p_2(z, x_1))\big] \\
\varphi_{\mathcal{X}_k} &= \exists y_1, \ldots, y_k \textstyle\bigwedge_i \big[x_0 S y_i \wedge \bigwedge_{j \neq i}(y_i \neq y_j) \wedge p_1(y_i, y_i)\big]
\end{aligned}$$

So e.g. the translation of $\mathcal{U}\big(p, q \rightarrow \mathcal{U}!(r, s)\big)$ becomes:

$$\begin{aligned}
\mathcal{U}\big(p, q \rightarrow \mathcal{U}!(r, s)\big)^\tau &\equiv \exists y\Big[x_0 S^+ y \wedge p(y, y) \wedge \forall z\big\langle x_0 S^+ z \wedge z S^+ y \rightarrow \big(q(z, y) \rightarrow \mathcal{U}!(r, s)^\tau(x_0, x_1/_{z, y})\big)\big\rangle\Big] \\
&\equiv \exists y\Big[x_0 S^+ y \wedge p(y, y) \wedge \forall z\big\langle x_0 S^+ z \wedge z S^+ y \wedge q(z, y) \rightarrow \\
&\rightarrow \exists y'\big[z S^+ y' \wedge (y S^* y' \vee y' S^+ y) \wedge r(y', y) \wedge \forall z'(z S^+ z' \wedge z' S^+ y' \rightarrow s(z', y))\big]\big\rangle\Big]
\end{aligned}$$

Definition 1.12 Let **O** be a set of two dimensional operators. The projection F^ρ of a $\mathbf{TL}(\mathbf{O},\mathcal{P})$–formula F is the $\mathbf{PL}(\mathcal{P},x_0)$–formula obtained by replacing in F^τ every dyadic predicate $p(y_1,y_2)$ by $p(y_1)$, and every occurrence of the free variable x_1 by x_0. A $\mathbf{TL}(\mathbf{O},\mathcal{P})$–formula F is valid in a model $\$ = \langle \mathbf{B},\eta,a\rangle$ if its projection F^ρ is valid in $\$$. Again F is equivalent to F' if F and F' are valid in the same models.

So the meaning of the above formula $\mathcal{U}(p, q \rightarrow \mathcal{U}!(r,s))$ is: There is a p–labelled node y such that on the path from x_0 to y for every q–labelled node z there is an r–labelled node y' *on this path to* y *or beyond* y such that between z and y' the predicate s holds.
The difference to the one dimensional formula $\mathcal{U}(p, q \rightarrow \mathcal{U}(r,s))$ can be graphically illustrated as in figure 1.

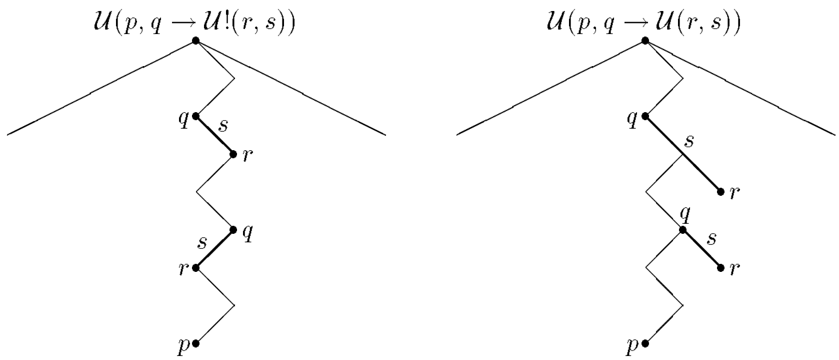


Figure 1: The difference between $\mathcal{U}$– and $\mathcal{U}!$–operator

Hence the operator $\mathcal{U}!$ can be seen as a kind of "path operator" which looks only in a given direction. The definitions of all of the other operators from example 1.11 are tailored to this intended meaning: A $\mathcal{U}$ defines a new direction, $\mathcal{S}$ leaves the given direction unchanged, while $\mathcal{X}_k$ eliminates the direction without defining a new one.

The reader may ask why in definition 1.11 the same operator symbols were chosen as in example 1.7. The reason can be found in the following lemma:

Lemma 1.13 For $F \in \mathbf{TL}(\mathcal{U}!,\mathcal{S},\mathcal{U},\mathcal{X}_k,\mathcal{P})$ define $F^{onedim} \in \mathbf{TL}(\mathcal{S},\mathcal{U},\mathcal{X}_k,\mathcal{P})$ as the result of eliminating every exclamation mark from F.
If in F every occurrence of a $\mathcal{U}!$–operator inside the scope of a $\mathcal{U}$– or $\mathcal{S}$–operator $\mathcal{O}$ is in the scope of an $\mathcal{X}_k$–operator which is also inside the scope of $\mathcal{O}$, then F is equivalent to F^{onedim}. Especially formulae without $\mathcal{U}!$–operators inside of $\mathcal{U}$– or $\mathcal{S}$–operators and formulae with no $\mathcal{U}!$ at all are equivalent to their one dimensional counterparts.

Proof: If there are no $\mathcal{U}!$–operators in F, the proof is immediate from the definition. If a $\mathcal{U}!$–operator in F satisfies the above condition, then either the first and second parameter are the same variable (if $\mathcal{U}!$ is inside of $\mathcal{X}_k$), or the second parameter is constantly x_1 (if $\mathcal{U}!$ is nested inside of $\mathcal{U}!$). Since in the one dimensional interpretation x_1 is identified with x_0, the additional condition $x_1 S^* y \vee y S^* x_1$ in the definition of $\mathcal{U}!$ is in both cases satisfied whenever an appropriate y can be found. So $\mathcal{U}!$ is equivalent to $\mathcal{U}$.

□

2 Two dimensional expressive completeness

The following proof is close to the proof by Hafer and Thomas[4]. Let $\mathcal{P}$ be fixed for this section, and $\theta^n = \{\psi_1(x_0), \ldots, \psi_k(x_0)\}$ be the finite set of formulae of $\mathbf{PL}^n(\mathcal{P}, x_0)$ guaranteed by lemma 1.4. Let $\mathcal{T}^n = \{p_{11}, \ldots, p_{1k}, p_{21}, \ldots, p_{bk}\}$ be $b * k$ new predicate symbols not in $\mathcal{P}$, and $\mathcal{P}^n = \mathcal{P} \cup \mathcal{T}^n$.

Definition 2.1 Let $\mathbf{B} = \langle \mathcal{N}, S \rangle$ be a tree and $\eta : \mathcal{P} \to 2^{\mathcal{N}}$ be an interpretation for $\mathcal{P}$. Then the <u>n–augmentation</u> η^n is the extension of η to domain $\mathcal{P}^n$, which satisfies the following condition for all $p_{ij} \in \mathcal{T}^n$ and all $a \in \mathcal{N}$:

$$\langle \mathbf{B}, \eta^n, a \rangle \models p_{ij}(x_0) \quad \text{iff} \quad \langle \mathbf{B}, \eta, a \rangle \models \exists y_1, \ldots, y_i \bigwedge_{\mu \neq \nu} [x_0 S y_\mu \wedge (y_\mu \neq y_\nu) \wedge \psi_j(y_\mu)]$$

This means that $p_{ij} \in \mathcal{T}^n$ is true in a node a if $\psi_j \in \theta^n$ is true in at least i successors of a. If $\$ = \langle \mathbf{B}, \eta, \xi \rangle$ is a model for $\mathbf{PL}(\mathcal{P}, \Im)$, then the n–augmented model $\$^n = \langle \mathbf{B}, \eta^n, \xi \rangle$ is a model for $\mathbf{PL}(\mathcal{P}^n, \Im)$.

Definition 2.2 Let $\$^n = \langle \mathbf{B}, \eta^n, a_0, \ldots, a_k \rangle$ be an n–augmented model. The <u>bough free</u> n–augmented model is $\hat{\hat{\$}}^n = \langle \hat{\hat{\mathbf{B}}}, \hat{\hat{\eta}}^n, a_0, \ldots, a_k \rangle$. Here $\hat{\hat{\mathbf{B}}}$ consists only of those nodes a of $\mathbf{B}$ for which $aS^* a_i$ for some $a_i \in \{a_0, \ldots, a_k\}$, the successor relation S on nodes is restricted appropriately, and $\hat{\hat{\eta}}^n$ is η^n with appropriately restricted range.

For $\$^n = \langle \mathbf{B}, \eta^n, x, y \rangle$, $\hat{\hat{\mathbf{B}}}$ must have one of the three forms indicated in figure 2.

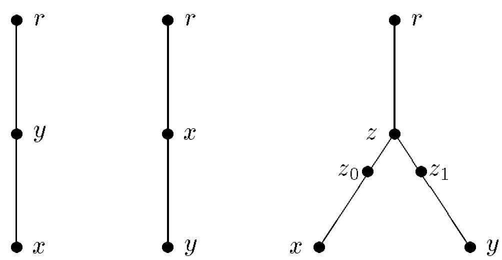


Figure 2: Three possible cases of bough free models for formulae with two parameters

The bough free n–augmented model contains all the information of the original model:

Lemma 2.3 Let $\Im = \{x_0, \ldots, x_k\}$. For every formula $\varphi_1(x_0, \ldots, x_k) \in \mathbf{PL}^n(\mathcal{P}, \Im)$ there is a formula $\varphi_2(x_0, \ldots, x_k) \in \mathbf{PL}^n(\mathcal{P}^n, \Im)$ such that for every model $\$$ it holds that $\$ \models \varphi_1$ iff $\hat{\hat{\$}}^n \models \varphi_2$.

The proof can be found in [6]. It is obtained by an appropriate version of the so–called Ehrenfeucht–Fraissé–game.

Definition 2.4 Let the <u>restriction</u> $\varphi^{[x_0, x_1]}$ of a formula $\varphi \in \mathbf{PL}(\mathcal{P}^n, x_0, x_1)$ to $[x_0, x_1]$ be the formula obtained by replacing every quantifier $\exists y(\ldots)$ in φ by $\exists y(x_0 S^+ y \wedge y S^+ x_1 \wedge \ldots)$.

The <u>relevant</u> nodes $a_0, \ldots, a_k$ with respect to nodes $b_0, \ldots, b_l$ of a bough free model are $b_0, \ldots, b_l$ as well as the root and all branching nodes (with more than one successor). Let $\{a_0, \ldots, a_k\}$ be the relevant nodes of a bough free model. Then there are exactly k tuples $T = \{t_1, \ldots, t_k\}$ such that $t_j = [a_{j_1}, a_{j_2}]$ and $a_{j_1} S^+ a_{j_2}$

and no other relevant node lies in between a_{j_1} and a_{j_2}. Note that the set of nodes between a_{j_1} and a_{j_2} is linearly ordered by S^*.

The following lemma is an extension of theorem 1 (p. 48) from Kamp[1] for the non linear case. A similar lemma can be found in Gabbay, Pnueli, Shelah, Stavi[5]. It shows that formulae of first order logic, speaking about bough free models, can be mapped onto formulae speaking about the points and linear parts which constitute this model.

Lemma 2.5 Let $\Im = \{x_0, \ldots, x_k\}$. For every formula $\varphi \in \mathbf{PL}^n(\mathcal{P}^n, \Im)$ there is a quantifier free formula $\psi \in \mathbf{PL}^0(\mathcal{P}^n, \Im)$ and k formulae $\psi_1^{[x_0,x_1]}, \ldots, \psi_k^{[x_0,x_1]} \in \mathbf{PL}^n(\mathcal{P}^n, x_0, x_1)$ such that for all bough free models $\widehat{\widehat{\$}}^n = \langle \widehat{\widehat{\mathbf{B}}}, \widehat{\widehat{\eta}}^n, a_0, \ldots, a_k \rangle$ with relevant nodes $\{a_0, \ldots, a_k\}$ it holds that

$$\begin{array}{lll} \langle \widehat{\widehat{\mathbf{B}}}, \widehat{\widehat{\eta}}^n, a_0, \ldots, a_k \rangle \models \varphi & \text{iff} \quad \langle \widehat{\widehat{\mathbf{B}}}, \widehat{\widehat{\eta}}^n, a_0, \ldots, a_k \rangle \models \psi & \text{and} \\ & \phantom{\text{iff}} \quad \langle \widehat{\widehat{\mathbf{B}}}, \widehat{\widehat{\eta}}^n, a_{j_0}, a_{j_1} \rangle \models \psi_j^{[x_0,x_1]} & \text{for all } t_j = [a_{j_0}, a_{j_1}] \in T. \end{array}$$

The proof of this lemma can be found in [6]. It is again done by the Ehrenfeucht–Fraissé–game. Note that lemma 2.5 depends on the fact that the underlying structures are trees; the argument is not valid for general structures.

For the proof of the following theorem we only need a special case of the above lemma.

Theorem 2.6 $\{\mathcal{U}!, \mathcal{S}, \mathcal{U}, \mathcal{X}_k\}$ is expressively complete.

Proof: We show by induction on n: for every $\varphi \in \mathbf{PL}^n(\mathcal{P}, x)$ there is a formula $F_\varphi \in \mathbf{TL}(\mathcal{U}!, \mathcal{S}, \mathcal{U}, \mathcal{X}_k, \mathcal{P})$ such that for every $\$$, $\$ \models \varphi$ iff $\$ \models F_\varphi$.

Case $n = 0$ is trivial ($p(x)$ becomes p, $\perp$ becomes $\perp$, $\rightarrow$ becomes $\rightarrow$). Since both languages are closed under boolean combinations, it suffices in the inductive step to consider $\varphi \equiv \exists y \psi(x, y)$ with $\psi \in \mathbf{PL}^n(\mathcal{P}, x, y)$.
φ is equivalent to the disjunction $\varphi_1 \vee \varphi_2 \vee \varphi_3$, where

$$\begin{array}{l} \varphi_1 \equiv \exists y \exists r(\mathrm{root}(r) \wedge rS^*y \wedge yS^*x \wedge \psi(x, y)) \\ \varphi_2 \equiv \exists y \exists r(\mathrm{root}(r) \wedge rS^*x \wedge xS^*y \wedge \psi(x, y)) \\ \varphi_3 \equiv \exists y \exists r z z_0 z_1(\mathrm{root}(r) \wedge rS^*z \wedge zSz_0 \wedge zSz_1 \wedge z_1 \neq z_2 \wedge z_0S^*x \wedge z_1S^*y \wedge \psi(x, y)) \end{array}$$

Here $\mathrm{root}(r)$ means $\neg\exists y(yS^+r)$. These cases correspond to the three cases of figure 2.
For φ_1, using lemma 2.3 and lemma 2.5, we can find formulae $\psi(x, y, r) \equiv \psi_x(x) \wedge \psi_y(y) \wedge \psi_r(r)$, $\psi^{[r,y]}(r, y)$, and $\psi^{[y,x]}(y, x)$, such that for every model $\$ = \langle \mathbf{B}, \eta, x, y, r \rangle$ and corresponding bough free n–augmented model $\widehat{\widehat{\$}}^n = \langle \widehat{\widehat{\mathbf{B}}}, \widehat{\widehat{\eta}}^n, x, y, r \rangle$,

$$\begin{array}{l} \langle \mathbf{B}, \eta, x, y, r \rangle \models \mathrm{root}(r) \wedge rS^*y \wedge yS^*x \wedge \psi(x, y) \quad \text{iff} \\ \langle \widehat{\widehat{\mathbf{B}}}, \widehat{\widehat{\eta}}^n, x, y, r \rangle \models \mathrm{root}(r) \wedge rS^*y \wedge yS^*x \wedge \psi_x(x) \wedge \psi_y(y) \wedge \psi_r(r) \quad \text{and} \\ \langle \widehat{\widehat{\mathbf{B}}}, \widehat{\widehat{\eta}}^n, r, y \rangle \models \psi^{[r,y]}(r, y) \quad \text{and} \\ \langle \widehat{\widehat{\mathbf{B}}}, \widehat{\widehat{\eta}}^n, y, x \rangle \models \psi^{[y,x]}(y, x). \end{array}$$

This in turn means that there are formulae

$$\begin{array}{l} \widehat{\widehat{\varphi}}_{\widehat{\widehat{11}}}(y) \equiv \exists r(\mathrm{root}(r) \wedge rS^*y \wedge \psi_y(y) \wedge \psi_r(r) \wedge \psi^{[r,y]}(r, y)) \text{ and} \\ \widehat{\widehat{\varphi}}_{\widehat{\widehat{12}}}(x) \equiv \exists y(yS^*x \wedge \psi_x(x) \wedge \widehat{\widehat{\varphi}}_{\widehat{\widehat{11}}}(y) \wedge \psi^{[y,x]}(y, x)), \end{array}$$

such that for every $\$ = \langle \mathbf{B}, \eta, x, y, r \rangle$

$$\begin{array}{l} \$ \models \mathrm{root}(r) \wedge rS^*y \wedge yS^*x \wedge \psi(x, y) \quad \text{iff} \\ \langle \widehat{\widehat{\mathbf{B}}}, \widehat{\widehat{\eta}}^n, y \rangle \models \widehat{\widehat{\varphi}}_{\widehat{\widehat{11}}}(y) \quad \text{and} \\ \langle \widehat{\widehat{\mathbf{B}}}, \widehat{\widehat{\eta}}^n, x \rangle \models \widehat{\widehat{\varphi}}_{\widehat{\widehat{12}}}(x) \end{array}$$

Hence, for every $\$ = \langle \mathbf{B}, \eta, x \rangle$, $\$ \models \varphi_1(x)$ iff $\widehat{\$}^{\widehat{n}} \models \widehat{\varphi_{12}}(x)$.
The formula $\widehat{\varphi_{11}}(y)$ is by Kamp's theorem and Gabbay's extension translatable into $\widehat{F_{\varphi_{11}}} \in \mathbf{TL}(\mathcal{S}, \mathcal{P}^n)$, since it is interpreted on a linear structure. The same argument holds for $\widehat{\varphi_{12}}(x)$, with $\widehat{\varphi_{11}}(y)$ replaced by a new predicate symbol $q(y)$. Now in $\widehat{F_{\varphi_{12}}}$ every occurrence of the new propositional variable q has to be replaced by $\widehat{F_{\varphi_{11}}}$. Call the result of this replacement $\widehat{F_{\varphi_1}}(x)$. Then for every $\$ = \langle \mathbf{B}, \eta, x \rangle$ it holds that $\$ \models \varphi_1(x)$ iff $\widehat{\$}^{\widehat{n}} \models \widehat{F_{\varphi_1}}$. Let F_j be the translation of $\varphi_j(x_0) \in \mathbf{PL}^n(\mathcal{P}, x_0)$, whose existence is guaranteed according to the induction hypothesis. If we replace in $\widehat{F_{\varphi_1}}$ every occurrence of an augmenting variable $p_{ij} \in \mathcal{T}^n$ by $\mathcal{X}_i F_j$, and call the result F_{φ_1} then clearly we have $\widehat{\$}^{\widehat{n}} \models \widehat{F_{\varphi_1}}$ iff $\$ \models F_{\varphi_1}$. Hence $\$ \models \varphi_1$ iff $\$ \models F_{\varphi_1}$ for every model $\$$.

The same considerations yield that φ_2 can be split such that

$$\langle \mathbf{B}, \eta, x, y, r \rangle \models \text{root}(r) \wedge rS^*x \wedge xS^+y \wedge \psi(x,y) \quad \text{iff}$$
$$\langle \widehat{\mathbf{B}}, \widehat{\eta}^{\widehat{n}}, x \rangle \models \exists r(\text{root}(r) \wedge rS^*x \wedge \psi_x(x) \wedge \psi_r(r) \wedge \psi^{[r,x]}(r,x)) \quad \text{and}$$
$$\langle \widehat{\mathbf{B}}, \widehat{\eta}^{\widehat{n}}, x \rangle \models \exists y(xS^+y \wedge \psi_y(y) \wedge \psi^{[x,y]}(x,y)).$$

The first of these formulae can be translated into $F_{21} \in \mathbf{TL}(\mathcal{S}, \mathcal{P}^n)$ as above. The second one can be translated into $\widehat{F_{22}} \in \mathbf{TL}(\mathcal{U}, \mathcal{P}^n)$ according to Kamp's and Gabbay's theorems. But now, in $\widehat{F_{22}}$ every $\mathcal{U}$–operator inside the scope of another $\mathcal{U}$–operator has to be replaced by a $\mathcal{U}!$–operator. This is necessary because we want to interpret the resulting formula not in $\widehat{\$}$ but in $\$$, to make all $\mathcal{U}$–operators "point in the same direction". Finally the augmenting variables have to be replaced as before to yield the formulae F_{21} and F_{22}. If F_2 stands for $(F_{21} \wedge F_{22})$, then again we obtain $\$ \models \varphi_2$ iff $\$ \models F_2$.

A similar argument reduces φ_3 with

$$\langle \mathbf{B}, \eta, x, y, r, z, z_0, z_1 \rangle \models \text{root}(r) \wedge rS^*z \wedge zSz_0 \wedge zSz_1 \wedge z_0 \neq z_1 \wedge z_0S^*x \wedge z_1S^*y \wedge \psi(x,y)$$

to φ_{31}–φ_{34}, where

$$\langle \widehat{\mathbf{B}}, \widehat{\eta}^{\widehat{n}}, z \rangle \models \exists r(\text{root}(r) \wedge rS^*z \wedge \psi_r(r) \wedge \psi_z(z) \wedge \psi^{[r,z]}(r,z)) \quad = \widehat{\varphi_{31}}(z)$$
$$\langle \widehat{\mathbf{B}}, \widehat{\eta}^{\widehat{n}}, z_1 \rangle \models \exists y(z_1S^*y \wedge \psi_1(z_1) \wedge \psi_y(y) \wedge \psi^{[z_1,y]}(z_1,y)) \quad = \widehat{\varphi_{32}}(z_1)$$
$$\langle \widehat{\mathbf{B}}, \widehat{\eta}^{\widehat{n}}, x \rangle \models \exists z_0(z_0S^*x \wedge \psi_2(z_0) \wedge \psi_X(x) \wedge \psi^{[z_0,x]}(z_0,x) \wedge \widehat{\varphi_{33}}(z_0)) \quad = \widehat{\varphi_{34}}(x),$$

with $\widehat{\varphi_{33}} = \exists z z_1(zSz_0 \wedge zSz_1 \wedge z_0 \neq z_1 \wedge \widehat{\varphi_{31}}(z) \wedge \widehat{\varphi_{32}}(z_1))$

For $\varphi_{31}(z)$ there exists a translation $\widehat{F_{31}}$ as above, for $\widehat{\varphi_{32}}(z_1)$ as in the case of φ_{22} a translation $\widehat{F_{32}}$. We can translate $\widehat{\varphi_{34}}(x)$ by replacing $\widehat{\varphi_{33}}(z_0)$ by a new predicate symbol $q(z_0)$ and get a formula $\widehat{F_{34}}$. Now we have to replace every inner $\mathcal{U}$ by $\mathcal{U}!$ in $\widehat{F_{32}}$, as well as the augmenting variables in $\widehat{F_{31}}$, $\widehat{F_{32}}$ and $\widehat{F_{34}}$. The resulting formulae F_{31}, F_{32} and F_{34} can be combined to yield the intended formula F_3 as follows: Every occurrence of q in F_{34} is replaced by $\mathcal{Y}F_{31} \wedge \mathcal{Y}\mathcal{X}_1F_{32} \wedge (F_{32} \rightarrow \mathcal{Y}\mathcal{X}_2F_{32})$. (Compare this formula with the one after example 1.7!)

Summarizing the achieved translations we have for every model $\$$:

$$\$ \models \varphi \quad \text{iff} \quad \$ \models \varphi_1 \vee \varphi_2 \vee \varphi_3 \quad \text{iff} \quad \$ \models F_1 \vee F_2 \vee F_3$$

□

3 One dimensional expressive completeness

Though the operator $\mathcal{U}!$ is rather easy to understand, from a theoretical viewpoint it seems not satisfying to have only a two dimensional expressively complete logic.

In this section we therefore show how to eliminate the $\mathcal{U}!$–operator from formulae by syntactical transformations.

Lemma 3.1 Let $F \equiv \big(B_1 \wedge \mathcal{U}!(C,D)\big) \vee \big(B_2 \wedge \neg\mathcal{U}!(C,D)\big)$, and

$$\mathrm{End}_1 \equiv B_1 \wedge \neg B_2 \wedge \mathcal{U}\Big[\big(B_1 \wedge B_2 \wedge C \wedge \mathcal{U}(A, B_1 \wedge B_2)\big) \vee \big(A \wedge \langle C \vee D \wedge \mathcal{U}(C,D)\rangle\big), B_1 \wedge B_2 \wedge D\Big]$$

$$\mathrm{End}_2 \equiv \neg B_1 \wedge B_2 \wedge \mathcal{U}\Big[\big(B_1 \wedge B_2 \wedge \neg C \wedge \neg D \wedge \mathcal{U}(A, B_1 \wedge B_2)\big) \vee \vee\big(A \wedge \neg C \wedge \langle \neg D \vee \neg\mathcal{U}(C,D)\rangle\big), B_1 \wedge B_2 \wedge \neg C\Big]$$

$$\mathrm{In}_1 \equiv B_1 \wedge \big(C \vee D \vee \mathcal{S}(B_2 \wedge C, B_2)\big)$$
$$\mathrm{At}_2 \equiv \neg B_1 \wedge B_2 \wedge \big(C \vee \mathcal{S}(B_2 \wedge C, B_2)\big)$$
$$\mathrm{In}_2 \equiv B_2 \wedge \big(\neg C \vee \mathcal{S}(B_1 \wedge \neg C \wedge \neg D, B_1)\big)$$
$$\mathrm{At}_1 \equiv B_1 \wedge \neg B_2 \wedge \big(\neg C \wedge \neg D \vee \mathcal{S}(B_1 \wedge \neg C \wedge \neg D, B_1)\big)$$

Then we have:

$$\begin{aligned}\models \mathcal{U}(A,F) \leftrightarrow \mathcal{U}(A \vee \mathrm{End}_1 \vee \mathrm{End}_2 \vee & \\ \vee\Big[B_1 \wedge \neg B_2 \wedge \mathcal{U}\Big[&\big(\mathrm{In}_1 \wedge \mathrm{End}_1\big) \vee \big(\mathrm{At}_2 \wedge \big\langle \mathrm{End}_2 \vee \\ &\vee\mathcal{U}[(\mathrm{At}_1 \wedge \mathrm{End}_1) \vee (\mathrm{At}_2 \wedge \mathrm{End}_2),\ \mathrm{In}_1 \vee \mathrm{At}_2 \vee \mathrm{In}_2 \vee \mathrm{At}_1]\big\rangle\big), \mathrm{In}_1\Big]\Big] \vee \\ \vee\Big[\neg B_1 \wedge B_2 \wedge \mathcal{U}\Big[&\big(\mathrm{In}_2 \wedge \mathrm{End}_2\big) \vee \big(\mathrm{At}_1 \wedge \big\langle \mathrm{End}_1 \vee \\ &\vee\mathcal{U}[(\mathrm{At}_1 \wedge \mathrm{End}_1) \vee (\mathrm{At}_2 \wedge \mathrm{End}_2),\ \mathrm{In}_1 \vee \mathrm{At}_2 \vee \mathrm{In}_2 \vee \mathrm{At}_1]\big\rangle\big), \mathrm{In}_2\Big]\Big], \\ & B_1 \wedge B_2)\end{aligned}$$

Proof: Consider the following abbreviations:

$$F_1 \equiv \mathcal{U}(A, B_1 \wedge B_2)$$
$$F_{21} \equiv \mathcal{U}(\mathrm{End}_1, B_1 \wedge B_2)$$
$$F_{22} \equiv \mathcal{U}(\mathrm{End}_2, B_1 \wedge B_2)$$
$$F_{31} \equiv \mathcal{U}\big(B_1 \wedge \neg B_2 \wedge \mathcal{U}[\mathrm{In}_1 \wedge \mathrm{End}_1, \mathrm{In}_1], B_1 \wedge B_2\big)$$
$$F_{32} \equiv \mathcal{U}\big(\neg B_1 \wedge B_2 \wedge \mathcal{U}[\mathrm{In}_2 \wedge \mathrm{End}_2, \mathrm{In}_2], B_1 \wedge B_2\big)$$
$$F_{41} \equiv \mathcal{U}\big(B_1 \wedge \neg B_2 \wedge \mathcal{U}[\mathrm{At}_2 \wedge \mathrm{End}_2, \mathrm{In}_1], B_1 \wedge B_2\big)$$
$$F_{42} \equiv \mathcal{U}\big(\neg B_1 \wedge B_2 \wedge \mathcal{U}[\mathrm{At}_1 \wedge \mathrm{End}_1, \mathrm{In}_2], B_1 \wedge B_2\big)$$
$$F_{51} \equiv \mathcal{U}\Big(B_1 \wedge \neg B_2 \wedge \mathcal{U}\big[\mathrm{At}_2 \wedge \mathcal{U}\langle(\mathrm{At}_1 \wedge \mathrm{End}_1) \vee (\mathrm{At}_2 \wedge \mathrm{End}_2), \mathrm{In}_1 \vee \mathrm{At}_2 \vee \mathrm{In}_2 \vee \mathrm{At}_1\rangle, \mathrm{In}_1\big], B_1 \wedge B_2\Big)$$
$$F_{52} \equiv \mathcal{U}\Big(\neg B_1 \wedge B_2 \wedge \mathcal{U}\big[\mathrm{At}_1 \wedge \mathcal{U}\langle(\mathrm{At}_1 \wedge \mathrm{End}_1) \vee (\mathrm{At}_2 \wedge \mathrm{End}_2), \mathrm{In}_1 \vee \mathrm{At}_2 \vee \mathrm{In}_2 \vee \mathrm{At}_1\rangle, \mathrm{In}_2\big], B_1 \wedge B_2\Big)$$

For every branch for which $\mathcal{U}(A,F)$ holds there must be a node x_1 below x_0 with A valid in x_1 and for all nodes y in between the formula F holds. The following cases arise:

(1) For all these y it holds that $B_1 \wedge B_2$ and therefore also F.

(2.1) There is exactly one y_1, in which $B_1 \wedge \neg B_2$ holds, for all other y it holds that $B_1 \wedge B_2$.

(2.2) There is exactly one y_1, in which $\neg B_1 \wedge B_2$ holds, for all other y it holds that $B_1 \wedge B_2$.

(3.1) There are several nodes $y_1, y_2, \ldots$, in which $B_1 \wedge \neg B_2$ holds, and no z with $\neg B_1 \wedge B_2$.

(3.2) There are several nodes $z_1, z_2 \ldots$, in which $\neg B_1 \wedge B_2$ holds, and no y with $B_1 \wedge \neg B_2$.

(4.1) There are several nodes $y_1, y_2, \ldots$, in which $B_1 \wedge \neg B_2$ holds, and exactly one z with $\neg B_1 \wedge B_2$, where z lies below $y_1, y_2, \ldots$.

(4.2) There are several nodes $z_1, z_2, \ldots$, in which $\neg B_1 \wedge B_2$ holds, and exactly one y with $B_1 \wedge \neg B_2$, where y lies below $z_1, z_2, \ldots$.

(5.1) There are several nodes $y_1, y_2, \ldots$, in which $B_1 \wedge \neg B_2$ holds, and several nodes $z_1, z_2, \ldots$, in which $\neg B_1 \wedge B_2$ holds, where y_1 lies above z_1.

(5.2) There are several nodes $y_1, y_2, \ldots$, in which $B_1 \wedge \neg B_2$ holds, and several nodes $z_1, z_2, \ldots$, in which $\neg B_1 \wedge B_2$ holds, where z_1 lies above y_1.

Let us consider the sequence of nodes between x_0 and x_1 in each of these cases to show that these cases correspond exactly to the formulae F_1–F_{52}:

Case (1) is obvious: Up to A holds $B_1 \wedge B_2$ and therefore in this case F_1 is valid.

Case (2.1): Assume that in y_1 the formula $B_1 \wedge \neg B_2 \wedge \mathcal{U}!(C, D)$ is valid. Then the node z required by $\mathcal{U}!(C, D)$ in which C holds, lies in between y_1 and x_1, or $z = x_1$, or $x_1 S^+ z$. Therefore one of the pictures from figure 3 fits:

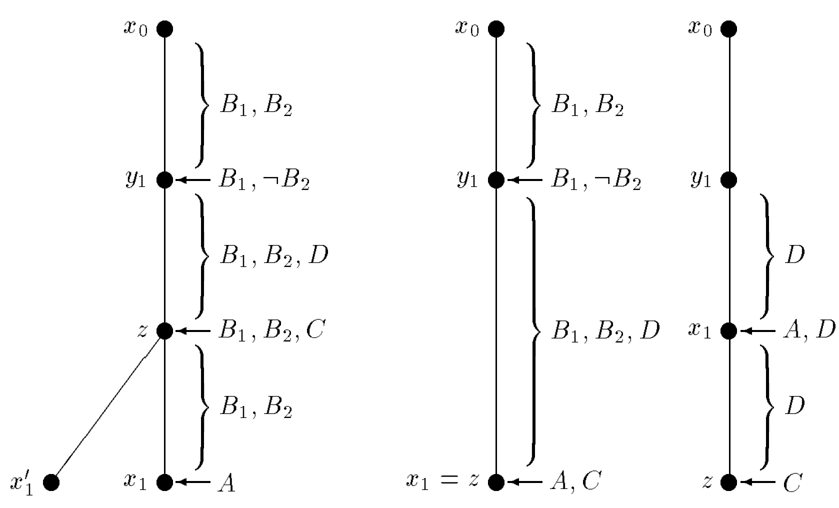


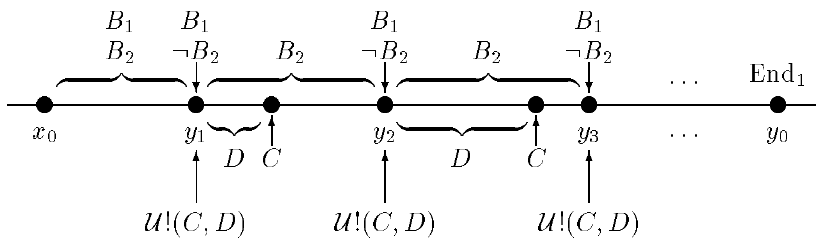
Figure 3: Three possible cases for z

Note that $\mathcal{U}(B_1 \wedge B_2 \wedge C \wedge \mathcal{U}(A, B_1 \wedge B_2), B_1 \wedge B_2 \wedge D)$ does not require the node in which A holds to be identical with x_1. This formula is also true if z is on the path from y_1 to some x_1' with $A(x_1')$. In this case we can consider the path $x_0, \ldots, x_1'$ instead of $x_0, \ldots, x_1$ for the evaluation of $\mathcal{U}(A, F)$.

Therefore under the assumption of case (2.1) in y_1 the formula End_1 resp. in x_0 the formula $\mathcal{U}(\text{End}_1, B_1 \wedge B_2)$ holds iff there is a branch with $\mathcal{U}(A, F)$.

Case (2.2): If in y_1 the formula $\neg B_1 \wedge B_2 \wedge \neg\mathcal{U}!(C, D)$ holds, we get by symmetry $\text{End}_2(y_1)$.

Case (3.1): Every chain of nodes between x_0 and x_1 with $B_1 \wedge \neg B_2$ ends in a y_0, such that the subtree below y_0 looks like in case (2.1), i.e. also $\text{End}_1(y_0)$ holds. We consider the path between x_0 and y_0. If the nodes, in which $B_1 \wedge \neg B_2$ holds, are y_1, y_2, ..., we have the following situation (with root to the left):



Now we have to describe the sequence of events between y_1 and y_0. On one hand for every node y with $y_1 S^+ y$ and $y S^* y_0$ such that neither C nor D holds in y there must be a former node (closer to the root) in which $C \wedge B_2$ was true, and since this node B_2 was true. On the other hand, if for every such y between y_1 and y_0 the formula $C \vee D \vee \mathcal{S}(B_2 \wedge C, B_2)$ is true, then at each $\neg B_2$ also $\mathcal{U}!(C, D)$ holds: $\text{End}_1(y_0)$ guarantees, that $\mathcal{U}!(C, D)$ is valid in y_0. Suppose there were a y_i with $\neg\mathcal{U}!(C, D)$. Then either $\neg\mathcal{U}!(C, \top)$ holds in y_i (which is impossible, because $\mathcal{U}!(C, \top)$ has to hold in y_0), or $\neg B_2 \wedge \mathcal{U}!(\neg C \wedge \neg D, \neg C)$ holds in y_i. Therefore there would be a y with $\neg C \wedge \neg D \wedge \mathcal{S}(\neg B_2, \neg C)$, hence also a y with $\neg(C \vee D \vee \mathcal{S}(B_2 \wedge C, B_2))$, which is a contradiction. We can conclude that in this case $\mathcal{U}(B_1 \wedge \neg B_2 \wedge \mathcal{U}(\text{In}_1 \wedge \text{End}_1, \text{In}_1), B_1 \wedge B_2)$ holds.

Case (3.2) is similar, we get $\mathcal{U}(\neg B_1 \wedge B_2 \wedge \mathcal{U}(\text{In}_2 \wedge \text{End}_2, \text{In}_2), B_1 \wedge B_2)$.

Case (4.1) differs from (3.1) in that in y_0 not End_1 but End_2 holds and therefore $\neg\mathcal{U}!(C, D)$. This means that the last $\mathcal{U}!(C, D)$ has to be finished *before* y_0 or at latest *in* y_0, and therefore $\mathcal{S}(B_2 \wedge C, B_2)$ or C holds in y_0. Thus we have $(\text{At}_2 \wedge \text{End}_2)$ in y_0, $B_1 \wedge \neg B_2 \wedge \mathcal{U}(\text{At}_2 \wedge \text{End}_2, \text{In}_1)$ in y_1 and F_{41} in x_0.

Case (4.2) again is similar to (4.1). $\text{End}_1$is valid in y_0, therefore also $\mathcal{U}!(C, D)$, thus for the last y_i such that $\neg\mathcal{U}!(C, D)$ there must be a node between y_i and y_0 (including), in which $(\neg C \wedge \neg D)$ holds, and after that no $\neg B_1$ occurs. This is what is expressed by $\text{At}_1(y_0)$.

Case (5.1): Here we have a chain, beginning with y_1, followed by y_2, y_3, ..., and z_1, z_2, ..., arbitrarily shuffled. For all y between y_1 and z_1 the formula In_1 holds, in z_1 the formula At_2 holds, after that up to the last element of the chain $\text{In}_1 \vee \text{At}_2 \vee \text{In}_2 \vee \text{At}_1$ is satisfied. The chain ends with a y_0 such that $(\text{At}_1 \wedge \text{End}_1)$, or with a z_0 such that $(\text{At}_2 \wedge \text{End}_2)$. This situation is described by the formula F_{51}.

Case (5.2) again is similar to (5.1).

Since (1)–(5.2) cover all possible cases, we have

$$\models \mathcal{U}(A,F) \;\leftrightarrow\; F_1 \vee F_{21} \vee F_{22} \vee F_{31} \vee F_{32} \vee F_{41} \vee F_{42} \vee F_{51} \vee F_{52}.$$

This is what was to be proved.

□

Note that on the right hand side of the equation of lemma 3.1 no $\mathcal{U}!$–operator occurs. This, together with the following lemma, gives a basis for eliminating all $\mathcal{U}!$–operators in a formula.

Lemma 3.2 The following equivalences are valid:

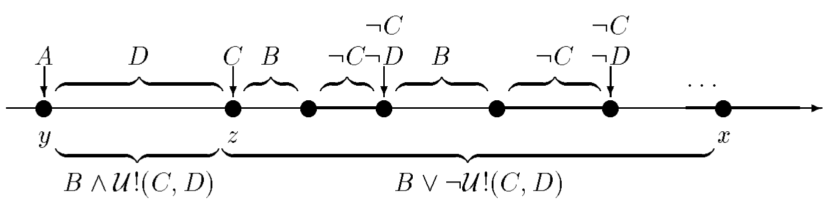
(i) $\mathcal{S}(A \wedge \mathcal{U}!(C,D), B \vee \mathcal{U}!(C,D)) \;\leftrightarrow\; \mathcal{S}(A,D) \wedge \langle C \vee D \wedge \mathcal{U}!(C,D)\rangle \vee$
$\vee \mathcal{S}\big[C \wedge \mathcal{S}(A,D), C \vee D \vee \neg\mathcal{S}(\neg B, \neg C)\big] \wedge \big(\neg\mathcal{S}(\neg B,\neg C) \vee \langle C \vee D \wedge \mathcal{U}!(C,D)\rangle\big)$

(ii) $\mathcal{S}(A \wedge \neg\mathcal{U}!(C,D), B \vee \mathcal{U}!(C,D)) \;\leftrightarrow\; \mathcal{S}(A, B \wedge \neg C) \wedge \langle \neg C \wedge (\neg D \vee \neg\mathcal{U}!(C,D))\rangle \vee$
$\vee \mathcal{S}\big[\neg C \wedge \neg D \wedge \mathcal{S}(A, B \wedge \neg C), C \vee D \vee \mathcal{S}(A \vee (B \wedge C), B)\big] \wedge$
$\wedge \mathcal{S}(B \wedge C, B) \vee \langle C \vee D \wedge \mathcal{U}!(C,D)\rangle$

(iii) $\mathcal{S}(A \wedge \mathcal{U}!(C,D), B \vee \neg\mathcal{U}!(C,D)) \;\leftrightarrow\; \mathcal{S}(A, B \wedge D) \wedge \langle C \vee D \wedge \mathcal{U}!(C,D)\rangle \vee$
$\vee \mathcal{S}\big[C \wedge \mathcal{S}(A, B \wedge D), \neg C \vee \mathcal{S}(A \vee B \wedge \neg C \wedge \neg D, B)\big] \wedge$
$\wedge \big(\mathcal{S}(B \wedge \neg C \wedge \neg D, B) \vee \langle \neg C \wedge (\neg D \vee \neg\mathcal{U}!(C,D))\rangle\big)$

(iv) $\mathcal{S}(A \wedge \neg\mathcal{U}!(C,D), B \vee \neg\mathcal{U}!(C,D)) \leftrightarrow \neg\mathcal{S}(\neg B \wedge \mathcal{U}!(C,D), \neg A \vee \mathcal{U}!(C,D)) \wedge$
$\Big[\mathcal{S}(\neg C \wedge \neg D \wedge \mathcal{S}(A,\neg C), \top) \vee \big(\mathcal{S}(A,\neg C) \wedge \langle \neg C \wedge (\neg D \vee \neg\mathcal{U}!(C,D))\rangle\big)\Big]$

Proof: These formulae are derived from Gabbay[7]. As an example we prove (iii):
The formula requires that for the current node x there exists a former node y with yS^+x such that $(A \wedge \mathcal{U}!(C,D))(y)$, i.e. there is a z such that $C(z)$ and $\forall t(yS^+t \wedge tS^+z \rightarrow D(t))$. There are two possibilities for z:
1) xS^*z. Then between y and x $\mathcal{U}!(C,D)$ holds and thus also B, therefore $\mathcal{S}(A, B \wedge D) \wedge \langle C \vee D \wedge \mathcal{U}!(C,D)\rangle$ is true.
2) zS^+x. Then we have the following situation:



In the area, in which $B \vee \neg\mathcal{U}!(C,D)$ holds, $\mathcal{S}(A,B) \vee \mathcal{S}(B \wedge \neg C \wedge \neg D, B) \vee \neg C$ is valid. The argument is the same as used in lemma 3.1. In x it holds that $\mathcal{S}(B \wedge \neg C \wedge \neg D, B)$, if the last $\neg\mathcal{U}!(C,D)$ was finished *before* x, or $\neg C \wedge \neg D$, if it ended *in* x, or $\neg C \wedge \neg\mathcal{U}!(C,D)$, if it will end *beyond* x.
Exactly this situation is expressed by the above disjunction (iii).

□

Note that in these formulae on the right hand side there is no $\mathcal{U}!$ inside of an $\mathcal{S}$ (the first conjunct of equation (iv) has to be replaced by the corresponding term via equation (i) to get this form). (i)–(iv) therefore can be used to pull out every formula $\mathcal{U}!(C,D)$ which occurs inside an $\mathcal{S}$–operator.

Theorem 3.3 For every formula $F \in \mathbf{TL}(\mathcal{U}!, \mathcal{S}, \mathcal{U}, \mathcal{P})$ there is an equivalent formula F', in which no $\mathcal{U}!$–operator occurs inside a $\mathcal{U}$ or $\mathcal{S}$.

Proof: is done by induction on the number of different subformulae $\mathcal{U}!(C,D)$ inside any $\mathcal{U}$, $\mathcal{S}$ in F, and for every number by subinduction on the depth of nesting of a particular formula $\mathcal{U}!(C,D)$ inside of $\mathcal{S}$.
Let F_1 and F_2 be boolean combinations of formulae with $\mathcal{U}!(C,D)$, such that C and D contain no $\mathcal{U}!$–operator. Then F_1 and F_2 can be rewritten using conjunctive and disjunctive form as

$$\begin{aligned} F_1 &\leftrightarrow (A_1 \wedge \mathcal{U}!(C,D)) \vee (A_2 \wedge \neg\mathcal{U}!(C,D)) \quad \text{and} \\ F_2 &\leftrightarrow (B_1 \vee \mathcal{U}!(C,D)) \wedge (B_2 \vee \neg\mathcal{U}!(C,D)) \end{aligned}$$

Therefore

$$\begin{aligned} \mathcal{S}(F_1,F_2) \leftrightarrow\ & \mathcal{S}\big[(A_1 \wedge \mathcal{U}!(C,D)) \vee (A_2 \wedge \neg\mathcal{U}!(C,D)), (B_1 \vee \mathcal{U}!(C,D)) \wedge (B_2 \vee \neg\mathcal{U}!(C,D)\big] \\ & \leftrightarrow \mathcal{S}[A_1 \wedge \mathcal{U}!(C,D), B_1 \vee \mathcal{U}!(C,D)] \wedge \mathcal{S}[A_1 \wedge \mathcal{U}!(C,D), B_2 \vee \neg\mathcal{U}!(C,D)] \vee \\ & \quad \vee \mathcal{S}[A_2 \wedge \neg\mathcal{U}!(C,D), B_1 \vee \mathcal{U}!(C,D)] \wedge \mathcal{S}[A_2 \wedge \neg\mathcal{U}!(C,D), B_2 \vee \neg\mathcal{U}!(C,D)] \end{aligned}$$

To each of these four $\mathcal{S}$–formulae the corresponding equivalence from lemma 3.2 can be applied to yield a formula with the same subformulae and a lower nesting of $\mathcal{U}!$ inside $\mathcal{S}$.

If $\mathcal{U}!$ occurs inside a $\mathcal{U}$, we can replace every direct occurrence of $\mathcal{U}!(C,D)$ in F_1 in $\mathcal{U}(F_1,F_2)$ by $\mathcal{U}(C,D)$, because of the following equivalence:

$$\models \mathcal{U}(A_1 \wedge \mathcal{U}!(C,D) \vee A_2 \wedge \neg\mathcal{U}!(C,D), F_2) \leftrightarrow \mathcal{U}(A_1 \wedge \mathcal{U}(C,D) \vee A_2 \wedge \neg\mathcal{U}(C,D), F_2)$$

Thus the only remaining case are $\mathcal{U}!$–operators in the second argument of a $\mathcal{U}$–operator. Since every such formula can be written as $\mathcal{U}(A,F_1)$, the occurrences of $\mathcal{U}!(C,D)$ can be eliminated using equation 3.1.

□

As a corollary of theorem 3.3 we get (using lemma 1.13)

Theorem 3.4 For every formula $F \in \mathbf{TL}(\mathcal{U}!,\mathcal{S},\mathcal{U},\mathcal{P})$ there is an equivalent formula $F' \in \mathbf{TL}(\mathcal{S},\mathcal{U},\mathcal{P})$.

Clearly this theorem generalises if the $\mathcal{X}_k$–operators are added. Therefore with theorem 2.6 we can conclude

Theorem 3.5 $\{\mathcal{U},\mathcal{S},\mathcal{X}_k\}$ is expressively complete.

4 Extensions

In this section we want to investigate on which structures other than the so far considered tree models our method of proving expressive completeness extends.

4.1 Ordered and arc–labelled trees

Definition 4.1 An <u>ordered tree</u> $\mathbf{B} = \langle \mathcal{N}, S, D\rangle$ consists of a set of nodes $\mathcal{N}$, a successor relation S and an additional ordering relation D on nodes with the same predecessor, such that

- $\langle \mathcal{N}, S\rangle$ is a tree according to definition 1.1
- D is a linear (irreflexive) order on the successor nodes of each node

Definition 4.2 An arc–labelled tree $\mathbf{B} = \langle \mathcal{N}, S_1, \ldots, S_b \rangle$ consists of a set of nodes $\mathcal{N}$ together with b distinguished successor relations $S_1, \ldots, S_b$, such that

- each S_i is functional, i.e. for every x there is at most one y with xS_iy
- $\langle \mathcal{N}, S \rangle$ is a tree according to definition 1.1, where xSy if $xS_1y \vee \ldots \vee xS_by$.

The first order language on ordered and arc–labelled trees uses predicates S^+, D and $S^+, S_1, \ldots, S_b$, respectively. Equality is in both cases definable. It is rather easy to see that these languages have the same expressive power as the first order language with the predicate S^* and the additional monadic predicates $\mathcal{A}_1(x), \ldots, \mathcal{A}_b(x)$ and $\mathcal{B}_1(x), \ldots, \mathcal{B}_b(x)$, respectively, where

$\mathcal{A}_k(x)$ if $\exists y_1, \ldots, y_k \left[\bigwedge_{\mu<k}(y_\mu D y_{\mu+1}) \wedge y_k Dx \right]$ and
$\mathcal{B}_k(x)$ if $\exists y[yS_kx]$.

Therefore for every formula of this language we can find an appropriate temporal formula with the operators $\mathcal{S}, \mathcal{U}, \mathcal{X}_k$ and $\mathcal{A}_1, \ldots, \mathcal{A}_b$ or $\mathcal{B}_1, \ldots, \mathcal{B}_b$, respectively. These new operators are "0–ary operators", i.e. temporal constants. Thus this set of operators is expressively complete for ordered and arc–labelled trees, respectively.

We can define "ordinary" unary modal operators $\mathcal{A}'$ and $\mathcal{B}'_1, \ldots, \mathcal{B}'_b$ via

$\varphi_{\mathcal{A}'} = \exists y(yDx_0 \wedge p_1(y))$
$\varphi_{\mathcal{B}'_k} = \exists y(yS_kx_0 \wedge p_1(y))$

The $\mathcal{B}'_k$–operators are similar to the inverse $\langle k \rangle$–operators of propositional dynamic logic PDL. On one hand, the $\mathcal{A}'$– and $\mathcal{B}'_k$–operators can be defined using $\mathcal{A}_k$ and $\mathcal{B}_k$, respectively, as follows:

$$\mathcal{A}'A \leftrightarrow \bigvee_{k<b} \left(\mathcal{A}_k \wedge \mathcal{Y}\mathcal{X}_1(\neg\mathcal{A}_k \wedge A)\right)$$
$$\mathcal{B}'_iA \leftrightarrow (\mathcal{B}_i \wedge \mathcal{Y}A)$$

(Remember that $\mathcal{X}_1A = \mathcal{U}(A, \bot)$ and $\mathcal{Y}A = \mathcal{S}(A, \bot)$.) On the other hand, we can define $\mathcal{A}_k$ and $\mathcal{B}_k$ by $\mathcal{A}'$ and $\mathcal{B}'_k$:

$$\mathcal{A}_k \leftrightarrow \mathcal{A}'\mathcal{A}' \ldots \mathcal{A}'\top \qquad (k \text{ times } \mathcal{A}')$$
$$\mathcal{B}_k \leftrightarrow \mathcal{B}'_k\top.$$

Besides that $\mathcal{X}_k$ can be defined via $\mathcal{A}'$ and via $\mathcal{B}'_k$:

$$\mathcal{X}_kA \equiv \mathcal{X}_1(A \wedge \mathcal{A}'(A \wedge \mathcal{A}'(\ldots)..) \qquad (k \text{ times A})$$
$$\mathcal{X}_kA \equiv \bigvee_P \left(\mathcal{X}_1(\mathcal{B}_{\mu_1} \wedge A) \wedge \ldots \wedge \mathcal{X}_1(\mathcal{B}_{\mu_k} \wedge A)\right)$$

where the latter disjunction is over all permutations P of k different μ_j. Therefore we have

Theorem 4.3 $\{\mathcal{S}, \mathcal{U}, \mathcal{A}'\}$ is expressively complete for ordered trees.
$\{\mathcal{S}, \mathcal{U}, \mathcal{B}'_1, \ldots, \mathcal{B}'_b\}$ is expressively complete for arc–labelled trees.

4.2 Unbounded branching trees

It is rather easy to see that no finite set of operators can be expressively complete if we give no upper bound on the branching degree of the nodes. Every operator uses only a fixed number of bound variables, whereas the statement "node x_0 has at least k different successors" requires k different variable names. But the above proofs also hold if we allow an *infinite* set of operators:

Theorem 4.4 $\{\mathcal{S}, \mathcal{U}\} \cup \{\mathcal{X}_k \mid k < \omega\}$ is expressively complete for unbounded ($< \omega$) branching trees.
$\{\mathcal{S}, \mathcal{U}\} \cup \{\mathcal{B}'_k \mid k < \omega\}$ is expressively complete for unbounded branching labelled trees.

4.3 Arborescences

Definition 4.5 An arborescence $\langle \mathcal{N}, S \rangle$ is a set of nodes $\mathcal{N}$ together with an irreflexive successor relation $S \subset \mathcal{N} \times \mathcal{N}$, such that the following holds:

- For every node there are at most b successors
- For every node there are at most b predecessors
- For every two different nodes there is a unique finite path connecting them, i.e. for $x_0 \neq x_1$ there is exactly one sequence $\langle y_0, y_1, \ldots, y_n \rangle$ such that $x_0 = y_0$, $x_1 = y_n$ and for every $\mu < n$ holds $y_\mu S y_{\mu+1}$ or $y_{\mu+1} S y_\mu$, and for all $\nu \neq \mu$ holds $y_\nu \neq y_\mu$.

(The third condition implies that there is no loop from x_0 to x_0). Symmetry tells us that we can construct a two dimensional logic with operators $\{\mathcal{U}!, \mathcal{S}!, \mathcal{U}, \mathcal{S}, \mathcal{X}_k, \mathcal{Y}_k\}$ which is expressively complete for arborescences. But the separating equations, which allowed us to eliminate $\mathcal{U}!$ from under $\mathcal{U}, \mathcal{S}$ fail to hold: they rely on the fact that the set of nodes y with yS^*x_0 is linearly ordered. We therefore leave it as an open question whether there is a one dimensional complete set of operators for arborescences.

References

[1] J.A.W. Kamp: *Tense Logic and the Theory of Linear Order*; Dissertation, University of California, Los Angeles (1968).

[2] D.M. Gabbay: *Expressive Functional Completeness in Tense Logic*; in: U. Mönnich (ed.): Aspects of Philosophical Logic; pp.91–117 Reidel, Dordrecht (1981).

[3] N. Immerman, D. Kozen: *Definability with Bounded Number of Bound Variables*; in: Proc. 2^{nd} IEEE LICS, pp.236–244; Edinburgh (1987). Reappeared in: Information and Computation **83**; pp.121–139 (1989).

[4] T. Hafer, W. Thomas: *Computation Tree Logic CTL* and Path Quantifiers in the Monadic Theory of the Binary Tree*; in: T. Ottmann (ed.): ICALP 1987, Springer LNCS **267**, (1988). Short version of T. Hafer: *On the Expressive Completeness of CTL**; Bericht Nr. 123, Schriften zur Informatik, RWTH Aachen, (Dez. 1986).

[5] D. Gabbay, A. Pnueli, S. Shelah, J. Stavi: *On the Temporal Analysis of Fairness*; in: Proc. 7^{th} ACM POPL, pp.163–173; Las Vegas (1980).

[6] B.-H. Schlingloff: *Zur temporalen Logik von Bäumen*; Dissertation, Institut für Informatik der Technischen Universität München, Report TUM–I9012; (1990).

[7] D. Gabbay: *The Declarative Past and Imperative Future: Executable Temporal Logic for Interactive Systems*; in: B.Banieqbal et al. (eds), Temporal logic in Specification, Springer LNCS **398**, pp.431–448 (1989).