On the modal logic of ‘elsewhere’ as a component in modal products

Christopher Hampson
Department of Informatics
King’s College London

Joint work with Agi Kurucz
The logic of ‘elsewhere’

(von Wright 1979)

- **Difference frames:** \( \mathcal{S} = (W, R) \)
  
  where \( W \) is a non-empty set of possible worlds and
  
  \[ xRy \text{ if and only if } x \neq y \]

- **Logics determined by frames:** Given a class of frames \( \mathcal{C} \), we define
  
  \[ \text{Log}(\mathcal{C}) = \{ \varphi \mid \mathcal{S} \models \varphi \text{ for all } \mathcal{S} \in \mathcal{C} \} \]

- Some well known unimodal logics include
  
  \[
  \begin{align*}
  K &= \text{Log \{all frames\}} \\
  K4 &= \text{Log \{transitive frames\}} \\
  K4.3 &= \text{Log \{linear frames\}} \\
  S5 &= \text{Log \{equivalence relations\}}
  \end{align*}
  \]
The logic of ‘elsewhere’

- **The logic of ‘elsewhere’**: 

  \[
  \text{Diff} = \text{Log} \{\text{all difference frames}\}
  \]

- **Axiomatisation**: \(\text{Diff}\) is the smallest logic containing all propositional tautologies and the axioms

  \[
  \square(p \rightarrow q) \rightarrow (\square p \rightarrow \square q)
  
  p \rightarrow \square \Diamond p
  
  \Diamond \Diamond p \rightarrow (p \lor \Diamond p)
  \]

  that is closed under

  - modus ponens \(\varphi, \varphi \rightarrow \psi / \psi\)
  - uniform substitution
  - and necessitation \(\varphi / \square \varphi\)

  \(\text{(Segerberg 1973)}\)

Note that there are frames for \(\text{Diff}\) that are **NOT** difference frames
The logic of ‘elsewhere’

- However, every frame for Diff is symmetric and pseudo-transitive

\[ \forall x \forall y \forall z (xRy \land yRz \rightarrow (x = z \lor xRz)) \]

Frames of this type are called pseudo-equivalence relations with possibly some irreflexive points.

- ... adding in these extra edges gives us a frame for S5.

\[ S5 = \text{Diff} + \Box p \rightarrow p \]

- Note that every frame for Diff is a p-morphic image of a difference frame.

- It is, perhaps then, little surprise that
  - both S5 and Diff have the finite model property, and
  - the validity problems for both S5 and Diff are coNP.
The logic of ‘elsewhere’

However, **Diff** is a little more expressive than **S5**.

- For one, we can express the **universal** modality and its dual with the abbreviations

\[
\forall \varphi = \varphi \land \Box \varphi \quad \text{and} \quad \exists \varphi = \varphi \lor \Diamond \varphi
\]

- But we can also express the ‘**exactly one**’ modality with the abbreviation

\[
\Diamond^{=1} \varphi = \exists (\varphi \land \Box \neg \varphi)
\]

something that is **not** expressible in **S5**.
Constructing products (a step-by-step guide)


**Step 1)** Take an $n$-modal logic $L_1$ and an $m$-modal logic $L_2$.

**Step 1)** Consider class of frames $\text{Fr}(L_1)$ for $L_1$ and the class of frames $\text{Fr}(L_2)$ for $L_2$.

Let us suppose

$\mathcal{F}_1 = (W_1, R^1_1, \ldots R^n_1)$ is an element of $\mathcal{F}_1$

$\mathcal{F}_2 = (W_2, R^1_2, \ldots R^m_2)$ is an element of $\mathcal{F}_2$

**Warning!** There may be some frames in $\text{Fr}(\text{Log}(C))$ that are not in $C$
Constructing products (a step-by-step guide)

**Step 2)** We define the **product frame** of $\mathcal{F}_1$ and $\mathcal{F}_2$ to be

$$\mathcal{F}_1 \times \mathcal{F}_2 = (W_1 \times W_2, R^1_{h_1}, \ldots, R^n_{h_1}, R^1_{v}, \ldots, R^m_{v})$$

where

$$(x, y) R^i_h(x', y') \iff x R^i_1 x' \text{ and } y = y'$$

and

$$(x, y) R^j_v(x', y') \iff x = x' \text{ and } y R^j_2 y'$$
Constructing products (a step-by-step guide)

**Step 3)** Take all these product frames and put them in their **product class**

\[ \text{Fr}(L_1) \times \text{Fr}(L_2) = \{ \mathcal{F}_1 \times \mathcal{F}_2 \mid \mathcal{F}_1 \in \text{Fr}(L_1) \text{ and } \mathcal{F}_2 \in \text{Fr}(L_2) \} \]

**Step 4)** Look at all the \((n + m)\)-modal formulas that are valid in this class of frames

\[ L_1 \times L_2 = \text{Log} (\text{Fr}(L_1) \times \text{Fr}(L_2)) \]

This is the so-called **product logic** of \(L_1\) and \(L_2\) that we are concerned with!

\(L_1 \times L_2\) can have frames that are **not** product frames!
A few general properties of products


• $L_1 \times L_2$ is determined by products of rooted frames.

$$L_1 \times L_2 = \text{Log} (\text{Fr}^r(L_1) \times \text{Fr}^r(L_2))$$

where $\text{Fr}^r(L_i)$ is the class of rooted frames for $L_i$.

• If $\mathcal{G}_i$ is a p-morphic image of $\mathcal{F}_i$ for $i = 1, 2$ then

$$\mathcal{G}_1 \times \mathcal{G}_2$$

is a p-morphic image of $\mathcal{F}_1 \times \mathcal{F}_2$.

• If $L_1$ and $L_2$ are Kripke complete modal logics, definable by a recursive set of first-order formulas, then

$$L_1 \times L_2$$

is recursively enumerable.
Minsky machines and undecidable products

Let $K_u$ be the bimodal logic, with modal operators $\Box$ and $\forall$, defined by

$$K_u = \text{Log} \{ \text{all frames of the form } (W, R, W \times W) \}$$

**Theorem:** The validity problem for the product logic $K_u \times \text{Diff}$ is undecidable

*(H–Kurucz 2012b)*

**Proof:** A **two-counter Minsky machine** is a finite sequence of instructions

$$M = (I_0, \ldots, I_T)$$

where each $I_t$, for $t < T$, is from the set

$$\{ \text{zero}_i, \text{inc}_i, \text{dec}_i(j) : i = 0, 1, j \leq T \}$$

and $I_t = \text{halt}$. The instructions tell the machine to either

- **reset** counter $i$ to zero – $\text{zero}_i$,
- **increment** counter $i$ by one – $\text{inc}_i$, or
- **decrement** (if it can) counter $i$ by one, else **jump** to instruction $j$ – $\text{dec}_i(j)$

*(Minsky 1967)*
Minsky machines and undecidable products

Let $p_0, p_1$ be propositional variables and consider the following formula:

$$\psi_{\text{inc}}(i) = \diamond^1_v (\neg p_i \land \diamond_h p_i) \land \forall_v (p_i \rightarrow \Box_h p_i)$$

$\psi_{\text{inc}}(i)$ increments the number of $p_i$s by exactly one!
Similarly for the following formula:

\[
\psi_{\text{dec}}(i) = \diamondsuit_v^{=1}(p_i \land \diamondsuit_h \neg p_i) \land \forall_v (\neg p_i \rightarrow \Box_h \neg p_i)
\]

\(\psi_{\text{dec}}(i)\) decrements the number of \(p_i\)s by exactly one!
Minsky machines and undecidable products

...as well as the formulas:

\[ \psi_{\text{fix}}(i) = \forall_v (p_i \leftrightarrow \diamond_h p_i) \]

and

\[ \psi_{\text{zero}}(i) = \forall_v \square_h \neg p_i \]

\( \psi_{\text{fix}}(i) \) and \( \psi_{\text{zero}}(i) \) fix and reset the number of \( p_i \)s, respectively!
Minsky machines and undecidable products

Using these formulas, we can encode every action of an arbitrary Minsky machine $M$,

- with each **vertical Diff-cluster** encoding the contents of the registers of $M$
- and the **horizontal K-component** encoding the flow of time.

We use variables $s_0, \ldots, s_T$ to denote the internal state of $M$ and stipulate the basic definition of the machine with the following formulas:

\[
\begin{align*}
&s_0 \land \forall_v \neg p_0 \land \forall_v \neg p_1 \\
&\forall_h \bigvee_{t \leq T} s_t \land \bigwedge_{t \neq t'} \leq T \forall_h (s_t \land s_{t'}) \\
&\bigwedge_{t \leq T} \forall_h (\Diamond_h s_t \rightarrow \Box_h s_t) \land \bigwedge_{i=0,1} \forall_h \forall_v (\Diamond_h p_i \rightarrow \Box_h p_i)
\end{align*}
\]

Notice that the third clause stipulates that all $K$-paths encode the same computation of $M$. 
Minsky machines and undecidable products

All we need now is to specify the ‘program’ for $M$.
We add formulas of the form

\[ \forall h \left( s_t \rightarrow \Box_h s_{t+1} \land \psi_{\text{zero}}(i) \land \psi_{\text{fix}}(1 - i) \right) \]

whenever the instruction $I_t$ of $M$ is $\text{zero}_i$.

and similarly,

\[ \forall h \left( s_t \rightarrow \Box_h s_{t+1} \land \psi_{\text{inc}}(i) \land \psi_{\text{fix}}(1 - i) \right) \]

whenever $I_t$ is $\text{inc}_i$. 
Minsky machines and undecidable products

However, for the cases where $I_t$ is $\text{dec}_i(j)$ we require two clauses: Firstly, we need

$$\forall_h((s_t \land \Diamond v p_i) \rightarrow \Diamond h s_{t+1} \land \psi_{\text{dec}}(i) \land \psi_{\text{fix}}(1 - i))$$

which describes the behaviour of $M$ when counter $i$ is non-empty,

and secondly, we need

$$\forall_h((s_t \land \Box v \neg p_i) \rightarrow \Diamond h s_{j} \land \psi_{\text{fix}}(0) \land \psi_{\text{fix}}(1))$$

which describes the behaviour of $M$ when counter $i$ is empty, that is, where $M$ fixes both registers and jumps to instruction $I_j$. 

AiML 2012  24.08.12
Minsky machines and undecidable products

Arguing inductively, we see that:

\[ \varphi_M \land \forall h \neg s_T \text{ is } K_u \times \text{Diff} \text{-satisfiable iff } M \text{ does not halt.} \]

Hence the undecidability of \( K_u \times \text{Diff} \) follows from the undecidability of the halting problem for 2-counter Minsky machines.

However, since \( K_u \) and \( \text{Diff} \) are both first-order definable, we are assured that \( K_u \times \text{Diff} \) is recursively enumerable.

What else can we prove with this approach?
Minsky machines and undecidable products

Consider the bimodal logics

\[ K_C = \text{Log} \{ \text{all frames of the form } (W, R, R^*) \} \]
\[ \text{PTL}_{X\square} = \text{Log} \{ (\omega, +1, <) \} \]

where \( R^* \) denote the **transitive closure** of \( R \).

(note that this condition is not first-order definable)

(... so we are not assured that their products are r.e!)

**Theorem:** Let \( L = \text{Log} (C) \) where \( C \) is any class of frames such that

- every frame in \( C \) is of the form \( \mathcal{F}_h \times \mathcal{F}_v \), where
  - \( \mathcal{F}_h = (W, R, R^*) \) with \( (W, R) \) being an irreflexive, intransitive tree,
  - \( \mathcal{F}_v \) is a difference frame.
- \( (\omega, +1, <) \times (\omega, \neq) \in C \).

Then \( L \) is **not recursively enumerable**
Corollary: The following logics are non-r.e

- $\text{PTL}_{\square} \times \text{Diff}$,
- $\text{KC} \times \text{Diff}$,
- $\text{PDL}_\text{I} \times \text{Diff}$
Open problems

• The logic \( S5 \times \text{Diff} \) is term definable in \( S5 \times \text{Lin} \),

  \[\text{(where Lin is linear temporal logic with 'future' and 'past')}\]

It is known that \( S5 \times \text{Lin} \) is decidable, so it follows that \( S5 \times \text{Diff} \) is decidable

...but does it have the finite model property?

  \[\text{(Reynolds 1997)}\]

• We proved that \( \text{Diff} \times \text{Diff} \) has neither the finite model property nor a finite axiomatisation,

  \[\text{(in fact, no logic between K \times \text{Diff} and S5 \times \text{Diff} is finitely axiomatisable)}\]

...but is it decidable?

  \[\text{(H–Kurucz 2012a)}\]

• We can apply the techniques outlined in this paper (together with a few 'tricks') to show that \( K4.3 \times \text{Diff} \) is undecidable

...but what about \( K4 \times \text{Diff} \)?

  \[\text{(H–Kurucz 2012a)}\]
Thank you for listening!
Some references


- C. Hampson and A. Kurucz. *Axiomatisation and decision problems of modal product logics with the difference operator*. (manuscript), 2012.