Temporary Unavailability Logic and General Modification Logic

Wouter Koolen

Contents

1 Introduction **2**
1.1 Caustic Sabotage 2
1.2 Temporary Unavailability 3
1.3 Outline of the article 4

2 TUL **4**
2.1 Formalisation 5
2.2 Translation to First Order Logic 7
2.3 Model Checking Complexity 8
2.4 Satisfiability 10
2.5 Several Considerations 10

3 GML **10**
3.1 Modification frames 11
3.2 Logic 11
3.3 Game 13
3.4 Modification Product 14
3.5 Reduction to ML-2 14
3.6 Satisfiability 17
3.7 Model Checking Complexity 17
3.8 Applications 17

4 Conclusion **20**
4.1 Open questions 20
4.2 Acknowledgements 21

1 Introduction

Modal logics are simple yet expressive devices to talk about relational structures. *Dynamic modal logics* extend standard modal logics with new operators that upon evaluation alter the subject relational structure. Modal logics are thoroughly covered in [BdRV01], and this paper assumes familiarity with standard and dynamic modal logic.

Sabotage logics are dynamic modal logics, that crumble the relational structure under the effect of certain new operators. The concept of sabotage in the context of graph algorithms and graph games was informally introduced in [vB05a]. It was subsequently formalised and extended in [Roh04], which covers three variants of sabotage logic in detail. This introduction is structured as follows. We first briefly review existing sabotage logics. Then we introduce a new concept of sabotage, based on temporary unavailability. Finally we provide the outline of the article.

1.1 Caustic Sabotage

[Roh04] defines three sabotage operators, which we briefly summarise. Fix a Kripke model $\mathfrak{M} = (W, R, V)$ with (W, R) a multi-graph and two worlds $w, v \in W$. The sabotage logics with their corresponding new operator are:

- $\diamond$ Sabotage Modal Logic (SML): SML Extends standard modal logic with an operator that removes a single edge from the relational structure. Formally, $\diamond\varphi$ is true at w when there is an edge e in R such that, after deleting e from R, φ is true at w.

- $\diamond\!\!\rightarrow$ Adjacent Sabotage Logic (ASL): ASL differs from SML in that edges can only be removed at the focus of evaluation. Formally, $\diamond\!\!\rightarrow \varphi$ holds at w when φ is true at w after deletion of some outgoing edge of w.

- $\diamond\!\!\Rightarrow$ Path Sabotage Logic (PSL): PSL is a variation of ASL, where edges are removed at a *second* focus of deletion. That is, $\diamond\!\!\Rightarrow \varphi$ is satisfied at (w, v) when φ is true at (w, u) after erasing some outgoing edge (v, u) of v from R.

We call a dynamic operator *caustic* if it reduces R to a proper subgraph. We call a dynamic modal logic *caustic* if all its dynamic operators are caustic. Clearly, $\diamond$, $\diamond\!\!\rightarrow$ and $\diamond\!\!\Rightarrow$ are caustic operators and SML, ASL and PSL are caustic logics.

[Roh04] gives various motivations for each sabotage logic. The travelling researcher problem, uncertainty elimination in epistemic logic and Euler's famous *Seven Bridges of Königsberg* problem are example domains for application of SML, ASL and PSL respectively.

[Roh04] also defines two-player games on multi-graphs for each type of sabotage. Runner, the optimistic player, tries to advance through the graph toward satisfaction of some fixed condition (like reaching a special node, or avoiding a set of nodes) while Blocker - the *saboteur* - chooses the edge to be removed, thus attempting to trap Runner in a sink.

[Roh04] distinguishes three versions of the model checking problem:

1. formula; considers the model fixed, and the formula variable.
2. program; considers the formula is fixed, and the model variable.
3. combined; considers both model and formula variable.

The complexities of these three model checking problems plus the satisfiability problem of the caustic sabotage logics are summarised in Table 1. The new sabotage operators strengthen standard modal logic on several (complexity) fronts.

- The satisfiability problems for SML, PSL and ASL are undecidable.
- The combined model checking problems for SML, PSL and ASL are all PSPACE-complete.

Interestingly, there is no difference in complexity between SML, PSL and ASL, even though, intuitively, this sentence lists them in order of decreasing expressive power.

1.2 Temporary Unavailability

This section proposes and motivates a new *non-caustic* sabotage operation. The first and foremost reason for this proposal is architectural interest, but I admit that I initially hoped that it would fill the gap between standard modal logic and the caustic sabotage logics. It turns out to do so, but in a way I did not expect. More about this in §2.3. Consider the following scenarios.

Scenario 1. When dealing with computer networks like the Internet, there are always connections that fail. But in general this is not because the computers or networks are physically destroyed by some malevolent force, but because of temporary failure. This means that broken entities might eventually be repaired, or might even automatically repair themselves. Furthermore, if failures are independent random events, it is very unlikely that many entities fail concurrently.

Scenario 2. A researcher is travelling toward an important conference. She is travelling by car and receives a radio message that there is a traffic jam ahead. If the jam is still some time ahead, then instead of rushing to the nearest train station to bypass the traffic jam, and arrive a little late for certain, she might take her chances and hope that the traffic jam will naturally resolve before she hits it.

Scenario 3. Epistemic agents typically gather knowledge by eliminating indistinguishability links in a graph of possible worlds. Rational agents with unbounded resources employ caustic removal of such edges. But forgetful (or even worse, memoryless) agents remove links that later resurface.

Note that, although we are using terminology from probability theory and temporal reasoning, we are not trying to model chance or time. We are just motivating the idea that compromised entities may eventually return to their normal state.

These scenarios suggest a different *non-caustic* sabotage concept. The structure does not crumble as before, but certain parts become temporarily unavailable. We will restrict attention to the deletion of edges, as it is in a sense simpler (to remove a world from a model, one also has to remove all incident edges) and it allows efficient reuse of ideas from [Roh04].

1.3 Outline of the article

The next section, §2, treats the simplest instance of non-caustic sabotage, namely temporary removal of arbitrary single edges. This results in the Temporary Unavailability Logic. The complexity of TUL is subsequently analysed along the lines of [Roh04]. In section §3 we present and motivates a generalisation of TUL called General Modification Logic or GML. Where TUL deals with temporary removal of single edges, allowing any individual edge to be removed, GML abstracts away from the actual way in which the model is altered, and introduces restricted accessibility between different alterations. The actual underlying mechanism is a variation of product update (see [BMS98]). We conclude with a comparison of all treated sabotage logics and a list of open problems in §4.

Table 1 Complexities of standard problems for sabotage logics, standard modal logic and first order logic.

Logic	Combined	Formula	Program	Satisfiability
ML	PTIME-compl.	in PTIME	in PTIME	PSPACE-compl.
ASL	PSPACE-compl.	in PTIME	in PTIME	undecidable
PSL	PSPACE-compl.	in PTIME	in PTIME	undecidable
SML	PSPACE-compl.	in PTIME	in PTIME	undecidable
FOL	PSPACE-compl.	PSPACE-compl.	in PTIME	undecidable

2 Temporary Unavailability Logic (TUL)

We will first deal with the simplest case of non-caustic sabotage. The sabotage operator that we will define removes a single edge from the model, just like in SML. However, when a nested sabotage operator is evaluated, the removed edge is restored before the new edge to remove is picked. As in [Roh04], we base our relational structures on multi-graphs, so it matters *how many* edges there are between a pair of nodes. Edges that have a multiplicity of two or more can never be eliminated by the sabotage operator, and thus have a different status than edges with a multiplicity of one.

2.1 Formalisation

Definition 2.1. Let Φ be a set of proposition letters. The language *TUL* is the smallest set of formulae containing all formulae generated by the the grammar

$$\varphi \quad ::= \quad \top \mid p \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \Diamond\varphi \mid \diamondsuit\varphi \tag{1}$$

The operator $\diamondsuit$ is called the *temporary unavailability* operator. Analogous to the dual operator $\Box$, we define $\boxdot\varphi := \neg\diamondsuit\neg\varphi$. The fragment of TUL that consists of $\diamondsuit$-free formulae is equivalent to the standard modal logic (ML).

Definition 2.2. Let O be a set of modal operators. The *operator depth* with respect to O, is given by $\mathrm{od}_O(\varphi)$ where

φ	$\mapsto$	$\mathrm{od}_O(\varphi)$
$\top$	$\mapsto$	0
p	$\mapsto$	0
$\neg\varphi$	$\mapsto$	$\mathrm{od}_O(\varphi)$
$\varphi \vee \psi$	$\mapsto$	$\max\{\mathrm{od}_O(\varphi), \mathrm{od}_O(\psi)\}$
$\triangle\varphi$	$\mapsto$	$\begin{cases}\mathrm{od}_O(\varphi)+1 & \text{if } \triangle \in O \\ \mathrm{od}_O(\varphi) & \text{otherwise}\end{cases}$

For a modal language with operator set O, we abbreviate $\mathrm{od}_O(\varphi)$ to $\mathrm{od}(\varphi)$. The modal language will always be clear from the context.

Definition 2.3. Let W be a non-empty set of worlds, $R : W \times W \to \mathbb{N}$ an accessibility multi-relation and $V : \Phi \to \wp(W)$ a valuation. We let a *model* be a tuple $\mathfrak{M} = (W^{\mathfrak{M}}, R^{\mathfrak{M}}, V^{\mathfrak{M}})$ as usual.

Definition 2.4. For $w, v \in W$, we write wRv and also $(w, v) \in R$ for $R(w, v) > 0$ and conversely $w\not{R}v$ for $R(w, v) = 0$. Also $|R| = \sum_{w,v \in W} R(w, v)$.

Definition 2.5. We define two multi-relation alteration operators, $+$ and $-$, that add a unit to or remove a unit from the multiplicity of a certain edge. For $(r, s), (t, u) \in W \times W$ let:

$$\big(R + (r, s)\big)(u, v) \quad = \quad \begin{cases} R(u, v) + 1 & \text{if } (u, v) = (r, s) \\ R(u, v) & \text{otherwise}\end{cases} \tag{2}$$

$$\big(R - (r, s)\big)(u, v) \quad = \quad \begin{cases} R(u, v) - 1 & \text{if } (u, v) = (r, s) \\ R(u, v) & \text{otherwise}\end{cases} \tag{3}$$

Note that the edge removal operator uses subtraction on the natural numbers, for which $0 - x = 0$. This should not matter, as we do not intend to use the function in this case.

Remark 2.6. (3) defines a right inverse of (2), for $\big(R + (r, s)\big) - (r, s) = R$.

Definition 2.7. We now lift the multi-relation operators to models. Again for $(r, s), (t, u) \in W \times W$ let:

$$\mathfrak{M} + (r, s) = (W, R + (r, s), V) \tag{4}$$
$$\mathfrak{M} - (r, s) = (W, R - (r, s), V) \tag{5}$$
$$\mathfrak{M}^{(t,u)}_{(r,s)} = (\mathfrak{M} + (r, s)) - (t, u) \tag{6}$$

Remark 2.8. In (6), when $(r, s) = (u, v)$ the model remains unaltered.

Definition 2.9. Now we can inductively define *TUL formula satisfaction* in a model $\mathfrak{M}$, world $w \in W$ and edge $(r, s) \in W \times W$:

$$\mathfrak{M}, w, (r, s) \Vdash \top \tag{7}$$
$$\mathfrak{M}, w, (r, s) \Vdash p \quad \Leftrightarrow \quad w \in V(p) \tag{8}$$
$$\mathfrak{M}, w, (r, s) \Vdash \neg\varphi \quad \Leftrightarrow \quad \mathfrak{M}, w, (r, s) \nVdash \varphi \tag{9}$$
$$\mathfrak{M}, w, (r, s) \Vdash \varphi \vee \psi \quad \Leftrightarrow \quad \mathfrak{M}, w, (r, s) \Vdash \varphi \quad \text{or} \quad \mathfrak{M}, w, (r, s) \Vdash \psi \tag{10}$$
$$\mathfrak{M}, w, (r, s) \Vdash \Diamond\varphi \quad \Leftrightarrow \quad \exists v : wRv \wedge \mathfrak{M}, v, (r, s) \Vdash \varphi \tag{11}$$
$$\mathfrak{M}, w, (r, s) \Vdash \diamondsuit\varphi \quad \Leftrightarrow \quad \exists (t, u) \in R : \mathfrak{M}^{(t,u)}_{(r,s)}, w, (t, u) \Vdash \varphi \tag{12}$$

Definition 2.10. And finally we define formula satisfaction in a pointed model

$$\mathfrak{M}, w \Vdash \varphi \quad \Leftrightarrow \quad \exists (r, s) \in R : \mathfrak{M} - (r, s), w, (r, s) \Vdash \varphi \tag{13}$$
$$\mathfrak{M}, w \Vdash_\forall \varphi \quad \Leftrightarrow \quad \forall (r, s) \in R : \mathfrak{M} - (r, s), w, (r, s) \Vdash \varphi \tag{14}$$

The choice for existential quantification as our primary definition is somewhat arbitrary, but it is in line with our preference for $\top$, $\vee$, $\Diamond$ over $\bot$,$\wedge$,$\Box$. There is some motivation for universal quantification too, especially when we want to reason about safety or security, that is, satisfaction irrespective of the first edge where disaster strikes.

Remark 2.11. Let $\mathfrak{M}$ be a TUL model, $w \in W$ and φ a TUL formula. During evaluation of $\mathfrak{M}, w \vDash \varphi$ we need to evaluate certain subformulae of φ in pointed models. Tracing the definitions given above, we see that in each nested evaluation $\mathfrak{N}, v, (r, s) \vDash \psi$, the actual model $\mathfrak{N}$ equals $\mathfrak{M} - (r, s)$. We could have chosen an alternative definition of truth where we drag the original model along, like:

$$\mathfrak{M} - (r, s), w \Vdash \diamondsuit\varphi \quad \Leftrightarrow \quad \exists (t, u) \in R : \mathfrak{M} - (t, u), w \Vdash \varphi \tag{15}$$

Our current definition of truth is specified in terms of arbitrary models to emphasise the fact that we are really *altering* the model. This will become a more important issue when we consider more advanced temporary unavailability operators in §3.

2.2 Translation to First Order Logic

Definition 2.12. Let $\mathfrak{M} = (W, R, V)$ be a TUL model. We define its corresponding First Order Logic (FOL) structure

$$\hat{\mathfrak{M}} = (W, \left\{P_p \mid p \in \Phi\right\}, R)$$

where $P_p = V(p)$.

Definition 2.13. Given a TUL formula φ, we inductively define its *translation to FOL* $\hat{\varphi}(x, y, z)$. The variables x, y, z in the FOL formula are used to represent the rôle of w, r, s respectively in the semantics of TUL.

φ	$\mapsto$	$\hat{\varphi}(x, y, z)$
$\top$	$\mapsto$	$\top$
p	$\mapsto$	$P_p(x)$
$\neg\varphi$	$\mapsto$	$\neg\hat{\varphi}(x, y, z)$
$\varphi \vee \psi$	$\mapsto$	$\hat{\varphi}(x, y, z) \vee \hat{\psi}(x, y, z)$
$\Diamond\varphi$	$\mapsto$	$\exists x' : xRx' \wedge \neg(x = y \wedge x' = z) \wedge \hat{\varphi}(x', y, z)$
$\diamondsuit\varphi$	$\mapsto$	$\exists (y', z') \in R : \hat{\varphi}(x, y', z')$

Proposition 2.14. *$\hat{\varphi}(x, y, z)$ is equivalent to a formula of FOL that uses only four variables.*

Proof. It is well-known that the ML side of the translation can be done using only two variables. Note that the rule for $\diamondsuit$ does not use y, z at all, so in particular does not pass them on to the inductive application of the translation. Hence we can repeatedly reuse a single pair y', z' for each (nested) occurrence of $\diamondsuit$. □

Remark 2.15. The reduction of TUL to first order logic does not yield formulae in the Loosely Guarded Fragment, as discussed in [vB05b]. The culprits are the definition for $\Diamond$, in which the existential quantifier is not restrained by a conjunction of atoms, and the definition for wD, where the existential quantifier is properly restrained by a single atom, but the variable x does not co-occur in the atom with the existentially quantified variables.

Proposition 2.16. *Let $\mathfrak{M} = (W, R, V)$ be a TUL model. For all $w, r, s, \in W$ we have*

$$\mathfrak{M} - (r, s), w \Vdash \varphi \quad \textit{iff} \quad \hat{\mathfrak{M}} \vDash \hat{\varphi}(w, r, s)$$

Proof. By induction on the structure of φ. The only interesting cases are $\Diamond$ and $\diamondsuit$.

- The formula is of the form $\Diamond\varphi$. We need to show

  $$\mathfrak{M} - (r, s), w \Vdash \Diamond\varphi \quad \text{iff} \quad \hat{\mathfrak{M}} \vDash \widehat{\Diamond\varphi}(w, r, s)$$

  that is

  $$\begin{aligned} &\exists v : w\,(R - (r, s))\, v \wedge \mathfrak{M} - (r, s), v \Vdash \varphi \quad \text{iff} \\ &\exists x' : xRx' \wedge \neg(x = y \wedge x' = z) \wedge \hat{\varphi}(x', y, z) \end{aligned} \tag{16}$$

  which follows by the induction hypothesis and renaming of variables.

- The formula is of the form $\diamondsuit\varphi$. We need to show

$$\mathfrak{M} - (r,s), w \Vdash \diamondsuit\varphi \quad \text{iff} \quad \hat{\mathfrak{M}} \vDash \widehat{\diamondsuit\varphi}(w,r,s)$$

Omitting the outer models and noting $(\mathfrak{M} - (r,s))^{(t,u)}_{(r,s)} = \mathfrak{M} - (t,u)$ that is

$$\exists(t,u) \in R : \mathfrak{M} - (t,u), w \Vdash \varphi \quad \text{iff} \quad \exists(y',z') \in R : \hat{\varphi}(x,y',z')$$

which also follows by the induction hypothesis and renaming of variables. □

2.3 Model Checking Complexity

We turn to the problem of TUL model checking. We consider the instance of model checking where both the model and the formula are considered as input. Given a formula φ, model $\mathfrak{M}$ and world w, how hard is it to determine whether $\mathfrak{M}, w \Vdash \varphi$? The translation to FOL of the previous section places the problem in PSPACE, but we can do better:

Proposition 2.17. *The complexity of TUL model checking is in PTIME.*

Proof. Let φ be a TUL formula and $\mathfrak{M} = (W,R,V)$ a TUL model. First observe that

1. There are at most $\left|\varphi\right|$ subformulae of φ.
2. There are at most $|W|^2$ many submodels of $\mathfrak{M}$ where a single edge's multiplicity has been decremented by one.

We give a procedure for determining truth of φ in all worlds of $\mathfrak{M}$ simultaneously in Algorithm 1. It is a dynamic programming algorithm, that constructs a table of partial solutions which are subsequently reused to answer more complicated questions fast. The algorithm performs $|\varphi| \cdot |W|^2 \cdot |W|$ evaluations, each requiring $|W|^2$ lookups in the worst case (which occurs at subformulae with $\diamondsuit$ as their main operator). The initial setup in line 1 and final sweep in line 22 do not surpass this amount of work. We conclude that the total needed time is

$$O\left(|\varphi| \cdot |W|^5\right) \quad \in \quad PTIME\left(|\varphi|, |\mathfrak{M}|\right) \qquad \square$$

Corollary 2.18. *Formula and program complexity for TUL model checking are in PTIME too.*

Corollary 2.19. *As TUL is an extension of ML, and ML model checking is PTIME complete (see Table 1), TUL model checking is PTIME complete as well.*

So in this respect TUL keeps closer to ML than to FOL. This is in a sense obvious. The caustic sabotage operators can potentially remove any subset of the edges, thus requiring consideration of exponentially many submodels. TUL logic can talk about at most $|W|^2$ submodels, thus remaining within polynomial time.

Algorithm 1 Dynamic programming algorithm for TUL model checking.

Construct $\varphi_1, \varphi_2, \ldots, \varphi_n$, a list of all subformulae of φ ordered by increasing formula length. Obviously φ itself is the longest subformula, so $\varphi_n = \varphi$.
Allocate a table T of truth-values of size $n \times |W|^2 \times |W|$.
for all φ_i **do** {the first dimension of T}
  for all $(r, s) \in W \times W$ **do** {the second dimension of T}
    for all $w \in W$ **do** {the third dimension of T}
      if $\varphi_i = \top$ **then**
        $T[i, (r, s), w] =$ **true**
      else if $\varphi_i = p$ **then**
        $T[i, (r, s), w] = V(p)$
      else if $\varphi_i = \neg\varphi_a$ **then** $\{a < i\}$
        $T[i, (r, s), w] =$ **not** $T[a, (r, s), w]$
      else if $\varphi_i = \varphi_a \vee \varphi_b$ **then** $\{a, b < i\}$
        $T[i, (r, s), w] = T[a, (r, s), w]$ **or** $T[b, (r, s), w]$
      else if $\varphi_i = \Diamond\varphi_a$ **then** $\{a < i\}$
        $T[i, (r, s), w] = 1$ iff there is some v s.t. $w(R{-}(r, s))v$ and $T[a, (r, s), v]$
      else if $\varphi_i = \diamondsuit\varphi_a$ **then** $\{a < i\}$
        $T[i, (r, s), w] = 1$ iff there are $t, u \in W$ s.t. $T[a, (t, u), w]$
      end if
    end for
  end for
end for
return 1 iff there are $w, r, s \in W$ s.t. $T[n, (r, s), w]$.

2.4 Satisfiability

TUL model checking can be done efficiently, as shown by the previous section, and in this respect TUL resembles standard ML. This similarity does not extend further, as shown by the following results. Multi-agent TUL extends TUL by including a sabotage modality for several agents.

Proposition 2.20. *Multi-agent TUL does not have the finite model property:*

Proof. The argument in [Roh04, definition of φ_∞ on p63] carries over to multi-agent TUL, for it uses only singly nested occurrences of the global sabotage operator. □

Proposition 2.21. *The satisfiability problem for multi-agent TUL is undecidable.*

Proof. The argument in [Roh04, section 3.3] that reduces Post's correspondence problem to SML using only singly nested global sabotage operators carries over to TUL. □

2.5 Several Considerations

The following points require some attention.

- In the introduction we talked about temporarily unavailable entities that will eventually be restored. The current semantics allows the case that a certain single edge remains unavailable for the entire evaluation of the formula. One could add a *fairness* condition, which states that each nested evaluation of the $\diamondsuit$ operator must take out a different edge.
- One could consider the ability to remove several edges. This requires an edge buffer together with an edge replacement protocol. We could think of for example the bag, queue or stack of fixed size. This allows the model to be altered in a more intricate fashion, while still allowing for the eventual return of removed edges.
- One could desire the ability to remove worlds. We have restricted ourself to edge removal, but we can model deleting a world by simultaneously suppressing all incident edges. We call this operation *blackout*.

These points are the motivation for the generalisation of TUL to GML in the next section.

3 General Modification Logic (GML)

This section defines the General Modification Logic, a new logic that incorporates the product concept. It is a proper generalisation of TUL, abstracting over the exact modification that is performed on the model. GML supports a.o. fairness, multiple edge removal and blackout.

3.1 Modification frames

Let $\mathbb{F}$ be the class of Kripke frames.

Definition 3.1. A function[1] $i : \mathbb{F} \to \mathbb{F}$ is called a *modification frame generator* if for all $\mathfrak{F} = (W, R) \in \mathbb{F}$ there are a set F, a directed graph $E \subseteq F \times F$ over F and directed multi-graphs $R_f \subseteq W \times W$ over W for all $f \in F$, such that

$$i(\mathfrak{F}) = \left(F, E, \left\{R_f\right\}_{f \in F}\right).$$

We lift i from frames to models thus: $i(\mathfrak{M}) = i(W, R)$. We call the structure $i(\mathfrak{M}) = \left(F, E, \left\{R_f\right\}_{f \in F}\right)$ the *modification frame* generated by i from $\mathfrak{M}$ and denote it by $\mathfrak{T}$ when i and $\mathfrak{M}$ are unambiguous. Furthermore, we call the elements of F *modifications*. We regard them as names for operations on R, and for each $f \in F$ we say that R_f is the *result* of f.

Example 3.2. For TUL, we can specify the corresponding modification frame generator i_{TUL} as follows

$$\begin{aligned} i_{TUL}(W, R) &= \left(F, E, \left\{R_f\right\}_{f \in F}\right) && (17)\\ F &= \{(a, b) \mid aRb\} && (18)\\ E &= F \times F && (19)\\ R_{(a,b)} &= R - (a, b) && (20) \end{aligned}$$

We take as modifications F all edges in the source frame (flattened from a multi-graph to a normal graph). The inter-modification accessibility graph E is the complete graph over F, and the result R_f (an accessibility relation over W) that corresponds to f is obtained by removing the edge f from R.

Example 3.3. Consider the Kripke frame $\mathfrak{F}$ in Figure 0a. Application of i_{TUL} to this frame yields the modification frame $\mathfrak{T}$ in Figure 0b. Each modification $f \in F$ (node of $\mathfrak{T}$) corresponds to an edge of $\mathfrak{F}$. The accessibility relation E between modifications is the complete graph, as shown by the undirected arcs. Reflexive arcs are omitted from the figure for brevity.

3.2 Logic

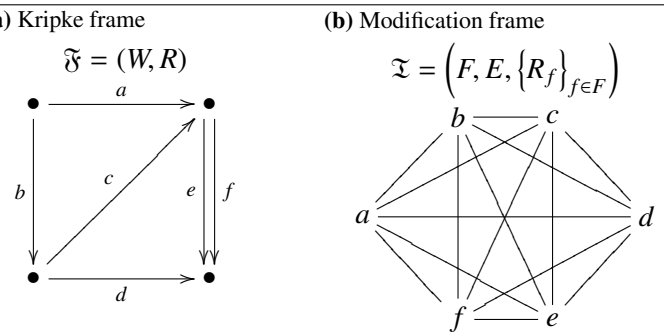
Now that we have defined the structures of interest, we turn to the logic.

Definition 3.4. Let Φ be a set of proposition letters. The language of General Modification Logic (GML) is the smallest set of formulae containing all formulae generated by the the grammar

$$\varphi \quad ::= \quad \top \mid p \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \Diamond\varphi \mid \diamondsuit\varphi \qquad (21)$$

[1] Technically, i is a class function. In particular, it is a formula in the language of set theory. If interest is limited to the computable case only, we can substitute any Turing complete notion of algorithm instead.

Figure 1 Example (a) source frame and (b) modification frame generated by i_{TUL}.



The operator $\diamondsuit$ is called the *modification operator*. Analogous to $\Box$, the dual operator of $\diamondsuit$, we define $\boxdot\varphi := \neg\diamondsuit\neg\varphi$. The fragment of GML that consists of $\diamondsuit$-free formulae is equivalent to the standard modal logic (ML).

Definition 3.5. The semantics of GML are defined relative to a modification frame generator i that we consider fixed to avoid subscripts. Fix a model $\mathfrak{M} = (W, R, V)$, and let $\mathfrak{T} = i(\mathfrak{M})$. Additionally fix a world $w \in W$ and a modification $f \in F$. Furthermore fix a GML formula φ. We inductively define truth in the pointed model, pointed frame combination $\mathfrak{M}, w, \mathfrak{T}, f$ by

$$\mathfrak{M}, w, \mathfrak{T}, f \Vdash \top \tag{22}$$

$$\mathfrak{M}, w, \mathfrak{T}, f \Vdash p \quad \Leftrightarrow \quad w \in V(p) \tag{23}$$

$$\mathfrak{M}, w, \mathfrak{T}, f \Vdash \neg\varphi \quad \Leftrightarrow \quad \mathfrak{M}, w, \mathfrak{T}, f \nVdash \varphi \tag{24}$$

$$\mathfrak{M}, w, \mathfrak{T}, f \Vdash \varphi \vee \psi \quad \Leftrightarrow \quad \mathfrak{M}, w, \mathfrak{T}, f \Vdash \varphi \quad \text{or} \quad \mathfrak{M}, w, \mathfrak{T}, f \Vdash \psi \tag{25}$$

$$\mathfrak{M}, w, \mathfrak{T}, f \Vdash \diamondsuit\varphi \quad \Leftrightarrow \quad \exists v \in W : wR_f v \wedge \mathfrak{M}, v, \mathfrak{T}, f \Vdash \varphi \tag{26}$$

$$\mathfrak{M}, w, \mathfrak{T}, f \Vdash \diamondsuit\varphi \quad \Leftrightarrow \quad \exists g \in E : fEg \wedge \mathfrak{M}, w, \mathfrak{T}, g \Vdash \varphi \tag{27}$$

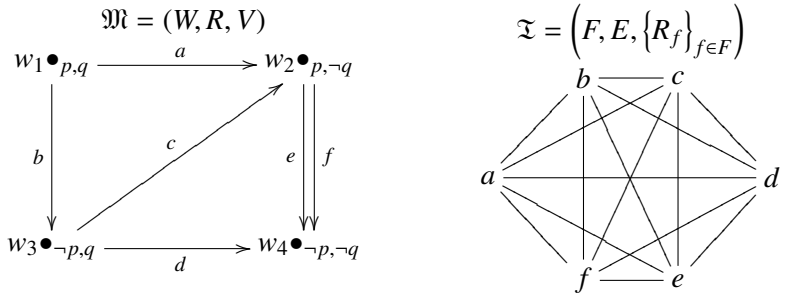
We call w the *first focus* or *current world* and f the *second focus* or *current modification*. (22) – (25) are just propositional logic with additional ballast. (26), defining the GML diamond operator, is similar to the corresponding clause in standard ML, but reads $wR_f v$ instead of wRv. This ensures that the first focus only uses edges that exist in R_f, the result of the current modification. (27), defining the GML modification operator, allows the second focus to make transitions in E, independent of the current world. Hence the GML modification operator behaves as a regular diamond in the modification frame.

Example 3.6. Consider the model $\mathfrak{M}$ of Figure 1a, which adds a valuation to the

frame of Figure 0a. Also let $\mathfrak{T}$ be $i_{TUL}(\mathfrak{M})$ as shown in Figure 1b. Then

$$\mathfrak{M}, w_1, \mathfrak{T}, a \Vdash p \wedge q \qquad \text{by propositional logic} \tag{28}$$
$$\mathfrak{M}, w_1, \mathfrak{T}, a \nVdash \Diamond p \qquad (w_1, w_2) = a \notin R_a \tag{29}$$
$$\mathfrak{M}, w_1, \mathfrak{T}, a \Vdash \Diamond q \qquad (w_1, w_3) = b \in R_a \tag{30}$$
$$\mathfrak{M}, w_1, \mathfrak{T}, b \Vdash \Diamond p \qquad (w_1, w_2) = a \in R_b \tag{31}$$
$$\mathfrak{M}, w_1, \mathfrak{T}, a \Vdash \diamondsuit\Diamond p \qquad (a, b) \in E \text{ and } (31) \tag{32}$$
$$\mathfrak{M}, w_2, \mathfrak{T}, f \Vdash \boxdot\Diamond(\neg p \wedge \neg q) \qquad (w_2, w_4) \text{ has multiplicity 2} \tag{33}$$

Figure 2 Example (a) model and (b) modification frame generated by i_{TUL}.



3.3 Game

These semantics can easily be interpreted as a two player formula game. The game starts with a GML formula φ, a Kripke model $\mathfrak{M} = (W, R, V)$ called the *original model*, a world w, a modification frame $\mathfrak{T}$ and a modification f. The players are called Verifier and Falsifier, and they try to do to φ what their name suggests. We call (W, R_f, V) the *current model*. The player to move is determined by the structure of the formula:

- $\top$. Verifier wins.
- $\neg\varphi$. Players exchange roles, the game continues with the formula φ.
- $\varphi \vee \psi$. Verifier picks one of φ, ψ to continue the game.
- $\Diamond\varphi$. Verifier makes a move (w, v) in the current model. The game continues with the formula φ and world v.
- $\diamondsuit\varphi$. Verifier makes a move (f, g) in the modification frame. The game continues with the formula φ and the modification g, hence with new current model (W, R_g, V).

Proposition 3.7. *Verifier has a winning strategy iff* $\mathfrak{M}, w, \mathfrak{T}, f \Vdash \varphi$.

Proof. An easy induction on the structure of φ. □

3.4 Modification Product

As pointed out in §3.2, the modal operators of GML are similar to the standard diamond operator. This section describes an operation called modification product, that maps a model and corresponding modification frame to a new structure, in which standard bi-agent modal logic suffices.

Definition 3.8. Starting from a Kripke model $\mathfrak{M} = (W, R, V)$ and the modification frame $\mathfrak{T} = i(\mathfrak{M}) = \left(F, E, \left\{R_f\right\}_{f\in F}\right)$ generated by i from $\mathfrak{M}$ we define the *modification product* by

$$\mathfrak{M} \times \mathfrak{T} = \left(W \times F, R_1^{\mathfrak{M}\times\mathfrak{T}}, R_2^{\mathfrak{M}\times\mathfrak{T}}, V^{\mathfrak{M}\times\mathfrak{T}}\right) \tag{34}$$

where

$$(w, f) \in V^{\mathfrak{M}\times\mathfrak{T}}(p) \quad \Leftrightarrow \quad w \in V(p) \tag{35}$$

$$(w, f)R_1^{\mathfrak{M}\times\mathfrak{T}}(v, g) \quad \Leftrightarrow \quad f = g \wedge wR_f v \tag{36}$$

$$(w, f)R_2^{\mathfrak{M}\times\mathfrak{T}}(v, g) \quad \Leftrightarrow \quad w = v \wedge fEg \tag{37}$$

Definition 3.9. For $f \in F$, we call the restriction of the modification product $\mathfrak{M}\times\mathfrak{T}$ to worlds in $W \times \{f\}$ the *modification image* of f. The only possible accessibility arrows for the second agent within the accessibility image of f are reflexive arrows, and these are present iff fEf.

The idea behind these definitions is this: the modification frame specifies a set of modifications. To form the modification product, we "apply" each modification to the original model and concatenate the resulting modification images. The accessibility relation for the first agent is determined by the result of each modification, and relates different worlds within the same modification image. On the other hand, the accessibility relation for the second agent relates images of the same original world between different modification images, for those modifications that are related in the modification frame. The valuations do not consider edges at all, and they are just replicated in each modification image.

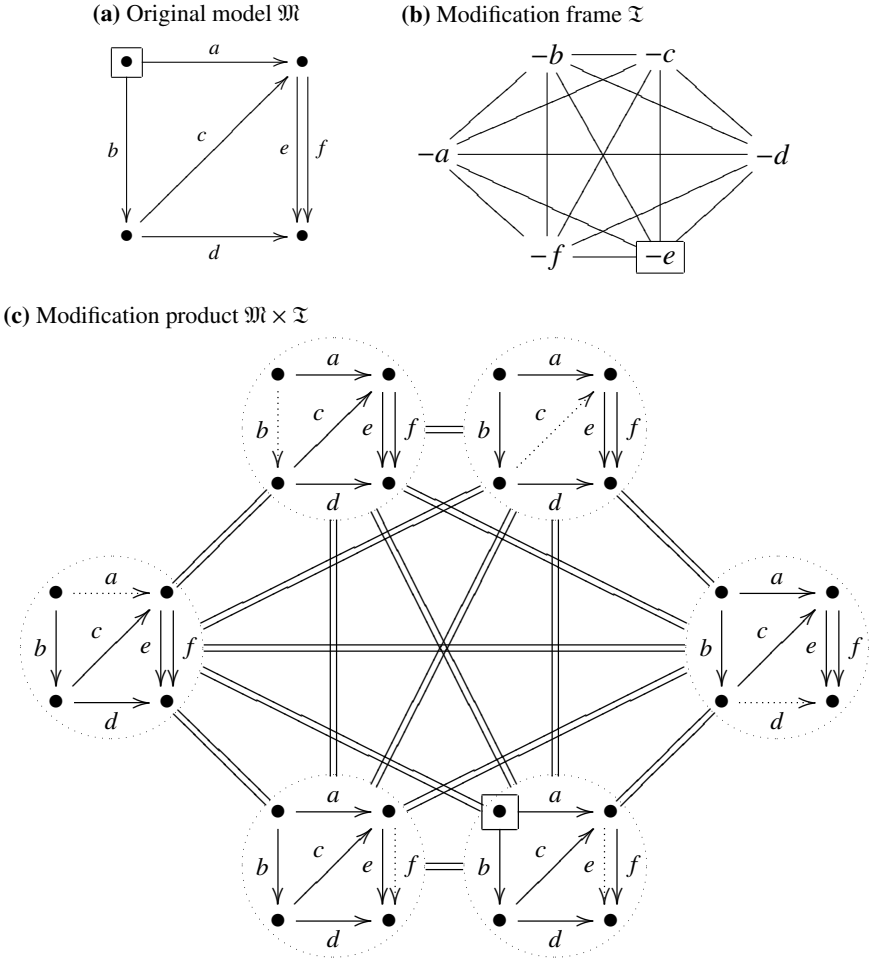
Example 3.10. We have shown a particular model and its TUL modification frame in Figure 2. The corresponding modification product is shown in Figure 3. To emphasise the accessibility relation, this figure omits the valuations. The valuation in the product model is formed by repeating copies of the valuation of the source model.

3.5 Reduction to ML-2

We reduce GML to ML-2, bi-agent standard modal logic.

Definition 3.11. The reduction of a GML formula φ to a ML-2 formula $\breve{\varphi}$ is given

Figure 3 Modification product construction. In (c), the individual modification images are enclosed by dotted circles, and the removed edge within each modification image is shown as a dotted arrow. Double edges between modification images of f and g abbreviate that (w, f) and (w, g) are related for all $w \in W$. As an example, when the current world and current modification are given by the nodes in rectangles in (a) and (b), then the resulting current world in the modification product is given by the node in the rectangle in (c).



by

φ	$\mapsto$	$\breve{\varphi}$
$\top$	$\mapsto$	$\top$
p	$\mapsto$	p
$\neg\varphi$	$\mapsto$	$\neg\breve{\varphi}$
$\varphi \vee \psi$	$\mapsto$	$\breve{\varphi} \vee \breve{\psi}$
$\Diamond\varphi$	$\mapsto$	$\Diamond_1\breve{\varphi}$
$\circledast\varphi$	$\mapsto$	$\Diamond_2\breve{\varphi}$

Proposition 3.12. *The reduction preserves truth, in other words*

$$\mathfrak{M}, w, \mathfrak{T}, f \Vdash \varphi \quad \Leftrightarrow \quad \mathfrak{M} \times \mathfrak{T}, (w, f) \Vdash \breve{\varphi} \tag{38}$$

Proof. By induction on the complexity of φ, and case distinction on structure

- $\varphi = \top$, $\varphi = \neg\psi$, $\varphi = \varphi_1 \vee \varphi_2$ are all trivial.
- $\varphi = p$. We need to show

  $$\mathfrak{M}, w, \mathfrak{T}, f \Vdash p \quad \Leftrightarrow \quad \mathfrak{M} \times \mathfrak{T}, (w, f) \Vdash p \tag{39}$$

  which both are equivalent to $w \in V(p)$.
- $\varphi = \Diamond\psi$. We need to show

  $$\mathfrak{M}, w, \mathfrak{T}, f \Vdash \Diamond\psi \quad \Leftrightarrow \quad \mathfrak{M} \times \mathfrak{T}, (w, f) \Vdash \Diamond_1\breve{\psi} \tag{40}$$

  well

  $$\begin{aligned}
  & \mathfrak{M} \times \mathfrak{T}, (w, f) \Vdash \Diamond_1\breve{\psi} && (41)\\
  \Leftrightarrow\ & \exists (v, g) : (w, f) R_1^{\mathfrak{M}\times\mathfrak{T}} (v, g) \wedge \mathfrak{M} \times \mathfrak{T}, (v, g) \Vdash \breve{\psi} && (42)\\
  \Leftrightarrow\ & \exists v : w R_f v \wedge \mathfrak{M} \times \mathfrak{T}, (v, f) \Vdash \breve{\psi} && (43)\\
  \overset{\Leftrightarrow}{\text{IH}}\ & \exists v : w R_f v \wedge \mathfrak{M}, v, \mathfrak{T}, f \Vdash \psi && (44)\\
  \Leftrightarrow\ & \mathfrak{M}, w, \mathfrak{T}, f \Vdash \Diamond\psi && (45)
  \end{aligned}$$

- $\varphi = \circledast\psi$. We need to show

  $$\mathfrak{M}, w, \mathfrak{T}, f \Vdash \circledast\psi \quad \Leftrightarrow \quad \mathfrak{M} \times \mathfrak{T}, (w, f) \Vdash \Diamond_2\breve{\psi} \tag{46}$$

  well

  $$\begin{aligned}
  & \mathfrak{M} \times \mathfrak{T}, (w, f) \Vdash \Diamond_2\breve{\psi} && (47)\\
  \Leftrightarrow\ & \exists (v, g) : (w, f) R_2^{\mathfrak{M}\times\mathfrak{T}} (v, g) \wedge \mathfrak{M} \times \mathfrak{T}, (v, g) \Vdash \breve{\psi} && (48)\\
  \Leftrightarrow\ & \exists g : f E g \wedge \mathfrak{M} \times \mathfrak{T}, (w, g) \Vdash \breve{\psi} && (49)\\
  \overset{\Leftrightarrow}{\text{IH}}\ & \exists g : f E g \wedge \mathfrak{M}, w, \mathfrak{T}, g \Vdash \psi && (50)\\
  \Leftrightarrow\ & \mathfrak{M}, w, \mathfrak{T}, f \Vdash \circledast\psi && (51)
  \end{aligned}$$

This covers all cases. □

3.6 Satisfiability

We saw that TUL, the simplest non-caustic sabotage logic, can be reduced to GML, from which it follows that the satisfiability problem for GML is generally undecidable.[2] One may think that this is in contradiction with the foregoing reduction of GML to ML, as the satisfiability problem for ML is certainly decidable! The problem with this reasoning is of course that most ML models are not images of GML models under the above reduction.

3.7 Model Checking Complexity

We already saw that model checking for TUL is in PTIME. This means that for certain classes of modification frame generators, the model checking problem for GML is in PTIME as well. We can generalise this result by imposing a polynomial bound on the size of the modification frame, and this will place combined formula complexity for GML in PTIME in this case.

Proposition 3.13. *Let i be a modification frame generator, and let there be an algorithm that for all finite $\mathfrak{F} = (W, R)$ computes $i(\mathfrak{F})$ in time $O(|W|^n)$ for some fixed n. (This implies that $|F| \in O(|W|^n)$.) Then the combined model checking problem for i-GML is in PTIME.*

Proof. Let $\mathfrak{M} = (W, R, V)$ be a Kripke model, φ a GML formula. We first compute $\mathfrak{T} = i(\mathfrak{M})$, which we can do in time polynomial in $|\mathfrak{M}|$ by assumption. We proceed by computing the modification product $\mathfrak{M} \times \mathfrak{T}$, which we can do in time polynomial in $|\mathfrak{M}|$, yielding a bi-agent Kripke model of polynomial size in $|\mathfrak{M}|$. We then compute $\check{\varphi}$, in time linear in $\left|\varphi\right|$. We finish by applying the model checking algorithm for standard modal logic, which runs in polynomial time in both model and formula size. Hence the entire procedure can be completed in polynomial time in $|\mathfrak{M}|$ and $\left|\varphi\right|$. □

Remark 3.14. This is a generalisation of Proposition 2.17.

3.8 Applications

We demonstrate the expressive power of GML by modelling the sabotage operators that we described in section §2.5, which lists possible extensions of TUL.

Fairness By taking E irreflexive, we get a fair sabotage operator. Evaluation of the sabotage operator requires making a move, picking an accessible modification, in the modification frame. When the accessibility relation E is irreflexive, we are forced to choose a *different* modification.

[2] It is not for certain degenerate modification frame generators, of course.

SML By taking F to contain any subrelation of R and fEg iff g contains a single edge less than f, we get the global sabotage modality of [Roh04]. Note that we identify modifications and results in this case.

$$\begin{aligned} i_{SML} &= \left(F, E, \left\{R_f\right\}_{f\in F}\right) && (52)\\ F &= \{f : R \times R \to \mathbb{N} \mid \forall x, y \in W : R(x,y) \geq f(x,y)\} && (53)\\ fEg &\Leftrightarrow \exists e \in f : g = f - e && (54)\\ R_f &= f && (55) \end{aligned}$$

This does not imply that we can do SML model checking in PTIME, for the set of modifications is exponentially large in $\mathfrak{M}$.

ASL We can not model adjacent sabotage, for the accessibility relation between modifications is not uniform, but in fact depends upon the focus of evaluation.

PSL To model path sabotage, we must make F even larger, including a modification for every subrelation of R like for SML, but now also labelling them with the current path deletion focus.

$$\begin{aligned} i_{PSL} &= \left(F, E, \left\{R_f\right\}_{f\in F}\right) && (56)\\ F &= W \times \{f : R \times R \to \mathbb{N} \mid \forall x, y \in W : R(x,y) \geq f(x,y)\} && (57)\\ (w,f)E(v,g) &\Leftrightarrow wfv \wedge g = f - (w,v) && (58)\\ R_{(w,f)} &= f && (59) \end{aligned}$$

Two modifications (w, f) and (v, g) are related if the edge (w, v) is present in f, and g is the result of deleting this edge from f.

Blackout The blackout operator simultaneously removes all edges incident to a certain world. It is used to model removing worlds in terms of edge deletions.

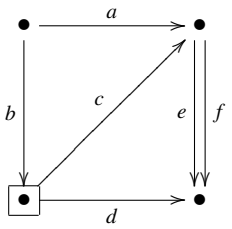
$$\begin{aligned} i_{BO} &= \left(F, E, \left\{R_f\right\}_{f\in F}\right) && (60)\\ F &= W && (61)\\ E &= F \times F && (62)\\ uR_w v &\Leftrightarrow uRv \wedge w \neq v && (63) \end{aligned}$$

Figure 4 shows an application of the fair blackout operator to the Idol model.

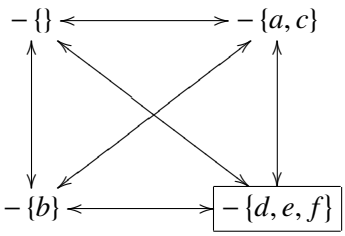
Protocols for multiple edge removal Given an algebraic specification S of a collection (say a bag, queue or stack), we take the modifications to be the states in the external behaviour of S (see [Fok00]). We then take R_f to be R minus all edges contained in the collection as indicated by f. Finally we take fEg if f has a primitive transition to g.

Figure 4 Example of the fair blackout operator.

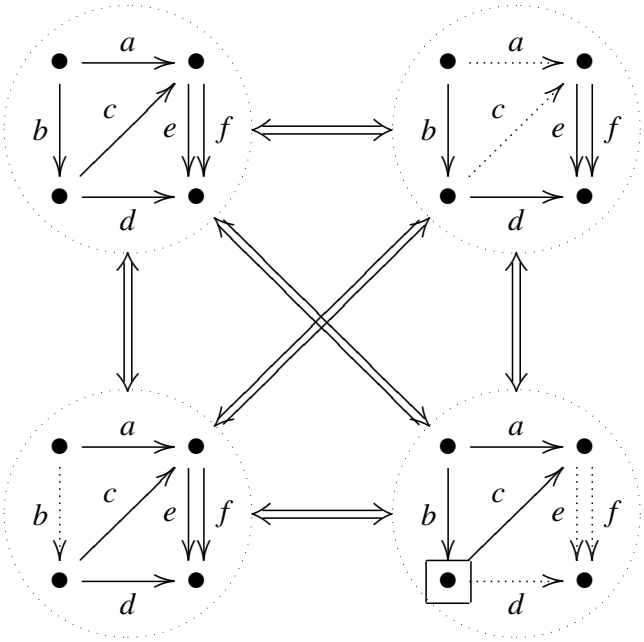
(a) Original (Idol) model $\mathfrak{M}$



(b) Modification frame $\mathfrak{T}$



(c) Modification product $\mathfrak{M} \times \mathfrak{T}$



4 Conclusion

We introduced and motivated TUL, a new sabotage modal logic. It lies in between ML and SML, for its model checking complexity is PTIME complete (like ML), but its satisfiability problem is undecidable (like SML).

We then generalised TUL and arrived at GML, which uses a modification frame generator to merge a Kripke model that represents the common starting point and a modification frame; an accessibility graph over modifications. We introduced modification frame generators to construct modification frames for arbitrary frames. We reduced GML to bi-agent standard modal logic, and proved that its model checking problem is PTIME complete under certain conditions on the modification frame generator.

Table 2 summarises the results of this paper, and puts them into context.

Table 2 Complexities of standard problems for sabotage logics and first order logic.

Logic	Combined	Formula	Program	Satisfiability
ML	PTIME-compl.	in PTIME	in PTIME	PSPACE-compl.
TUL	PTIME-compl.	in PTIME	in PTIME	undecidable
GML[3]	PTIME-compl.	in PTIME	in PTIME	undecidable
GML[4]	?	?	?	undecidable
ASL	PSPACE-compl.	in PTIME	in PTIME	undecidable
PSL	PSPACE-compl.	in PTIME	in PTIME	undecidable
SML	PSPACE-compl.	in PTIME	in PTIME	undecidable
FOL	PSPACE-compl.	PSPACE-compl.	in PTIME	undecidable

4.1 Open questions

The following interesting questions have remained unanswered:

- Under which conditions on the modification frame (generator) is the satisfiability problem of GML decidable? When F is a singleton set and $E = \emptyset$ we are working in standard modal logic, for which the satisfiability problem is decidable. When i models temporary unavailability of single edges, we already enter the realm of undecidability.
- A somewhat related question is that of the finite model property. Which modification frame generators do have this property?
- Which modification frame generators have bisimulation invariance? [Roh04, p62] shows that global sabotage allows us to distinguish between loops and paths of otherwise indistinguishable worlds, thus losing bisimulation invariance.

[4]With polynomial bound on the modification frame.
[4]In general.

4.2 Acknowledgements

The author would like to thank Prof. Dr. Johan van Benthem for his comments. He also appreciated the remarks by Drs. Steven de Rooij, correcting grammar, style and punctuation.

References

[BdRV01] Patrick Blackburn, Maarten de Rijke, and Yde Venema. *Modal Logic*. Cambridge University Press, 2001.

[BMS98] Alexandru Baltag, Lawrence S. Moss, and Slawomir Solecki. The logic of public announcements, common knowledge, and private suspicions. In *TARK '98: Proceedings of the 7th conference on Theoretical aspects of rationality and knowledge*, pages 43–56, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.

[Fok00] Wan Fokkink. *Introduction to Process Algebra*. Springer, 2000.

[Roh04] Philipp Rohde. *On Games and Logics over Dynamically Changing Structures*. PhD thesis, Rheinisch-Westfälischen Technischen Hochschule Aachen, 2004.

[vB05a] Johan van Benthem. An essay on sabotage and obstruction, essays in honor of Jörg Siekmann on the occasion of his 69th birthday. In D. Hutter, editor, *Mechanizing Mathematical Reasoning*, pages 268–276. Springer Verlag, 2005.

[vB05b] Johan van Benthem. Guards, bounds, and generalized semantics. *Journal of Logic, Language and Information*, 14(3):263–279, 2005.