Complete Game Logic with Sabotage

Noah Abou El Wafa
noah.abouelwafa@kit.edu
Karlsruhe Institute of Technology
Karlsruhe, Germany
Carnegie Mellon University
Pittsburgh, USA

André Platzer
platzer@kit.edu
Karlsruhe Institute of Technology
Karlsruhe, Germany
Carnegie Mellon University
Pittsburgh, USA

ABSTRACT

We introduce *Sabotage Game Logic* ($\mathsf{GL_s}$), a simple and natural extension of Parikh's Game Logic with a single additional primitive, which allows players to lay traps for the opponent to avoid. $\mathsf{GL_s}$ can be used to model infinite sabotage games, in which players can change the rules during game play. In contrast to Game Logic, which is strictly less expressive, $\mathsf{GL_s}$ is exactly as expressive as the modal μ-calculus. This reveals a close connection between the entangled nested recursion inherent in modal fixpoint logics and adversarial dynamic rule changes characteristic for sabotage games.

Additionally we present a natural Hilbert-style proof calculus for $\mathsf{GL_s}$ and prove completeness. The completeness of an extension of Parikh's calculus for Game Logic follows.

CCS CONCEPTS

• **Theory of computation → Logic**; **Proof theory**; **Modal and temporal logics**.

KEYWORDS

game logic, μ-calculus, proof theory, completeness, expressiveness, sabotage games

1 INTRODUCTION

Games such as the Ehrenfeucht-Fraïssé game are invaluable tools in the study of logic [19] and some deep results about logic can be proved with the help of games [1, 27]. Logic can even be given meaning via games in the form of game-theoretical semantics [28].

On the contrary, logical methods are frequently used to study games [15]. Logic and games meet most directly in logics specifically designed for the study of games, such as Game Logic (GL) due to Parikh [36], which allows reasoning about the existence of winning strategies in a game. This requires giving exact meaning to general games, a nontrivial task, except for games that are limited to a fixed number of rounds. Nested alternating least and greatest fixpoints can provide the correct denotational semantics for games, when they are used to reflect the alternating responsibilities of the respective players at their decision points in the dynamic games [36].

Fixpoints generally play an important role in logic, for example in modal fixpoint logics such as the modal μ-calculus (L_μ). L_μ is where logic, games and fixpoints begin to converge. In fact Game Logic can be expressed in the modal μ-calculus using alternating fixpoint formulas to directly capture the semantics of alternating game play. However this first encounter is imperfect. After 25 years it was shown that GL is in fact strictly less expressive than L_μ [8].

The purpose of this paper is to remedy this situation by unifying the three fundamental concepts of logic, games, and fixpoints in a small and natural extension of Game Logic, which is shown to be *equivalent* to the fixpoint logic L_μ and to have a complete proof calculus. This identification of fixpoints with games eliminates the difference between interactive game play and alternating fixpoints. The key insight behind this paper is that because Game Logic can already express sufficient adversarial dynamics to express the alternating fixpoints of L_μ and is merely lacking a suitable way of referring to fixpoints by their respective fixpoint variables, this can be alleviated in a parsimonious and purely game-theoretic way. This is done in *Sabotage Game Logic* ($\mathsf{GL_s}$), a new extension of GL. In sabotage game logic reference can be expressed, not through the unstructured use of fixpoint variables as is done in the modal μ-calculus, but by using a simple game operator $\sim a$ that *changes* the rules of subsequent game play. Playing the game $\sim a$ has the effect that the game a is reserved exclusively for the Angel player in the future. This can be used to change the rules of a game dynamically according to rules that are explicit in the original game. This simple and natural mechanism of imperative game play is expressively equivalent to the functional mechanism of unstructured nested named recursion with the fixpoint variables in the alternating fixpoints of L_μ. The role the sabotage $\sim a$ plays in establishing the equiexpressiveness reveals an interesting connection between games with sabotage and the nesting of fixpoints in the modal μ-calculus which have previously been studied separately.

Formulas of the modal μ-calculus are frequently easiest to understand through their corresponding validity or model-checking parity games [35]. This is complicated by the unstructured goto-like action a fixpoint variable induces. Sabotage game logic avoids this problem, as $\mathsf{GL_s}$ formulas describe two-player games built up from simple connectives and, instead of fixpoint variables, players only need to consider the previously committed acts of sabotage, making sabotage game logic a very intuitive logic with very high expressive power. By the equivalence of L_μ and $\mathsf{GL_s}$, many desirable properties of the modal μ-calculus, such as decidability and small model property, can be transferred to sabotage game logic for free. Moreover completeness of an axiomatization of $\mathsf{GL_s}$ can be obtained through the translation. This is in contrast to the original axiomatization for game logic, for which completeness is still open after four decades [30]. $\mathsf{GL_s}$ promises to be a useful tool for understanding GL. This is evidenced by the completeness of an extension of Parikh's axiomatization for GL obtained from the complete proof calculus for $\mathsf{GL_s}$. To the best of our knowledge this is the only complete proof calculus for game logic to date. The embedding from sabotage game logic to the modal μ-calculus also suggests the possibility that the same property can be expressed

significantly more concisely in sabotage game logic than in the modal μ-calculus.

In summary, the contributions of this paper are threefold. Firstly, $\mathsf{GL_s}$, a new minimal, natural, concise and intuitive extension of game logic well-suited to logically studying sabotage games, is introduced. Secondly, it is shown that $\mathsf{GL_s}$ is expressively exactly as powerful as the modal μ-calculus and, consequently, many desirable logical properties of L_μ transfer to $\mathsf{GL_s}$. Thirdly, a sound proof calculus for $\mathsf{GL_s}$ is presented, proved complete and completeness is transferred to obtain a complete extension of Parikh's GL calculus.

Outline. The required preliminaries are recalled in Section 2. Section 3 introduces sabotage game logic ($\mathsf{GL_s}$) and another extension recursive game logic (RGL) of game logic that will play an intermediary role in translating between the modal μ-calculus and sabotage game logic. In essence RGL adds completely recursive games to game logic by naively importing the notion of fixpoint variable reference from modal μ-calculus. Section 4 briefly recalls the definitions of some modal fixpoint logics such as the modal μ-calculus formally. In Section 5 the expressive power of fragments of recursive game logic is compared to modal fixpoint logics. Subsequently, proof calculi for sabotage game logic and the fragment right-linear game logic of recursive game logic are introduced and completeness for right-linear game logic is proved in Section 6. Finally in Section 7 equiexpressiveness of sabotage game logic and the modal μ-calculus is established and completeness of the proof calculus for $\mathsf{GL_s}$ is proved. Completeness of an extension of Parikh's calculus for GL is obtained as a corollary. All proofs are in Appendix D.

Related Work. Sabotage games have been considered to model algorithms under adversarial conditions and in learning [26, 44]. Previous work on using modal logic for the sabotage game using Sabotage Modal Logic (SML) [6, 44] differs from sabotage game logic. Unlike in $\mathsf{GL_s}$, where obstruction is modelled as changing the meaning of the game described syntactically, in SML sabotage is described as changing the structure of interpretation. The sabotage μ-calculus was investigated for modelling infinite sabotage games [41]. In contrast to $\mathsf{GL_s}$ of the model changing nature satisfiability problem for SML is undecidable and lacks the finite model property [32].

For examples of applications of Game Logic see [37, 38]. The relation of the games, game logic, fixpoints and modal μ-calculus has been considered in [8, 10, 17, 20, 25, 29, 35, 36, 42]. Equiexpressiveness and relative completeness of game logic and modal μ-calculus in the first-order case was shown in [45]. The modal μ-calculus and its relation to model checking is well-studied [11, 12, 21, 39]. Completeness for game logic was conjectured [36]. A completeness proof based on a cut-free complete calculus for L_μ [2] was suggested [23]. It was recently shown not to work [30].

2 PRELIMINARIES

Effectivity Function. An *effectivity function* [22] is a monotone function $w : \mathcal{P}(X) \to \mathcal{P}(X)$. It will be used for the denotational semantics of a game, where $w(Y)$ denotes the set of all states (winning region) from which a given player can win into the region $Y \subseteq X$. These are naturally monotone, as any point in the winning region for a set $A \subseteq B$ is also in the winning region for the larger goal B. Let $\mathcal{W}(X)$ be the set of such effectivity functions ordered by point-wise inclusion, i.e. $w \subseteq u$ if $w(A) \subseteq u(A)$ for all $A \subseteq X$.

Definition 2.1.
(1) Given a set $A \subseteq X$ the intersection effectivity function is $A_?(B) = A \cap B$ and the constant effectivity function $\overline{A}(B) = A$.
(2) For an effectivity function $w \in \mathcal{W}(X)$ its dual is $w^{\mathrm{d}}(A) = X \setminus w(X \setminus A)$.
(3) An effectivity function $w \in \mathcal{W}(X)$ is *relational* if there is a relation $R \subseteq X \times X$ such that
$$w(A) = R \circ A = \{r : \exists s \in A \;\; rRs\}$$
(4) For $w \in \mathcal{W}(X)$ let
$$\mu A.w(A) = \bigcap \{A \in \mathcal{P}(X) : w(A) \subseteq A\}$$
(5) For a function $\Delta : \mathcal{W}(X) \to \mathcal{W}(X)$ let
$$\mu w.\Delta(w) = \bigcap \{p \in \mathcal{W}(X) : \Delta(w) \subseteq w\}$$
be the least fixpoint as usual.

As usual $\mu A.w(A)$ and $\mu q.\Delta(q)$ are the least fixpoints of w and Δ respectively, provided Δ is monotone [4]. In the sequel it will be necessary to work with both fixpoints of monotone functions on a power set and fixpoints of monotone functions on the set of monotone functions on a power set. Under some conditions the latter can be viewed pointwise as the former.

Lemma 2.2. *Suppose $\Delta : \mathcal{W}(X) \to \mathcal{W}(X)$ is monotone and $\Delta(w)(A) = \Delta(\overline{w(A)})(A)$ for all $w \in \mathcal{W}(X)$ and all $A \in \mathcal{P}(X)$.*
(1) $(\mu u.\Delta(u))(A) = \mu B.(\Delta(\overline{B})(A))$
(2) $(\nu u.\Delta(u))(A) = \nu B.(\Delta(\overline{B})(A))$

See proof on page 16.

Neighbourhood and Kripke Structures. Both game logic and the modal μ-calculus can be interpreted over arbitrary coalgebraic structures [16]. While the modal μ-calculus is commonly interpreted over Kripke structures, game logic was originally interpreted over the more general class of neighbourhood models [34]. The results in this paper hold for both classes of models equally.

Definition 2.3. A *monotone neighbourhood structure* is a triple $\mathcal{N} = (|\mathcal{N}|, v_{\mathcal{N}}, \rho_{\mathcal{N}})$ consisting of a set $|\mathcal{N}|$ and functions
$$v_{\mathcal{N}} : \mathbb{A} \to \mathcal{P}(|\mathcal{N}|) \qquad \rho_{\mathcal{N}} : \mathbb{G} \to \mathcal{W}(|\mathcal{N}|)$$
assigning a valuation to atomic propositions from $\mathbb{G}$ and a effectivity function to atomic games from $\mathbb{G}$. The structure $\mathcal{N}$ is a *Kripke structure* if each $\rho_{\mathcal{N}}(a)$ is relational.

3 EXTENSIONS OF GAME LOGIC

Throughout the paper fix a countably infinite set $\mathbb{A}$ of propositional constants, a countably infinite set $\mathbb{V}$ of fixpoint variables and a countably infinite set $\mathbb{G}$ of atomic games.

3.1 Sabotage Game Logic

Sabotage game logic ($\mathsf{GL_s}$) is an extension of game logic defined by adding the atomic games $\sim a$. Formulas and games of $\mathsf{GL_s}$ are given

by the following grammar:

$$\varphi ::= P \mid \neg\varphi \mid \varphi \vee \psi \mid \langle\alpha\rangle\varphi$$
$$\alpha ::= a \mid {\sim}a \mid ?\varphi \mid \alpha \cup \beta \mid \alpha;\beta \mid \alpha^* \mid \alpha^{\mathrm{d}}$$

The formula $\langle\alpha\rangle\varphi$ expresses that player Angel has a winning strategy in the game α to reach one of the states in which φ is true. The test game $?\varphi$ is lost prematurely by Angel unless the formula φ is true in the current state. The choice game $\alpha \cup \beta$ allows Angel to choose between playing α or β. To play the sequential game $\alpha;\beta$ is to play β after α unless one of the players has already lost while playing α. The repetition game α^* allows Angel to decide after each round of α whether she wants to stop playing or repeat α unless one of the two players has already lost. The dual operator $^{\mathrm{d}}$ switches the roles of the players. Any decision or test within the scope of a dual operator is not taken by Angel, but her opponent Demon.

The additional primitive ${\sim}a$ is the trap-setting game. When ${\sim}a$ is played, Angel lays a trap at a. It has the effect that should Demon, anytime in the subsequent play, try to play a (by reaching a^{d}), he falls into the trap and loses the game prematurely. The trap stays in effect throughout the formula until it is turned into a trap for Demon by playing ${\sim}a^{\mathrm{d}}$ according to the rules of the game. However once a trap has been set for an atomic game it can only change hands, but will not be played normally again.

Viewed differently, if a player plays ${\sim}a$ the player claims the atomic game a for herself. The opponent is not allowed to play it and forfeits the game by trying. Formally ${\sim}a$ means that the game a belongs to Angel until the next time the game ${\sim}a^{\mathrm{d}}$ is played. Playing ${\sim}a^{\mathrm{d}}$ dually means the game a belongs to Demon until it returns to Angel. The effect of the claim is that the rules for playing a and a^{d} in the future *change* as in Table 1.

Table 1: Effect of Rule Changes

Owner of a	Rules for a	Rules for a^{d}
Neither $\emptyset$	a played normally	a^{d} played normally
Angel $\diamond$	a skipped	Angel wins a^{d}
Demon $\square$	Demon wins a	a^{d} skipped

Abbreviations and Conventions. As in game logic define the dual game connectives for demons choice, test, or repetition. That is let $\alpha \cap \beta$ abbreviate $(\alpha^{\mathrm{d}} \cup \beta^{\mathrm{d}})^{\mathrm{d}}$. This leaves the choice of whether to play α or β to Angel's opponent. Analogously let $!\varphi$ stand for the tests Demon needs to pass $(?\varphi)^{\mathrm{d}}$ and $\alpha^\times$ for demonic repetition $(\alpha^{\mathrm{d}*})^{\mathrm{d}}$. The propositional connectives $\wedge, \rightarrow, \leftrightarrow$ and $\top, \bot$ are defined as usual. Sequential composition ; binds stronger than choice $\cup, \cap$ and conjunction and disjunction bind stronger than implication.

Infinite Plays. A subtlety in Game Logic is that it is possible for the two players to play indefinitely. For example Angel could potentially choose to repeat a $(?\top)^*;?\bot$ game indefinitely, instead of choosing to stop the repetition and thereby losing prematurely due to the rules of the ? game. This is desirable as it allows modelling infinite games. Intuitively the semantics will be defined so that the player who causes the game to be infinite (by repeating a subgame infinitely often that is not contained in another subgame that is repeated infinitely often) loses.

Game Logic. Syntactically Parikh's Game logic (GL) [36] is the fragment of $\mathsf{GL_s}$ without games of the form ${\sim}a$ and ${\sim}a^{\mathrm{d}}$. The semantics of sabotage game logic will be defined so that they agree with the usual semantics. Hence sabotage game logic is a genuine extension of game logic.

Examples. To illustrate the role of the trap-setting game consider the following two games

$$({\sim}a \cap {\sim}a^{\mathrm{d}});a \qquad ({\sim}a \cup {\sim}a^{\mathrm{d}});a;!\bot$$

Demon has a winning strategy in the first game. He simply chooses to play the game ${\sim}a^{\mathrm{d}}$ first, thereby laying a trap for Angel at a. Subsequently Angel will need to try to play a. However the trap Demon set causes her to lose at this point. In the second game in contrast, Angel has a winning strategy. If Angel chooses first to play ${\sim}a$, she sets a trap for Demon at a. This also means that Angel can skip the following game of a and go straight to playing $!\bot$. At this point Demon loses the game, since he can not pass the test $\bot$.

Example: The Poison Game in $\mathsf{GL_s}$. The poison game is a graph game introduced [18] to capture the notion of a perfect kernel. To begin the game Angel picks a vertex on the graph. The players alternate to choose adjacent vertices, tracing a path through the graph. However Demon poisons every vertex he chooses to move to. Angel loses if she chooses to go to a poisoned vertex, but Demon is immune to the poison he leaves behind. It is Angel's objective to survive and Demon's to poison Angel. The value of the game lies in the fact that Angel has a winning strategy in the poison game played on a progressively and outwardly finite directed graph iff the graph has a local kernel [18].

Various approaches to capture the poison game in modal logic have been suggested [6, 9]. However the previously considered logics have undecidable satisfiability problems. In $\mathsf{GL_s}$, instead of viewing the poisoning as a model change, which is difficult to capture by a well-behaved logic [3], poisoning can be understood as a describable rule change.

Formally we list the vertices of the graph as $a_1, \ldots, a_n$ and view them simultaneously as atomic games to be interpreted as moving to the vertex of the same name. Now consider the games $\alpha_\diamond \equiv a_1 \cup \ldots \cup a_n$ and $\alpha_\square \equiv ((a_1;{\sim}a_1) \cup \ldots \cup (a_n;{\sim}a_n))^{\mathrm{d}}$. In $\alpha_\diamond$ Angel chooses which vertex to go to next. (The semantics enforce that she loses should she try to go to a non-adjacent vertex.) Analogously $\alpha_\square$ allows Demon to make a move, which is followed by him poisoning the vertex he goes to. The two players now play the game

$$\varphi \equiv \langle(\alpha_\square;\alpha_\diamond)^\times\rangle\top.$$

interpreted over an n-element graph (W, E) viewed as a Kripke frame $\mathcal{K}$ with states $|\mathcal{K}| = W$ and the reachability relations

$$xR_{w_i}y \quad \text{if} \quad (x,y) \in E \text{ and } y = w_i.$$

In words, the formula $\langle a_i\rangle\top$ holds in exactly those states from which vertex w_i is reachable along an E-edge. The formula φ interpreted in this way describes the poison game, as φ is satisfiable in this structure (so there is a state from which Angel can win the

game $(\alpha_{\Box};\alpha_{\diamond})^{\times}$) iff Angel has a winning strategy in the original poison game.

3.2 Recursive Game Logic

We introduce a second extension, *recursive game logic* (RGL), of game logic that allows arbitrary recursive games. This large increase in expressive power allows RGL to serve as the essential technical intermediary connecting $\mathsf{GL_s}$ to the modal μ-calculus. Recursive Game Logic (RGL) is defined by the grammar

$$\varphi ::= P \mid \neg\varphi \mid \varphi \vee \psi \mid \langle\alpha\rangle\varphi$$
$$\alpha ::= a \mid x \mid \alpha^{\mathrm{d}} \mid ?\varphi \mid \alpha \cup \beta \mid \alpha;\beta \mid \mathrm{r}x.\alpha$$

where $P \in \mathbb{A}, a \in \mathbb{G}, x \in \mathbb{V}$. The additional restriction is that variables x are not free in games $?\varphi$ and can only be bound by $\mathrm{r}x.\alpha$ if x appears only in the scope of an even number of $^{\mathrm{d}}$ in α. Syntactically the only difference between recursive game logic and game logic is that repetition games α^* have been replaced by named subgames of the form $\mathrm{r}x.\alpha$ and games x to recursively call these named subgames have been added. The ordinary repetition game α^* can be viewed as an abbreviation for $\mathrm{r}x.(\alpha;x \cup ?\top)$

Intuitively a recursive game $\mathrm{r}x.\alpha$ is played just like α until the variable x is encountered. In this case the game is interrupted and the players will begin another subgame of $\mathrm{r}x.\alpha$. At some stage the players may finish playing this subgame. They then continue in the state they reached to play the original game that was interrupted previously.

The abbreviations for the usual propositional symbols and the demonic connectives are defined just as in $\mathsf{GL_s}$. Additionally the dual version of a named subgame is defined as $\imath x.\alpha$ to mean $(\imath x.\alpha^{\mathrm{d}})^{\mathrm{d}}$. This game is played similarly to $\mathrm{r}x.\varphi$. The only difference is which of the players is held responsible if the game is played infinitely long. If the largest subgame repeated infinitely often during a play in a game of the form $\mathrm{r}x.\alpha$, then Angel loses the game. If the largest such game is of the form $\imath x.\alpha$, Demon loses.

3.2.1 Examples. An example of a fully recursive game is $\mathrm{r}x.(?\top \cup a;x;b)$. Angel can win this game relative to the winning condition that some formula φ holds in the final state exactly if there is some n such that after n rounds of playing a she can win a game of n consecutive rounds of b into φ. This game can not be described in game logic, which lacks facilities to retain the number of games b that still have to be played after Angel chooses the left side in her choice once.

3.3 Semantics of Game Logics

A denotational semantics for recursive game logic and sabotage game logic can be defined in a simple and compositional way. Superficially both semantics are quite different from the usual semantics of game logic. However it will be shown that for GL formulas the semantics of RGL and $\mathsf{GL_s}$ agree with the original semantics of GL.

3.3.1 Semantics of Recursive Game Logic. Because recursive game logic contains games of the form $\mathrm{r}x.a;(x \cup c);b$ unlike in game logic the semantics of such a game can no longer be defined as the fixpoint of a function between power sets, as the play of c that will take place after Angel chooses to play b for the first time must be taken into account.

Formally the semantics of recursive game logic with respect to both a monotone neighbourhood structure and a valuation. A *valuation* is a function $I : \mathbb{V} \to \mathcal{W}(|\mathcal{N}|)$ assigning an interpretation to every variable x. Given a valuation I a variable $x \in \mathbb{V}$ and a $w \in \mathcal{W}(|\mathcal{N}|)$ let $I[x \mapsto w]$ denote the valuation that agrees with I, except that $I[x \mapsto w](x) = w$.

Definition 3.1. For any monotone neighbourhood structure $\mathcal{N}$ and any valuation I define the *semantics* $\mathcal{N}[\![\varphi]\!]^I \in \mathcal{P}(|\mathcal{N}|)$ and $\mathcal{N}[\![\alpha]\!]^I \in \mathcal{W}(|\mathcal{N}|)$ by mutual induction on RGL formulas φ and RGL games α:

$$\mathcal{N}[\![P]\!]^I = v_{\mathcal{N}}(P) \qquad \mathcal{N}[\![\neg\varphi]\!]^I = |\mathcal{N}| \setminus \mathcal{N}[\![\varphi]\!]^I$$
$$\mathcal{N}[\![\varphi \vee \psi]\!]^I = \mathcal{N}[\![\varphi]\!]^I \cup \mathcal{N}[\![\psi]\!]^I \qquad \mathcal{N}[\![\langle\alpha\rangle\varphi]\!]^I = \mathcal{N}[\![\alpha]\!]^I(\mathcal{N}[\![\varphi]\!]^I)$$
$$\mathcal{N}[\![a]\!]^I = \rho_{\mathcal{N}}(a) \qquad \mathcal{N}[\![x]\!]^I = I(x)$$
$$\mathcal{N}[\![?\varphi]\!]^I(A) = \mathcal{N}[\![\varphi]\!]^I \cap A \qquad \mathcal{N}[\![\alpha;\beta]\!]^I = \mathcal{N}[\![\alpha]\!]^I \circ \mathcal{N}[\![\beta]\!]^I$$
$$\mathcal{N}[\![\mathrm{r}x.\alpha]\!]^I = \mu u.\mathcal{N}[\![\alpha]\!]^{I[x\mapsto u]} \qquad \mathcal{N}[\![\alpha \cup \beta]\!]^I = \mathcal{N}[\![\alpha]\!]^I \cup \mathcal{N}[\![\beta]\!]^I$$
$$\mathcal{N}[\![\alpha^{\mathrm{d}}]\!]^I = (\mathcal{N}[\![\alpha]\!]^I)^{\mathrm{d}}$$

For closed formulas the superscript I is dropped. As usual the notation $\mathcal{N} \vDash \varphi$ means that $\mathcal{N}[\![\varphi]\!]^I = |\mathcal{N}|$ for all valuations I. Moreover write $\vDash \varphi$ if $\mathcal{N} \vDash \varphi$ for all *monotone neighbourhood structures* $\mathcal{N}$, and $\vDash_K \varphi$ if $\mathcal{N} \vDash \varphi$ for all *Kripke structures* $\mathcal{N}$.

The semantics of named subgames are well-defined and the meaning of games $\mathrm{r}x.\alpha$ can be seen to be extremal fixpoints by monotonicity of the function $u \mapsto \mathcal{N}[\![\alpha]\!]^{I[x\mapsto u]}$. The proof is postponed to Lemma 3.6.

3.3.2 Semantics of Sabotage Game Logic. The semantics of a game of sabotage game logic depends on the traps that players have already laid in the run of the game so far. To keep track of these, games and formulas of sabotage game logic must be evaluated in a context. A *context* is a function $c : \mathbb{G} \to \{\emptyset, \diamond, \Box\}$ indicating which player has previously laid a trap (see Table 1). All contexts are assumed to have finite support, that is $c(a) = \emptyset$ for all but finitely many a. Let C be the set of all contexts and let $c_\emptyset(a) = \emptyset$ for all a be the constant context without any traps laid. For any set $U \subseteq |\mathcal{N}| \times C$ and any context $c \in C$ let $U\restriction c = \{\omega : (\omega, c) \in U\}$ be the projection on $|\mathcal{N}|$.

To interpret an atomic game a it is necessary to consider the context in which it is played. If one of the players has already laid a trap the normal rules no longer apply. Formally any effectivity function $w : \mathbb{G} \to \mathcal{W}(|\mathcal{N}|)$ is *lifted* to $\widehat{w}(a) \in \mathcal{W}(|\mathcal{N}| \times C)$ by defining $(\omega, c) \in \widehat{w}(a)(U)$ iff

(1) $c(a) = \emptyset$ and $\omega \in w(a)(U\restriction c)$ or
(2) $c(a) = \diamond$ and $(\omega, c) \in U$

Hence Angel can win the game a from a position ω in context c, into the set U if a is not claimed (i.e. $c(a) = \emptyset$) and additionally she can win a game of a played according to the usual rules into $U\restriction c$. If a belongs to Angel (i.e. $c(a) = \diamond$), she can also win if the current state ω and context c are already in U. However if a belongs to Demon (i.e. $c(a) = \Box$), Angel has already lost. This formalizes the effects of the rule changes as described in Table 1. For any context c write $\overline{c}$ for the dual context where angelic traps become demonic

traps and vice versa:

$$\overline{c}(a) = \begin{cases} \emptyset & \text{if } c(a) = \emptyset \\ \square & \text{if } c(a) = \diamond \\ \diamond & \text{if } c(a) = \square \end{cases}$$

For a set $A \subseteq W \times C$ write $A^{\mathsf{C}} = \{(\omega, c) : (\omega, \overline{c}) \notin A\}$. For a function $w \in \mathcal{W}(W \times C)$ define the *sabotage dual*

$$w^{\mathsf{D}}(A) = w(A^{\mathsf{C}})^{\mathsf{C}}.$$

The sabotage dual extends the notion of the ordinary dual to sabotage games. In particular $(\omega, c) \in \widehat{w}(a)^{\mathsf{D}}(U)$ iff

(1) $c(a) = \emptyset$ and $\omega \in w(a)^{\mathsf{d}}(\{v : (v, c) \in U\})$ or
(2) $c(a) = \diamond$ or
(3) $c(a) = \square$ and $(\omega, c) \in U$

The semantics of formulas and games of recursive game logic with respect to a monotone neighbourhood structure is defined by mutual induction.

Definition 3.2. For $\mathsf{GL_s}$ formulas and any monotone neighbourhood structure $\mathcal{N}$ the *semantics* is defined as a set $\mathcal{N}[\![\varphi]\!]_{\mathsf{s}} \in \mathcal{P}(|\mathcal{N}| \times C)$ as follows

$$\mathcal{N}[\![P]\!]_{\mathsf{s}} = v_{\mathcal{N}}(P) \times C \qquad \mathcal{N}[\![\langle\alpha\rangle\varphi]\!]_{\mathsf{s}} = \mathcal{N}[\![\alpha]\!]_{\mathsf{s}}(\mathcal{N}[\![\varphi]\!]_{\mathsf{s}})$$
$$\mathcal{N}[\![\neg\varphi]\!]_{\mathsf{s}} = \mathcal{N}[\![\varphi]\!]_{\mathsf{s}}{}^{\mathsf{C}} \qquad \mathcal{N}[\![\varphi \vee \psi]\!]_{\mathsf{s}} = \mathcal{N}[\![\varphi]\!]_{\mathsf{s}} \cup \mathcal{N}[\![\psi]\!]_{\mathsf{s}}$$

The *semantics* of games α is defined as an effectivity function $\mathcal{N}[\![\alpha]\!]_{\mathsf{s}} \in \mathcal{W}(|\mathcal{N}| \times C)$ by

$$\begin{aligned} \mathcal{N}[\![a]\!]_{\mathsf{s}} &= \widehat{\rho_{\mathcal{N}}(a)} \\ \mathcal{N}[\![\sim a]\!]_{\mathsf{s}}(A) &= \{(\omega, c) : (\omega, c\tfrac{\diamond}{a}) \in A\} \\ \mathcal{N}[\![?\varphi]\!]_{\mathsf{s}}(A) &= \mathcal{N}[\![\varphi]\!]_{\mathsf{s}} \cap A \\ \mathcal{N}[\![\alpha \cup \beta]\!]_{\mathsf{s}} &= \mathcal{N}[\![\alpha]\!]_{\mathsf{s}} \cup \mathcal{N}[\![\beta]\!]_{\mathsf{s}} \\ \mathcal{N}[\![\alpha^*]\!]_{\mathsf{s}}(A) &= \mu B.A \cup \mathcal{N}[\![\alpha]\!]_{\mathsf{s}}(B) \\ \mathcal{N}[\![\alpha;\beta]\!]_{\mathsf{s}} &= \mathcal{N}[\![\alpha]\!]_{\mathsf{s}} \circ \mathcal{N}[\![\beta]\!]_{\mathsf{s}} \\ \mathcal{N}[\![\alpha^{\mathsf{d}}]\!]_{\mathsf{s}} &= \mathcal{N}[\![\alpha]\!]_{\mathsf{s}}{}^{\mathsf{D}} \end{aligned}$$

The semantics of $\sim a$ illustrates the role of the context. Playing the game $\sim a$ changes the context and assigns $\diamond$ the game a to keep track of the trap Angel has laid there.

Unlike for sabotage modal logic [5] the semantics is not defined in terms of a changing model. Instead the state space is enlarged to contain the states of the structure and independently keep track of the acts of model change or, here, traps laid. The definition is similar in spirit to the modified semantics for the sabotage μ-calculus [5]. Unlike in the definition of the modal μ-calculus augmented with sabotage [41] the traps set will persist throughout multiple repetitions of a game α^* instead of being reset without cause.

3.3.3 Dual Normal Form. For some proofs it is important that negation is only applied to propositional atoms and the duality operator is only applied to atomic games and free variables. Formulas and games that satisfy this condition are said to be in *normal form*.

Definition 3.3. By mutual recursion on $\mathsf{GL_s}$ formulas and games define the *syntactic complement* $\overline{\varphi}$ of a sabotage game logic formula and the *syntactic dual* α^{d} of a sabotage game logic game as follows:

$$\overline{P} = \neg P \qquad \overline{\neg P} = P$$
$$\overline{\varphi \wedge \psi} = \overline{\varphi} \vee \overline{\psi} \qquad \overline{\varphi \vee \psi} = \overline{\varphi} \wedge \overline{\psi}$$
$$\overline{\langle\alpha\rangle\varphi} = \langle\alpha^{\mathsf{d}}\rangle\overline{\varphi} \qquad (\alpha;\beta)^{\mathsf{d}} = \alpha^{\mathsf{d}};\beta^{\mathsf{d}}$$
$$(a)^{\mathsf{d}} = a^{\mathsf{d}} \qquad (a^{\mathsf{d}})^{\mathsf{d}} = a$$
$$(\sim a)^{\mathsf{d}} = \sim a^{\mathsf{d}} \qquad (\sim a^{\mathsf{d}})^{\mathsf{d}} = \sim a$$
$$(?\varphi)^{\mathsf{d}} = !\varphi \qquad (!\varphi)^{\mathsf{d}} = ?\varphi$$
$$(\alpha \cup \beta)^{\mathsf{d}} = \alpha^{\mathsf{d}} \cap \beta^{\mathsf{d}} \qquad (\alpha \cap \beta)^{\mathsf{d}} = \alpha^{\mathsf{d}} \cup \beta^{\mathsf{d}}$$
$$(\alpha^*)^{\mathsf{d}} = (\alpha^{\mathsf{d}})^{\times} \qquad (\alpha^{\times})^{\mathsf{d}} = (\alpha^{\mathsf{d}})^*$$

The syntactic complement and dual semantically correspond to set complements and dual functions:

Lemma 3.4 (Duality). $\mathcal{N}[\![\overline{\varphi}]\!]_{\mathsf{s}} = \mathcal{N}[\![\varphi]\!]_{\mathsf{s}}{}^{\mathsf{C}}$ *for any* $\mathsf{GL_s}$ *formula* φ *and* $\mathcal{N}[\![\alpha^{\mathsf{d}}]\!]_{\mathsf{s}} = \mathcal{N}[\![\alpha]\!]_{\mathsf{s}}{}^{\mathsf{D}}$ *for any game* α.

See proof on page 16.

By inductively replacing negations in $\neg\varphi$ by $\overline{\varphi}$ and duals α^{d} by α^{d} any $\mathsf{GL_s}$ formula and any game can be transformed into an equivalent formula or game in normal form.

Definition 3.3 can be easily modified for formulas of recursive game logic by defining

$$(x)^{\mathsf{d}} = x \quad (x^{\mathsf{d}})^{\mathsf{d}} = x \quad (\mathsf{r}x.\alpha)^{\mathsf{d}} = \mathsf{l}x.(\alpha)^{\mathsf{d}} \quad (\mathsf{l}x.\alpha)^{\mathsf{d}} = \mathsf{r}x.(\alpha)^{\mathsf{d}}$$

Again the syntactic complement and dual semantically correspond to set complements and dual functions.

Lemma 3.5. $\mathcal{N}[\![\overline{\varphi}]\!]^I = |\mathcal{N}| \setminus \mathcal{N}[\![\varphi]\!]^{I^{\mathsf{d}}}$ *for any* RGL *formula* φ *and* $\mathcal{N}[\![\alpha^{\mathsf{d}}]\!]^I = (\mathcal{N}[\![\alpha]\!]^{I^{\mathsf{d}}})^{\mathsf{d}}$ *for any* RGL *game* α.

See proof on page 16.

As was the case for $\mathsf{GL_s}$, through inductively replacing negation $\neg\varphi$ and duality α^{d} by their syntactic versions $\overline{\varphi}$ and $(\alpha)^{\mathsf{d}}$ every RGL formula and every RGL game can be turned into an equivalent formula or game in normal form respectively, since bound variables appear only within the scope of an even number of $^{\mathsf{d}}$ operators.

In the sequel we assume all formulas and games of $\mathsf{GL_s}$ *and* RGL *are in* ***normal form.*** Because every formula is equivalent to one in normal form this does not restrict the generality.

3.3.4 Semantic Compatibility. Using the normal form, observe that the semantics of named games are indeed extremal fixpoint. This relies on the monotonicity:

Lemma 3.6. *If* $\mathsf{r}x.\alpha$ *is a game of recursive game logic, then* $F : q \mapsto \mathcal{N}[\![\alpha]\!]^{I[x\mapsto q]}$ *is monotone.*

Proof. The game α is equivalent to a formula in normal form, in which x^{d} does not appear, since it must be in the scope of an even number of $^{\mathsf{d}}$ operators in α. Monotonicity of F follows by induction on such a α. □

Unlike for game logic the semantics of repetition games above was not defined as the fixpoint of a set operator. However Definition 3.1 agrees with the definition of the standard semantics for game logic for all GL games.

Lemma 3.7. *If* α *is a* GL *game then*

$$\mathcal{N}[\![\alpha^*]\!]^I(A) = \mu B.A \cup \mathcal{N}[\![\alpha]\!]^I(B)$$

For a proof see Lemma B.2 in Appendix B.

Although Definitions 3.1 and 3.2 are quite different the semantics of recursive game logic and of sabotage game logic agree on their common fragment, GL.

Proposition 3.8. *If φ is a formula and α a game of* GL *then*

$$\mathcal{N}[\![\varphi]\!] = \mathcal{N}[\![\varphi]\!]_s \restriction c_\emptyset \qquad \mathcal{N}[\![\alpha]\!](U \restriction c_\emptyset) = \mathcal{N}[\![\alpha]\!]_s(U) \restriction c_\emptyset.$$

Proof. Use Lemma 3.7 for the case of fixpoints in a mutual induction on formulas and games of game logic. □

Write $\mathcal{N} \vDash \varphi$ to abbreviate $\mathcal{N}[\![\varphi]\!]_s \supseteq |\mathcal{N}| \times \{c_\emptyset\}$. This captures the intended semantics of φ as being evaluated when no traps have been set initially, by requiring the formula to hold in every state in the special context $c_\emptyset$ in which no game has been claimed. Moreover write $\vDash \varphi$ if $\mathcal{N} \vDash \varphi$ for all *monotone neighbourhood structures* $\mathcal{N}$ and $\vDash_K \varphi$ if $\mathcal{N} \vDash \varphi$ for all *Kripke structures* $\mathcal{N}$. This overloading of notation for game logic formulas is justified by Proposition 3.8.

4 MODAL FIXPOINT LOGICS

The Modal μ-Calculus. This section recalls two modal fixpoint logics. Of particular interest is the *modal μ-calculus* (L_μ) [11], because of its desirable logical properties. It has decidable satisfiability and model checking problems, the finite model property and comes with a natural complete proof calculus. The syntax of L_μ is given by the following grammar:

$$\varphi ::= P \mid \neg P \mid x \mid \langle a \rangle \varphi \mid [a]\varphi \mid \varphi \vee \psi \mid \varphi \wedge \psi \mid \mu x.\varphi \mid \nu x.\varphi$$

for $P \in \mathbb{A}, a \in \mathbb{G}, x \in \mathbb{V}$. The modal μ-calculus extends basic (multi)-modal logic with fixpoint operators $\mu x.\varphi$ and $\nu x.\varphi$. These denote the least and greatest fixpoints of φ in the sense that $\mu x.\varphi(x)$ is equivalent to $\varphi(\mu x.\varphi)$. The syntax enforces that fixpoint variables x can appear only positively in order to ensure that the semantics of fixpoint operators $\mu x.\varphi$ denote the desired extremal fixpoints.

Fixpoint Logic with Chop. An interesting extension of the modal μ-calculus is fixpoint logic with chop [33]. Although it lacks some of the nice properties of modal μ-calculus, its high expressiveness is useful to establish a close correspondence with the game logics from the previous section via a natural translation. The syntax of fixpoint logic with chop (FLC) [33] is given by the following grammar:

$$\varphi ::= \mathrm{id} \mid P \mid \neg P \mid x \mid \langle a \rangle \varphi \mid [a]\varphi \mid \varphi \vee \psi \mid \varphi \wedge \psi \mid \varphi \circ \psi \mid \mu x.\varphi \mid \nu x.\varphi$$

for $P \in \mathbb{A}, a \in \mathbb{G}, x \in \mathbb{V}$. Fixpoint logic with chop is conceptually close to the modal μ-calculus. However fixpoint variables do not range over predicates (elements of $\mathcal{P}(|\mathcal{N}|)$) anymore, but over predicate transformers (monotone functions in $\mathcal{W}(|\mathcal{N}|)$) instead. Consequently formulas denote predicate transformers which admit a natural notion of concatenation $\circ$ and identity transformation id. As in the modal μ-calculus the definition syntactically restricts to positive appearances of x, to ensure the well-definedness of the semantics of the fixpoint operator.

Semantics of Fixpoint Logic with Chop. The *semantics* of fixpoint logic with chop is defined with respect to monotone neighbourhood structures and a valuation $I : \mathbb{V} \to \mathcal{W}(|\mathcal{N}|)$. By structural induction on formulas φ define the set $\mathcal{N}[\![\varphi]\!]^I \in \mathcal{W}(|\mathcal{N}|)$

$$\mathcal{N}[\![\mathrm{id}]\!]^I = \mathrm{id} \qquad \mathcal{N}[\![x]\!]^I = I(x)$$
$$\mathcal{N}[\![\neg P]\!]^I = |\mathcal{N}| \setminus v_{\mathcal{N}}(P) \qquad \mathcal{N}[\![P]\!]^I = v_{\mathcal{N}}(P)$$
$$\mathcal{N}[\![\varphi \vee \psi]\!]^I = \mathcal{N}[\![\varphi]\!]^I \cup \mathcal{N}[\![\psi]\!]^I \qquad \mathcal{N}[\![\varphi \wedge \psi]\!]^I = \mathcal{N}[\![\varphi]\!]^I \cap \mathcal{N}[\![\psi]\!]^I$$
$$\mathcal{N}[\![\langle a \rangle \varphi]\!]^I = \rho_{\mathcal{N}}(a) \circ \mathcal{N}[\![\varphi]\!]^I \qquad \mathcal{N}[\![[a]\varphi]\!]^I = \rho_{\mathcal{N}}(a)^{\mathrm{d}} \circ \mathcal{N}[\![\varphi]\!]^I$$
$$\mathcal{N}[\![\mu x.\varphi]\!]^I = \mu q.\mathcal{N}[\![\varphi]\!]^{I[x \mapsto q]} \qquad \mathcal{N}[\![\nu x.\varphi]\!]^I = \nu q.\mathcal{N}[\![\varphi]\!]^{I[x \mapsto q]}$$
$$\mathcal{N}[\![\varphi \circ \psi]\!]^I = \mathcal{N}[\![\varphi]\!]^I \circ \mathcal{N}[\![\psi]\!]^I$$

The semantics of μ and ν formulas denotes extremal fixpoints, since the semantics are monotone:

Lemma 4.1. *The function $F : q \mapsto \mathcal{N}[\![\varphi]\!]^{I[x \mapsto q]}$ is monotone.*

Proof. Monotonicity holds, because by definition of the syntax $\neg x$ can not appear in a formula. □

The semantics of a formula of fixpoint logic with chop is defined as a *monotone* function. To assign a truth value the function can be evaluated at $\emptyset$. Write $\mathcal{N} \vDash \varphi$ if $\mathcal{N}[\![\varphi]\!]^I(\emptyset) = |\mathcal{N}|$ for all I. By monotonicity of the semantics this ensures that $\mathcal{N} \vDash \varphi$ iff $\mathcal{N}[\![\varphi]\!]^I(U) = |\mathcal{N}|$ for all I and all $U \subseteq |\mathcal{N}|$. Moreover write $\vDash \varphi$ if $\mathcal{N} \vDash \varphi$ for all *monotone neighbourhood structures* $\mathcal{N}$ and $\vDash_K \varphi$ if $\mathcal{N} \vDash \varphi$ for all *Kripke structures* $\mathcal{N}$.

The semantics of L_μ formulas with respect to the FLC semantics coincide with the usual semantics of modal μ-calculus. (See Lemma B.1 in Appendix B.)

Negation in Fixpoint Logic with Chop. The negation of a formula of fixpoint logic with chop is defined syntactically as usual:

$$\overline{P} = \neg P \qquad \overline{\langle a \rangle \varphi} = [a]\overline{\varphi} \qquad \overline{\varphi \vee \psi} = \overline{\varphi} \wedge \overline{\psi} \qquad \overline{\mu x.\varphi} = \nu x.\overline{\varphi}$$
$$\overline{\neg P} = P \qquad \overline{[a]\varphi} = \langle a \rangle \overline{\varphi} \qquad \overline{\varphi \wedge \psi} = \overline{\varphi} \vee \overline{\psi} \qquad \overline{\nu x.\varphi} = \mu x.\overline{\varphi}$$
$$\overline{x} = x$$

The syntactic definition of negation corresponds semantically to complementation:

Lemma 4.2. $\mathcal{N}[\![\overline{\varphi}]\!]^I(\emptyset) = |\mathcal{N}| \setminus \mathcal{N}[\![\varphi]\!]^{I^c}(\emptyset)$ *for all L_μ formulas φ, where I^c is the pointwise complement of I.*

Proof. By a straightforward induction on formulas. □

The Modal $*$-Calculus. Restricting the fixpoints in FLC to structured ones as they appear in game logic yields a logic we call the modal $*$-calculus, which is the exact modal fixpoint logic equivalent of game logic. The syntax of *modal $*$-calculus* (L_*) is defined as

$$\varphi ::= \mathrm{id} \mid P \mid \neg\varphi \mid \varphi \vee \psi \mid \langle a \rangle \varphi \mid \varphi \circ \psi \mid \varphi^*$$

This can be viewed as a fragment of FLC by interpreting $\neg\varphi$ as $\overline{\varphi}$ and φ^* as an abbreviation for $\mu x.\mathrm{id} \vee \varphi \circ x$ where x is some fresh variable. Disjunctions $\varphi \vee \psi$ do not strictly need to be added as primitives, since they are definable in L_* as $(\varphi \circ \bot)^* \circ \psi$.

5 EXPRESSIVENESS

The semantics of game logic and modal μ-calculus are in many ways similar and game logic can express large parts of the modal μ-calculus. In particular it spans the entire fixpoint alternation hierarchy of the modal μ-calculus [7]. Nevertheless, game logic is less expressive than modal μ-calculus [8]. This section introduces natural translations to show that at the level of fixpoint logic with chop and recursive game logic, modal fixpoint logics and game logics can be identified completely. As a consequence the exact modal fixpoint logic corresponding to game logic and the fragment of recursive game logic corresponding to modal μ-calculus can be determined.

A formula φ of RGL is *well-named* if it does not bind the same variable twice and no variable appears both free and bound. Every formula is equivalent to a well-named formula by bound renaming.

5.1 Equiexpressiveness of FLC and RGL

For formulas φ, ψ of fixpoint logic with chop denote by $\varphi\frac{\psi}{x}$ the formula obtained from φ by syntactically replacing all free occurrences of x in φ by ψ. The same notation is used for syntactic substitution in formulas and games of game logics.

5.1.1 Translation from fixpoint logic with chop to recursive game logic. To express any formula φ of FLC equivalently in recursive game logic a translation $\varphi^\#$ of any FLC formula φ in the form of a RGL game is defined inductively:

$$(\mathrm{id})^\# = ?\top \qquad (x)^\# = x$$
$$(P)^\# = ?P;!\bot \qquad (\neg P)^\# = ?\neg P;!\bot$$
$$(\varphi \vee \psi)^\# = \varphi^\# \cup \psi^\# \qquad (\varphi \wedge \psi)^\# = \varphi^\# \cap \psi^\#$$
$$(\langle a\rangle\varphi)^\# = a;\varphi^\# \qquad ([a]\varphi)^\# = a^\mathrm{d};\varphi^\#$$
$$(\mu x.\varphi)^\# = \mathsf{r}x.\varphi^\# \qquad (\nu x.\varphi)^\# = \mathsf{l}x.\varphi^\#$$
$$(\varphi \circ \psi)^\# = \varphi^\#;\psi^\#$$

The RGL formula corresponding to φ is $\varphi^\sharp \equiv \langle\varphi^\#\rangle\bot$.

Proposition 5.1 (Correct $\#$). *For any* FLC *formula φ the translation satisfies $\mathcal{N}[\![\varphi]\!]^I = \mathcal{N}[\![\varphi^\#]\!]^I$. Hence $\mathcal{N}[\![\varphi]\!]^I(\emptyset) = \mathcal{N}[\![\varphi^\sharp]\!]^I$.*

Proof. By structural induction on φ. □

5.1.2 Translation from recursive game logic to fixpoint logic with chop. Conversely any formula of recursive game logic can be expressed equivalently in fixpoint logic with chop. To do this, fix two fresh variables u, v. Intuitively the purpose of these variables is to mark the end of the game, so that it can later be replaced by its continuation. The difference between the two variables is that v marks games that end in fixpoint variables, while u marks the end of all other games. This distinction will only be important later when considering a particular subclass of formulas.

For any rlGL formula φ and any RGL game α define FLC formulas $\varphi^\flat$ and $\alpha^\flat$ by structural induction:

$$(P)^\flat = P \qquad (\neg P)^\flat = \neg P$$
$$(\varphi \vee \psi)^\flat = \varphi^\flat \vee \psi^\flat \qquad (\varphi \wedge \psi)^\flat = \varphi^\flat \wedge \psi^\flat$$
$$(\langle\alpha\rangle\varphi)^\flat = \alpha^\flat\tfrac{\varphi^\flat}{\mathsf{u},\mathsf{v}}$$
$$(a)^\flat = \langle a\rangle\mathsf{u} \qquad (a^\mathrm{d})^\flat = [a]\mathsf{u}$$
$$(?\psi)^\flat = \psi^\flat \wedge \mathsf{u} \qquad (!\psi)^\flat = \neg(\psi)^\flat \vee \mathsf{u}$$
$$(\alpha \cup \beta)^\flat = \alpha^\flat \vee \beta^\flat \qquad (\alpha \cap \beta)^\flat = \alpha^\flat \wedge \beta^\flat$$
$$(x)^\flat = x \circ \mathsf{v} \qquad (\alpha;\beta)^\flat = \alpha^\flat\tfrac{\beta^\flat}{\mathsf{u},\mathsf{v}}$$
$$(\mathsf{r}x.\alpha)^\flat = (\mu x.\alpha^\flat\tfrac{\mathrm{id}}{\mathsf{u},\mathsf{v}}) \circ \mathsf{u} \qquad (\mathsf{l}x.\alpha)^\flat = (\nu x.\alpha^\flat\tfrac{\mathrm{id}}{\mathsf{u},\mathsf{v}}) \circ \mathsf{u}$$

Note that $\varphi\frac{\psi}{\mathsf{u},\mathsf{v}}$ denotes the formula obtained by replacing all appearances of u and v in φ by ψ. This is different from successive substitution $\varphi\frac{\psi}{\mathsf{u}}\frac{\psi}{\mathsf{v}}$. The substitutions here are always admissible, that is no fixpoint construct captures a free variable. In fact none of the variables that are substituted (u, v) even appears in the context of a fixpoint in the translation. (By choice of u, v, the variables do not appear in the original formula.)

Proposition 5.2 (Correct $\flat$). *For any well-named* RGL *formula φ and any well-named* RGL *game α*

(1) $\mathcal{N}[\![\varphi]\!]^I = \mathcal{N}[\![\varphi^\flat]\!]^I(A)$ *for any* $A \subseteq |\mathcal{N}|$
(2) $\mathcal{N}[\![\alpha]\!]^I \circ w = \mathcal{N}[\![\alpha^\flat]\!]^{I[\mathsf{u},\mathsf{v}\mapsto w]}$

See proof on page 16.

Corollary 5.3 (Roundtrip). $\vDash \varphi \leftrightarrow \varphi^{\#^\flat}$ *and* $\vDash \psi \leftrightarrow \psi^{\flat^\#}$ *for all well-named* FLC *formulas φ and all well-named* RGL *formulas ψ*

5.2 The Modal μ-Calculus as a Game Logic

In this section the precise extension of game logic that corresponds to modal μ-calculus is identified. Although the lack of fixpoint variables of the modal μ-calculus in game logic was remedied by introducing named subgames, the modal μ-calculus can only be understood as a game logic, in which games are played in a tail-recursive way. This is required to capture the regularity of the modal μ-calculus in the context of recursive game logic.

A game α of recursive game logic is *right-linear* if no subgame $\beta;\gamma$ of α has a free fixpoint variable in β. A formula φ of recursive game logic is *right-linear* if all its subgames are right-linear. The fragment of recursive game logic consisting only of right-linear formulas and games is called *right-linear game logic* (rlGL).

The translation $\#$ transforms formulas of the modal μ-calculus to right-linear game logic, since the only sequential games $\alpha;\beta$ introduced in the translation of α are of the form $a, a^\mathrm{d}, ?P$ or $?\overline{P}$. We can modify $\flat$ to ensure that it carries out a converse translation. For any rlGL formula φ and any rlGL game α define L_μ formulas $\varphi^\natural$ and $\alpha^\natural$ by structural induction. The definition of $\alpha^\natural$ is identical to the definition of $\alpha^\flat$, except for the following cases

$$(x)^\natural = x \qquad (\alpha;\beta)^\natural = \alpha^\natural\tfrac{\beta^\natural}{\mathsf{u}} \qquad (\mathsf{r}x.\alpha)^\natural = \mu x.\alpha^\natural$$

Note that $\varphi^{\natural}$ is a modal μ-calculus formula, as it does not mention $\circ$. This is a generalization of the translation employed in [23].

Proposition 5.4 (Correct $\natural$). *For any well-named* closed *right-linear* RGL *formula* φ *the* L_μ *formula* $\varphi^\flat$ *satisfies* $\mathcal{N}[\![\varphi^{\natural\flat}]\!] = \mathcal{N}[\![\varphi^\flat]\!]$. *Moreover* $\mathcal{N}[\![\varphi]\!] = \mathcal{N}[\![\varphi^\flat]\!](\emptyset)$.

See proof on page 17.

Corollary 5.5 (Equiexpressiveness for L_μ). *Right-linear game logic and modal μ-calculus are equiexpressive.*

Proof. It is easy to see that $\varphi^\sharp$ is a formula of right-linear game logic if φ is a formula in modal μ-calculus. This shows that right-linear game logic is at least as expressive as modal μ-calculus. The converse follows with Proposition 5.4. □

5.3 Game Logic as a Fixpoint Logic

Recall that the modal $*$-calculus is the fragment of the fixpoint logic with chop, which contains no fixpoints except in the form φ^*. Because the fixpoint structure in the modal $*$-calculus mirrors the structure in game logic, the translations between RGL and FLC also show the equiexpressiveness of the modal $*$-calculus and GL. This identifies the exact modal fixpoint logic corresponding to Parikh's original game logic.

The technical notion of formula separability will be used for the proof. A formula φ of the modal μ-calculus is *separable* if it contains fixpoints only in the forms $\mu x.(\psi \vee \rho)$ and $\nu x.(\psi \wedge \rho)$ where ρ does not mention x and ψ has no variable other than x free. Let L_s denote the set of separable formulas of the modal μ-calculus.

Lemma 5.6. (1) *If φ is a L_* formula, then $\varphi^\sharp$ is a* GL *formula.*
(2) *If φ is a well-named* GL *formula, then $\varphi^\natural$ is a L_s formula.*
(3) *Any L_s formula is equivalent to a L_* formula.*

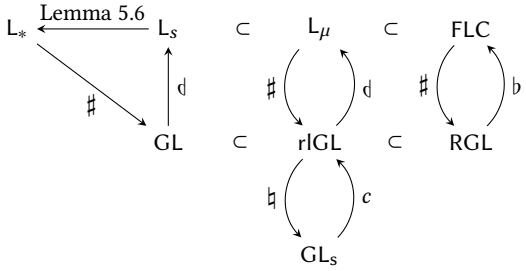
See proof on page 17.

Theorem 5.7 (Equiexpressiveness for GL). *Game logic (*GL*), the modal $*$-calculus (L_*) and the separable fragment of the modal μ-calculus (L_s) are equiexpressive.*

The equivalence between the separable fragment of modal μ-calculus and game logic has been shown [14, Theorem 3.3.10]. Theorem 5.7 adds to this equivalence the modal $*$-calculus. It is still open whether game logic is equivalent to the two variable fragment. With the above this can be reduced to the question of whether every L_μ formula is expressible in L_*.

5.4 Summary of Expressiveness

The following illustrates the relations of the logics considered:



The equivalences for sabotage game logic will be proved in the sequel. All inclusions in the illustration are strict. Game Logic is strictly less expressive than the modal μ-calculus [8], hence it is also less expressive than right-linear game logic. Fixpoint logic with chop and thus recursive game logic are strictly more expressive than the modal μ-calculus and right-linear game logic [33].

In the definition of RGL and sabotage game logic the games could contain tests of arbitrary formulas. For example $\langle ?(\langle a\rangle P)\rangle P$ is a well-formed Game Logic formula. This *rich-test* version is in contrast to the *poor-test* version in which only tests of literals (i.e. formulas P and $\neg P$) are allowed.

Corollary 5.8 (Tests). *The poor-test versions of game logic,right-linear game logic and recursive game logic are equiexpressive with their respective rich-test versions.*

Proof. This can be seen by translating into the corresponding fragment of fixpoint logic with chop via $\flat$ or $\natural$, since the backward translation via $\sharp$ yields an equivalent (Corollary 5.3) poor-test formula, since the translation $\sharp$ only introduces tests of literals. □

6 PROOF CALCULI

This section introduces natural proof calculi for right-linear game logic and sabotage game logic. We also recall Kozen's original calculus for the modal μ-calculus and its monotone variant, since completeness for the above game logics is obtained from completeness for the modal μ-calculus.

6.1 Proof Calculi for the Modal μ-Calculus

Because we consider the modal μ-calculus interpreted generally over neighbourhood structures, we recall the *monotone modal μ-calculus* mL_μ [24], the restriction of Kozen's calculus for the modal μ-calculus for neighbourhood structures. The monotone modal μ-calculus consists of all propositional tautologies together with all instances of the following axioms:

$$\text{(fp)}\ \varphi\tfrac{\mu x.\varphi}{x} \to \mu x.\varphi$$

$$(\alpha)\ \sigma x.\varphi \leftrightarrow \sigma y.\varphi\tfrac{y}{x} \quad (y \text{ fresh}, \sigma \in \{\mu,\nu\})$$

The rules of the proof calculus are:

$$\text{(MP)}\ \frac{\varphi \quad \varphi \to \psi}{\psi} \qquad (\mu)\ \frac{\varphi\frac{\psi}{x} \to \psi}{\mu x.\varphi \to \psi} \qquad (\mathrm{M}_a)\ \frac{\varphi \to \psi}{\langle a\rangle\varphi \to \langle a\rangle\psi}$$

We write $\mathsf{mL}_\mu \vdash \varphi$ if there is a Hilbert style proof of φ in the monotone modal μ-calculus. Note that this is a subset of Kozen's proof calculus for the modal μ-calculus [31]. Adding the following axioms yields the full Kozen calculus.

$$([\wedge])\ [a]\varphi \wedge [a]\psi \leftrightarrow [a](\varphi \wedge \psi)$$

$$(\mathrm{K})\ [a](\varphi \vee \psi) \to \langle a\rangle\varphi \vee [a]\psi$$

$$([\top])\ [a]\top$$

We write $\mathsf{L}_\mu \vdash \varphi$ if there is a Hilbert-style proof in this calculus of the formula φ. Both proof calculi are complete:

Proposition 6.1 (Enqvist, Seifan, Venema [24]). *The monotone modal μ-calculus is sound and complete with respect to monotone neighbourhood structures. That is $\mathsf{mL}_\mu \vdash \varphi$ iff $\models \varphi$ for L_μ formulas φ*

Proposition 6.2 (Walukiewicz [46]). *Kozen's calculus is sound and complete with respect to Kripke structures. That is $\mathsf{L}_\mu \vdash \varphi$ iff $\models_K \varphi$ for L_μ formulas φ.*

6.2 Proof Calculi for Game Logics

Parikh proposed a similar Hilbert-style proof calculus for game logic [36], which can be extended to right-linear game logic. Consider the axioms

$$(\neg)\ \langle \alpha^{\mathsf{d}}\rangle\varphi \leftrightarrow \neg\langle\alpha\rangle\neg\varphi$$

$$(?)\ \langle ?\varphi\rangle\psi \leftrightarrow \varphi \wedge \psi$$

$$(\cup)\ \langle \alpha \cup \beta\rangle\varphi \leftrightarrow \langle\alpha\rangle\varphi \vee \langle\beta\rangle\psi$$

$$(;)\ \langle \alpha;\beta\rangle\varphi \leftrightarrow \langle\alpha\rangle\langle\beta\rangle\varphi$$

$$(\mathrm{fp}_G)\ \langle \alpha\tfrac{\mathsf{r}x.\alpha}{x}\rangle\varphi \to \langle \mathsf{r}x.\alpha\rangle\varphi$$

$$(\alpha_G)\ \langle \sigma x.\alpha\rangle\varphi \leftrightarrow \langle \sigma y.\alpha\tfrac{y}{x}\rangle\varphi \quad (y \text{ fresh}, \sigma \in \{\mathsf{r}, \imath\})$$

and the rules

$$(\mathrm{MP}_G)\ \frac{\varphi \quad \varphi \to \psi}{\psi} \qquad (\mathrm{M}_G)\ \frac{\varphi \to \psi}{\langle\alpha\rangle\varphi \to \langle\alpha\rangle\psi}$$

$$(\mu_G)\ \frac{\langle \alpha\frac{\beta;?\psi;!\bot}{x}\rangle\varphi \to \langle\beta\rangle\psi}{\langle \mathsf{r}x.\alpha\rangle\varphi \to \langle\beta\rangle\psi}$$

We write $\mathsf{rIGL} \vdash \varphi$ to mean that there is a Hilbert style proof of the formula φ consisting only of *right-linear game logic* formulas. A more general proof calculus for full recursive game logic is of interest as well. (Note however that μ_G is not sound for this language, since the soundness proof relies on right-linearity.) Because there can not be a recursive and complete such calculus, we restrict attention to the calculus for right-linear game logic.

The new rule μ_G is an adaptation of the Park fixpoint induction rule to the setting of right-linear games. Completeness requires the rule M_G only for games of the form a, a^{d} and x, since the more general rule is derivable. With the more general definition however it is clear that if $\mathsf{GL} \vdash \varphi$, then by substituting free occurrences of x across the entire proof also $\mathsf{GL} \vdash \varphi\frac{\alpha}{x}$.

If there is a proof of $\mathsf{rIGL} \vdash \varphi$ consisting only of game logic formulas we write $\mathsf{GL} \vdash \varphi$. Observe that this is equivalent to Parikh's calculus for game logic [36]. The rule μ_G restricted to GL formulas is interderivable with the standard game logic induction rule

$$(\mu^*_G)\ \frac{\rho \vee \langle\alpha\rangle\psi \to \psi}{\langle\alpha^*\rangle\rho \to \psi}$$

in the context of the axioms $\cup$, ;, ?, ! and M_G. Similarly the axiom fp_G restricted to GL formulas is interderivable with

$$(*_G)\ \varphi \vee \langle\alpha\rangle\langle\alpha^*\rangle\varphi \to \langle\alpha^*\rangle\varphi$$

in the presence of axioms $\cup$, $\neg$, ; and ?.

The following axioms and rule are derivable from $\neg$:

$$(!)\ \langle !\varphi\rangle\psi \leftrightarrow (\varphi \to \psi)$$

$$(\cap)\ \langle \alpha \cap \beta\rangle\varphi \leftrightarrow \langle\alpha\rangle\varphi \wedge \langle\beta\rangle\psi$$

$$(\nu_G)\ \frac{\langle\beta\rangle\psi \to \langle \alpha\frac{\beta;?\psi;!\bot}{x}\rangle\rho}{\langle\beta\rangle\psi \to \langle \nu x.\alpha\rangle\rho}$$

Rule ν_G captures the coinductive nature of the greatest fixpoints that define Demons's games, similarly to how the induction rule μ_G formalizes the inductive nature of Angel's games.

Originally Parikh's calculus was designed for game logic interpreted over monotone neighbourhood structures. An extension of Parikh's calculus with the axioms $[\wedge]$, K and $[\top]$ from Kozen's calculus turns it into a suitable axiomatization for game logic over Kripke structures. The appearance of a in these axioms ranges only over *atomic games* and not over arbitrary games for which these axioms would be unsound. If there is a proof of φ in the rIGL calculus from the axioms $[\wedge]$, K and $[\top]$ we write $\mathsf{rIGL} + G \vdash \varphi$. The proof calculus is sound for monotone neighbourhood structures and Kripke structures:

Lemma 6.3 (Soundness). *For any formula φ of* rIGL

(1) *if* $\mathsf{rIGL} \vdash \varphi$ *then* $\models \varphi$
(2) *if* $\mathsf{rIGL} + G \vdash \varphi$ *then* $\models_K \varphi$

Proof. For most axioms and rules the proof of soundness is exactly the same as for game logic [36]. Soundness of μ_G follows as in game logic with Lemma 3.7. □

Lemma 6.4. *The right-linearity axiom and the reverse fixpoint axiom are derivable in the* rIGL *calculus for* rIGL

$$(\mathrm{RL})\ \langle \alpha\tfrac{\beta}{x}\rangle\varphi \leftrightarrow \langle \alpha\tfrac{\beta;?\varphi;!\bot}{x}\rangle\varphi \quad (\alpha \text{ is right-linear})$$

$$(\mathrm{rfp}_G)\ \langle \mathsf{r}x.\alpha\rangle\varphi \to \langle \alpha\tfrac{\mathsf{r}x.\alpha}{x}\rangle\varphi$$

Proof. Axiom RL can be derived by induction on α. For axiom rfp_G by RL and μ_G, it suffices to prove

$$\langle \alpha\tfrac{\alpha\frac{\mathsf{r}x.\alpha}{x}}{x}\rangle\varphi \to \langle \alpha\tfrac{\mathsf{r}x.\alpha}{x}\rangle\varphi$$

in the calculus. By induction one proves more generally for any right-linear game β that

$$\langle \beta\tfrac{\alpha\frac{\mathsf{r}x.\alpha}{x}}{x}\rangle\varphi \to \langle \beta\tfrac{\mathsf{r}x.\alpha}{x}\rangle\varphi.$$

For the case that $\beta \equiv x$ this holds by axiom fp_G. □

6.2.1 Proof Calculus for Sabotage Game Logic. In addition to the proof calculus for right-linear game logic, the proof calculus for game logic can be extended to a proof calculus for sabotage game logic. This involves adding axioms for the rule-change games:

$(\sim)\ \langle \sim a\rangle\langle \alpha \frac{a;\vec{\beta}}{\vec{x}} \frac{a^{\mathrm{d}};\vec{\gamma}}{\vec{y}} \frac{\vec{\delta}}{\vec{z}}\rangle\varphi \leftrightarrow \langle \alpha \frac{\sim a;\vec{\beta}}{\vec{x}} \frac{!\bot}{\vec{y}} \frac{\sim a;\vec{\delta}}{\vec{z}}\rangle\varphi$ (†)

$(\wr)\ \langle \sim a^{\mathrm{d}}\rangle\langle \alpha \frac{a;\vec{\beta}}{\vec{x}} \frac{a^{\mathrm{d}};\vec{\gamma}}{\vec{y}} \frac{\vec{\delta}}{\vec{z}}\rangle\varphi \leftrightarrow \langle \alpha \frac{?\bot}{\vec{x}} \frac{\sim a^{\mathrm{d}};\vec{\gamma}}{\vec{y}} \frac{\sim a^{\mathrm{d}};\vec{\delta}}{\vec{z}}\rangle\varphi$ (†)

$(\uplus)\ \langle \eta; \sim a_i; \beta_i\rangle(\langle \alpha \frac{\beta}{w}\rangle\varphi \leftrightarrow \langle \alpha \frac{(\cup_{i=1}^{n} a_i;\beta_i);\beta}{w}\rangle\varphi)$ $(\eta = \sim a_1^{\mathrm{d}}; \ldots; \sim a_n^{\mathrm{d}}$, and ‡)

$(\approx)\ \langle \sim a; \alpha \frac{a}{x}\rangle\varphi \leftrightarrow \langle \sim a; \alpha \frac{a;\sim a}{x}\rangle\varphi$

$(\simeq)\ \langle \alpha \frac{\vec{\eta}}{\vec{x}} \frac{\vec{\delta};\beta}{\vec{y}}\rangle\varphi \leftrightarrow \langle \alpha \frac{?\top}{\vec{x}} \frac{\vec{\delta};\beta}{\vec{y}}\rangle\varphi$ $(a, a^{\mathrm{d}} \notin \alpha, \varphi$ and $\eta_i, \delta_i \in \{\sim a, \sim a^{\mathrm{d}}\})$

$(\sim_1)\ \langle \alpha \frac{a;\beta}{x} \frac{\sim a^{\mathrm{d}}}{y}\rangle\varphi \leftrightarrow \langle \alpha \frac{a;\beta}{x} \frac{\sim a^{\mathrm{d}};\eta}{y}\rangle\varphi$ $(\eta \in \{\sim b, \sim b^{\mathrm{d}}\}$ and $\sim a, b, b^{\mathrm{d}} \notin \alpha, \varphi)$

The notation in axioms should be understood schematically. In $\sim$ and $\wr$ the game α is a $\mathsf{GL_s}$ game with two lists of free variables $\vec{x}, \vec{y}, \vec{z}$ occuring only right-linearly[1] and $\vec{\beta}, \vec{\gamma}, \vec{\delta}$ are lists of sabotage game logic games of the same dimensions that are substituted into α. The axioms $\sim$ and $\wr$ require the side condition † that $a, a^{\mathrm{d}}, \sim a, \sim a^{\mathrm{d}}$ do not appear in α and that a, a^{d} do not appear in φ. The axioms $\sim$ and $\wr$ capture the effect of a rule change deep within a formula. The axiom $\uplus$ allows to reason about more complex branching behaviour. It requires as a side condition ‡ saying that each β_i is a formula of the form $\sim b_1; \ldots; \sim b_j; \sim b_{j+1}^{\mathrm{d}}; \ldots \sim b_m^{\mathrm{d}}$, such that sabotages of the atomic games a_i, b_j appear in α only in the form $\eta; \sim a_i; \beta_i$. Axiom $\simeq$ allows the uniform removal of unneeded sabotages.

The proof calculus for sabotage game logic consists of the six axioms $\sim$, $\wr$, $\uplus$, $\simeq$, $\sim_1$ and $\approx$ together with $\neg$, ?, $\cup$, ;, $*_G$, MP_G, μ_G^* and M_G. If there is a Hilbert-style proof of φ in this calculus consisting only of $\mathsf{GL_s}$ formulas, write $\mathsf{GL_s} \vdash \varphi$. The proof calculus for $\mathsf{GL_s}$ can also be adjusted to work for Kripke structures. This extension adds the axioms $[\wedge]$, K and $[\top]$. Write $\mathsf{GL_s} + G \vdash \varphi$ if there is a proof of φ in this extension.

Lemma 6.5 (Soundness). *For any formula φ of* $\mathsf{GL_s}$

(1) *if* $\mathsf{GL_s} + G \vdash \varphi$ *then* $\vDash_K \varphi$
(2) *if* $\mathsf{GL_s} \vdash \varphi$ *then* $\vDash \varphi$

Proof. Soundness of the common part of the proof calculus goes through exactly as for game logic [36]. Soundness of $\sim$ and $\wr$ is proved by induction on α using the side condition †. □

See full proof on page 17.

6.3 Equivalence of Proof Calculi

This section shows that the translations between right-linear game logic and the modal μ-calculus shows not only equiexpressiveness, but also that the proof calculi are equivalent. This will enable the transfer of completeness from modal μ-calculus to right-linear game logic.

Rank. In order to flatten the mutual inductions that arise naturally, because of the mutually inductive definition of formulas and games, a well-order on all formulas is required. This order is given by the rank of a formula of game logic defined in Appendix A. Many of the proofs about the calculi for game logic are carried out by induction on the rank of a formula.

[1] No variable appears on the left hand side of a composition, in the scope of a loop or in a test.

Correctness of Translation. The translations between right-linear game logic and the modal μ-calculus have been proved sound semantically. In order to use these to relate the proof calculi, the soundness of the translation needs to be proved in the calculus itself. Since each calculus can only talk about formulas in its respective language the correct notion of correctness here is that of Corollary 5.3.

Lemma 6.6 (Provable Correctness). (1) $\mathsf{rlGL} \vdash \varphi \leftrightarrow \varphi^{\natural\sharp}$ *for any well-named* rlGL *formula* φ

(2) *and* $\mathsf{mL}_\mu \vdash \varphi \leftrightarrow \varphi^{\sharp\flat}$ *for any well-named* L_μ *formula* φ

See proof on page 18.

The key then is that proofs in the modal μ-calculus can be turned into right-linear game logic proofs. Since the modal μ-calculus is complete and has the same expressive power as right-linear game logic it follows that any formula φ is provable up to translation.

Theorem 6.7 (Equivalence of Calculi). *Right-linear game logic and the modal μ-calculus prove the same formulas modulo translation.*

(1) $\mathsf{rlGL} \vdash \varphi$ *iff* $\mathsf{mL}_\mu \vdash \varphi^{\natural}$ *for closed well-named* rlGL *formulas* φ
(2) $\mathsf{L}_\mu \vdash \varphi$ *iff* $\mathsf{rlGL} \vdash \varphi^{\sharp}$ *for well-named* L_μ *formulas* φ

Proof. For the forward direction of (1) observe that Lemma 6.3 and Corollary 5.5 imply $\vDash \varphi^{\natural}$. Hence $\mathsf{mL}_\mu \vdash \varphi^{\natural}$ follows from Proposition 6.1. The forward direction of (2) is proved in the appendix as Lemma D.2.

For the backward direction of (1) consider $\mathsf{mL}_\mu \vdash \varphi^{\natural}$. By the forward direction of (2) also $\mathsf{rlGL} \vdash \varphi^{\natural\sharp}$. Then by Lemma 6.6 conclude $\mathsf{rlGL} \vdash \varphi$. The backward implication of (2) is similar. □

6.4 Completeness for rlGL

Because the translations are semantically correct and preserve provability by the results of Section 6.3, completeness of rlGL follows.

Theorem 6.8 (rlGL Completeness). *For a closed* rlGL *formula* φ

(1) $\mathsf{rlGL} + G \vdash \varphi$ *iff* $\vDash_K \varphi$
(2) $\mathsf{rlGL} \vdash \varphi$ *iff* $\vDash \varphi$

Proof. (2) By α_G assume without loss of generality that φ is well-named. Consider the following chain of equivalences

$$\begin{aligned} & \vDash \varphi \\ \text{iff} \quad & \vDash \varphi^{\natural} && \text{(Corollary 5.5)} \\ \text{iff} \quad & \mathsf{mL}_\mu \vdash \varphi^{\natural} && \text{(Proposition 6.1)} \\ \text{iff} \quad & \mathsf{rlGL} \vdash \varphi^{\natural\sharp} && \text{(Theorem 6.7)} \\ \text{iff} \quad & \mathsf{rlGL} \vdash \varphi && \text{(Lemma 6.6 and MP)} \end{aligned}$$

This proves the equivalence.

(1) A slight modification of the proof of Lemma D.2 shows that $\mathsf{rlGL} + G \vdash \psi^{\sharp}$ provided $\mathsf{L}_\mu \vdash \psi$ for a closed L_μ formula ψ. All that needs to be added is that the $\sharp$-translation of the axioms $[\wedge]$, K and $[\top]$ are again instances of the same axiom in $\mathsf{rlGL} + G$. The same argument as for (2) then applies to show completeness of rlGL. □

7 SABOTAGE GAME LOGIC AND MODAL μ

In this section the equiexpressiveness of rlGL and modal μ-calculus are exploited to obtain the desired equiexpressiveness of sabotage game logic with the modal μ-calculus as well as completeness of sabotage game logic through translations turning sabotage modalities into named subgames and vice versa.

7.1 Embedding of $\mathsf{GL_s}$ into rlGL

The difficulty in embedding $\mathsf{GL_s}$ into rlGL is that the ownership information about previously committed acts of sabotage must be taken into account. This can be done here by coding this information on the claimed atomic games into the nesting structure of the fixpoint variables. To simplify this coding, we use simultaneous fixpoints, which do not add to the expressive power. This is captured by the following proposition, known as Bekić's Theorem,

Theorem 7.1 ([4, Lemma 1.4.2]). *For variables $x_1, \dots, x_n$ and rlGL games $\alpha_1, \dots, \alpha_n$ there are rlGL games $\beta_1, \dots, \beta_n$ such that*

$$\begin{pmatrix} \mathcal{N}[\![\beta_1]\!]^I \\ \mathcal{N}[\![\beta_2]\!]^I \\ \vdots \\ \mathcal{N}[\![\beta_m]\!]^I \end{pmatrix} = \mu \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{pmatrix} . \begin{pmatrix} \mathcal{N}[\![\alpha_1]\!]^{I[\vec{x}\mapsto\vec{w}]} \\ \mathcal{N}[\![\alpha_2]\!]^{I[\vec{x}\mapsto\vec{w}]} \\ \vdots \\ \mathcal{N}[\![\alpha_m]\!]^{I[\vec{x}\mapsto\vec{w}]} \end{pmatrix}$$

We write $\mathsf{r}_i(x_1, \dots, x_n).(\alpha_1, \dots, \alpha_n)$ for this β_i.

The Embedding. We fix for every possible context $c \in C$ a fresh variable y_c. For any formula φ and any game α of sabotage game logic we define a translation α^c depending on the context c. The context allows the translation to depend on the state of ownership of atomic games. Moreover the translation of games will contain free variables y_c. Those mark the end of the game and keep track of the context in which this end has been reached. This allows a compositional definition of the translation. Let

$$(P)^c = ?P; !\bot \qquad (\neg P)^c = ?\neg P; !\bot$$

$$(\varphi \vee \psi)^c = \varphi^c \cup \psi^c \qquad (\varphi \wedge \psi)^c = \varphi^c \cap \psi^c$$

$$(a)^c = \begin{cases} a; y_c & \text{if } c(a) = \emptyset \\ y_c & \text{if } c(a) = \diamond \\ ?\bot & \text{if } c(a) = \Box \end{cases} \qquad (a^{\mathrm{d}})^c = \begin{cases} a^{\mathrm{d}}; y_c & \text{if } c(a) = \emptyset \\ !\bot & \text{if } c(a) = \diamond \\ y_c & \text{if } c(a) = \Box \end{cases}$$

$$(\sim a)^c = y_{c\frac{\diamond}{a}} \qquad (\sim a^{\mathrm{d}})^c = y_{c\frac{\Box}{a}}$$

$$(?\varphi)^c = \varphi^c \cap y_c \qquad (!\varphi)^c = \overline{\varphi}^c \cup y_c$$

$$(\alpha \cup \beta)^c = \alpha^c \cup \beta^c \qquad (\alpha \cap \beta)^c = \alpha^c \cap \beta^c$$

The translation of atomic games and sabotage games illustrates the importance of translating relative to a context. The translation of formulas $\langle\alpha\rangle\varphi$ and games $\alpha;\beta$ and α^* are slightly more involved. For the first two define

$$(\langle\alpha\rangle\varphi)^c = \alpha^{c\,\frac{\varphi^\cdot;?\bot}{y_\cdot}}$$

$$(\alpha;\beta)^c = \alpha^{c\,\frac{\beta^\cdot}{y_\cdot}}$$

where the notation $\alpha^{c\,\frac{\beta^\cdot}{y_\cdot}}$ means that any instance of a variable y_e is replaced by β^e, the translation of β with respect to the context e.

For a fixpoint game α^* list all the atomic games $a_1, \dots, a_m$ such that either $c(a) \neq \emptyset$ or $\sim a$ or $\sim a^{\mathrm{d}}$ appears in α. List $c_1, \dots, c_m$ all possible contexts that satisfy $c_i(a) = \emptyset$ for all $a \notin \{a_1, \dots, a_m\}$. Define the translation of repetition games

$$(\alpha^*)^{c_i} = \mathsf{r}_i(z_{c_1}, \dots, z_{c_n}).(y_{c_1} \cup \alpha^{c_1}\tfrac{z_{c_\cdot}}{y_\cdot}, \dots, y_{c_n} \cup \alpha^{c_n}\tfrac{z_{c_\cdot}}{y_\cdot})$$

$$(\alpha^\times)^{c_i} = \mathsf{l}_i(z_{c_1}, \dots, z_{c_n}).(y_{c_1} \cap \alpha^{c_n}\tfrac{z_{c_\cdot}}{y_{c_\cdot}}, \dots, y_{c_n} \cap \alpha^{c_n}\tfrac{z_{c_\cdot}}{c_\cdot;y_\cdot})$$

where the $z_{c_\cdot}$ are fresh variables. Observe that the translation of any $\mathsf{GL_s}$ formula or game is a right-linear rlGL game.

The next proposition shows that the translation is correct.

Proposition 7.2. *For any formula φ of sabotage game logic*

$$\mathcal{N}[\![\varphi]\!]_{\mathsf{s}}\restriction c = \mathcal{N}[\![\varphi^c]\!](\emptyset).$$

and for any game α of sabotage game logic

$$\mathcal{N}[\![\alpha]\!]_{\mathsf{s}}(U)\restriction c = \mathcal{N}[\![\alpha^c]\!]^I(\emptyset)$$

where $I(y_e) = U\restriction e$.

See proof on page 19.

The translation of a sabotage game logic formulas into a formula of right-linear game logic potentially grows very quickly. The upper bound on the length of the translation of a fixpoint game obtained from the proof above is

$$|(\alpha)^c| \leq (C \cdot |\alpha|)^{3^\ell \uparrow\uparrow k},$$

where k is the fixpoint nesting depth of α and ℓ is the number of atomic games in α.[2] This comes from the fact that the translation of any game α considers all of the up to 3^ℓ-many relevant contexts. The only known transformation from vectorial fixpoints to non-vectorial nested fixpoints as in Theorem 7.1 grows exponentially in the formula size. Consequently every fixpoint leads to a doubly exponential blow-up in length. In [13] it is shown that reducing vectorial fixpoints to non-vectorial fixpoints is at least as hard as solving parity games, for which the existence of a polynomial time algorithm is a longstanding open question. It was conjectured [12] that a vectorial fixpoint formula can be exponentially smaller than the shortest equivalent non-vectorial formula.

While it is unclear to what extent this upper bound is optimal, it suggests that complex formulas of the modal μ-calculus may in general be expressed much more succinctly in sabotage game logic.

7.2 Embedding of rlGL into $\mathsf{GL_s}$

The challenge of the converse translation from rlGL to $\mathsf{GL_s}$ is that the arbitrarily nested named games of rlGL need to be turned into structured repetition games of sabotage game logic. It turns out that using sabotage, players can force the behaviour of nested named games onto structured repetition games. To facilitate this, fix fresh atomic games b_x, c_x for every variable x. By induction define the

[2] We use Knuth's up-arrow notation for k-fold iterated exponentiation.

translation of a rlGL formula and game into $\mathsf{GL_s}$:

$$(P)^\natural = P \qquad (\neg P)^\natural = \neg P$$
$$(\varphi \vee \psi)^\natural = \varphi^\natural \vee \psi^\natural \qquad (\varphi \wedge \psi)^\natural = \varphi^\natural \wedge \psi^\natural$$
$$(\langle\alpha\rangle\varphi)^\natural = \langle\alpha^\natural\rangle\varphi^\natural$$
$$(a)^\natural = a \qquad (a^{\mathrm{d}})^\natural = a^{\mathrm{d}}$$
$$(?\varphi)^\natural = ?\varphi^\natural \qquad (!\varphi)^\natural = !\varphi^\natural$$
$$(\alpha \cup \beta)^\natural = \alpha^\natural \cup \beta^\natural \qquad (\alpha \cap \beta)^\natural = \alpha^\natural \cap \beta^\natural$$
$$(\alpha;\beta)^\natural = \alpha^\natural;\beta^\natural \qquad (x)^\natural = x$$

$$(\mathrm{r}x.\alpha)^\natural = \delta;(c_x;\delta^{\mathrm{d}};\alpha^\natural\tfrac{\delta}{x})^*;b_x \qquad \text{where } \delta \equiv {\sim}b_x^{\mathrm{d}};{\sim}c_x$$
$$(\imath x.\alpha)^\natural = \delta^{\mathrm{d}};(c_x^{\mathrm{d}};\delta;\alpha^\natural\tfrac{\delta^{\mathrm{d}}}{x})^\times;b_x^{\mathrm{d}} \qquad \text{where } \delta \equiv {\sim}b_x^{\mathrm{d}};{\sim}c_x$$

While only the translation of closed formulas of right-linear game logic is of interest, the translation is defined more generally for open formulas into the extension of sabotage game logic with free variables.

Intuitively the translation of the fixpoints uses rule changes to remove the non-determinism from the repetition game. In a normal α^* game it is Angel's choice whether to continue playing the game α or not. However in general $\mathrm{r}x.\alpha$ games, this choice is made differently. If the variable x is reached the game must be repeated. If the game ends without reaching this variable it must not be repeated. The translation enforces this deterministic behaviour of the repetition game in a * game. Although the *-game theoretically allows Angel to stop prematurely, Demon will have changed the rules (${\sim}b_x^{\mathrm{d}}$), so that Angel would automatically lose afterwards. Similarly if Angel does not stop when she ought to, she will also lose immediately, due to a demonic claim on ${\sim}c_x^{\mathrm{d}}$. This ensures correctness of the translation.

Proposition 7.3 (Correct $\natural$). *For any* closed, *well-named formula φ of right-linear game logic the translation $\varphi^\natural$ is a formula of sabotage game logic with*

$$\mathcal{N}[\![\varphi^\natural]\!]_{\mathsf{s}}\restriction c = \mathcal{N}[\![\varphi]\!].$$

See proof on page 20.

Theorem 7.4 (Equiexpressiveness). *Sabotage game logic, right-linear game logic and the modal μ-calculus are equiexpressive.*

Proof. By Propositions 7.2 and 7.3 and Corollary 5.5. □

The equiexpressiveness of sabotage game logic and the modal μ-calculus means that sabotage game logic inherits many of the nice properties of the modal μ-calculus for free.

Theorem 7.5 (Meta properties). *Sabotage game logic has the small model property and the satisfiability problem is decidable.*

Proof. The modal μ-calculus has these properties [40, 43], and they can be transferred to sabotage game logic by Theorem 7.4. □

7.3 Proof Transformations

This section shows that the translation also respects the proof calculus. Combined with the semantic correctness of the translation this allows the transfer of completeness from rlGL to $\mathsf{GL_s}$. The key fact needed about the translation is that the sabotage paraphrasing of the named recursive games, *provably* behaves the same as the extremal fixpoint it denotes.

Lemma 7.6. *For any formula φ and any game α of right-linear game logic the following hold:*

(1) $\overline{\varphi^\natural} \equiv \overline{\varphi}^\natural$
(2) $\mathsf{GL_s} \vdash (\langle\alpha\frac{\mathrm{r}x.\alpha}{x}\rangle\varphi \to \langle\mathrm{r}x.\alpha\rangle\varphi)^\natural$
(3) $\mathsf{GL_s} \vdash (\langle\mathrm{r}x.\alpha\rangle\varphi)^\natural \to \langle\beta\rangle\psi$ *if* $\mathsf{GL_s} \vdash \langle\alpha^\natural\frac{\beta;?\psi;!\top}{x}\rangle\varphi^\natural \to \langle\beta\rangle\psi$

See proof on page 21.

Lemma 7.7. *Suppose φ is a formula of sabotage game logic then* $\mathsf{GL_s} \vdash (\langle\varphi^{c_\emptyset}\rangle\bot)^\natural \to \varphi$.

See proof on page 21.

Since the translation $\natural$ provably behaves as intended, rlGL proofs can be translated completely to $\mathsf{GL_s}$.

Proposition 7.8. *Suppose φ is a* closed *right-linear game logic formula and* rlGL $\vdash \varphi$ *then* $\mathsf{GL_s} \vdash \varphi^\natural$.

See proof on page 23.

Theorem 7.9 (Sabotage game logic Completeness). *Sabotage game logic is sound and complete. That is for all* $\mathsf{GL_s}$ *formulas φ:*

$$\mathsf{GL_s} \vdash \varphi \qquad \textit{iff} \qquad \vDash \varphi$$

See proof on page 23.

7.4 A Completion of Parikh's Calculus

Fix infinite disjoint sets $\mathbb{G}_1 \cup \mathbb{G}_2 \cup \mathbb{G}_3 = \mathbb{G}$. For all $b \in \mathbb{G}_2$ fix a unique $\tilde{b} \in \mathbb{G}_3$. Consider the set $\mathfrak{A}$ consisting of all formulas obtained from instances of the axioms $\sim, \wr, \approx$ and $\wr\!\wr$ of sabotage game logic which contain atomic games only from $\mathbb{G}_2$ by replacing all ${\sim}b$ and all ${\sim}b^{\mathrm{d}}$ by $\tilde{b}$ and $(\tilde{b})^{\mathrm{d}}$ respectively. Taken as axioms these *game logic* formulas suffice to complete Parikh's proof calculus for game logic. Write $\mathsf{GL} + \mathfrak{A} \vdash \varphi$ if there is a proof of φ in Parikh's calculus from these axioms.

Theorem 7.10 (Game logic Completeness). *For any* GL *formula φ with atomic games from $\mathbb{G}_1$:*

(1) $\vDash \varphi$ *iff* $\mathsf{GL} + \mathfrak{A} \vdash \varphi$
(2) $\vDash_K \varphi$ *iff* $\mathsf{GL} + G + \mathfrak{A} \vdash \varphi$

Proof. *Soundness:* Suppose $\mathsf{GL} + \mathfrak{A} \vdash \varphi$ and consider a monotone neighbourhood structure $\mathcal{N}$. Replacing every appearance of some $\tilde{b}$ in a proof of $\mathsf{GL} + \mathfrak{A} \vdash \varphi$ by ${\sim}b$ gives a proof for $\mathsf{GL_s} \vdash \varphi$. Hence by soundness (Lemma 6.5) conclude that $\vDash \varphi$.

Completeness: Suppose φ is a valid formula. As in the proof of Theorem 7.9 obtain a proof of $\mathsf{GL_s} \vdash \varphi$. By the construction of this proof (Proposition 7.8) it is clear that the proof contains games of the form ${\sim}a, {\sim}a^{\mathrm{d}}$ only for atomic games that do not already appear in φ. Without loss of generality assume the proof contains games of the form ${\sim}b, {\sim}b^{\mathrm{d}}$ only for $b \in \mathbb{G}_2$.

The $\mathsf{GL_s}$ proof of φ can be transformed into a game logic proof by uniformly replacing every game of the form $\sim a$ by $\tilde{b}$ and every $\sim a^{\mathrm{d}}$ by $\tilde{b}^{\mathrm{d}}$. Instances of the axioms $\sim, \wr, \approx$ and $\wr\!\wr$ in the original proof become instances of axioms in $\mathfrak{A}$. Hence the modified version of the proof is a valid GL proof of φ from the axioms in $\mathfrak{A}$. □

The results in this section pave the way for using the proof calculus for $\mathsf{GL_s}$ to address the question of completeness of Parikh's axiomatization for game logic through a proof transformation by eliminating instances of axioms from $\mathfrak{A}$.

8 CONCLUSION

This paper studies how logic, games, and fixpoints meet by introducing two different extensions of Game Logic. The first, sabotage game logic, allows players to sabotage their opponent, while the second, recursive game logic, adds recursive games.

Not only is sabotage game logic ($\mathsf{GL_s}$) well-suited to describe and investigate games with rule changes by logical means, but perhaps surprisingly, sabotage game logic has a number of additional advantages over game logic. Unlike game logic, the extension sabotage game logic allows exactly the right amount of state to increase its expressive power to match the modal μ-calculus, without sacrificing the desirable logical properties of game logic.

The virtue of recursive game logic (RGL) is that it allows the description of games featuring arbitrarily nested recursive games. Unlike ordinary Game Logic, the extended version is significantly more expressive than the modal μ-calculus, although it remains syntactically close to GL. We have identified the fragment of RGL that corresponds exactly to the modal μ-calculus in expressiveness and transferred completeness of modal μ-calculus to obtain a complete and natural proof calculus for this fragment.

Additionally, it was shown that sabotage game logic and the modal μ-calculus are equivalent in expressiveness via a translation going through the right-linear fragment of recursive game logic. Completeness of the natural Hilbert style proof calculus for sabotage game logic $\mathsf{GL_s}$ was obtained as a consequence. This is in contrast to game logic GL for which completeness of the natural proof calculus is not known [30]. Completeness of sabotage game logic was used to obtain the completeness of a modest extension of Parikh's proof calculus for game logic GL.

Future Research. The completeness of sabotage game logic suggests an interesting approach to studying proof calculi for game logic. It reduces the problem of the completeness of Parikh's axiomatization of game logic to eliminating instances of the new axioms in a proof. Equiexpressiveness with L_μ indicates that atomic games for sabotage are worth studying further.

The translation from $\mathsf{GL_s}$ into L_μ leads to a non-elementary blow-up in formula length. This raises the question whether this increase is necessary and if better algorithms exist that directly target the model checking and satisfiability problems of $\mathsf{GL_s}$.

ACKNOWLEDGMENTS

This project was supported by an Alexander von Humboldt Professorship.

REFERENCES

[1] S. Abramsky and P.-A. Mellies. 1999. Concurrent games and full completeness. In *Proceedings. 14th Symposium on Logic in Computer Science (Cat. No. PR00158)*. 431–442. https://doi.org/10.1109/LICS.1999.782638

[2] Bahareh Afshari and Graham E. Leigh. 2017. Cut-free completeness for modal mu-calculus. In *32nd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2017, Reykjavik, Iceland, June 20-23, 2017*. IEEE Computer Society, 1–12. https://doi.org/10.1109/LICS.2017.8005088

[3] Carlos Areces, Raul Fervari, Guillaume Hoffmann, and Mauricio Martel. 2017. Undecidability of Relation-Changing Modal Logics. In *Dynamic Logic. New Trends and Applications - First International Workshop, DALI 2017, Brasilia, Brazil, September 23-24, 2017, Proceedings (Lecture Notes in Computer Science, Vol. 10669)*, Alexandre Madeira and Mario R. F. Benevides (Eds.). Springer, 1–16. https://doi.org/10.1007/978-3-319-73579-5_1

[4] André Arnold and Damian Niwinski. 2001. *Rudiments of mu-calculus*. Studies in Logic and the Foundations of Mathematics, Vol. 146. North Holland Publishing Co., Amsterdam.

[5] Guillaume Aucher, Johan van Benthem, and Davide Grossi. 2015. Sabotage Modal Logic: Some Model and Proof Theoretic Aspects. In *Logic, Rationality, and Interaction - 5th International Workshop, LORI 2015 Taipei, Taiwan, October 28-31, 2015, Proceedings (Lecture Notes in Computer Science, Vol. 9394)*, Wiebe van der Hoek, Wesley H. Holliday, and Wen-Fang Wang (Eds.). Springer, 1–13. https://doi.org/10.1007/978-3-662-48561-3_1

[6] Guillaume Aucher, Johan van Benthem, and Davide Grossi. 2018. Modal logics of sabotage revisited. *J. Log. Comput.* 28, 2 (2018), 269–303. https://doi.org/10.1093/LOGCOM/EXX034

[7] Dietmar Berwanger. 2003. Game Logic is Strong Enough for Parity Games. *Studia Logica* 75, 2 (2003), 205–219. https://doi.org/10.1023/A:1027358927272

[8] Dietmar Berwanger, Erich Grädel, and Giacomo Lenzi. 2007. The Variable Hierarchy of the μ-calculus is strict. *Theory Comput. Syst.* 40, 4 (2007), 437–466.

[9] Francesca Zaffora Blando, Krzysztof Mierzewski, and Carlos Areces. 2020. The Modal Logics of the Poison Game. In *Knowledge, Proof and Dynamics*, Fenrong Liu, Hiroakira Ono, and Junhua Yu (Eds.). Springer, 3–23.

[10] Julian C. Bradfield. 1996. The Modal mu-calculus Alternation Hierarchy is Strict. In *CONCUR '96, Concurrency Theory, 7th International Conference, Pisa, Italy, August 26-29, 1996, Proceedings (LNCS, Vol. 1119)*, Ugo Montanari and Vladimiro Sassone (Eds.). Springer, 233–246. https://doi.org/10.1007/3-540-61604-7_58

[11] Julian C. Bradfield and Colin Stirling. 2006. Modal Mu Calculi. In *Handbook of Modal Logic*, Patrick Blackburn, Johann van Benthem, and Frank Wolter (Eds.). Elsevier, 721–756.

[12] Julian C. Bradfield and Igor Walukiewicz. 2018. The mu-calculus and Model Checking. In *Handbook of Model Checking*, Edmund M. Clarke, Thomas A. Henzinger, Helmut Veith, and Roderick Bloem (Eds.). Springer, 871–919. https://doi.org/10.1007/978-3-319-10575-8_26

[13] Florian Bruse, Oliver Friedmann, and Martin Lange. 2015. On guarded transformation in the modal μ-calculus. *Log. J. IGPL* 23, 2 (2015), 194–216. https://doi.org/10.1093/JIGPAL/JZU030

[14] Facundo Carreiro. 2015. *Fragments of fixpoint logics*. Ph. D. Dissertation. Ph. D. thesis, University of Amsterdam.

[15] Krishnendu Chatterjee, Thomas A. Henzinger, and Nir Piterman. 2007. Strategy Logic. In *CONCUR 2007 - Concurrency Theory, 18th International Conference, CONCUR 2007, Lisbon, Portugal, September 3-8, 2007, Proceedings (Lecture Notes in Computer Science, Vol. 4703)*, Luís Caires and Vasco Thudichum Vasconcelos (Eds.). Springer, 59–73. https://doi.org/10.1007/978-3-540-74407-8_5

[16] Corina Cîrstea, Clemens Kupke, and Dirk Pattinson. 2011. EXPTIME Tableaux for the Coalgebraic mu-Calculus. *Log. Methods Comput. Sci.* 7, 3 (2011). https://doi.org/10.2168/LMCS-7(3:3)2011

[17] Pierre Clairambault. 2009. Least and Greatest Fixpoints in Game Semantics. In *Foundations of Software Science and Computational Structures, 12th International Conference, FOSSACS 2009, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2009, York, UK, March 22-29, 2009. Proceedings (Lecture Notes in Computer Science, Vol. 5504)*, Luca de Alfaro (Ed.). Springer, 16–31. https://doi.org/10.1007/978-3-642-00596-1_3

[18] Pierre Duchet and Henry Meyniel. 1993. Kernels in directed graphs: a poison game. *Discret. Math.* 115, 1-3 (1993), 273–276. https://doi.org/10.1016/0012-365X(93)90496-G

[19] Andrzej Ehrenfeucht. 1961. An application of games to the completeness problem for formalized theories. *Fundamenta Mathematicae* 49 (1961), 129–141. https://api.semanticscholar.org/CorpusID:118038695

[20] E. Allen Emerson and Charanjit S. Jutla. 1991. Tree Automata, Mu-Calculus and Determinacy (Extended Abstract). In *32nd Annual Symposium on Foundations of Computer Science, San Juan, Puerto Rico, 1-4 October 1991*. IEEE Computer Society, 368–377. https://doi.org/10.1109/SFCS.1991.185392

[21] E. Allen Emerson, Charanjit S. Jutla, and A. Prasad Sistla. 2001. On model checking for the μ-calculus and its fragments. *Theor. Comput. Sci.* 258, 1-2 (2001), 491–522.

[22] Sebastian Enqvist, Helle Hvid Hansen, Clemens Kupke, Johannes Marti, and Yde Venema. 2019. Completeness for Game Logic. In *34th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2019, Vancouver, BC, Canada, June 24-27, 2019*. IEEE, 1–13. https://doi.org/10.1109/LICS.2019.8785676

[23] Sebastian Enqvist, Fatemeh Seifan, and Yde Venema. 2018. Completeness for the modal μ-calculus: Separating the combinatorics from the dynamics. *Theor. Comput. Sci.* 727 (2018), 37–100. https://doi.org/10.1016/j.tcs.2018.03.001

[24] Sebastian Enqvist, Fatemeh Seifan, and Yde Venema. 2019. Completeness for μ-calculi: A coalgebraic approach. *Ann. Pure Appl. Log.* 170, 5 (2019), 578–641. https://doi.org/10.1016/J.APAL.2018.12.004

[25] Alessandro Facchini, Yde Venema, and Fabio Zanasi. 2013. A Characterization Theorem for the Alternation-Free Fragment of the Modal µ-Calculus. In *Proceedings of the Twenty-Eighth Annual IEEE Symposium on Logic in Computer Science (LICS 2013)* (New Orleans, USA). IEEE Computer Society Press, 478–487.

[26] Nina Gierasimczuk, Lena Kurzen, and Fernando R. Velázquez-Quesada. 2009. Games for Learning: A Sabotage Approach. In *Proceedings of the Second Multi-Agent Logics, Languages, and Organisations Federated Workshops, Turin, Italy, September 7-10, 2009 (CEUR Workshop Proceedings, Vol. 494)*, Matteo Baldoni, Cristina Baroglio, Jamal Bentahar, Guido Boella, Massimo Cossentino, Mehdi Dastani, Barbara Dunin-Keplicz, Giancarlo Fortino, Marie-Pierre Gleizes, João Leite, Viviana Mascardi, Julian A. Padget, Juan Pavón, Axel Polleres, Amal El Fallah Seghrouchni, Paolo Torroni, and Rineke Verbrugge (Eds.). CEUR-WS.org. https://ceur-ws.org/Vol-494/famaspaper5.pdf

[27] Yuri Gurevich and Leo Harrington. 1982. Trees, Automata, and Games. In *Proceedings of the 14th Annual ACM Symposium on Theory of Computing, May 5-7, 1982, San Francisco, California, USA*, Harry R. Lewis, Barbara B. Simons, Walter A. Burkhard, and Lawrence H. Landweber (Eds.). ACM, 60–65. https://doi.org/10.1145/800070.802177

[28] Jaakko Hintikka. 1982. Game-theoretical semantics: insights and prospects. *Notre Dame Journal of Formal Logic* 23, 2 (1982), 219–241.

[29] David Janin and Igor Walukiewicz. 1996. On the Expressive Completeness of the Propositional mu-Calculus with Respect to Monadic Second Order Logic. In *CONCUR '96, Concurrency Theory, 7th International Conference, Pisa, Italy, August 26-29, 1996, Proceedings (LNCS, Vol. 1119)*, Ugo Montanari and Vladimiro Sassone (Eds.). Springer, 263–277. https://doi.org/10.1007/3-540-61604-7_60

[30] Johannes Kloibhofer. 2023. A note on the incompleteness of Afshari & Leigh's system Clo. https://doi.org/10.48550/arXiv.2307.06846 arXiv:2307.06846 [math.LO]

[31] Dexter Kozen. 1983. Results on the Propositional μ-Calculus. *Theor. Comput. Sci.* 27, 3 (1983), 333–354. https://doi.org/10.1016/0304-3975(82)90125-6

[32] Christof Löding and Philipp Rohde. 2003. Model Checking and Satisfiability for Sabotage Modal Logic. In *FST TCS 2003: Foundations of Software Technology and Theoretical Computer Science, 23rd Conference, Mumbai, India, December 15-17, 2003, Proceedings (Lecture Notes in Computer Science, Vol. 2914)*, Paritosh K. Pandya and Jaikumar Radhakrishnan (Eds.). Springer, 302–313. https://doi.org/10.1007/978-3-540-24597-1_26

[33] Markus Müller-Olm. 1999. A Modal Fixpoint Logic with Chop. In *STACS 99, 16th Annual Symposium on Theoretical Aspects of Computer Science, Trier, Germany, March 4-6, 1999, Proceedings (Lecture Notes in Computer Science, Vol. 1563)*, Christoph Meinel and Sophie Tison (Eds.). Springer, 510–520. https://doi.org/10.1007/3-540-49116-3_48

[34] Sara Negri. 2017. Proof Theory for Non-normal Modal Logics: The Neighbourhood Formalism and Basic Results. *FLAP* 4, 4 (2017). http://www.collegepublications.co.uk/downloads/ifcolog00013.pdf

[35] Damian Niwinski and Igor Walukiewicz. 1996. Games for the mu-Calculus. *Theor. Comput. Sci.* 163, 1&2 (1996), 99–116. https://doi.org/10.1016/0304-3975(95)00136-0

[36] Rohit Parikh. 1983. Propositional Game Logic. In *FOCS*. 195–200. https://doi.org/10.1109/SFCS.1983.47

[37] Marc Pauly and Rohit Parikh. 2003. Game Logic - An Overview. *Stud Logica* 75, 2 (2003), 165–182. https://doi.org/10.1023/A:1027354826364

[38] André Platzer. 2015. Differential Game Logic. *ACM Trans. Comput. Log.* 17, 1 (2015), 1. https://doi.org/10.1145/2817824

[39] Vaughan R. Pratt. 1981. A Decidable mu-Calculus: Preliminary Report. In *FOCS*. IEEE Computer Society, 421–427. https://doi.org/10.1109/SFCS.1981.4

[40] Vaughan R. Pratt. 1981. A Decidable mu-Calculus: Preliminary Report. In *22nd Annual Symposium on Foundations of Computer Science, Nashville, Tennessee, USA, 28-30 October 1981*. IEEE Computer Society, 421–427. https://doi.org/10.1109/SFCS.1981.4

[41] Philipp Rohde. 2006. On the μ-Calculus Augmented with Sabotage. In *Foundations of Software Science and Computation Structures, 9th International Conference, FOSSACS 2006, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2006, Vienna, Austria, March 25-31, 2006, Proceedings (Lecture Notes in Computer Science, Vol. 3921)*, Luca Aceto and Anna Ingólfsdóttir (Eds.). Springer, 142–156. https://doi.org/10.1007/11690634_10

[42] Colin Stirling. 1996. Games and Modal Mu-Calculus. In *Tools and Algorithms for Construction and Analysis of Systems, Second International Workshop, TACAS '96, Passau, Germany, March 27-29, 1996, Proceedings (Lecture Notes in Computer Science, Vol. 1055)*, Tiziana Margaria and Bernhard Steffen (Eds.). Springer, 298–312. https://doi.org/10.1007/3-540-61042-1_51

[43] Robert S. Streett and E. Allen Emerson. 1989. An Automata Theoretic Decision Procedure for the Propositional Mu-Calculus. *Inf. Comput.* 81, 3 (1989), 249–264.

[44] Johan van Benthem. 2005. An Essay on Sabotage and Obstruction. In *Mechanizing Mathematical Reasoning, Essays in Honor of Jörg H. Siekmann on the Occasion of His 60th Birthday (Lecture Notes in Computer Science, Vol. 2605)*, Dieter Hutter and Werner Stephan (Eds.). Springer, 268–276. https://doi.org/10.1007/978-3-540-32254-2_16

[45] Noah Abou El Wafa and André Platzer. 2022. First-Order Game Logic and Modal Mu-Calculus. arXiv:2201.10012 [cs.LO]

[46] Igor Walukiewicz. 1995. Completeness of Kozen's Axiomatisation of the Propositional mu-Calculus. In *Proceedings, 10th Annual IEEE Symposium on Logic in Computer Science, San Diego, California, USA, June 26-29, 1995*. IEEE Computer Society, 14–24. https://doi.org/10.1109/LICS.1995.523240

A THE RANK

We define the rank of a recursive game logic formula and a recursive game logic game by (simultaneous) structural induction on the formula

$$\begin{aligned}
\operatorname{rank}(P) &= 0\\
\operatorname{rank}(\neg P) &= 0\\
\operatorname{rank}(\varphi \vee \psi) &= \max\{\operatorname{rank}(\varphi), \operatorname{rank}(\psi)\} + 1\\
\operatorname{rank}(\varphi \wedge \psi) &= \max\{\operatorname{rank}(\varphi), \operatorname{rank}(\psi)\} + 1\\
\operatorname{rank}(\langle\alpha\rangle\varphi) &= \operatorname{rank}(\alpha) + \operatorname{rank}(\varphi) + 1\\
\operatorname{rank}(a) &= 0\\
\operatorname{rank}(a^{\mathrm{d}}) &= 0\\
\operatorname{rank}(x) &= 0\\
\operatorname{rank}(?\varphi) &= \operatorname{rank}(\varphi) + 2\\
\operatorname{rank}(?\varphi) &= \operatorname{rank}(\varphi) + 3\\
\operatorname{rank}(\alpha \cup \beta) &= \max\{\operatorname{rank}(\alpha), \operatorname{rank}(\beta)\} + 2\\
\operatorname{rank}(\alpha \cap \beta) &= \max\{\operatorname{rank}(\alpha), \operatorname{rank}(\beta)\} + 2\\
\operatorname{rank}(\alpha;\beta) &= \operatorname{rank}(\alpha) + \operatorname{rank}(\beta) + 2\\
\operatorname{rank}(\mathrm{r}x.\alpha) &= \operatorname{rank}(\alpha) + 1\\
\operatorname{rank}(\imath x.\alpha) &= \operatorname{rank}(\alpha) + 1
\end{aligned}$$

B FIXPOINT LEMMAS

For any valuation I and any $A \in \mathcal{P}(|\mathcal{N}|)$ we let $I[A]$ be the modified valuation with $I[A](x) = I(x)(A)$.

Lemma B.1. *Suppose φ is a* FLC *formula without composition and $A \in \mathcal{P}(|\mathcal{N}|)$. Then*

$$\mathcal{N}[\![\varphi]\!]^I(A) = \mathcal{N}[\![\varphi]\!]^{I[A]}(A).$$

Moreover

(1) $\mathcal{N}[\![\mu x.\varphi]\!]^I(A) = \mu B.(\mathcal{N}[\![\varphi]\!]^{I[x\mapsto\overline{B}]}(A))$
(2) $\mathcal{N}[\![\nu x.\varphi]\!]^I(A) = \nu B.(\mathcal{N}[\![\varphi]\!]^{I[x\mapsto\overline{B}]}(A))$

Proof. We prove this by induction on φ. The only interesting case is for formulas of the form $\mu x.\varphi$. We define the following monotone functions

$$\Delta_1(w) = \mathcal{N}[\![\varphi]\!]^{I[x\mapsto w]} \qquad \Delta_2(w) = \mathcal{N}[\![\varphi]\!]^{I[A][x\mapsto w]}$$

and prove that they satisfy the assumptions of Lemma 2.2. Two applications of the inductive hypothesis yield

$$\begin{aligned}
\Delta_1(w)(B) &= \mathcal{N}[\![\varphi]\!]^{I[x\mapsto w]}(B) = \mathcal{N}[\![\varphi]\!]^{I[x\mapsto w][B]}(B)\\
&= \mathcal{N}[\![\varphi]\!]^{I[x\mapsto\overline{w(B)}][B]}(B) = \mathcal{N}[\![\varphi]\!]^{I[x\mapsto\overline{w(B)}]}(B)\\
&= \Delta_1(\overline{w(B)})(B)
\end{aligned}$$

for all $w \in \mathcal{W}(|\mathcal{N}|)$ and all $B \in \mathcal{P}(|\mathcal{N}|)$. The case for Δ_2 is similar.

Note also that $I[A][x \mapsto \overline{B}] = I[x \mapsto \overline{B}][A]$ and hence by the induction hypothesis $\Delta_1(\overline{B})(A) = \Delta_2(\overline{B})(A)$ for all $B \in \mathcal{P}(|\mathcal{N}|)$.

By Lemma 2.2 we compute

$$\begin{aligned}
\mathcal{N}[\![\mu x.\varphi]\!]^I(A) &= \mu w.\Delta_1(w)(A) = \mu B.(\Delta_1(\overline{B})(A))\\
&= \mu B.(\Delta_2(\overline{B})(A)) = \mathcal{N}[\![\mu x.\varphi]\!]^{I[A]}(A).
\end{aligned}$$

The case for greatest fixpoints is symmetric. □

A similar result holds for right-linear game logic.

Lemma B.2. *Suppose φ is a formula and α a game of right-linear game logic and $A, B \in \mathcal{P}(|\mathcal{N}|)$. Then*

$$\mathcal{N}[\![\alpha]\!]^I(A) = \mathcal{N}[\![\alpha]\!]^{I[A]}(A).$$

Moreover

(1) $\mathcal{N}[\![\mathrm{r}x.\alpha]\!]^I(A) = \mu B.\mathcal{N}[\![\alpha]\!]^{I[x\mapsto\overline{B}]}(A)$
(2) $\mathcal{N}[\![\imath x.\alpha]\!]^I(A) = \nu B.\mathcal{N}[\![\alpha]\!]^{I[x\mapsto\overline{B}]}(A)$

Proof. We prove this by structural induction on formulas and games simultaneous. in which the duality operator a^{d} is only applied to atomic modalities a.

The interesting cases are for where α is a test, a composition or a fixpoint.

(1) If the game is of the form $?\varphi$ then

$$\mathcal{N}[\![?\varphi]\!]^I(A) = \mathcal{N}[\![\varphi]\!]^I \cap A = \mathcal{N}[\![\varphi]\!]^{I[A]} \cap A = \mathcal{N}[\![\varphi]\!]^{I[A]}(A),$$

since φ does not have any free variables by definition of right-linear games.

(2) If the game is of the form $\alpha;\beta$ then

$$\begin{aligned}
\mathcal{N}[\![\alpha;\beta]\!]^I(A) &= \mathcal{N}[\![\alpha]\!]^I(A) \circ \mathcal{N}[\![\beta]\!]^I(A)\\
&= \mathcal{N}[\![\alpha]\!]^{I[A]}(\mathcal{N}[\![\beta]\!]^I(A))\\
&= \mathcal{N}[\![\alpha]\!]^{I[A]}(\mathcal{N}[\![\beta]\!]^{I[A]}(A))\\
&= \mathcal{N}[\![\alpha;\beta]\!]^{I[A]}(A)
\end{aligned}$$

where the second equality uses that α does not have any free variables and the third equality is by induction hypothesis.

(3) If the game is of the form $\mathrm{r}x.\alpha$. We define the following monotone functions

$$\Delta_1(w) = \mathcal{N}[\![\alpha]\!]^{I[x\mapsto w]} \qquad \Delta_2(w) = \mathcal{N}[\![\alpha]\!]^{I[A][x\mapsto w]}$$

and prove that they satisfy the assumptions of Lemma 2.2. Two applications of the inductive hypothesis yield

$$\begin{aligned}
\Delta_1(w)(B) &= \mathcal{N}[\![\alpha]\!]^{I[x\mapsto w]}(B) = \mathcal{N}[\![\alpha]\!]^{I[x\mapsto w][B]}(B)\\
&= \mathcal{N}[\![\alpha]\!]^{I[x\mapsto\overline{w(B)}][B]}(B) = \mathcal{N}[\![\alpha]\!]^{I[x\mapsto\overline{w(B)}]}(B)\\
&= \Delta_1(\overline{w(B)})(B)
\end{aligned}$$

for all $w \in \mathcal{W}(|\mathcal{N}|)$ and all $B \in \mathcal{P}(|\mathcal{N}|)$. The case for Δ_2 is similar.

Note also that $I[A][x \mapsto \overline{B}] = I[x \mapsto \overline{B}][A]$ and hence by the induction hypothesis $\Delta_1(\overline{B})(A) = \Delta_2(\overline{B})(A)$ for all $B \in \mathcal{P}(|\mathcal{N}|)$.

By Lemma 2.2 we compute

$$\begin{aligned}
\mathcal{N}[\![\mu x.\alpha]\!]^I(A) &= \mu w.\Delta_1(w)(A) = \mu B.(\Delta_1(\overline{B})(A))\\
&= \mu B.(\Delta_2(\overline{B})(A)) = \mathcal{N}[\![\mu x.\alpha]\!]^{I[A]}(A).
\end{aligned}$$

The case for greatest fixpoints is analogous. □

Lemma B.3. *Suppose ψ is an* FLC *formula with no free variables other than x and in which x is not bound. Then $\psi \equiv \psi\frac{\mathrm{id}}{x} \circ x$.*

Proof. By induction on the formula ψ. The only interesting case is for formulas of the form $\mu y.\psi$. By Lemma B.1

$$\begin{aligned}\mathcal{N}[\![(\mu y.\psi\tfrac{\mathrm{id}}{x})\circ x]\!]^I(A) &= \mu B.(\mathcal{N}[\![\psi\tfrac{\mathrm{id}}{x}]\!]^{I[y\mapsto\overline{B}]}(\mathcal{N}[\![x]\!]^I(A)))\\ &= \mu B.(\mathcal{N}[\![\varphi]\!]^{I[y\mapsto\overline{B}]}(A))\\ &= \mathcal{N}[\![\mu y.\psi]\!]^I(A)\end{aligned}$$

where the second equality is by the induction hypothesis. □

C DERIVED AXIOMS FOR $\mathsf{GL_s}$

To facilitate proofs in $\mathsf{GL_s}$ we use some derived axioms, which are immediate consequences of the original axioms of $\mathsf{GL_s}$ for convenience.

(W) $\langle\sim a;a^{\mathrm{d}}\rangle\bot$

(C) $\langle\sim a\rangle\langle a\rangle\varphi \leftrightarrow \langle\sim a\rangle\varphi$

(⅄) $\langle\sim a\rangle(\varphi\vee\psi) \leftrightarrow \langle\sim a\rangle\varphi \vee \langle\sim a\rangle\psi$

(⍲) $\langle\sim a\rangle(\varphi\wedge\psi) \leftrightarrow \langle\sim a\rangle\varphi \wedge \langle\sim a\rangle\psi$

(θ) $\langle\sim a\rangle\varphi \leftrightarrow \varphi$ (if a, a^{d} not in φ)

(P) $\langle\alpha;\sim a\rangle\varphi \leftrightarrow \langle\sim a;\alpha\rangle\varphi$ ($\{a, a^{\mathrm{d}}, \sim a, \sim a^{\mathrm{d}}\}$ not in α)

(≈) $\langle\sim a;\sim a\rangle\varphi \leftrightarrow \langle\sim a\rangle\varphi$

(ℵ) $\langle\sim a;\sim a^{\mathrm{d}}\rangle\varphi \leftrightarrow \langle\sim a^{\mathrm{d}}\rangle\varphi$

Proof. Axiom W is derived from the ∼ instance

$$\langle\sim a\rangle\langle x\tfrac{a^{\mathrm{d}}}{x}\rangle\bot \leftrightarrow \langle x\tfrac{!\bot}{x}\rangle\bot.$$

Axiom C is exactly the instance

$$\langle\sim a\rangle\langle x\tfrac{a}{x}\rangle\varphi \leftrightarrow \langle x\tfrac{\sim a}{x}\rangle\varphi$$

of ∼. For ⅄ use the ∼-instance

$$\langle\sim a\rangle\langle(x\cup y)\tfrac{?\varphi}{x}\tfrac{?\psi}{y}\rangle\top \leftrightarrow \langle(x\cup y)\tfrac{\sim a;?\varphi}{x}\tfrac{\sim a;?\psi}{y}\rangle\top$$

and similarly for ⍲. Axiom θ is derived from the instance

$$\langle\sim a\rangle\langle?\top\rangle\varphi \leftrightarrow \langle?\top\rangle\varphi$$

or ∼. For axiom P use the instance

$$\langle\sim a\rangle\langle(\alpha;x)\tfrac{?\varphi}{x}\rangle\top \leftrightarrow \langle(\alpha;x)\tfrac{\sim a;?\varphi}{x}\rangle\top$$

of ∼. Axiom ≈ and ℵ are instances of ≃ with $\alpha\equiv x;y$. □

D PROOFS

Proof of Lemma 2.2. We prove the case for the least fixpoint. Let $w=\mu u.\Delta(u)$. For the $\subseteq$ inclusion we pick any $B\in\mathcal{P}(X)$ with $\Delta(\overline{B})(r)\subseteq B$ and show that $w(A)\subseteq B$. Define $u\in\mathcal{W}(X)$

$$u(C)=\begin{cases}B & \text{if } C=A\\ X & \text{otherwise.}\end{cases}$$

Note that $\Delta(u)\subseteq u$ because

$$\Delta(u)(A)=\Delta(\overline{u(A)})(A)=\Delta(\overline{B})(A)=B=u(A).$$

Hence $w\subseteq u$ and in particular $w(A)\subseteq u(A)=B$.

For the $\supseteq$ inclusion note that $\Delta(w)\subseteq w$. Hence

$$\Delta(\overline{w(A)})(A)=\Delta(w)(A)\subseteq w(A)$$

for all A. □

Proof of Lemma 3.4. For games of the form $\sim\alpha$ observe

$$\begin{aligned}\mathcal{N}[\![(\sim a)^{\mathrm{d}}]\!]_{\mathrm{s}}(A^{\mathsf{C}}) &= \{(\omega,c):(\omega,c\tfrac{\Box}{a})\in A^{\mathsf{C}}\}\\ &= \{(\omega,c):(\omega,\overline{c}\tfrac{\Diamond}{a})\notin A\}\\ &= \{(\omega,c):(\omega,c\tfrac{\Diamond}{a})\in A\}^{\mathsf{C}} = (\mathcal{N}[\![\sim a]\!]_{\mathrm{s}}(A))^{\mathsf{C}}\end{aligned}$$

The general claim is proved by mutual induction on the definition of formulas and games. We do the interesting case for games of the formα^*. Observe

$$\begin{aligned}\mathcal{N}[\![\alpha^*]\!]_{\mathrm{s}}{}^{\mathsf{D}}(A) &= \mathcal{N}[\![\alpha^*]\!]_{\mathrm{s}}(A^{\mathsf{C}})^{\mathsf{C}} = (\mathcal{N}[\![\alpha]\!]_{\mathrm{s}}(\mathcal{N}[\![\alpha^*]\!]_{\mathrm{s}}(A^{\mathsf{C}}))\cup A^{\mathsf{C}})^{\mathsf{C}}\\ &= \mathcal{N}[\![\alpha^{\mathrm{d}}]\!]_{\mathrm{s}}(\mathcal{N}[\![\alpha^*]\!]_{\mathrm{s}}(A^{\mathsf{C}})^{\mathsf{C}})\cap A\\ &= \mathcal{N}[\![\alpha^{\mathrm{d}}]\!]_{\mathrm{s}}(\mathcal{N}[\![\alpha^*]\!]_{\mathrm{s}}{}^{\mathsf{D}}(A))\cap A\end{aligned}$$

Hence by maximality $\mathcal{N}[\![\alpha^*]\!]_{\mathrm{s}}{}^{\mathsf{D}}(A)\subseteq\mathcal{N}[\![\alpha^{\mathrm{d}\times}]\!]_{\mathrm{s}}(A)$.

For the reverse inclusion note

$$\begin{aligned}\mathcal{N}[\![\alpha^{\mathrm{d}\times}]\!]_{\mathrm{s}}(A)^{\mathsf{C}} &= (\mathcal{N}[\![\alpha^{\mathrm{d}}]\!]_{\mathrm{s}}(\mathcal{N}[\![\alpha^{\mathrm{d}\times}]\!]_{\mathrm{s}}(A))\cap A)^{\mathsf{C}}\\ &= \mathcal{N}[\![\alpha]\!]_{\mathrm{s}}(\mathcal{N}[\![\alpha^{\mathrm{d}\times}]\!]_{\mathrm{s}}(A^{\mathsf{C}})^{\mathsf{C}})\cup A^{\mathsf{C}}\\ &= \mathcal{N}[\![\alpha]\!]_{\mathrm{s}}(\mathcal{N}[\![\alpha^{\mathrm{d}\times}]\!]_{\mathrm{s}}{}^{\mathsf{D}}(A))\cup A^{\mathsf{C}}\end{aligned}$$

Hence by minimality $\mathcal{N}[\![\alpha]\!]_{\mathrm{s}}(A^{\mathsf{C}})\subseteq\mathcal{N}[\![\alpha^{\mathrm{d}\times}]\!]_{\mathrm{s}}(A)^{\mathsf{C}}$. The required $\mathcal{N}[\![\alpha^*]\!]_{\mathrm{s}}{}^{\mathsf{D}}(A)\supseteq\mathcal{N}[\![\alpha^{\mathrm{d}\times}]\!]_{\mathrm{s}}(A)$ follows by taking $^{\mathsf{C}}$. □

Proof of Lemma 3.5. We do the cases for tests $?\varphi$ and fixpoints $\mathrm{r}x.\alpha$ in a mutual induction on formulas and games explicitly. For a test

$$\begin{aligned}(\mathcal{N}[\![?\varphi]\!]^{I^{\mathrm{d}}})^{\mathrm{d}}(A) &= |\mathcal{N}|\setminus(\mathcal{N}[\![?\varphi]\!]^{I^{\mathrm{d}}}(|\mathcal{N}|\setminus A))\\ &= |\mathcal{N}|\setminus(\mathcal{N}[\![\varphi]\!]^{I^{\mathrm{d}}}\cap|\mathcal{N}|\setminus A)\\ &= |\mathcal{N}|\setminus(\mathcal{N}[\![\varphi]\!]^{I^{\mathrm{d}}})\cup A\\ &= \mathcal{N}[\![\overline{?\varphi}]\!]^I(A)\end{aligned}$$

The last equality holds, since any formula appearing in a test does not have free variables by definition.

Consider games of the form $\mathrm{r}x.\alpha$.

$$\begin{aligned}(\mathcal{N}[\![\mathrm{r}x.\alpha]\!]^{I^{\mathrm{d}}})^{\mathrm{d}} &= (\mu w.\mathcal{N}[\![\alpha]\!]^{I^{\mathrm{d}}[x\mapsto w]})^{\mathrm{d}}\\ &= \nu w.(\mathcal{N}[\![\alpha]\!]^{I[x\mapsto w]^{\mathrm{d}}})^{\mathrm{d}}\\ &= \nu w.\mathcal{N}[\![\alpha^{\mathrm{d}}]\!]^{I[x\mapsto w]}\\ &= \mathcal{N}[\![(\mathrm{r}x.\alpha)^{\mathrm{d}}]\!]^{I[x\mapsto w]}\end{aligned}$$

□

Proof of Proposition 5.2. This is shown by a mutual induction on formulas φ and games α of RGL. Most cases of the induction are straightforward. We do the interesting cases.

First we consider formulas of the form $\langle\alpha\rangle\psi$.

$$\begin{aligned}\mathcal{N}[\![(\langle\alpha\rangle\psi)^\flat]\!]^I(A) &= \mathcal{N}[\![\alpha^\flat \tfrac{\psi^\flat}{\mathsf{u},\mathsf{v}}]\!]^I(A) = \mathcal{N}[\![\alpha^\flat]\!]^{I[\mathsf{u},\mathsf{v}\mapsto\mathcal{N}[\![\varphi^\flat]\!]^I]}(A)\\ &= \mathcal{N}[\![\alpha]\!]^I(\mathcal{N}[\![\varphi^\flat]\!]^I(A)) = \mathcal{N}[\![\alpha]\!]^I(\mathcal{N}[\![\varphi]\!]^I)\\ &= \mathcal{N}[\![\langle\alpha\rangle\psi]\!]^I\end{aligned}$$

The well-namedness assumption is used to ensure that the substitution does not capture free variables. The case for the games of the form $\alpha;\beta$ is very similar, as it is a similar composition.

For a game of the form x observe

$$\mathcal{N}[\![x]\!]^I \circ w = I(x) \circ w = \mathcal{N}[\![x \circ \mathsf{v}]\!]^{I[\mathsf{u},\mathsf{v}\mapsto w]} = \mathcal{N}[\![x^\flat]\!]^{I[\mathsf{u},\mathsf{v}\mapsto w]}.$$

Finally we also consider the case of games $\mathsf{r}x.\alpha$. with the induction hypothesis we compute

$$\begin{aligned}\mathcal{N}[\![(\mathsf{r}x.\alpha)^\flat]\!]^{I[\mathsf{u},\mathsf{v}\mapsto w]} &= \mathcal{N}[\![\mathsf{r}x.\alpha \tfrac{\mathrm{id}}{\mathsf{u},\mathsf{v}}{}^\flat \circ \mathsf{u}]\!]^{I[\mathsf{u},\mathsf{v}\mapsto w]}\\ &= (\mu w.\mathcal{N}[\![\varphi^\flat \tfrac{\mathrm{id}}{\mathsf{u},\mathsf{v}}]\!]^{I[\mathsf{u},\mathsf{v}\mapsto\mathrm{id}][x\mapsto w]}) \circ w\\ &= (\mu w.\mathcal{N}[\![\varphi^\flat]\!]^{I[\mathsf{u},\mathsf{v}\mapsto\mathrm{id}][x\mapsto w]}) \circ w\\ &= (\mu w.\mathcal{N}[\![\varphi]\!]^{I[x\mapsto w]} \circ \mathrm{id}) \circ w\\ &= \mathcal{N}[\![\mathsf{r}x.\alpha]\!]^I \circ w\end{aligned}$$ □

Proof of Proposition 5.4. For the purposes of this proof we call a valuation I *constant* if $I(x)$ is constant for all x except u and v. We prove by structural induction on *all* well-named normal form rlGL formula φ all well-named rlGL games α that

$$\mathcal{N}[\![\varphi^\natural]\!]^I = \mathcal{N}[\![\varphi^\flat]\!]^I \qquad \text{and} \qquad \mathcal{N}[\![\alpha^\natural]\!]^I = \mathcal{N}[\![\alpha^\flat]\!]^I.$$

for all constant valuations I.

Most cases of the induction are straightforward. For formulas of the form $\langle\alpha\rangle\varphi$

$$\begin{aligned}\mathcal{N}[\![(\langle\alpha\rangle\varphi)^\natural]\!]^I &= \mathcal{N}[\![\alpha^\natural \tfrac{\varphi^\natural}{\mathsf{u}}]\!]^I = \mathcal{N}[\![\alpha^\natural]\!]^{I[\mathsf{u},\mathsf{v}\mapsto\mathcal{N}[\![\varphi^\natural]\!]^I]}\\ &= \mathcal{N}[\![\alpha^\flat]\!]^{I[\mathsf{u},\mathsf{v}\mapsto\mathcal{N}[\![\varphi^\flat]\!]^I]} = \mathcal{N}[\![(\langle\alpha\rangle\varphi)^\flat]\!]^I\end{aligned}$$

The well-namedness property of the formula ensures that α does not bind a variable that is free in φ, so that the substitution above does not capture variables. For variables x note that

$$\mathcal{N}[\![x^\natural]\!]^I = I(x) = I(x) \circ I(\mathsf{v}) = \mathcal{N}[\![x \circ \mathsf{v}]\!]^I = \mathcal{N}[\![x^\flat]\!]^I$$

because I is constant. The argument for games of the form $\alpha;\beta$ is similar.

Finally we also consider the case of games $vx.\alpha$. Using Lemmas B.1 and B.2 we can compute the fixpoint pointwise

$$\begin{aligned}\mathcal{N}[\![(vx.\alpha)^\natural]\!]^I(A) &= \mathcal{N}[\![vx.\alpha^\natural]\!]^I(A) = \mu B.(\mathcal{N}[\![\alpha^\natural]\!]^{I[x\mapsto\overline{B}]}(A))\\ &= \mu B.(\mathcal{N}[\![\alpha^\flat]\!]^{I[x\mapsto\overline{B}]}(A))\\ &= \mu B.(\mathcal{N}[\![\alpha]\!]^{I[x\mapsto\overline{B}]}(I(\mathsf{u})(A)))\\ &= \mathcal{N}[\![vx.\alpha]\!]^I(I(\mathsf{u})(A))\\ &= \mathcal{N}[\![(vx.\alpha)^\flat]\!]^I(A)\end{aligned}$$

The third equality is by the induction hypothesis. The fourth and the sixth equalities are by (2) of Proposition 5.2.

The moreover follows with Proposition 5.2. □

Proof of Lemma 5.6. (1) The only way the $\sharp$ translation can give rise to a non-GL formula is through translation of fixpoints. The translation of the $*$ fixpoints in L_* formulas are clearly into repetition games.

(2) First observe that if φ is separable then $\varphi\frac{\psi}{\mathsf{u}}$ is also separable if the free variables of ψ are never bound in φ and u is not bound in φ.
We prove by simultaneous structural induction that for any formula φ and any game α of GL the translations $\varphi^\natural$ and $\alpha^\natural$ are separable L_μ formulas.
Most cases are straightforward for formulas $\langle\alpha\rangle\varphi$ and games $\alpha;\beta$ we use the induction hypothesis and the observation above, which applies by the assumption that φ is well-named.
The most interesting case is for games α is of the form $\alpha^* = \mathsf{r}x.(?\top \cup \alpha; x)$, where x is not in α. The translation is

$$(\alpha^*)^\natural = \mu x.(\mathsf{u} \vee \alpha^\natural \tfrac{x}{\mathsf{u}}).$$

Because α has no free variables as a GL game only u is free in $\alpha^\natural$. Hence the translation is separable.

(3) We prove by structural induction on a formula of the modal μ-calculus that provided it is separable, there is an equivalent L_* formula. The only interesting case is for fixpoint formulas. Consider a separable formula φ. Pick separable formulas ψ, ρ such that $\varphi \leftrightarrow \mu x.(\rho \vee \psi)$ where x is not free in ρ, only x is free in ψ. By renaming we ensure that x is not bound in ψ. Note that by Lemma B.3 semantically $\psi \equiv \psi\frac{\mathrm{id}}{x} \circ x$.
By the induction hypothesis pick L_* formulas ψ', ρ' equivalent to ψ, ρ respectively. We claim that φ is equivalent to the L_* formula $(\psi'\frac{\mathrm{id}}{x})^* \circ \rho'$. Semantically

$$\begin{aligned}(\psi'\tfrac{\mathrm{id}}{x})^* \circ \rho' &\equiv \mu x.(\mathrm{id} \vee \psi'\tfrac{\mathrm{id}}{x} \circ x) \circ \rho\\ &\equiv \mu x.(\mathrm{id} \vee \psi\tfrac{\mathrm{id}}{x} \circ x) \circ \rho\\ &\equiv \mu x.(\mathrm{id} \vee \psi) \circ \rho\end{aligned}$$

Because $\mathrm{id} \vee \psi$ is a modal μ-calculus formula without composition by Lemma B.3 we compute

$$\begin{aligned}&\mathcal{N}[\![\mu x.(\mathrm{id} \vee \psi) \circ \rho]\!]^I(A)\\ =&\mu B.(\mathcal{N}[\![\mathrm{id} \vee \psi]\!]^{I[x\mapsto\overline{B}]}(\mathcal{N}[\![\rho]\!]^I(A)))\\ =&\mu B.(\mathcal{N}[\![\rho]\!]^I(A) \cup \mathcal{N}[\![\psi\tfrac{\mathrm{id}}{x} \circ x]\!]^{I[x\mapsto\overline{B}]}(\mathcal{N}[\![\rho]\!]^I(A)))\\ =&\mu B.(\mathcal{N}[\![\rho]\!]^{I[x\mapsto\overline{B}]}(A) \cup \mathcal{N}[\![\psi\tfrac{\mathrm{id}}{x} \circ x]\!]^{I[x\mapsto\overline{B}]}(A))\\ =&\mu B.(\mathcal{N}[\![\rho \vee \psi]\!]^{I[x\mapsto\overline{B}]}(A))\\ =&\mathcal{N}[\![\mu x.(\rho \vee \psi)]\!]^I(A)\end{aligned}$$

The third equality holds because x is not free in ρ and by the fact that $\mathcal{N}[\![x]\!]^{I[x\mapsto\overline{B}]}(C) = B$ is constant. □

Proof of Lemma 6.5. Soundness of the common part of the proof calculus goes through exactly as for game logic [36].

We say a set $A \subseteq |\mathcal{N}| \times C$ is a-invariant if

$$\mathcal{N}[\![\sim a]\!]_{\mathsf{s}}(A) = \mathcal{N}[\![\sim a^\mathsf{d}]\!]_{\mathsf{s}}(A) = A.$$

Observe that $\mathcal{N}[\![\varphi]\!]_{\mathsf{s}}$ is $\sim a$-invariant as it does not mention a, and a^{d}. Hence it suffices to show

$$\mathcal{N}[\![\sim a]\!]_{\mathsf{s}}(\mathcal{N}[\![\alpha \tfrac{a;\vec{\beta}}{\vec{x}} \tfrac{a^{\mathrm{d}};\vec{\gamma}}{\vec{y}} \tfrac{\vec{\delta}}{\vec{z}}]\!]_{\mathsf{s}}(A)) = \mathcal{N}[\![\alpha \tfrac{\sim a;\vec{\beta}}{\vec{x}} \tfrac{!\bot}{\vec{y}} \tfrac{\sim a;\vec{\delta}}{\vec{z}}]\!]_{\mathsf{s}}(A)$$

for all $\sim a$-invariant sets A and all α satisfying †. We prove this by induction on α. Most cases of the induction are routine. If α is a loop this is immediate by $\sim a$-invariance, since α does not contain variables.

For $\uplus$ let

$$E = |\mathcal{N}| \times \bigcup_{i=1}^{n} \{c \in C : c(a_i) = \diamond, \forall k \neq i\ c(a_k) = \Box, \\ \forall k \leq j\ c(b_k) = \diamond, \forall k > j\ c(b_k) = \Box\}$$

where $\beta_i \equiv \sim b_1^i; \ldots \sim b_j^i; \sim b_{j+1}^{i\,\mathrm{d}}; \ldots \sim b_m^{i\,\mathrm{d}}$. It is easy to see by induction on α that

$$\begin{aligned}
&\mathcal{N}[\![\alpha \tfrac{(a_1;\beta_1 \cup \ldots \cup a_n;\beta_n);\beta}{w}]\!]_{\mathsf{s}}(U) \cap E \\
&= \mathcal{N}[\![\alpha \tfrac{(a_1;\beta_1 \cup \ldots \cup a_n;\beta_n);\beta}{w}]\!]_{\mathsf{s}}(U \cap E) \cap E \\
&= \mathcal{N}[\![\alpha \tfrac{\beta}{w}]\!]_{\mathsf{s}}(U \cap E) \cap E \\
&= \mathcal{N}[\![\alpha \tfrac{\beta}{w}]\!]_{\mathsf{s}}(U) \cap E
\end{aligned}$$

for all U.

For $\simeq$ observe by induction on α that $\mathcal{N}[\![\alpha \tfrac{\vec{\eta}}{\vec{x}} \tfrac{\vec{\delta};\beta}{\vec{y}}]\!]_{\mathsf{s}}$ is a function that maps a-invariant sets to a invariant sets. Again by induction on α now observe that it equals $\mathcal{N}[\![\alpha \tfrac{?\top}{\vec{x}} \tfrac{\vec{\delta};\beta}{\vec{y}}]\!]_{\mathsf{s}}$ on a-invariant sets.

For $\sim_1$ let $E = \{U \subseteq |\mathcal{N}| \times C : \text{if } (\omega, c) \in U \text{ and } c(a) = \Box \text{ then } (\omega, c\tfrac{\diamond}{b} \in U) \Leftrightarrow (\omega, c\tfrac{\Box}{b} \in U)\}$. By induction on α simultaneously prove that

$$\mathcal{N}[\![\alpha \tfrac{a;\beta}{x} \tfrac{\sim a^{\mathrm{d}}}{y}]\!]_{\mathsf{s}} = \mathcal{N}[\![\alpha \tfrac{a;\beta}{x} \tfrac{\sim a^{\mathrm{d}};\eta}{y}]\!]_{\mathsf{s}}$$

as functions from E to itself.

For $\approx$ let $E = |\mathcal{N}| \times \{c \in C : c(a) \neq \emptyset\}$. By induction on α it is straightforward to prove the following equalities

$$\begin{aligned}
\mathcal{N}[\![\alpha \tfrac{a}{x}]\!]_{\mathsf{s}}(U) \cap E &= \mathcal{N}[\![\alpha \tfrac{a}{x}]\!]_{\mathsf{s}}(U \cap E) \cap E \\
&= \mathcal{N}[\![\alpha \tfrac{a;\sim a}{x}]\!]_{\mathsf{s}}(U \cap E) \cap E \\
&= \mathcal{N}[\![\alpha \tfrac{a;\sim a}{x}]\!]_{\mathsf{s}}(U) \cap E
\end{aligned}$$

for all U. □

Proof of Lemma 6.6. For (1) we prove only the backward implication. The forward direction will later follow from Theorem 6.8 and corollary 5.3 and unlike the backward implication is not required for the proof of Theorem 6.8. We prove the implication for all ρ by induction on the rank of φ. We distinguish based on the shape of the formula.

If φ is a proposition constant of the form P then we can prove

$$\mathsf{rIGL} \vdash \langle ?P; !\bot \rangle \to P$$

by ;, ? and !. Similarly for $\neg P$. If φ is a conjunction the equivalence is an instance of $\cup$. If φ is of the form $\langle\alpha\rangle\psi$ we distinguish on the shape of α:

(1) Case a. The desired implication

$$\mathsf{rIGL} \vdash \langle a \rangle \psi^{\mathrm{d}\sharp} \to \langle a \rangle \psi$$

is derivable with an application of M_G from the induction hypothesis $\mathsf{rIGL} \vdash \psi^{\mathrm{d}\sharp} \to \psi$. Similarly for a^{d} and variables x.

(2) Case $\beta;\gamma$. Then by induction hypothesis since $\langle\beta\rangle\langle\gamma\rangle\psi$ is of lower rank

$$\mathsf{rIGL} \vdash (\langle\beta\rangle\langle\gamma\rangle\psi)^{\mathrm{d}\sharp} \to \langle\beta\rangle\langle\gamma\rangle\psi.$$

Since $(\langle\beta\rangle\langle\gamma\rangle\psi)^{\mathrm{d}} \equiv (\langle\beta;\gamma\rangle\psi)^{\mathrm{d}}$ the implication is derivable with a use of ;.

(3) Case $?\rho$. Observe that $(\langle?\rho\rangle\psi)^{\mathrm{d}} = (\rho \wedge \psi)^{\mathrm{d}}$ and $\mathsf{rIGL} \vdash \langle?\rho\rangle\psi \leftrightarrow \rho \wedge \psi$. Hence the implication follows by induction hypothesis on the lower rank formula $\rho \wedge \psi$. Analogously for $!\rho$.

(4) Case $\beta \cup \gamma$. The induction hypothesis on the lower rank formula $\langle\alpha\rangle\psi \wedge \langle\beta\rangle\psi$ can be used by $\cup$ because $(\langle\alpha\rangle\psi \wedge \langle\beta\rangle\psi)^{\mathrm{d}} \leftrightarrow (\langle\alpha \cup \beta\rangle\psi)^{\mathrm{d}}$. The case $\beta \cap \gamma$ is similar.

(5) Case $\mathsf{r}x.\alpha$. Note that by the inductive hypothesis applied to the lower rank formula $\langle\alpha\rangle\psi$

$$\mathsf{rIGL} \vdash \langle \alpha^{\mathrm{d}\sharp} \tfrac{\psi^{\mathrm{d}\sharp}}{\mathsf{u}} \rangle \bot \to \langle\alpha\rangle\psi$$

By substituting x by $\mathsf{r}x.\alpha; ?\psi; !\bot$

$$\mathsf{rIGL} \vdash \langle \alpha^{\mathrm{d}\sharp} \tfrac{\psi^{\mathrm{d}\sharp}}{\mathsf{u}} \tfrac{\mathsf{r}x.\alpha;?\psi;!\bot}{x} \rangle \bot \to \langle \alpha \tfrac{\mathsf{r}x.\alpha;?\psi;!\bot}{x} \rangle \psi$$

and an application of RL and fp_G yields

$$\mathsf{rIGL} \vdash \langle \alpha^{\mathrm{d}\sharp} \tfrac{\psi^{\mathrm{d}\sharp}}{\mathsf{u}} \tfrac{\mathsf{r}x.\alpha;?\psi;!\bot}{x} \rangle \bot \to \langle \mathsf{r}x.\alpha \rangle \psi$$

The implication follows with an application of μ_G, since the translation of $\langle \mathsf{r}x.\alpha \rangle \psi$ is

$$(\langle \mathsf{r}x.\alpha \rangle \psi)^{\mathrm{d}\sharp} \equiv \langle \mu x. \alpha^{\mathrm{d}\sharp} \tfrac{\psi^{\mathrm{d}\sharp}}{\mathsf{u}} \rangle \bot.$$

(6) Case $\imath x.\alpha$. By definition of the translations

$$\mathsf{rIGL} \vdash (\langle \imath x.\alpha \rangle \psi)^{\mathrm{d}\sharp} \to \langle \alpha^{\mathrm{d}\sharp} \tfrac{\psi^{\mathrm{d}\sharp}}{\mathsf{u}} \tfrac{(\langle \imath x.\alpha \rangle \psi)^{\mathrm{d}\sharp}}{x} \rangle \bot$$

is a consequence of rfp_G and consequently

$$\mathsf{rIGL} \vdash (\langle \imath x.\alpha \rangle \psi)^{\mathrm{d}\sharp} \to \langle \alpha^{\mathrm{d}\sharp} \tfrac{\psi^{\mathrm{d}\sharp}}{\mathsf{u}} \tfrac{(\langle \imath x.\alpha \rangle \psi)^{\mathrm{d}\sharp};?\bot;!\bot}{x} \rangle \bot$$

by RL. By the induction hypothesis applied to α and substituting x by $(\langle \imath x.\alpha \rangle \psi)^{\mathrm{d}\sharp}; ?\bot; !\bot$ this implies

$$\mathsf{rIGL} \vdash (\langle \imath x.\alpha \rangle \psi)^{\mathrm{d}\sharp} \to \langle \alpha \tfrac{(\langle \imath x.\alpha \rangle \psi)^{\mathrm{d}\sharp}}{x} \rangle \psi$$

Hence by ν_G the implication follows.

The equivalence in (2) follows immediately from Proposition 6.1 and Corollary 5.3 □

Lemma D.1. $\mathsf{rIGL} \vdash \overline{\varphi}^{\sharp} \leftrightarrow \overline{\varphi^{\sharp}}$ *for φ a closed L_μ formula.*

Proof of Lemma D.1. We prove the equivalence by induction on the formula φ.

(1) If the formula is of the form id, then we need to prove the equivalence

$$\mathsf{rlGL} \vdash \langle ?\top; !\bot\rangle\bot \leftrightarrow \langle !\bot\rangle\top$$

This is immediate from ;, $\cup$, $\cap$

(2) If the formula is of the form P we prove the equivalence

$$\mathsf{rlGL} \vdash \langle ?\neg P; !\bot\rangle\bot \leftrightarrow \langle !P; ?\bot\rangle\top$$

by ;,? and !.

(3) If the formula is of the form $\varphi \vee \psi$ the equivalence follows from the induction hypothesis with $\cap$. Similarly for formulas of the form $\varphi \wedge \psi$

(4) If the formula is of the form $\langle a\rangle\varphi$ the equivalence follows from the induction hypothesis with ; and an application of M_G.

(5) If the formula is of the form $\mu x.\psi$. The equivalence we need to prove is $\mathsf{rlGL} \vdash \langle \imath x.\overline{\psi}^\sharp\rangle\bot \leftrightarrow \langle \imath x.\psi^{\sharp^\mathrm{d}}\rangle\top$.

For the forward direction $\mathsf{rlGL} \vdash \langle \imath x.\overline{\psi}^\sharp\rangle\bot \to \langle \overline{\psi}^\sharp \frac{\imath x.\overline{\psi}^\sharp}{x}\rangle\bot$ is an instance of rfp_G. By induction hypothesis

$$\mathsf{rlGL} \vdash \langle \imath x.\overline{\psi}^\sharp\rangle\bot \to \langle \psi^{\sharp^\mathrm{d}} \frac{\imath x.\overline{\psi}^\sharp}{x}\rangle\top$$

An application of RL allows to deduce the desired implication with ν_G. The reverse implication is analogous. □

Lemma D.2. *For closed L_μ formulas φ, if $\mathsf{mL}_\mu \vdash \varphi$ then $\mathsf{rlGL} \vdash \varphi^\sharp$.*

Proof of Lemma D.2. First note that

$$\mathsf{rlGL} \vdash (\varphi \vee \psi)^\sharp \leftrightarrow \varphi^\sharp \vee \psi^\sharp \qquad \mathsf{rlGL} \vdash (\varphi \wedge \psi)^\sharp \leftrightarrow \varphi^\sharp \wedge \psi^\sharp$$

Indeed these are instances of $\cup$ and $\cap$ respectively. By Lemma D.1 $\mathsf{rlGL} \vdash \neg P^\sharp \leftrightarrow (\neg P)^\sharp$. Thus if φ is a propositional tautology $\mathsf{rlGL} \vdash \varphi^\sharp$ is provably reducible to a propositional tautology and therefore provable in $\mathsf{rlGL}\vdash$.

Note also by Lemma D.1

$$(\sharp)\ (\varphi \to \psi)^\sharp \leftrightarrow (\varphi^\sharp \to \psi^\sharp)$$

is an instance of $\cup$ after expanding the abbreviations.

From $\sharp$ it follows that the $\sharp$-translation of any instance of fp is provably equivalent to an instance of fp_G. Hence the translations of all axioms of the monotone modal μ-calculus are provable in right-linear game logic. We now proceed to show the claim by induction on the length of the proof.

If the last step of the proof of $\mathsf{mL}_\mu \vdash \psi$ is an instance of MP of the kind

$$\mathrm{M}_a\frac{\varphi \qquad \varphi \to \psi}{\psi} \qquad \text{then} \qquad \mathrm{M}_G\frac{\varphi^\sharp \qquad \varphi^\sharp \to \psi^\sharp}{\psi^\sharp}$$

is an instance of MP_G. Because $\sharp$ provably distributes over implications we can derive

$$\mathrm{M}_G\frac{\varphi^\sharp \qquad \sharp,\mathrm{MP}_G\dfrac{(\varphi \to \psi)^\sharp}{\varphi^\sharp \to \psi^\sharp}}{\psi^\sharp}$$

If the last step of a proof of $\mathsf{mL}_\mu \vdash \langle a\rangle\varphi \to \langle a\rangle\psi$ is an instance of rule M_a of the kind

$$\mathrm{M}_a\frac{\varphi \to \psi}{\langle a\rangle\varphi \to \langle a\rangle\psi} \qquad \text{then} \qquad \mathrm{M}_G\frac{\varphi^\sharp \to \psi^\sharp}{\langle a\rangle\varphi^\sharp \to \langle a\rangle\psi^\sharp}$$

is an instance of M_G. We can derive

$$\sharp,\mathrm{MP}_G\frac{(\varphi \to \psi)^\sharp}{\mathrm{M}_G\dfrac{\varphi^\sharp \to \psi^\sharp}{;,\mathrm{MP}_G\dfrac{\langle a\rangle\varphi^\sharp \to \langle a\rangle\psi^\sharp}{\sharp,\mathrm{MP}_G\dfrac{(\langle a\rangle\varphi)^\sharp \to (\langle a\rangle\psi)^\sharp}{(\langle a\rangle\varphi \to \langle a\rangle\psi)^\sharp}}}}$$

The induction hypothesis yields a proof for $\mathsf{rlGL} \vdash (\langle a\rangle\varphi \to \langle a\rangle\psi)^\sharp$. If the last step of the proof is an instance of μ of the kind

$$\mu\frac{\varphi\frac{\psi}{x} \to \psi}{\mu x.\varphi \to \psi}$$

then by the induction hypothesis and $\sharp$ observe $\mathsf{rlGL} \vdash \langle \varphi^\sharp \frac{\psi^\sharp}{x}\rangle\bot \to \langle \psi^\sharp\rangle\bot$. By an application of RL deduce $\mathsf{rlGL} \vdash \langle \varphi^\sharp \frac{\psi^\sharp;?\bot;!\bot}{x}\rangle\bot \to \langle \psi^\sharp\rangle\bot$. The implication $\mathsf{rlGL} \vdash (\mu x.\varphi \to \psi)^\sharp$ follows by μ_G with $\sharp$. □

Lemma D.3 (Substitution). *If $\alpha, \beta_1, \ldots, \beta_n$ are rlGL games such that β_i does not mention freely any variable x such that x_i appears in α in a context where x is bound, then*

$$\mathcal{N}[\![\alpha]\!]^I\frac{\mathcal{N}[\![\beta_\cdot]\!]^I}{x_\cdot} = \mathcal{N}[\![\alpha\frac{\beta_\cdot}{x_\cdot}]\!]^I.$$

Proof of Lemma D.3. By a straightforward induction on α. □

Proof of Proposition 7.2. By simultaneous induction on formulas and games of sabotage game logic. For tests this uses the observation that the translation φ^c of a formula does not have free variables.

We consider the interesting cases. For formulas of the form $\langle\alpha\rangle\varphi$ note that by the induction hypothesis

$$\mathcal{N}[\![\langle\alpha\rangle\varphi]\!]_\mathrm{s}\restriction c = \mathcal{N}[\![\alpha^c]\!]^I(\emptyset)$$

where $I(y_e) = \mathcal{N}[\![\varphi^e]\!](\emptyset) = \mathcal{N}[\![\varphi^e;?\bot]\!]$. Hence $\mathcal{N}[\![\langle\alpha\rangle\varphi]\!]_\mathrm{s}\restriction c = \mathcal{N}[\![\alpha^c\frac{\varphi^\cdot;?\bot}{y_\cdot}]\!](\emptyset)$ by Lemma D.3 as required. The case for games of the kind $\alpha;\beta$ is similar.

We also explicitly treat games of the form α^*. Let $a_1, \ldots, a_m$ be the list of atomic games in α such that either $c(a) \neq \emptyset$ or $\sim a$ or $\sim a^\mathrm{d}$ appears in α. List $c_1, \ldots, c_m$ all possible contexts that satisfy $c_i(a) = \emptyset$ for all $a \notin \{a_1, \ldots, a_m\}$.

Consider first the $\subseteq$ inclusion. Fix $B_i = \mathcal{N}[\![(\alpha^*)^{c_i}]\!]^I(\emptyset)$. By definition of the translation, Lemma D.3 the inductive hypothesis

$$\begin{aligned} B_i &= \mathcal{N}[\![y_i \cup \alpha^{c_i}\frac{(\alpha^*)^{c_i}}{y_\cdot}]\!]^I(\emptyset) \\ &= U\restriction c_i \cup \mathcal{N}[\![\alpha^{c_i}]\!]^I\frac{B_\cdot}{y_\cdot}(\emptyset) \\ &= U\restriction c_i \cup \mathcal{N}[\![\alpha]\!]_\mathrm{s}(B)\restriction c_i \end{aligned}$$

where $B = \bigcup_{i=1}^n B_i \times \{c_i\}$. Hence $\mathcal{N}[\![\alpha^*]\!]_\mathrm{s}(U)\restriction c_i \subseteq B_i$ follows by pointwise minimality.

Consider next the $\supseteq$ inclusion. Define $A_i = \mathcal{N}[\![\alpha^*]\!]_{\mathrm{s}}(U)\restriction c_i$. By induction hypothesis

$$\begin{aligned} A_i &= U\restriction c_i \cup \mathcal{N}[\![\alpha]\!]_{\mathrm{s}}(\mathcal{N}[\![\alpha^*]\!]_{\mathrm{s}}(U))\restriction c_i \\ &= U\restriction c_i \cup \mathcal{N}[\![\alpha]\!]^{I_i}(\emptyset) \end{aligned}$$

where $I_i(y_{c_i}) = \mathcal{N}[\![\alpha^*]\!]_{\mathrm{s}}(U)\restriction c_i$ for all i. The $\supseteq$ inclusion follows from Theorem 7.1 by minimality. □

Proof of Proposition 7.3. Note that the proof of Lemma 7.6 does note rely on this lemma. For convenience we therefore freely use Lemma 7.6 in this proof.

The identity is proved by structural induction on the rank of a closed formula. Most cases are straightforward. The interesting cases are for least and greatest fixpoint formulas. So consider a formula $\mathrm{r}x.\alpha$. Observe that for $Z = \mathcal{N}[\![\mathrm{r}x.\alpha^\natural]\!]_{\mathrm{s}}$ by (2) of Lemma 7.6 and soundness

$$\mathcal{N}[\![\alpha^\natural]\!]_{\mathrm{s}}^{I[x\mapsto Z]} \subseteq Z$$

Hence by induction hypothesis and minimality $\mathcal{N}[\![\mathrm{r}x.\alpha]\!] \subseteq Z$. Analogously compute

$$\mathcal{N}[\![\alpha^\natural]\!]_{\mathrm{s}}^{I[x\mapsto W]} = \mathcal{N}[\![\alpha]\!]^{I[x\mapsto W]} \subseteq W$$

where $W = \mathcal{N}[\![\mathrm{r}x.\alpha]\!]\times C$. The reverse inclusion $\mathcal{N}[\![\varphi]\!] \supseteq Z$ follows by minimality of Z.

The case for greatest fixpoints follows from the least fixpoint with (1) of Lemma 7.6. □

Lemma D.4. *If α is a game of right-linear game logic, $\delta \equiv \sim b^{\mathrm{d}}_x;\sim c_x$ are fresh and β is a $\mathrm{GL_s}$ game , then*

$$\mathrm{GL_s} \vdash \langle \alpha^\natural \tfrac{\delta;\beta}{x}\rangle\varphi \leftrightarrow \langle \delta^{\mathrm{d}}; \alpha^\natural \tfrac{\delta}{x}; (c_x;\delta;\beta\cup b_x)\rangle\varphi$$

Proof of Lemma D.4. For the purposes of this proof we call a $\mathrm{GL_s}$ game α with free variables *controlled* if it is right-linear and it contains iteration games only in the form

$$\eta;(c;\eta^{\mathrm{d}};\alpha\tfrac{\eta}{x})^*;b \qquad \text{and} \qquad \eta^{\mathrm{d}};(c^{\mathrm{d}};\eta;\alpha\tfrac{\eta^{\mathrm{d}}}{x})^\times;b^{\mathrm{d}}$$

for $\eta \equiv \sim b^{\mathrm{d}};\sim c$, where neither $b, c, \sim b, \sim c$ nor their duals appear in α.

Fix a fresh variable y and first define for every controlled $\mathrm{GL_s}$ game α a modification α_U with respect to a set $U \subseteq \mathbb{V}$ by induction on α as follows

$$w_U = \begin{cases} x & \text{if } x \in U \\ w;y & \text{otherwise} \end{cases} \qquad (\alpha;\beta)_U = \alpha;(\beta)_U$$

$$a_U = a;y \qquad (a^{\mathrm{d}})_U = a^{\mathrm{d}};y$$

$$(\sim a)_U = \sim a;y \qquad (\sim a^{\mathrm{d}})_U = \sim a^{\mathrm{d}};y$$

$$(?\varphi)_U = ?\varphi;y \qquad (!\varphi)_U = !\varphi;y$$

$$(\alpha\cup\beta)_U = \alpha_U \cup \alpha_U \qquad (\alpha\cap\beta)_U = \alpha_U \cap \alpha_U$$

and for controlled loop games:

$$(\eta;(c;\eta^{\mathrm{d}};\alpha\tfrac{\eta}{x})^*;b)_U = \eta;(c;\eta^{\mathrm{d}};\alpha_{U\cup\{x\}}\tfrac{\eta}{x})^*;b$$

$$(\eta^{\mathrm{d}};(c^{\mathrm{d}};\eta;\alpha\tfrac{\eta^{\mathrm{d}}}{x})^\times;b^{\mathrm{d}})_U = \eta^{\mathrm{d}};(c^{\mathrm{d}};\eta;\alpha_{U\cup\{x\}}\tfrac{\eta^{\mathrm{d}}}{x})^\times;b^{\mathrm{d}}$$

where $\eta \equiv \sim b^{\mathrm{d}};\sim c$.

By induction on the definition observe that

$$\mathrm{GL_s} \vdash \langle \alpha_U \tfrac{\sigma_\cdot;\gamma}{u_\cdot}\tfrac{\gamma}{y}\rangle\varphi \leftrightarrow \langle \alpha\tfrac{\sigma_\cdot}{u_\cdot};\gamma\rangle\varphi \tag{1}$$

for all right-linear $\mathrm{GL_s}$ σ,γ, where $u_\cdot$ ranges over all elements of U.

The only interesting case is for controlled loop games. Before we give a proof we show that the following pairwise equivalences are provable for any ψ

$$\begin{aligned} & \langle \eta^{\mathrm{d}}; \alpha_{U\cup\{x\}}\tfrac{\eta}{x}\tfrac{\sigma_\cdot;\gamma}{u_\cdot}\tfrac{\gamma}{y}\rangle\psi \\ \text{iff}\quad & \langle \eta^{\mathrm{d}}; \alpha_{U\cup\{x\}}\tfrac{\eta}{x}\tfrac{\eta^{\mathrm{d}};\sigma_\cdot;\gamma}{u_\cdot}\tfrac{\eta^{\mathrm{d}};\gamma}{y}\rangle\psi \\ \text{iff}\quad & \langle \eta^{\mathrm{d}}; \alpha_{U\cup\{x\}}\tfrac{\eta;\tilde\gamma}{x}\tfrac{\eta^{\mathrm{d}};\sigma_\cdot;\tilde\gamma}{u_\cdot}\tfrac{\eta^{\mathrm{d}};\tilde\gamma}{y}\rangle\psi \\ \text{iff}\quad & \langle \eta^{\mathrm{d}}; \alpha\tfrac{\eta}{x}\tfrac{\eta^{\mathrm{d}};\sigma_\cdot}{u_\cdot};\tilde\gamma\rangle\psi \end{aligned}$$

where $\tilde\gamma \equiv (c\cup b;\gamma)$. The first two equivalence use $\sim$ and $\wr$ and the third is by induction hypothesis and using ϑ since η sets fresh variables not in $\sigma_\cdot$. We now turn to the proof of the case of (1) for controlled repetition. This amounts to proving

$$\mathrm{GL_s} \vdash \langle \eta;(c;\eta^{\mathrm{d}};\alpha_{U\cup\{x\}}\tfrac{\eta}{x}\tfrac{\sigma_\cdot;\gamma}{u_\cdot}\tfrac{\gamma}{y})^*;b\rangle\varphi \leftrightarrow \langle \eta;(c;\eta^{\mathrm{d}};\alpha\tfrac{\eta}{x}\tfrac{\sigma_\cdot}{u_\cdot})^*;b;\gamma\rangle\varphi \tag{2}$$

We first prove the forward implication. For this purpose let $\rho = \langle c;(c;\eta^{\mathrm{d}};\alpha\tfrac{\eta}{x}\tfrac{\sigma_\cdot}{u_\cdot})^*;b;\gamma\rangle\varphi \vee \langle b\rangle\varphi$. The following chain of equivalences is provable:

$$\begin{aligned} & \langle c;c;\eta^{\mathrm{d}};\alpha_{U\cup\{x\}}\tfrac{\eta}{x}\tfrac{\sigma_\cdot;\gamma}{u_\cdot}\tfrac{\gamma}{y}\rangle\rho \\ \text{then}\quad & \langle c;c;\eta^{\mathrm{d}};\alpha\tfrac{\eta}{x}\tfrac{\eta^{\mathrm{d}};\sigma_\cdot}{u_\cdot};\tilde\gamma\rangle\rho \\ \text{then}\quad & \langle c;c;\eta^{\mathrm{d}};\alpha_{U\cup\{x\}}\tfrac{\eta;\tilde\gamma;\tilde\rho}{x}\tfrac{\eta^{\mathrm{d}};\sigma_\cdot;\tilde\gamma;\tilde\rho}{u_\cdot}\tfrac{\eta^{\mathrm{d}};\tilde\gamma;\tilde\rho}{y}\rangle\varphi \\ \text{then}\quad & \langle c;c;\eta^{\mathrm{d}};\alpha_{U\cup\{x\}}\tfrac{\eta;\tilde\rho_0}{x}\tfrac{\eta^{\mathrm{d}};\sigma_\cdot;\tilde\rho_0}{u_\cdot}\tfrac{\tilde\rho_0}{y}\rangle\varphi \\ \text{then}\quad & \langle c;c;\eta^{\mathrm{d}};\alpha\tfrac{\eta}{x}\tfrac{\eta^{\mathrm{d}};\sigma_\cdot}{u_\cdot};\tilde\rho_0\rangle\varphi \\ \text{then}\quad & \langle c;\tilde\rho_0\rangle\varphi \\ \text{then}\quad & \rho \end{aligned}$$

where $\tilde\rho \equiv (c;\tilde\rho_0)\cup b$ and $\tilde\rho_0 \equiv (c;\eta^{\mathrm{d}};\alpha\tfrac{\eta}{x}\tfrac{\sigma_\cdot}{u_\cdot})^*;b;\gamma$. Hence we also get:

$$\mathrm{GL_s} \vdash \langle b\rangle\varphi \vee \langle c;c;\eta^{\mathrm{d}};\alpha_{U\cup\{x\}}\tfrac{\eta}{x}\tfrac{\sigma_\cdot;\gamma}{u_\cdot}\tfrac{\gamma}{y}\rangle\rho \to \rho$$

By applying μ^*_G this yields

$$\mathrm{GL_s} \vdash \langle (c;c;\eta^{\mathrm{d}};\alpha_{U\cup\{x\}}\tfrac{\eta}{x}\tfrac{\sigma_\cdot;\gamma}{u_\cdot}\tfrac{\gamma}{y})^*;b\rangle\varphi \to \rho.$$

By applying M_G we obtain

$$\mathrm{GL_s} \vdash \langle \eta;(c;c;\eta^{\mathrm{d}};\alpha_{U\cup\{x\}}\tfrac{\eta}{x}\tfrac{\sigma_\cdot;\gamma}{u_\cdot}\tfrac{\gamma}{y})^*;b\rangle\varphi \to \langle\eta\rangle\rho.$$

Applying $\approx$ we can remove the additional c and obtain:

$$\mathrm{GL_s} \vdash \langle \eta;(c;\eta^{\mathrm{d}};\alpha_{U\cup\{x\}}\tfrac{\eta}{x}\tfrac{\sigma_\cdot;\gamma}{u_\cdot}\tfrac{\gamma}{y})^*;b\rangle\varphi \to \langle\eta\rangle\rho.$$

Finally using $\forall$, $\wr$ and $\sim$ on the succedent of the implication we get the forward implication of (2).

We next prove the backward implication (2). For this purpose let $\chi = \langle c;(c;\eta^{\mathrm{d}};\alpha_{U\cup\{x\}}\tfrac{\eta}{x}\tfrac{\sigma_\cdot;\gamma}{u_\cdot}\tfrac{\gamma}{y})^*;b\rangle\varphi \vee \langle b;\gamma\rangle\varphi$. The following chain

of equivalences is provable:

$$\begin{aligned}
& \langle c;c;\eta^{\mathrm{d}};\alpha\tfrac{\eta}{x}\tfrac{\sigma.}{u.}\rangle\chi \\
\text{then}\quad & \langle c;c;\eta^{\mathrm{d}};\alpha\tfrac{\eta}{x}\tfrac{\eta^{\mathrm{d}};\sigma.}{u.}\rangle\chi \\
\text{then}\quad & \langle c;c;\eta^{\mathrm{d}};\alpha_{U\cup\{x\}}\tfrac{\eta;\tilde\chi}{x}\tfrac{\eta^{\mathrm{d}};\sigma.;\tilde\chi}{u.}\tfrac{\eta^{\mathrm{d}};\tilde\chi}{y}\rangle\varphi \\
\text{then}\quad & \langle c;c;\eta^{\mathrm{d}};\alpha_{U\cup\{x\}}\tfrac{\eta;\tilde\gamma;\tilde\chi_0}{x}\tfrac{\eta^{\mathrm{d}};\sigma.\tilde\gamma;\tilde\chi_0}{u.}\tfrac{\eta^{\mathrm{d}};\tilde\gamma;\tilde\chi_0}{y}\rangle\varphi \\
\text{then}\quad & \langle c;c;\eta^{\mathrm{d}};\alpha\tfrac{\eta}{x}\tfrac{\eta^{\mathrm{d}};\sigma.}{u.};\tilde\gamma;\tilde\chi_0\rangle\varphi \\
\text{then}\quad & \langle c;c;\eta^{\mathrm{d}};\alpha_{U\cup\{x\}}\tfrac{\eta;\tilde\gamma}{x}\tfrac{\eta^{\mathrm{d}};\sigma.\tilde\gamma}{u.}\tfrac{\eta^{\mathrm{d}};\tilde\gamma}{y};\tilde\chi_0\rangle\varphi \\
\text{then}\quad & \langle c;c;\eta^{\mathrm{d}};\alpha_{U\cup\{x\}}\tfrac{\eta}{x}\tfrac{\sigma.\gamma}{u.}\tfrac{\gamma}{y};\tilde\chi_0\rangle\varphi \\
\text{then}\quad & \langle c;\tilde\chi_0\rangle\varphi \\
\text{then}\quad & \chi
\end{aligned}$$

where $\tilde\chi \equiv (c;\tilde\chi_0)\cup(b;\gamma)$ and $\tilde\chi_0 \equiv (c;\eta^{\mathrm{d}};\alpha_{U\cup\{x\}}\tfrac{\eta}{x}\tfrac{\sigma.;\gamma}{u.}\tfrac{\gamma}{y})^*;b$. Hence

$$\mathsf{GL_s}\vdash\langle b;\gamma\rangle\varphi\vee\langle c;c;\eta^{\mathrm{d}};\alpha\tfrac{\eta}{x}\tfrac{\sigma.}{u.}\rangle\chi\to\chi$$

The remainder of the proof of the backward implication of (2) is analogous to the proof of the forward implication.

Finally the pairwise equivalences of the following formulas is provable in $\mathsf{GL_s}$:

$$\begin{aligned}
& \langle\delta^{\mathrm{d}};\alpha^\natural\tfrac{\delta}{x};(c_x;\delta;\beta\cup b_x)\rangle\varphi \\
\text{iff}\quad & \langle\delta^{\mathrm{d}};(\alpha^\natural)_{\{x\}}\tfrac{\delta;(c_x;\delta;\beta\cup b_x)}{x}\tfrac{(c_x;\delta;\beta\cup b_x)}{y}\rangle\varphi \\
\text{iff}\quad & \langle(\alpha^\natural)_{\{x\}}\tfrac{\delta;(c_x;\delta;\beta\cup b_x)}{x}\tfrac{\delta^{\mathrm{d}}(c_x;\delta;\beta\cup b_x)}{y}\rangle\varphi \\
\text{iff}\quad & \langle(\alpha^\natural)_{\{x\}}\tfrac{\delta;\beta}{x}\tfrac{?\top}{y}\rangle\varphi \\
\text{iff}\quad & \langle\alpha^\natural\tfrac{\delta;\beta}{x}\rangle\varphi
\end{aligned}$$

where the first and the fourth equivalence are by (1). The second and third equivalences use $\sim$, $\wr$ and $\approx$. □

Proof of Lemma 7.6. (1) is a straightforward structural induction.

(2) The translation of $\langle\alpha\tfrac{\mathrm{r}x.\alpha}{x}\rangle\varphi$ with $\natural$ is

$$\langle\alpha^\natural\tfrac{\delta;(c_x;\delta^{\mathrm{d}};\alpha^\natural\frac{\delta}{x})^*;b_x}{x}\rangle\varphi^\natural$$

where $\delta\equiv\sim b_x^{\mathrm{d}};\sim c_x$. For readability we drop the variable subscript. By Lemma D.4 this provably implies

$$\langle\delta^{\mathrm{d}};\alpha^\natural\tfrac{\delta}{x};(c;\delta;(c;\delta^{\mathrm{d}};\alpha^\natural\tfrac{\delta}{x})^*;b\cup b)\rangle\varphi^\natural$$

By $\cup$, ;, ? and M_G this provably implies

$$\langle\delta^{\mathrm{d}};\alpha^\natural\tfrac{\delta}{x};(c;\delta;(c;\delta^{\mathrm{d}};\alpha^\natural\tfrac{\delta}{x})^*\cup?\top);b\rangle\varphi^\natural$$

By rfp_G and M_G this provably implies

$$\langle\delta^{\mathrm{d}};\alpha^\natural\tfrac{\delta}{x};(c;\delta;c;\delta^{\mathrm{d}};\alpha^\natural\tfrac{\delta}{x};(c;\delta^{\mathrm{d}};\alpha^\natural\tfrac{\delta}{x})^*\cup?\top);b\rangle\varphi^\natural$$

By P, C and $\aleph$:

$$\langle\delta^{\mathrm{d}};\alpha^\natural\tfrac{\delta}{x};(c;\delta^{\mathrm{d}};\alpha^\natural\tfrac{\delta}{x};(c;\delta^{\mathrm{d}};\alpha^\natural\tfrac{\delta}{x})^*\cup?\top);b\rangle\varphi^\natural$$

Hence by $*_G$ and M_G

$$\langle\delta^{\mathrm{d}};\alpha^\natural\tfrac{\delta}{x};(c;\delta^{\mathrm{d}};\alpha^\natural\tfrac{\delta}{x})^*;b\rangle\varphi^\natural$$

By $\aleph$ and C this in turn provably implies the translation

$$\langle\delta;c;\delta^{\mathrm{d}};\alpha^\natural\tfrac{\delta}{x};(c;\delta^{\mathrm{d}};\alpha^\natural\tfrac{\delta}{x})^*;b\rangle\varphi^\natural$$

By M_G and $*_G$ this in turn provably implies the translation

$$(\langle\mathrm{r}x.\alpha\rangle\varphi)^\natural\equiv\langle\delta;(c;\delta^{\mathrm{d}};\alpha^\natural\tfrac{\delta}{x})^*;b\rangle\varphi^\natural$$

as required.

(3) Let $\rho\equiv\langle c;\delta^{\mathrm{d}};\beta\rangle\psi\vee\langle b\rangle\varphi^\natural$. Assume $\mathsf{GL_s}\vdash\langle\alpha^\natural\frac{\beta;?\psi;!\bot}{x}\rangle\varphi^\natural\to\langle\beta\rangle\psi$. By M_G and some propositional reasoning deduce

$$\mathsf{GL_s}\vdash\langle b\rangle\varphi^\natural\vee\langle c;\delta^{\mathrm{d}};\alpha^\natural\tfrac{\delta;\beta;?\psi;!\bot}{x}\rangle\varphi^\natural\to\rho$$

By Lemma D.4 (and $\aleph$) this implies (by M_G and Θ)

$$\mathsf{GL_s}\vdash\langle b\rangle\varphi^\natural\vee\langle c;\delta^{\mathrm{d}};\alpha^\natural\tfrac{\delta}{x};(c;\delta;\beta;?\psi;!\bot\cup b)\rangle\varphi^\natural\to\rho$$

Since b,c do not appear in β,ψ this implies

$$\mathsf{GL_s}\vdash\langle b\rangle\varphi^\natural\vee\langle c;\delta^{\mathrm{d}};\alpha^\natural\tfrac{\delta}{x}\rangle\rho\to\rho$$

By applying μ_G^* it follows that

$$\mathsf{GL_s}\vdash\langle(c;\delta^{\mathrm{d}};\alpha^\natural\tfrac{\delta}{x})^*;b\rangle\varphi^\natural\to\rho$$

is provable. By M_G it follows that

$$\mathsf{GL_s}\vdash\langle\delta;(c;\delta^{\mathrm{d}};\alpha^\natural\tfrac{\delta}{x})^*;b\rangle\varphi^\natural\to\langle\delta\rangle(\langle c;\delta^{\mathrm{d}};\beta\rangle\psi\vee\langle b\rangle\varphi^\natural)$$

By C, $\aleph$, Θ, $\forall$ and W this provably implies

$$\mathsf{GL_s}\vdash\langle\delta;(c;\delta^{\mathrm{d}};\alpha^\natural\tfrac{\delta}{x})^*;b\rangle\varphi^\natural\to\langle\beta\rangle\psi$$

as required. □

Proof of Lemma 7.7. We can without loss of generality fix a finite set U of atomic games and prove the lemma only for formulas restricted to atomic games from U. We fix for every atomic game $a\in U$ three fresh atomic games $a_\emptyset$, $a_\diamond$ and $a_\square$. The role of these games is to capture the context syntactically, that is we maintain the invariant $c(a)=i$ iff $c(a_i)=\diamond$. We say c is a U-context if $c(a)=\emptyset$ for all $a\notin U$. Let C_U be the set of all U-contexts.

Fix some context c and list all elements of U as

$$a_1,\dots,a_n,b_1,\dots,b_m,c_1,\dots,c_k$$

such that $c(a_i)=\diamond$, $c(b_i)=\square$ and $c(c_i)=\emptyset$ for all i. Define the games

$$\begin{aligned}
\eta_c &\equiv \sim a_1;\dots;\sim a_n;\sim b_1^{\mathrm{d}};\dots;\sim b_m^{\mathrm{d}} \\
\zeta_c &\equiv \sim(a_1)_\emptyset^{\mathrm{d}};\sim(a_1)_\diamond;\sim(a_1)_\square^{\mathrm{d}};\dots;\sim(a_n)_\emptyset^{\mathrm{d}};\sim(a_n)_\diamond;\sim(a_n)_\square^{\mathrm{d}} \\
&\quad\sim(c_1)_\emptyset^{\mathrm{d}};\sim(c_1)_\diamond^{\mathrm{d}};\sim(c_1)_\square;\dots;\sim(c_k)_\emptyset^{\mathrm{d}};\sim(c_k)_\diamond^{\mathrm{d}};\sim(c_k)_\square \\
&\quad\sim(c_1)_\emptyset;\sim(c_1)_\diamond^{\mathrm{d}};\sim(c_1)_\square^{\mathrm{d}};\dots;\sim(c_k)_\emptyset;\sim(c_k)_\diamond^{\mathrm{d}};\sim(c_k)_\square^{\mathrm{d}} \\
\xi_c &\equiv (a_1)_\diamond;\dots;(a_n)_\diamond;(b_1)_\square;\dots;(b_m)_\square;(c_1)_\emptyset;\dots;(c_k)_\emptyset
\end{aligned}$$

For a C_U-indexed family α_c of $\mathsf{GL_s}$ games we define the guarded version

$$\check\alpha. = \bigcup_{c\in C_U}(\xi_c;\zeta_c;\alpha_c)$$

For a single game α we mean by $\check\alpha=\check\beta.$ where $\beta_c=\alpha$ for all $c\in C_U$.

We define for every formula φ and every game α of $\mathsf{GL_s}$ we define modified versions $\widehat{\varphi}$ and $\widehat{\alpha}$ by induction on the definition as follows

$$\widehat{P} \equiv P \qquad \widehat{\neg P} \equiv \neg P$$
$$\widehat{\varphi \vee \psi} \equiv \widehat{\varphi} \vee \widehat{\psi} \qquad \widehat{\varphi \wedge \psi} \equiv \widehat{\varphi} \wedge \widehat{\psi}$$
$$\widehat{\langle\alpha\rangle\varphi} \equiv \langle\widehat{\alpha}\rangle\widehat{\varphi} \qquad \widehat{\alpha;\beta} \equiv \widehat{\alpha};\widehat{\beta}$$
$$\widehat{a} \equiv a_\emptyset; a \cup a_\diamond \qquad \widehat{a^\mathrm{d}} \equiv a_\emptyset; a^\mathrm{d} \cup a_\diamond; !\bot \cup a_\square$$
$$\widehat{\sim a} \equiv \sim a^\mathrm{d}_\emptyset; \sim a_\diamond; \sim a^\mathrm{d}_\square \qquad \widehat{\sim a^\mathrm{d}} \equiv \sim a^\mathrm{d}_\emptyset; \sim a^\mathrm{d}_\diamond; \sim a_\square\,.$$
$$\widehat{?\varphi} \equiv ?\widehat{\varphi} \qquad \widehat{!\varphi} \equiv !\widehat{\varphi}$$
$$\widehat{\alpha \cup \beta} \equiv \widehat{\alpha} \cup \widehat{\beta} \qquad \widehat{\alpha \cap \beta} \equiv \widehat{\alpha} \cap \widehat{\beta}$$
$$\widehat{\alpha^*} \equiv (\check{\widehat{\alpha}})^* \qquad \widehat{\alpha^\times} \equiv (\check{\widehat{\alpha}})^\times$$

Before we prove the lemma we make the following observations

(1) $\mathsf{GL_s} \vdash \langle\zeta_c\rangle\widehat{\varphi} \to \langle\eta_c\rangle\varphi$
(2) $\mathsf{GL_s} \vdash \langle\zeta_c\rangle\langle\widehat{\alpha};\check{\beta}\rangle\top \to \langle\zeta_c\rangle\langle\widehat{\alpha};\beta\rangle\bot$
(3) $\mathsf{GL_s} \vdash \langle\zeta_c\rangle(\langle\check{\beta}.\rangle\varphi \leftrightarrow \langle\beta_c\rangle\varphi)$

for all formulas φ and all games α.

For observation (1) define first a second modification $(\cdot)^E$ exactly like $\widehat{\cdot}$ except that

$$(\sim a)^E \equiv \sim a^\mathrm{d}_\emptyset; \sim a_\diamond; \sim a^\mathrm{d}_\square; \sim a \qquad (\sim a^\mathrm{d})^E \equiv \sim a^\mathrm{d}_\emptyset; \sim a^\mathrm{d}_\diamond; \sim a_\square; \sim a^\mathrm{d}.$$

Then by $\sim_1$ observe $\mathsf{GL_s} \vdash \langle\zeta_c\rangle(\widehat{\varphi} \leftrightarrow \varphi^E)$. Define a further modification $(\cdot)^F$ exactly like $(\cdot)^E$ except that

$$(a)^F \equiv (a_\emptyset \cup a_\diamond \cup a_\square); a \qquad (a^\mathrm{d})^F \equiv (a_\emptyset \cup a_\diamond \cup a_\square); a^\mathrm{d}$$

Using $\forb$ it is not hard to deduce that $\mathsf{GL_s} \vdash \langle\zeta_c\rangle(\varphi^E \leftrightarrow \varphi^F)$. Again applying $\forb$ yields that $\mathsf{GL_s} \vdash \langle\zeta_c\rangle(\varphi^F \leftrightarrow \varphi^G)$, where $(\cdot)^F$ exactly like $(\cdot)^E$ except that

$$(a^*)^G \equiv (\alpha^F)^* \qquad (\alpha^\times)^G \equiv (\alpha^F)^\times.$$

Finally the required implication follows with $\simeq$ since no $a_\emptyset, a_\diamond, a_\square$ appears. Observation (2) can also be proved with $\forb$ and (3) is immediate.

We finally prove the main claim of the lemma. By simultaneous induction on formulas φ and games α of sabotage game logic we prove

(1) $\mathsf{GL_s} \vdash \langle\varphi^{c\natural}\rangle\bot \to \langle\zeta_c\rangle\widehat{\varphi}$
(2) $\mathsf{GL_s} \vdash \langle\alpha^{c\natural}\frac{\zeta_{c.};\beta_{c.}}{y_{c.}}\rangle\bot \to \langle\zeta_c\rangle\langle\widehat{\alpha};\check{\beta}.\rangle\bot$

for all contexts c and all games $\beta_{c.}$. The lemma is easily deduced from the case for $c = c_\emptyset$ with observation 1.

(1) If the formula is of the form P the implication to show is $\mathsf{GL_s} \vdash \langle ?P; !\bot\rangle\bot \to \langle\zeta_c\rangle P$. The implication is provably by $\mathfrak{Y}$, ; and ?. The case for $\neg P$ is analogous.
(2) If the formula is of the form $\varphi \vee \psi$ or $\varphi \wedge \psi$ then the implication is by the induction hypothesis, $\cup$, $\cap$, $\forall$ and $\mathbb{A}$.
(3) If the formula is of the form $\langle\alpha\rangle\psi$ we first apply the induction hypothesis to φ and obtain

$$\mathsf{GL_s} \vdash \langle\alpha^{c\natural}\frac{\varphi^{c\natural};?\bot}{y_{c.}}\rangle\bot \to \langle\alpha^{c\natural}\frac{\zeta_{c.};?\widehat{\varphi};!\bot}{y_{c.}}\rangle\bot.$$

Now applying the induction hypothesis for α combined with (2) we get

$$\mathsf{GL_s} \vdash \langle\alpha^{c\natural}\frac{\zeta_{c.};?\widehat{\varphi};!\bot}{y_{c.}}\rangle\bot \to \langle\zeta_c\rangle\langle\widehat{\alpha};?\widehat{\varphi};!\bot\rangle\bot.$$

Putting this together we get

$$\mathsf{GL_s} \vdash \langle(\langle a\rangle\varphi)^{c\natural}\rangle\bot \to \langle\zeta_c\rangle\langle\widehat{a}\rangle\widehat{\varphi}.$$

as required.

(4) If the game is atomic of the form a there are three cases. If $c(a) = \emptyset$ we need to show

$$\mathsf{GL_s} \vdash \langle a;\zeta_c;\beta_c\rangle\bot \to \langle\zeta_c\rangle\langle\widehat{a};\check{\beta}.\rangle\bot$$

This is easy to see using P, (3) and $\sim$.
If $c(a) = \diamond$ note that the required

$$\mathsf{GL_s} \vdash \langle\zeta_c;\beta_c\rangle\bot \to \langle\zeta_c\rangle\langle\widehat{a};\check{\beta}.\rangle\bot$$

is easily provable with $\sim$ since $\sim a_\diamond$ appears in ζ_c and using observation (3).
If $c(a) = \square$ then $(\langle a^c\rangle\bot)^\natural$ is provably false, so the implication holds vacuously.

(5) If the game is of the form a^d there are again three cases. The case for $c(a) = \emptyset$ is analogous to the previous case.
If $c(a) = \diamond$ the antecedent is valid, so we need to show $\mathsf{GL_s} \vdash \langle\zeta_c\rangle\langle\widehat{a^\mathrm{d}}\rangle\langle\check{\beta}.\rangle\bot$. This follows with $\sim$ since ζ_c contains $\sim a_\diamond$.
If $c(a) = \square$ the proof is analogous to the case of $c(a) = \diamond$ for games a.

(6) If the game is of the form $\sim a$ the claim is

$$\mathsf{GL_s} \vdash \langle\zeta_{c\frac{\diamond}{a}};\beta_{c\frac{\diamond}{a}}\rangle\bot \to \langle\zeta_c\rangle\langle\widehat{\sim a};\check{\beta}.\rangle\bot.$$

This is easy to see by rearranging the sabotages with P and observation (3).
The case for $\sim a^\mathrm{d}$ is analogous.

(7) If the game is of the form $?\varphi$, $!\varphi$, $\beta \cup \gamma$ or $\beta \cap \gamma$ the claim is easy to derive from the induction hypothesis using $\forall$ and $\mathbb{A}$.
(8) If the game is of the form $\alpha;\gamma$ note that by induction hypothesis on α:

$$\mathsf{GL_s} \vdash \langle(\alpha;\gamma)^{c\natural}\frac{\zeta_{c.}\beta_{c.}}{y_{c.}}\rangle\bot \to \langle\zeta_c\rangle\langle\widehat{\alpha};\bigcup_{e\in C_U}(\xi_e;\zeta_e;\gamma^{e\natural}\frac{\zeta_{c.}\beta_{c.}}{y_{c.}})\rangle\bot$$

Applying the induction hypothesis on γ this shows:

$$\mathsf{GL_s} \vdash \langle(\alpha;\gamma)^{c\natural}\frac{\zeta_{c.}\beta_{c.}}{y_{c.}}\rangle\bot \to \langle\zeta_c\rangle\langle\widehat{\alpha};\bigcup_{e\in C_U}(\xi_e;\zeta_e;\zeta_e;\widehat{\gamma};\check{\beta}.)\rangle\bot$$

Now $\approx$ and (3) yield the required implication.
(9) If the game is of the form α^* list the contexts $c_1, \ldots, c_n$ as in the definition of α^{*c}.
We need to show:

$$\mathsf{GL_s} \vdash (\langle \mathrm{r}_i(z_{c_1},\ldots,z_{c_n}).(y_{c_1} \cup \alpha^{c_1}\frac{z_{c.}}{y_{c.}},\ldots,y_{c_n} \cup \alpha^{c_n}\frac{z_{c.}}{y_{c.}})\rangle\bot)^\natural\frac{\zeta_{c.};\beta_{c.}}{y_{c.}}$$
$$\to \langle\zeta_{c_i}\rangle(\langle\check{\widehat{\alpha}}^*;\check{\beta}_{c.}\rangle\bot)$$

By the inductive generalization (3) of Lemma 7.6 to the vectorial fixpoints from Theorem 7.1 this reduces to proving (for all $i = 1, \ldots, n$):

$$\mathsf{GL_s} \vdash \langle\zeta_{c_i};\beta_{c_i} \cup (\alpha^{c_i})^\natural\frac{\zeta_{c.};\check{\widehat{\alpha}}^*;\check{\beta}_{c.};?\bot}{y_{c.}}\rangle\bot \to \langle\zeta_{c_i}\rangle(\langle\check{\widehat{\alpha}}^*;\check{\beta}_{c.}\rangle\bot)$$

By the induction hypothesis and (3) on α this reduces to

$$\mathsf{GL_s} \vdash \langle \zeta_{c_i}; \beta_{c_i} \cup \zeta_{c_i}; \widehat{\alpha}; \check{\widehat{\alpha}}^*; \check{\beta}_{c.}; ?\bot\rangle\bot \to \langle \zeta_{c_i}\rangle(\langle \widehat{(\alpha^*)}; \check{\beta}_{c.}\rangle\bot)$$

Using (3) this again reduces to

$$\mathsf{GL_s} \vdash \langle \zeta_{c_i}\rangle(\langle \check{\beta}_{c.} \cup \check{\widehat{\alpha}}; \check{\widehat{\alpha}}^*; \check{\beta}_{c.}; ?\bot\rangle\bot \to \langle \check{\widehat{\alpha}}^*; \check{\beta}_{c.}\rangle\bot)$$

This now easily follows from an instance of $*_G$ with M_G.

(10) If the game is of the form $\alpha^\times$ list the contexts $c_1, \ldots, c_n$ as in the definition of $\alpha^{\times c}$. Let

$$\gamma_{c_i} \equiv (\lrcorner_i(z_{c_1}, \ldots, z_{c_n}).(y_{c_1} \cap \alpha^{c_1} \tfrac{w_{c.}; z_{c.}}{y_{c.}}, \ldots, y_{c_n} \cap \alpha^{c_n} \tfrac{w_{c.}; z_{c.}}{y_{c.}}))^{\natural \frac{\zeta_{c.}; \beta_{c.}}{y_{c.}}}$$

And let $\delta_i = \gamma_{c_i} \frac{\zeta_{c.}}{w_{c.}}$. By (2) of Lemma 7.6 and $\neg$

$$\mathsf{GL_s} \vdash \langle \eta_{c_i}\rangle\bot \to \langle \zeta_{c_i}; \beta_{c_i} \cap (\alpha^{c_i})^{\natural \frac{\zeta_{c.}; \delta_{c.}}{y_{c.}}}\rangle\bot. \tag{3}$$

(Although it is written as a vectorial fixpoint, formally it is a single variable fixpoint as defined by Theorem 7.1 and Lemma 7.6 applies.) By applying M_G

$$\mathsf{GL_s} \vdash \langle \zeta_{c_i}; \delta_{c_i}\rangle\bot \to \langle \zeta_{c_i}; \beta_{c_i} \cap \zeta_{c_i}; (\alpha^{c_i})^{\natural \frac{\zeta_{c_i}; \delta_{c.}}{y_{c.}}}\rangle\bot.$$

By the induction hypothesis on α and $\approx$ we obtain

$$\mathsf{GL_s} \vdash \langle \zeta_{c_i}; \delta_{c_i}\rangle\bot \to \langle \zeta_{c_i}; \beta_{c_i} \cap \zeta_{c_i}; \widehat{\alpha}; \check{\delta}_{c.}\rangle\bot.$$

Combining these for all i and using M_G

$$\mathsf{GL_s} \vdash \langle \check{\delta}.\rangle\bot \to \langle \check{\beta}_{c.} \cap \bigcup_{e \in C_U} (\xi_e; \zeta_e; \widehat{\alpha}); \check{\delta}_{c.}\rangle\bot.$$

By $*_G$ this implies

$$\mathsf{GL_s} \vdash \langle \check{\delta}.\rangle\bot \to \langle \widehat{(\alpha^*)}; \check{\beta}_{c.}\rangle\bot.$$

Using M_G and observation (3)

$$\mathsf{GL_s} \vdash \langle \zeta_{c_i}; \delta_{c_i}\rangle\bot \to \langle \zeta_{c_i}; \widehat{(\alpha^*)}; \check{\beta}_{c.}\rangle\bot.$$

By $\simeq$ the game $\zeta_{c_i}; \delta_{c_i}$ is equivalent to $\alpha^{c\natural \frac{\zeta_{c.}; \beta_{c.}}{y_{c.}}}$. □

Proof of Proposition 7.8. Let π be a proof of φ in rlGL. By replacing every free variable in any formula in π by a fixed atomic game a we turn π into a rlGL proof of φ which does not mention formulas which have free variables. Next replace every formula φ in this proof by $\varphi^\natural$ and call the resulting proof π'. This proof can now be transformed into a $\mathsf{GL_s}$ proof for $\mathsf{GL_s} \vdash \varphi^\natural$.

By definition and (1) of Lemma 7.6 it is clear that the translation of most proof rules and axioms remain proof rules and axioms respectively. For the fp_G axiom we use (2) of Lemma 7.6 and include a derivation of this axiom. For the rule μ_G we can extend the proof appropriately by (3) of Lemma 7.6. □

Proof of Theorem 7.9. Soundness holds by Lemma 6.5. If φ is a valid formula of $\mathsf{GL_s}$, the translation $\langle \varphi^{c_\emptyset}\rangle\bot$ is valid by Proposition 7.2. Moreover by Theorem 6.8 and Proposition 7.8 the formula $\mathsf{GL_s} \vdash (\langle \varphi^{c_\emptyset}\rangle\bot)^\natural$ is provable and therefore by Lemma 7.7 also $\mathsf{GL_s} \vdash \varphi$. □