LIF

Laboratoire d'Informatique Fondamentale
de Marseille

Unité Mixte de Recherche 6166
CNRS – Université de Provence – Université de la Méditerranée

Controllers from proofs: An alternative approach to control synthesis for μ-calculus specifications

Nicolas Baudru and Peter Niebert

Rapport/Report 23-2004

October 21, 2004

Controllers from proofs: An alternative approach to control synthesis for μ-calculus specifications

Nicolas Baudru and Peter Niebert

LIF – Laboratoire d'Informatique Fondamentale de Marseille
UMR 6166
CNRS – Université de Provence – Université de la Méditerranée

Laboratoire d'Informatique Fondamentale (LIF) de Marseille
Université de Provence – CMI
39, rue Joliot-Curie / F-13453 Marseille Cedex 13

[baudru,niebert]@cmi.univ-mrs.fr

Abstract/Résumé

Given a "plant" $\mathcal{S}$, *the control synthesis problem* can be understood as the search for a component $\mathcal{C}$ such that the combined system $\mathcal{S} \times \mathcal{C}$ meets a specification φ. The properties may range from simple reachability constraints to arbitrary finite state branching time properties such as specified in the μ-calculus. Previous works typically combine the (finite state transition) system $\mathcal{S}$ and the specification of the desired behavioural property φ into a specification $\psi(\mathcal{S}, \varphi)$ of the controller $\mathcal{C}$ to be synthesized.

In a twist to this setting, we show how the control synthesis problem can be presented as *controllability problem* for the uncontrolled system $\mathcal{S}$, i.e. we give a simple transformation of the control goal φ specification into a property $\varsigma(\varphi)$ such that any system $\mathcal{S}$ satisfies $\varsigma(\varphi)$ iff a controller $\mathcal{C}$ exists such that $\mathcal{S} \times \mathcal{C}$ satisfies φ. $\mathcal{S} \models \varsigma(\varphi)$ can then be established by model-checking and we also show how to extract $\mathcal{C}$ from the result (tableau) of the model-checking process.

The proof technique used for the main theorem is based on transformations of proofs (tableaus) and may have an interest of its own. Beyond model-checking, and beyond previous works on control synthesis for the μ-calculus, our approach may also be applied to the algorithmic construction of controllers for provably controllable infinite state systems. However, we see the main potential advantage in opening an easy way of implementing general control synthesis in existing model-checking frameworks (like Spin, Murϕ or CADP), including application of model-checking heuristics that may provide reduced size controllers.

Keywords: control synthesis, μ-calculus, model-checking, approximations, disjunctive formulas, control formulas.

Etant donné un système $\mathcal{S}$, le problème de *la synthèse de contrôleurs* peut être compris comme la recherche d'un composant $\mathcal{C}$ tel que le système combiné $\mathcal{S} \times \mathcal{C}$ rencontre une spécification φ. Les propriétés considérées peuvent aussi bien être de simples contraintes d'accessibilité que des propriétés temporelles arborescentes à état finis spécifiées dans le μ-calcule. Les travaux précédents combinent un système (de transition à états finis) $\mathcal{S}$ et une spécification de la propriété comportementale désirée φ en une spécification $\psi(\mathcal{S}, \varphi)$ du contrôleur $\mathcal{C}$ à synthétiser.

Notre approche diffère des précédentes: nous montrons que le problème de la synthèse de contrôleurs peut être présenté comme *un problème de contrôlabilité* du système à contrôler $\mathcal{S}$: nous présentons une simple transformation de la spécification φ exprimant le but du contrôle en une propriété $\varsigma(\varphi)$ telle qu'un système $\mathcal{S}$ satisfait $\varsigma(\varphi)$ ssi un contrôleur $\mathcal{C}$ tel que $\mathcal{S} \times \mathcal{C}$ satisfait φ existe. $\mathcal{S} \models \varsigma(\varphi)$ peut être alors établie par model-checking et nous montrons comment extraire $\mathcal{C}$ à partir du résultat (tableaux) du processus de model-checking.

La technique de preuve utilisée pour le théorème principal est basée sur des transformations de preuves (tableaux) et présente un intérêt en soi. Au-delà du model-checking, et au-delà des précédents travaux sur la synthèse de contrôleurs pour le μ-calcule, notre approche peut être aussi utilisée pour la construction algorithmique de contrôleurs pour des systèmes contrôlables avec un espace d'états infini. De plus, nous verrons que le principal avantage de notre méthode est qu'elle permet une implémentation facile du problème de la synthèse de contrôleurs dans des domaines particuliers du model-checking (comme Spin, Murϕ ou CADP), incluant des heuristiques qui peuvent réduire la taille des contrôleurs.

Mots-clés : synthèse de contrôleur, μ-calcule, model-checking, approximations, formules disjunctives, formules de contrôles.

Relecteurs/Reviewers: Luigi Santocanale and Rémi Morin.

1 Introduction

Control synthesis addresses the question of finding (if it exists) a controller $\mathcal{C}$ for a "plant" $\mathcal{S}$ such that the combination $\mathcal{S} \times \mathcal{C}$ of plant and controller satisfies a desired behavioural property φ, written $\mathcal{S} \times \mathcal{C} \models \varphi$. Originally, it was introduced by Ramadge and Wonham [9] for the problem of avoiding undesired states where it was shown how such controllers can be computed. The verification community has subsequently identified this question as being related to synthesizing parts of an open system. A huge body of work both fundamental and pragmatic has emerged from this question. In particular, it has been found that the properties amenable to automatic synthesis can be extended to arbitrary finite state branching time properties such as the μ-calculus [2].

These works, but also older works for synthesizing processes from a context and a specification [1,6] typically transform the problem $\mathcal{S} \times ? \models \varphi$ into an equivalent problem $? \models \psi(\mathcal{S}, \varphi)$, i.e. a specification $\psi(\mathcal{S}, \varphi)$ that a transition system must fulfill to be a controller for $\mathcal{S}$ and the desired property φ. Then, a satisfiability problem has to be solved for $\psi(\mathcal{S}, \varphi)$, which – if the system is controllable as desired – yields a controller.

In this work, we show that a twist to this problem is possible where the existence of a controller for some system $\mathcal{S}$ and behavioural property φ is rewritten as a property $\varsigma(\varphi)$ of $\mathcal{S}$, i.e. $\mathcal{S} \models \varsigma(\varphi)$ iff there exist a $\mathcal{C}$ such that $\mathcal{S} \times \mathcal{C} \models \varphi$. It is noteworthy that this comes with a transformation $\varsigma(\varphi)$ which is completely independent of $\mathcal{S}$ and that the existence of a controller is now presented as a verification problem (rather than a synthesis problem). In contrast to previous work it is therefore not excluded that $\mathcal{S}$ is an infinite state system, in which human aided proof may replace fully automatic model-checking. However, throughout this work we assume full observability of the plant states. We intend to consider partial obserbability in future work.

Both model-checking or human aided proof produces as a side effect a *tableau* or *proof object* we call derivation graph here. In a second step, we show how to construct a controller directly from the derivation graph of $\mathcal{S} \models \varsigma(\varphi)$, the controller thus has at most the size (or, for infinite state systems, at most the algorithmic space and time complexity) of the derivation graph. In the case of $\mathcal{S}$ finite, we obtain confirmation of the published bounds on the decision and size of the controllers. However, the smaller proofs give smaller controllers. This opens a very appealing perspective: The collected knowledge about heuristic accelerations in model-checking (abstractions, symmetry reduction, symbolic state space representations like BDDs, ...) may then service the search for smaller proofs and smaller controllers, essential for applicability in bounded resource situations. We intend to further explore this question in the future.

The formal framework of this work relies on transition systems, synchronous products and the μ-calculus as specification logic. More precisely, we consider μ-calculus formulas in a normal form called *disjunctive formulas* [5]. The normal

form, avoiding conjunctions between behavioural parts of the specification[1], is essential for an easy translation $\varsigma(\varphi)$. Janin and Walukiewicz [5] have shown that any μ-calculus formula can be transformed into an equivalent disjunctive formula, at the price of an exponential blowup and the introduction of alternations (between least and greatest fixpoints) in the formula. However, from a pragmatic perspective it is possible to code many interesting control problems directly as a disjunctive formula, as we will discuss. In summary, the translation of an arbitrary μ-calculus formula φ to a controllability-formula $\varsigma(\varphi)$ is done in two steps, first the passage to an equivalent disjunctive formula φ' (technically difficult and with an exponential blowup, yet already extensively treated in the literature) followed by a second – and much easier step – of the transformation of disjunctive φ' to a control formula. In this work we only need to discuss the second step.

Given disjunctive formulas, our translation ς encodes the capabilities of the controller. We obtain a μ-calculus formula $\varsigma(\varphi)$ that has a tightly similar structure and – depending on the control problem – at most a linear blowup in size (with sharing).

We assume a model-checking procedure that computes a derivation graph (tableau/proof object), a low level circular structure consisting of pairs (s, α), s a state and α a "sub-formula" (in a certain sense) of $\varsigma(\varphi)$, a subgoal in the justification why $\varsigma(\varphi)$ holds at the initial state of $\mathcal{S}$. In [10], the existence of derivation graphs is shown to be an alternative semantics for the μ-calculus, even for infinite state systems.

For finite state systems, if $\mathcal{S} \models \varsigma(\varphi)$ then the model checking algorithm will find a derivation graph for it. From this derivation graph, we construct at the same time $\mathcal{C}$ and a derivation graph proving $\mathcal{S} \times \mathcal{C} \models \varphi$. In fact, every derivation graph gives thus a controller and the freedom in the construction of the derivation graph provides a corresponding freedom for the construction of a controller.

If to the contrary $\mathcal{S} \not\models \varsigma(\varphi)$ then – due to closure under negation of the μ-calculus – according to [10], there exists a derivation graph for $\mathcal{S} \models \overline{\varsigma(\varphi)}$. We then show how to construct from this derivation graph for any finite state or infinite state system $\mathcal{C}$ a derivation graph for $\mathcal{S} \times \mathcal{C} \models \overline{\varphi}$, hence $\mathcal{S} \times \mathcal{C} \not\models \varphi$ and no controller exists.

Both directions of our proof are thus based on transformations of derivation graphs which turn out once more to be an excellent tool for reasoning about the μ-calculus.

The rest of this article is structured as follows: In Section 2, we formalize the control synthesis problem treated in this paper, control synthesis for synchronous discrete event systems with full observability. In Section 3, we introduce the μ-calculus and the normal forms important to this work. In Section 4, we recall known notions from model checking as well as the derivation graph characterisation of [10]. Section 5 contains the main technical contribution, translation

[1] In terms of tree automata, the formulas correspond to non-deterministic automata after the elimination of alternations

to control formulas, controller synthesis and correctness proofs. In Section 6, we discuss the applicability of our approach to an action based setting.

In the future we intend to explore extensions of the current framework to partial observability and we intend to explore the approach experimentally in combinations with model checking heuristics.

2 The control synthesis problem

We address the control synthesis problem for synchronous discrete event systems under the assumption of complete observability. I.e. we assume that the controller has by observation of events complete information of the system state and that the system and the controller "take turns". This setting applies to a number of industrial control problems, notably error handling in production plants. In Section 6, we discuss applicability to variants of this basic control problem.

As formal framework, we introduce the notion of *controllable state transition system*.

Definition 1. A *transition system* over a set of actions Σ is a tuple $\mathcal{T} = (S, s_0, \delta)$ where S is a set of states, $s_0 \in S$ is the *initial state* and $\delta \subseteq S \times \Sigma \times S$ is the transition relation. We say that $\mathcal{T}$ is *deterministic* iff for each $s \in S$ and $a \in \Sigma$ there is at most one $s' \in S$ such that $(s, a, s') \in \delta$.

The set of accessible states $A \subseteq S$ is inductively defined as the least set such that $s_0 \in A$ and for each $s \in A$, $(s, a, s') \in \delta$ implies $s' \in A$.

The *synchronous product* of two transition systems $\mathcal{T}_1 = (S_1, s_0^1, \delta_1), \mathcal{T}_2 = (S_2, s_0^2, \delta_2)$ is the transition system $\mathcal{T}_1 \times \mathcal{T}_2 = (S_1 \times S_2, (s_0^1, s_0^2), \delta_\times)$, where $((s_1, s_2), a, (s_1', s_2')) \in \delta_\times$ iff $(s_1, a, s_1') \in \delta_1$ and $(s_2, a, s_2') \in \delta_2$.

A *controllable state transition system* $\mathcal{S} = (S, C, s_0, \delta)$ is a deterministic transition system (S, s_0, δ) with a subset $C \subseteq S$ of controllable states.

A controller for $\mathcal{S}$ is a transition system $\mathcal{T} = (S', s_0', \delta')$ such that for any accessible state (s, s') in the synchronous product $\mathcal{S} \times \mathcal{T}$ with $s \notin C$ and $(s, a, s_1) \in \delta_1$ there also exists $s_1' \in S'$ such that $(s', a, s_1') \in \delta'$.

Determinism is the formalization of complete observability: A given path labeling $a_1 \dots a_n$ in the synchronous product $\mathcal{S} \times \mathcal{T}$ corresponds to only one state of the system $\mathcal{S}$. The definition of a controllable state transition system and a controller informally describes as "controlling" restricting the transitions in controllable states.

3 The propositional μ-calculus

As a logic for the specification of behavioural properties of transition systems, we use the propositional μ-calculus as introduced by Kozen [7]. Behavioural properties are properties that combine static state properties with a means of specifying the dynamic behaviour of the system in passing from one state to another.

In order to specify the state properties of a transition system, we assume a set $\mathcal{A}$ of atomic propositions that hold at certain states and not at others, i.e. we consider for a transition system $\mathcal{T} = (S, s_0, \delta)$ a function $v : \mathcal{A} \to 2^S$.

In addition to the set $\mathcal{A} = \{p, q, r, \ldots\}$ of atomic propositions we assume a disjoint set of variables $\mathcal{V} = \{X, Y, Z, \ldots\}$ for recursive definitions. The variant of the μ-calculus we use in this work is then defined to be the least set of formulas of one of the following forms :

$$X|p|\neg p|\bigwedge \Phi|\bigvee \Phi|\Box\alpha|\Diamond\alpha|\mu X.\alpha|\nu X.\alpha$$

where $p \in \mathcal{A}$ is some atomic proposition, Φ is itself a finite set of formulas, α is a formula, and $X \in \mathcal{V}$ is some variables. We also write binary conjunctions ($\alpha_1 \wedge \alpha_2 = \bigwedge\{\alpha_1, \alpha_2\}$) and disjunctions ($\alpha_1 \vee \alpha_2 = \bigvee\{\alpha_1, \alpha_2\}$).

In the following, α, β, α_1 ... will denote formulas., X,Y, X_1 ... variables and p, q, p_1, ... the atomic propositions. We will use σ to denote either the operator μ or the operator ν. The operators μ,ν bind variables and we assume the notions of sub-formulas, free occurrences of variables and substitutions $\alpha[\beta/X]$ (of all free occurrences of X in α by β) as standard and obvious.

Semantics

Let $\mathcal{T} = (S, s_0, \delta)$ be a transition system over Σ, $v : \mathcal{A} \to 2^S$ be a valuation mapping atomic propositions to subsets of states of S and $e : \mathcal{V} \to 2^S$ be an environment mapping variables to subsets of states of S. In analogy to syntactic substitutions, we introduce the notation $e[X \mapsto U]$ for the environment with $e[X \mapsto U](X) = U$ and $e[X \mapsto U](Y) = e(Y)$ for $Y \neq X$.

The satisfaction of a formula with respect to $\mathcal{T}$, v and e is inductively defined as follows:

- $(\mathcal{T}, s) \models^v_e p$ iff $s \in v(p)$. $(\mathcal{T}, s) \models^v_e \neg p$ iff $s \notin v(p)$.
- $(\mathcal{T}, s) \models^v_e X$ iff $s \in e(X)$.
- $(\mathcal{T}, s) \models^v_e \bigwedge \Phi$ iff for all $\alpha \in \Phi$ it holds that $(\mathcal{T}, s) \models^v_e \alpha$.
- $(\mathcal{T}, s) \models^v_e \bigvee \Phi$ iff for some $\alpha \in \Phi$ it holds that $(\mathcal{T}, s) \models^v_e \alpha$.
- $(\mathcal{T}, s) \models^v_e \Box\alpha$ iff for all $a \in \Sigma, s' \in S$ with $(s, a, s') \in \delta$ we have $(\mathcal{T}, s') \models^v_e \alpha$.
- $(\mathcal{T}, s) \models^v_e \Diamond\alpha$ iff there exists $a \in \Sigma, s' \in S$ with $(s, a, s') \in \delta$ and $(\mathcal{T}, s') \models^v_e \alpha$.
- $(\mathcal{T}, s) \models^v_e \mu X.\alpha$ iff $s \in \bigcap\{U \subseteq S \mid U \supseteq \{s' \mid (\mathcal{T}, s') \models^v_{e[X \mapsto U]} \alpha\}\}$.
- $(\mathcal{T}, s) \models^v_e \nu X.\alpha(X)$ iff $s \in \bigcup\{U \subseteq S \mid U \subseteq \{s' \mid (\mathcal{T}, s') \models^v_{e[X \mapsto U]} \alpha\}\}$.

Note that $(\mathcal{T}, s) \models^v_e \alpha$ iff $(\mathcal{T}, s) \models^v_{e'} \alpha$ for all environments e and e' and all states s as soon as the formula α is closed. So, we write $(\mathcal{T}, s) \models^v \alpha$ to refer to the semantics of a closed formula. We say that a *transition system* $\mathcal{T}$ *with valuation* v *satisfies* α if $(\mathcal{T}, s_0) \models^v \alpha$, i.e. if α is satisfied at the initial state.

Despite the absence of explicit negation in the logic, it is immediate to find for each formula its *dual formula* $\overline{\alpha}$ expressing its negation:

$$\begin{array}{ll} \overline{p} = \neg p & \overline{\neg p} = p \\ \overline{X} = X & \overline{\Box\alpha} = \Diamond\overline{\alpha} \\ \overline{\bigwedge \Phi} = \bigvee\{\overline{\alpha} \mid \alpha \in \Phi\} & \overline{\bigvee \Phi} = \bigwedge\{\overline{\alpha} \mid \alpha \in \Phi\} \\ \overline{\mu X.\alpha} = \nu X.\overline{\alpha} & \overline{\nu X.\alpha} = \mu X.\overline{\alpha} \end{array}$$

Proposition 2. *(1)* $\overline{\overline{\alpha}} = \alpha$. *(2) For a closed formula* α *we have* $(\mathcal{T}, s) \models^v \overline{\alpha}$ *iff* $(\mathcal{T}, s) \not\models^v \alpha$.

Disjunctive formulas

For algorithmic manipulation and the formulation of proof systems, certain normal forms are beneficial compared to the syntactically full μ-calculus. This is particularly the case for so called *guarded* [7] *disjunctive formulas* [5], a subclass of formulas that does not impose any semantic restriction. That is, any formula of the μ-calculus can be transformed into a disjunctive formula – at the price of a potential exponential blowup.

There are two parts to this normal form.

Definition 3. We say that a fixpoint of the form $\sigma X.\alpha$ is *guarded* if all free occurrences of the variable X in α appear in sub-formulas of α of the form $\Box\beta$ or $\Diamond\beta$. We say that a formula α is *guarded* if all fixpoints of α are guarded.

Next, we introduce a new operator[2] $\rightarrow$ and its dual operator $\Rightarrow$ defined below:

$$\rightarrow \Phi = \bigwedge\{\Diamond\alpha \mid \alpha \in \Phi\} \wedge \Box \bigvee \Phi \text{ and } \Rightarrow \Phi = \bigvee\{\Box\alpha \mid \alpha \in \Phi\} \vee \Diamond \bigwedge \Phi$$

where Φ is a finite set of formulas of μ-calculus. These operators can thus either be seen as macros based on existing operators or as actual syntactic extensions with the semantics defined according to the above syntactic definition.

We can use these new operators to replace the usual ones: For instance, the formula $\Diamond\alpha$ can be rewritten as $\rightarrow \{\alpha, \top\}$ and the formula $\Box\alpha$ as $\rightarrow \{\alpha\} \vee \rightarrow \emptyset$, where $\top$ is some formula that is always true, e.g. $\top := \bigwedge \emptyset$ is a valid choice.

We furthermore extend the notation of dual formulas to sets of formulas with $\rightarrow$ and $\Rightarrow$:

$$\overline{\rightarrow \Phi} \quad = \quad \Rightarrow \{\overline{\alpha} \mid \alpha \in \Phi\} \qquad \overline{\Rightarrow \Phi} \quad = \quad \rightarrow \{\overline{\alpha} \mid \alpha \in \Phi\}$$

This extension is coherent with the syntactic definitions of $\rightarrow$ and $\Rightarrow$ and obviously preserves the validity of Proposition 2.

Note also, that the notion of guarded and closed formulas extends to formulas with the new operators in an obvious way.

Now we can present the special set of formulas that we will use in the sequel of this paper:

Definition 4. (Janin, Walukiewicz) The set of *disjunctive formulas* is defined inductively as follows.

- all variables X,
- all disjunctions of disjunctive formulas,

[2] The version of the μ-calculus used here has no action names in modalities. The operator as described in [5] is written $\xrightarrow{a}$ and is restricted to transitions labeled a in the modalities $\langle a \rangle$ rather than $\Diamond$ and $[a]$ rather than $\Box$.

– all conjunctions of the form $\bigwedge L \wedge \rightarrow \Phi$ where Φ is a finite set of disjunctive formulas and L is a finite set of literals (p or $\neg p$),
– all formulas of the form $\mu X.\beta$ or $\nu X.\beta$ where β is a disjunctive formula.

Note that there is a slight adaptation from [5] in this definition. The latter allows also conjunctions $\bigwedge L$ with no component $\rightarrow \Phi$. It is however no problem to "add" such components without changing the semantics: The formula $\nu X.(\rightarrow \emptyset \vee \rightarrow \{X\})$ is true at any state in any transition system. Hence, we can rewrite the conjunction $\bigwedge L$ as $(\bigwedge L \wedge \rightarrow \emptyset) \vee (\bigwedge L \wedge \rightarrow \{\nu X.(\rightarrow \emptyset \vee \rightarrow \{X\})\})$.

The following Proposition summarizes the essential results of works by Kozen [7] and Janin and Walukiewicz [5] respectively on transformations of μ-calculus formulas into normal forms.

Proposition 5. *(Kozen,Walukiewicz,Janin) For each (closed) formula in the propositional μ-calculus, there exists an (semantically) equivalent (closed) guarded disjunctive formula.*

In the following we consider only closed guarded disjunctive formulas.

4 Model checking and derivation graphs

There is a solid body of literature on the model checking problem [4,3,11] and the satisfiability problem [10] for the μ-calculus. The principle task of model checking algorithms is to check whether $(\mathcal{T}, s_0) \models^v \varphi$. However, by nature model checking algorithms can provide diagnostic information, like "counter examples" in linear time model checking.

In the case of branching time logics like the propositional μ-calculus, the nature of diagnostic information is less obvious from a user's perspective, but most model checking procedures can provide at little additional cost an object resembling a "proof", which we will formalize as *derivation graph* below. In the next section, we will use these derivation graphs for controller synthesis.

Pre-derivation graphs

If it were not for the fixpoints, the semantics given in Section 3 immediately allows to define a kind of proof objects, which we will call pre-derivation graphs for now:

Definition 6. Let $\mathcal{T} = (S, s_0, \delta)$ be a transition system over a set of actions Σ, $v : \mathcal{A} \rightarrow S$ be a valuation of $\mathcal{T}$ and α be a closed formula. A *pre-derivation graph* for $\mathcal{T}$, α and v is a (labeled) graph $\mathcal{G} = (V, \longrightarrow)$ where V is a set of *nodes*, pairs $\langle s, \beta \rangle$ where $s \in S$ is a state and β a formula is a set of nodes and $\longrightarrow \subseteq V \times (\Sigma \cup \{\varepsilon\}) \times V$ is a set of edges (also called the derivation relation) such that $\mathcal{G}$ satisfies the following conditions:

(v) If $\langle s, p \rangle \in V$ then $s \in v(p)$ and if $\langle s, \neg p \rangle \in V$ then $s \notin v(p)$.

($\vee$) If $\langle s, \bigvee \Phi\rangle \in V$ then there exists a unique $\beta \in \Phi$ such that $\langle s, \beta\rangle \in V$ and $\langle s, \bigvee \Phi\rangle \xrightarrow{\varepsilon} \langle s, \beta\rangle$.

($\wedge$) If $\langle s, \bigwedge \Phi\rangle \in V$ then for all $\beta \in \Phi$, $\langle s, \beta\rangle \in V$ and $\langle s, \bigwedge \Phi\rangle \xrightarrow{\varepsilon} \langle s, \beta\rangle$.

($\Diamond$) If $\langle s, \Diamond\beta\rangle \in V$ then there exists a transition $(s, a, s') \in \delta$ such that $\langle s', \beta\rangle \in V$ and $\langle s, \Diamond\beta\rangle \xrightarrow{a} \langle s', \beta\rangle$.

($\Box$) If $\langle s, \Box\beta\rangle \in V$ then for all transitions $(s, a, s') \in \delta$, $\langle s', \beta\rangle \in V$ and $\langle s, \Box\beta\rangle \xrightarrow{a} \langle s', \beta\rangle$.

($\rightarrow$) If $\langle s, \rightarrow \Phi\rangle \in V$ then for all formulas $\beta \in \Phi$, there exists a transition $(s, a_\beta, s'_\beta) \in \delta$ such that $\langle s'_\beta, \beta\rangle \in V$ and $\langle s, \rightarrow \Phi\rangle \xrightarrow{a_\beta} \langle s'_\beta, \beta\rangle$ and on other hand for all transitions $(s, a, s') \in \delta$, there exists some $\beta \in \Phi$ with $\langle s', \beta\rangle \in V$ and $\langle s, \rightarrow \Phi\rangle \xrightarrow{a} \langle s', \beta\rangle$.

($\Rightarrow$) If $\langle s, \Rightarrow \Phi\rangle \in V$ then either there exists a formula $\beta \in \Phi$ such that for all transitions $(s, a, s') \in \delta$, $\langle s', \beta\rangle \in V$ and $\langle s, \Rightarrow \Phi\rangle \xrightarrow{a} \langle s', \beta\rangle$ or there exists a transition $(s, a, s') \in \delta$ such that for all formulas $\beta \in \Phi$, $\langle s', \beta\rangle \in V$ and $\langle s, \Rightarrow \Phi\rangle \xrightarrow{a} \langle s', \beta\rangle$.

(μ) If $\langle s, \mu X.\beta\rangle \in V$ then $\langle s, \beta[\mu X.\beta/X]\rangle \in V$ and $\langle s, \mu X.\beta\rangle \xrightarrow{\varepsilon} \langle s, \beta[\mu X.\beta/X]\rangle$.

(ν) If $\langle s, \nu X.\beta\rangle \in V$ then $\langle s, \beta[\nu X.\beta/X]\rangle \in V$ and $\langle s, \nu X.\beta\rangle \xrightarrow{\varepsilon} \langle s, \beta[\nu X.\beta/X]\rangle$.

Supposing that a formula α contains no fixpoint then it is obvious that $\mathcal{T}, s \models \alpha$, if a derivation graph containing the node $\langle s, \alpha\rangle$ exists. Moreover, the nodes $\langle s', \beta\rangle$ can be restricted to $s' \in S$ and β a sub-formula of α. Supposing that α only contains the greatest fixpoint operator νX, then if $\mathcal{T}, s \models \alpha$ holds, a pre-derivation graph can be defined in an inductive manner and conversely a from a pre-derivation graph assigning $U = \{s \mid \langle s, \nu X.\beta\rangle \in V\}$ the semantic definition of immediately yields $\mathcal{T}, s \models \nu X.\beta$ for all $s \in U$, a reasoning due to Park [8].

Approximations of Fixpoints

However, least fixpoints are not correctly characterized by the existence of pre-derivation graphs. Without reexploring the full complexity of the issue, it is useful for the understanding to recall an approximation approach to the semantics of fixpoints:

Let $\mu^0 X.\alpha = \bigvee \emptyset$ and $\mu^{n+1} X.\alpha = \alpha[\mu^n X.\alpha/X]$ be approximations of a least fixpoint $\mu X.\alpha$ and let $U_n = \{s \mid (\mathcal{T}, s) \models \mu^n X.\alpha\}$. Monotonicity implies that $\emptyset = U_0 \subseteq U_1 \subseteq U_2 \ldots \subseteq U_n \subseteq U_{n+1}$ and since $U_n \subseteq S$ are finite sets, there is some k with $U_k = U_{k+1}$ which is easily seen to be the actual least fixpoint. This is the essential content of Kleene's fixpoint theorem. Dually, greatest fixpoints are approximated by $\nu^0 X.\alpha = \bigwedge \emptyset$ and $\nu^{n+1} X.\alpha = \alpha[\nu^n X.\alpha/X]$. In [10], it is shown (for multiple fixpoints in a formula) how to extend this reasoning to the generalized Kleene theorem with transfinite approximations where μ^n is generalized to μ^κ, κ an ordinal (for limit ordinals λ like e.g. ω we define $U_\lambda = \bigcup_{\kappa<\lambda} U_\kappa$). This shows that approximations are not in principle limited to finite state systems. Supposing accordingly a syntactic modification where the least fixpoints are annotated with ordinals, this gives rise to an alternative version of pre-derivation graphs with the following to rules for approximated least fixpoints:

(1) if $\langle s, \mu^{\kappa+1}X.\alpha\rangle \in V$ then also $\langle s, \alpha[\mu^n X.\alpha/X]\rangle \in V$ and $\langle s, \mu^{\kappa+1}X.\alpha\rangle \xrightarrow{\varepsilon} \langle s, \alpha[\mu^n X.\alpha/X]\rangle \in V$; (2) if λ is a limit ordinal and $\langle s, \mu^{\lambda}X.\alpha\rangle \in V$ then for some $\kappa < \lambda$ also $\langle s, \mu^{\kappa}X.\alpha\rangle \in V$ and $\langle s, \mu^{\lambda}X.\alpha\rangle \xrightarrow{\varepsilon} \langle s, \mu^{\kappa}X.\alpha\rangle \in V$.

From here, [10] obtained a remarkable abstraction: The ordinals in such a pre-derivation graph witness of a certain termination condition. They formalize it as follows:

Definition 7. In a pre-derivation graph $\mathcal{G}$, a least fixpoint formula $\mu X.\beta$ is *regenerated* from a state s to a state s' if there is a non-zero length path from $\langle s, \mu X.\beta\rangle$ to $\langle s', \mu X.\beta\rangle$ in $\mathcal{G}$ such that any intermediate node $\langle s'', \gamma\rangle$ on the path is such that $\mu X.\beta$ is a sub-formula of γ.

A least fixpoint $\mu X.\beta$ is *infinitely regenerated* if there exists an infinite sequence $s_1, s_2, \ldots$ of (not necessarily pairwise different[3]) states of S such that $\mu X.\beta$ is regenerated from s_i to s_{i+1} for every $i \geq 1$.

Definition 8. A *derivation graph* is a pre-derivation graph with no infinitely regenerated least fixpoint.

The notion of a derivation graph thus avoids explicit approximation ordinals retaining their essential role of termination witnesses. Before stating their theorem, we additionally observe that the set of formulas needed in the (pre-)derivation graphs is limited to the *Fischer-Ladner closure* of the initial formula, a notion of sub-formulas that is compatible with the fixpoint rules:

Definition 9. The *Fischer-Ladner closure* of a formula α, is the smallest set $FL(\alpha)$ of formulas satisfying the following constraints:

1. $\alpha \in FL(\alpha)$,
2. if $\beta \vee \gamma \in FL(\alpha)$ then $\beta, \gamma \in FL(\alpha)$, if $\beta \wedge \gamma \in FL(\alpha)$ then $\beta, \gamma \in FL(\alpha)$,
3. if $\Diamond\beta \in FL(\alpha)$ then $\beta \in FL(\alpha)$, if $\Box\beta \in FL(\alpha)$ then $\beta \in FL(\alpha)$,
4. if $\mu X.\beta \in FL(\alpha)$ then $\beta[\mu X.\beta/X] \in FL(\alpha)$, if $\nu X.\beta(X) \in FL(\alpha)$ then $\beta[\nu X.\beta/X] \in FL(\alpha)$,
5. if $\rightarrow \Phi \in FL(\alpha)$ then $\Phi \subseteq FL(\alpha)$, if $\Rightarrow \Phi \in FL(\alpha)$ then $\Phi \subseteq FL(\alpha)$.

Example 10. The Fischer-Ladner closure of the formula $\mu X.(p\wedge \rightarrow \{\neg q, X\})$ consists of the following five formulas: $\mu X.(p\wedge \rightarrow \{\neg q, X\})$, $p\wedge \rightarrow \{\neg q, \mu X.(p\wedge \rightarrow \{\neg q, X\})\}$, $p, \rightarrow \{\neg q, \mu X.(p\wedge \rightarrow \{\neg q, X\})\}$, $\neg q$.

Proposition 11. *The cardinality of the Fischer-Ladner closure of a formula α is bounded by the (printed) length of α: $|FL(\alpha)| \leq |\alpha|$.*

The Fischer-Ladner closure of a closed/guarded/disjunctive formula contains only closed/guarded/disjunctive formulas.

Note that $|FL(\alpha)|$ may be much smaller than $|\alpha|$ due to the implicit sharing of common sub-formulas.

Now we can state the desired result concerning derivation graphs:

[3] On a finite state transition system, the definition is equivalent to a cyclic regeneration from s back to the same s.

Proposition 12 (Streett-Emerson [10]). *For a transition system $\mathcal{T} = (S, s_0, \delta)$, valuation v and closed formula α it holds that $(\mathcal{T}, s) \models^v \alpha$ iff there exists a derivation graph $\mathcal{G} = (V, \longrightarrow)$ with $\langle s, \alpha \rangle \in V$. Moreover, $|V| \leq |S| \cdot |FL(\alpha)|$ and $| \longrightarrow | \leq |\delta| \cdot |FL(\alpha)|$.*

Our above discussion ignores some technical difficulties due to simultaneous fixpoints, but the idea we want to recall is that the characterisation is based on approximations.

Model checking generates derivation graphs

The aim of the following discussion is to give hints why algorithms like [4,3] can easily be adapted to actually compute derivation graphs as a by product. By nature, they compute a satisfaction relation that can serve as a set of nodes V of a derivation graph and the backward propagation over rules shows that the algorithms readily compute pre-derivation graphs. But even for systems where a property holds, an arbitrary pre-derivation graph is rarely a derivation graph: Disjunctions require a choice that assures finite regeneration of least fixpoints.

The algorithms in question actually evaluate fixpoints for finite transition systems based on approximations (for both greatest and least fixpoints). A naïve view would be to consider a nested of fixpoints like a hierarchy of loops that have to be iterated until stabilisation (at some finite ordinal). For a state s a formula $\beta[\mu X.\gamma/X]$ containing an outermost least fixpoint sub-formula $\mu X.\gamma$ this means that there is a least n where $\mathcal{T}, s \models \beta[\mu^n X.\gamma/X]$ is found by backward propagation, e.g. if $\beta = \beta_1 \vee \beta_2$ then the smallest n_1 such that $\mathcal{T}, s \models \beta_1[\mu^{n_1} X.\gamma/X]$ or (or and) a smalles n_2 such that $\mathcal{T}, s \models \beta_2[\mu^{n_2} X.\gamma/X]$. Then n is the minimum of n_1 and n_2 and (in accord with [10]) the β_i with the smaller n_i as justification $\stackrel{\varepsilon}{\longrightarrow}$ is a safe choice for obtaining a derivation graph.

5 Synthesizing a controller through a control formula

In Section 2, we generally introduced the notion of a controller. We still have to make precise the notion of a controller achieving a property (in the μ-calculus). First, we have to define what it means for a proposition to hold at a state in a synchronous product.

Let $\mathcal{S} = (S, s_0, \delta)$ be a controllable states transition system over a set of actions Σ, v a valuation from $\mathcal{A}$ to 2^S and $\mathcal{T} = (S', s'_0, \delta')$ be another transition system over Σ. Then for the synchronous product $\mathcal{S} \times \mathcal{T}$, we define $v_\times$ to be the valuation from $\mathcal{A}$ to $2^{S \times S'}$ such that $v_\times(p) = \{(s, s') \in S \times S' \mid s \in v(p)\}$.

For a given μ-calculus formula α and valuation v, the control synthesis problem is now to find a controller $\mathcal{T}$ for $\mathcal{S}$, such that $\mathcal{S} \times \mathcal{T} \models^{v_\times} \alpha$.

Our approach to this problem consists of three steps:

- Derive a *control formula* $\varsigma(\alpha)$ from α, such that a controller $\mathcal{T}$ as required exists iff $\mathcal{S}$ satisfies $\varsigma(\alpha)$

– Verify that $\mathcal{S}$ satisfies $\varsigma(\alpha)$ by model checking and if it holds, obtain a derivation graph $\mathcal{G}$.
– Obtain a transition system $\mathcal{T}_\mathcal{G}$ from the derivation graph that is provably a controller achieving α.

Control formula. Let α be a closed and guarded disjunctive formula over $\mathcal{A}$ and $\mathcal{V}$. We moreover assume an extension $\mathcal{A}_c = \mathcal{A} \cup \{c\}$ where c is a fresh atomic proposition intended to represent controllability: For a given valuation $v : \mathcal{A} \longrightarrow 2^S$, let $v_c : \mathcal{A}_c \longrightarrow 2^S$ such that $v_c(c) = C$ and $v_c(p) = v(p)$ if $p \neq c$.

We inductively build from α a new formula $\varsigma(\alpha)$ (not necessary in a disjunctive form) called *control formula*:

– $\varsigma(l) = l$ for all literals l, $\varsigma(X) = X$ for all variables X,
– $\varsigma(\bigvee \Phi) = \bigvee\{\varsigma(\alpha) \mid \alpha \in \Phi\}$, $\varsigma(\bigwedge \Phi) = \bigwedge\{\varsigma(\alpha) \mid \alpha \in \Phi\}$,
– $\varsigma(\rightarrow \Phi) = (c \wedge \bigwedge\{\Diamond\varsigma(\alpha) \mid \alpha \in \Phi\}) \vee (\neg c \wedge \rightarrow \{\varsigma(\alpha) \mid \alpha \in \Phi\})$,
– $\varsigma(\mu X.\alpha) = \mu X.\varsigma(\alpha)$, $\varsigma(\nu X.\alpha) = \nu X.\varsigma(\alpha)$.

The idea is the formula α expresses the property that the controlled system $\mathcal{S} \times \mathcal{T}$ must satisfy whereas the control formula $\varsigma(\alpha)$ expresses whether we can control the controllable system $\mathcal{S}$ in order to satisfy α.

Example 13. Let $\nu X.\mu Y.((\rightarrow \{X\} \wedge p) \vee \rightarrow \{Y\})$ be a disjunctive formula. The corresponding control formula is:

$$\nu X.\mu Y.(([(c \wedge \Diamond X) \vee (\neg c \wedge \rightarrow \{X\})] \wedge p) \vee [(c \wedge \Diamond Y) \vee (\neg c \wedge \rightarrow \{Y\})])$$

Proposition 14. *Let α be a closed guarded disjunctive formula. Then $\varsigma(\alpha)$ is closed and guarded (but not disjunctive) and $|FL(\varsigma(\alpha))| \leq O(|FL(\alpha)|)$.*

From derivation graphs to controllers. We explain here how we turn a derivation graph $\mathcal{G}$ into a controller $\mathcal{T}_\mathcal{G}$.

Let $\mathcal{G} = (V, \longrightarrow)$ be a derivation graph for a transition system $\mathcal{S} = (S, s_0, \delta)$ over Σ, a formula α and a valuation v. We say that a node $\langle s, \beta\rangle$ derives to a node $\langle s', \beta'\rangle$ by ε-transitions and we write $\langle s, \beta\rangle \overset{\varepsilon}{\rightsquigarrow} \langle s', \beta'\rangle$ if $\langle s, \beta\rangle = \langle s', \beta'\rangle$ or $\langle s, \beta\rangle \xrightarrow{\varepsilon} \ldots \xrightarrow{\varepsilon} \langle s', \beta'\rangle$. Then $\langle s, \beta\rangle \overset{a}{\rightsquigarrow} \langle s', \beta'\rangle$ for an action $a \in \Sigma$ if there is a node $\langle s'', \beta''\rangle \in V$ with $\langle s, \beta\rangle \overset{\varepsilon}{\rightsquigarrow} \langle s'', \beta''\rangle \xrightarrow{a} \langle s', \beta'\rangle$ in $\mathcal{G}$.

Proposition 15. *Let S be a deterministic transition system, α a closed guarded disjunctive formula over $\mathcal{A}$ and $\mathcal{V}$, and v be a valuation from $\mathcal{A}$ to 2^S. Let $\mathcal{G} = (V, \longrightarrow)$ be a derivation graph for S, α and v (if it exists). Then for all nodes $\langle s, \beta\rangle \in V$ there exists a unique $\langle s, \rightarrow \Phi\rangle \in V$ such that $\langle s, \beta\rangle \overset{\varepsilon}{\rightsquigarrow} \langle s, \rightarrow \Phi\rangle$.*

The transition system $\mathcal{T}_\mathcal{G}$ over Σ is the tuple $(S_\mathcal{G}, \langle s_0, \alpha\rangle, \delta_\mathcal{G})$ where $S_\mathcal{G}$ and simultaneously $\delta_\mathcal{G}$ are inductively defined to be the least set/relation such that $\langle s_0, \alpha\rangle \in S_\mathcal{G}$ and for all $\langle s, \beta\rangle \in S_\mathcal{G}$ and $a \in \Sigma$ if $\langle s, \beta\rangle \overset{a}{\rightsquigarrow} \langle s', \beta'\rangle$ in $\mathcal{G}$ then $\langle s', \beta'\rangle \in S_\mathcal{G}$ and$(\langle s, \beta\rangle, a, \langle s', \beta'\rangle) \in \delta_\mathcal{G}$.

Now we can state the main theorem of this work:

Theorem 16. *Let $\mathcal{S}$ be a controllable state transition system, α a disjunctive, guarded and closed μ-calculus formula, v an evaluation, v_c, v_x defined accordingly.*

1. *If $\mathcal{S} \models^{v_c} \varsigma(\alpha)$ then for all derivation graphs $\mathcal{G}$ for $\mathcal{S}$ and $\varsigma(\alpha)$, $\mathcal{T}_\mathcal{G}$ is a controller for $\mathcal{S}$ and $\mathcal{S} \times \mathcal{T}_\mathcal{G} \models^{v_\times} \alpha$.*
2. *If there exists a controller $\mathcal{C}$ such that $\mathcal{S} \times \mathcal{C} \models^{v_\times} \alpha$, then $\mathcal{S} \models^{v_c} \varsigma(\alpha)$.*

The rest of this section is devoted to the proof of Theorem 16 and is done in several steps.

Proof of Theorem 16, part (1)

Lemma 17. *All accessible states in $\mathcal{S} \times \mathcal{T}_\mathcal{G}$ are of the form $(s, \langle s, \beta\rangle)$.*

Proof. Let $\mathcal{S} \times \mathcal{T}_\mathcal{G}$ be the synchronous product $(S \times S_\mathcal{G}, (s_0, \langle s_0, \alpha\rangle), \delta_\times)$. We use an induction on the distance from $(s_0, \langle s_0, \alpha\rangle)$ in $\mathcal{S} \times \mathcal{T}_\mathcal{G}$. First the initial state $(s_0, \langle s_0, \alpha\rangle)$ satisfies the invariant. Suppose that $(s_1, \langle s_1, \beta\rangle)$ is an accessible state of $\mathcal{S} \times \mathcal{T}_\mathcal{G}$ and $((s_1, \langle s_1, \beta\rangle), a, (s_2, \langle s_3, \gamma\rangle) \in \delta_\times$. Then $(s_1, a, s_2) \in \delta$ and $(\langle s_1, \beta\rangle, a, \langle s_3, \gamma\rangle) \in \delta_\mathcal{G}$. By construction of $\mathcal{T}_\mathcal{G}$, $\langle s_1, \beta\rangle \overset{a}{\rightsquigarrow} \langle s_3, \gamma\rangle$ in $\mathcal{G}$. It follows from the definition of $\mathcal{G}$ that $(s_1, a, s_3) \in \delta$ and then $s_2 = s_3$ because $\mathcal{S}$ is deterministic.

Proposition 18. *$\mathcal{T}_\mathcal{G}$ is a controller for $\mathcal{S}$ provided α is a control formula.*

Proof. Let $\mathcal{S} \times \mathcal{T}_\mathcal{G}$ be the synchronous product $(S \times S_\mathcal{G}, (s_0, \langle s_0, \alpha\rangle), \delta_\times)$. Suppose that $(s, \langle s, \beta\rangle)$ is accessible in $\mathcal{S} \times \mathcal{T}_\mathcal{G}$, $s \notin C$ and $(s, a, s') \in \delta$. Since α is a control formula there is a formula $\gamma \in FL(\alpha)$ such that $\gamma = (c \wedge \bigwedge\{\Diamond\gamma' \mid \gamma' \in \Phi\}) \vee (\neg c \wedge \rightarrow \Phi)$ and $\langle s, \beta\rangle \overset{\varepsilon}{\rightsquigarrow} \langle s, \gamma\rangle \in \mathcal{G}$. Moreover $\langle s, \gamma\rangle \xrightarrow{\varepsilon} \langle s, \neg c \wedge \rightarrow \Phi\rangle \xrightarrow{\varepsilon} \langle s, \rightarrow \Phi\rangle$ because $s \notin C$. Then following the definition of $\mathcal{G}$ there is a formula $\gamma' \in \Phi$ such that $\langle s, \rightarrow \Phi\rangle \xrightarrow{a} \langle s', \gamma'\rangle$ and this because $(s, a, s') \in \delta$. It follows that $\langle s, \beta\rangle \overset{a}{\rightsquigarrow} \langle s', \gamma'\rangle$ that is $(\langle s, \beta\rangle, a, \langle s', \gamma'\rangle) \in \delta_\mathcal{G}$. Consequently $((s, \langle s, \beta\rangle), a, (s', \langle s', \gamma'\rangle)) \in \delta_\times$.

Lemma 19. *If $\mathcal{S} \models^{v_c} \varsigma(\alpha)$ then for any derivation graph $\mathcal{G}$ for $\mathcal{S}$, $\varsigma(\alpha)$ and v_c is such that $\mathcal{S} \times \mathcal{T}_\mathcal{G} \models^{v_\times} \alpha$.*

Proof. Let $\mathcal{S} = (S, s_0, \delta)$ be the controllable state transition system and let $\mathcal{G} = (V, \longrightarrow)$ be a derivation graph for $\mathcal{S}$, $\varsigma(\alpha)$ and v_c with $(s_0, \varsigma(\alpha)) \in V$. Let $\mathcal{S} \times \mathcal{T}_\mathcal{G}$ be the synchronous product $(S \times S_\mathcal{G}, (s_0, \langle s_0, \varsigma(\alpha)\rangle), \delta_\times)$. We construct from the derivation graph $\mathcal{G}$ a graph $\mathcal{G}_\times = (V_\times, \longrightarrow_\times)$ as it follows:

1. $V_\times = \{\langle (s, \langle s, \beta\rangle), \gamma\rangle \in S \times S_\mathcal{G} \times FL(\alpha) \mid \langle s, \beta\rangle \overset{\varepsilon}{\rightsquigarrow} \langle s, \varsigma(\gamma)\rangle$ and $\langle s, \varsigma(\gamma)\rangle \in V\}$,
2. $\langle (s, \langle s, \beta\rangle), \gamma\rangle \xrightarrow{\varepsilon}_\times \langle (s, \langle s, \beta\rangle), \gamma'\rangle$ iff $\langle s, \varsigma(\gamma)\rangle \xrightarrow{\varepsilon} \langle s, \varsigma(\gamma')\rangle$,
3. $\langle (s, \langle s, \beta\rangle), \rightarrow \Phi\rangle \xrightarrow{a}_\times \langle (s', \langle s', \varsigma(\gamma')\rangle), \gamma'\rangle$ iff $\langle s, \varsigma(\rightarrow \Phi)\rangle \overset{a}{\rightsquigarrow} \langle s', \varsigma(\gamma')\rangle$.

We prove now that $\mathcal{G}_\times$ is really a derivation graph for $\mathcal{S} \times \mathcal{T}_\mathcal{G}$, α and $v_\times$. Let $\langle (s, \langle s, \beta \rangle), \gamma \rangle$ be a node of $V_\times$.

Condition (v): γ is a proposition p. Then $\varsigma(\gamma) = p$ and by (1) $\langle s, \beta \rangle \overset{\varepsilon}{\rightsquigarrow} \langle s, p \rangle$ and $\langle s, p \rangle \in V$. Since $\mathcal{G}$ is a derivation graph $s \in v(p)$. It follows that $(s, \langle s, \beta \rangle) \in v_\times(p)$.

Condition ($\vee$): γ is of the form $\bigvee \Phi$. Then $\varsigma(\bigvee \Phi) = \bigvee \{\varsigma(\gamma') \mid \gamma' \in \Phi\}$ and by (1) $\langle s, \beta \rangle \overset{\varepsilon}{\rightsquigarrow} \langle s, \varsigma(\gamma) \rangle$ and $\langle s, \varsigma(\gamma) \rangle \in V$. Since $\mathcal{G}$ is a derivation graph there exists a unique $\gamma' \in \Phi$ such that $\langle s, \varsigma(\gamma') \rangle \in V$ and $\langle s, \varsigma(\gamma) \rangle \xrightarrow{\varepsilon} \langle s, \varsigma(\gamma') \rangle$. It follows by (1) that $\langle (s, \langle s, \beta \rangle), \gamma' \rangle \in V_\times$ and by (2) that $\langle (s, \langle s, \beta \rangle), \gamma \rangle \xrightarrow{\varepsilon} \langle (s, \langle s, \beta \rangle), \gamma' \rangle$.

Conditions ($\wedge$), (μ) and (ν): the proofs are similar to the previous case ($\vee$). Moreover we have not to deal with the conditions ($\Diamond$), ($\Box$) and ($\Rightarrow$) because α is in a disjunctive form and then these operators never appear in formulas of $FL(\alpha)$. The last case to deal with is the condition ($\rightarrow$).

Condition ($\rightarrow$): γ is of the form $\rightarrow \Phi$. Then $\varsigma(\rightarrow \Phi) = (c \wedge \bigwedge \{\Diamond \varsigma(\gamma') \mid \gamma' \in \Phi\}) \vee (\neg c \wedge \rightarrow \{\varsigma(\gamma') \mid \gamma' \in \Phi\})$. By (1) $\langle s, \beta \rangle \overset{\varepsilon}{\rightsquigarrow} \langle s, \varsigma(\gamma) \rangle$ and $\langle s, \varsigma(\gamma) \rangle \in V$.

If s is an uncontrollable state (that is $s \notin C$): Then $\langle s, \varsigma(\gamma) \rangle \overset{\varepsilon}{\rightsquigarrow} \langle s, \rightarrow \{\varsigma(\gamma') \mid \gamma' \in \Phi\} \rangle$. Following the condition ($\rightarrow$): On one hand, for all formulas $\gamma' \in \Phi$, there exists a transition $(s, a_{\gamma'}, s_{\gamma'}) \in \delta$ such that $\langle s_{\gamma'}, \varsigma(\gamma') \rangle \in V$ and $\langle s, \rightarrow \{\varsigma(\gamma') \mid \gamma' \in \Phi\} \rangle \xrightarrow{a_{\gamma'}} \langle s_{\gamma'}, \varsigma(\gamma') \rangle$. This means that $\langle s, \beta \rangle \overset{a_{\gamma'}}{\rightsquigarrow} \langle s_{\gamma'}, \varsigma(\gamma') \rangle$ i.e. $(\langle s, \beta \rangle, a_{\gamma'}, \langle s_{\gamma'}, \varsigma(\gamma') \rangle) \in \mathcal{T}_\mathcal{G}$. Consequently for all $\gamma' \in \Phi$ there exists in the product $\mathcal{S} \times \mathcal{T}_\mathcal{G}$ a transition $((s, \langle s, \beta \rangle), a_{\gamma'}, (s_{\gamma'}, \langle s_{\gamma'}, \varsigma(\gamma') \rangle)) \in \delta_\times$ such that by (1) $\langle (s_{\gamma'}, \langle s_{\gamma'}, \varsigma(\gamma') \rangle), \gamma' \rangle \in V_\times$ and by (3) $\langle (s, \langle s, \beta \rangle), \rightarrow \Phi \rangle \xrightarrow{a_{\gamma'}}_\times \langle (s_{\gamma'}, \langle s_{\gamma'}, \varsigma(\gamma') \rangle), \gamma' \rangle$. On other hand for all transitions $((s, \langle s, \beta \rangle), a, (s', \langle s', \beta' \rangle))$ in $\delta_\times$ we have $(s, a, s') \in \delta$, $\langle s', \beta' \rangle \in V$ and $\langle s, \beta \rangle \overset{\varepsilon}{\rightsquigarrow} \langle s, \rightarrow \{\varsigma(\gamma') \mid \gamma' \in \Phi\} \rangle \xrightarrow{a} \langle s', \beta' \rangle$. Then there exists some $\gamma' \in \Phi$ such that $\varsigma(\gamma') = \beta'$. Consequently by (1) $\langle (s', \langle s', \varsigma(\gamma') \rangle), \gamma' \rangle \in V_\times$ and by (3) $\langle (s, \langle s, \beta \rangle), \rightarrow \Phi \rangle \xrightarrow{a}_\times \langle (s', \langle s', \varsigma(\gamma') \rangle), \gamma' \rangle$.

If s is a controllable state: Then for all formulas $\gamma' \in \Phi$ $\langle s, \varsigma(\gamma) \rangle \overset{\varepsilon}{\rightsquigarrow} \langle s, \Diamond \varsigma(\gamma') \rangle$. Following the condition ($\Diamond$), there exists a transition $(s, a_{\gamma'}, s_{\gamma'}) \in \delta$ such that $\langle s_{\gamma'}, \varsigma(\gamma') \rangle \in V$ and $\langle s, \Diamond \varsigma(\gamma') \rangle \xrightarrow{a_{\gamma'}} \langle s_{\gamma'}, \varsigma(\gamma') \rangle$. This means that $\langle s, \beta \rangle \overset{a_{\gamma'}}{\rightsquigarrow} \langle s_{\gamma'}, \varsigma(\gamma') \rangle$ i.e. $(\langle s, \beta \rangle, a_{\gamma'}, \langle s_{\gamma'}, \varsigma(\gamma') \rangle) \in \mathcal{T}_\mathcal{G}$. Consequently for all formulas $\gamma' \in \Phi$ $((s, \langle s, \beta \rangle), a_{\gamma'}, (s_{\gamma'}, \langle s_{\gamma'}, \varsigma(\gamma') \rangle)) \in \delta_\times$. Then by (1) $\langle (s_{\gamma'}, \langle s_{\gamma'}, \varsigma(\gamma') \rangle), \gamma' \rangle \in V_\times$ and by (3) $\langle (s, \langle s, \beta \rangle), \rightarrow \Phi \rangle \xrightarrow{a_{\gamma'}}_\times \langle (s_{\gamma'}, \langle s_{\gamma'}, \varsigma(\gamma') \rangle), \gamma' \rangle$. On the other hand, following the construction of $\mathcal{T}_\mathcal{G}$, all possible transitions from $(s, \langle s, \beta \rangle)$ are exactly the previous set $\{((s, \langle s, \beta \rangle), a_{\gamma'}, (s_{\gamma'}, \langle s_{\gamma'}, \varsigma(\gamma') \rangle)) \in \delta_\times \mid \gamma' \in \Phi\}$. It follows that the condition ($\rightarrow$) is satisfied.

Infinitely regenerated condition. It remains to prove the global condition on the least fixpoint. We show that if $\mathcal{G}_\times$ is not a derivation graph because of an infinite regeneration of a least fixpoint, then $\mathcal{G}$ already contains an infinite regeneration and is not a derivation graph – contradicting assumptions.

Suppose that there exists a regeneration from a node $\langle (s_1, \langle s_1, \beta_1 \rangle), \mu X.\gamma \rangle$ to a node $\langle (s_2, \langle s_2, \beta_2 \rangle), \mu X.\gamma \rangle$ then based on the construction of $G_\times$ from $\mathcal{G}$ there exists a regeneration from $\langle s_1, \varsigma(\mu X.\gamma) \rangle$ to $\langle s_2, \varsigma(\mu X.\gamma) \rangle$. Structurally, the

only difference in the derivations concerns sub-formulas $\rightarrow \Phi$, which have less structure than their translations $\varsigma(\rightarrow \Phi)$. However, if $\mu X.\gamma$ occurs as sub-formula of some formula $\beta \in \Phi$ then $\varsigma(\mu X.\gamma)$ occurs accordingly as sub-formula of $\varsigma(\beta)$ in $\varsigma(\rightarrow \Phi)$. Useful properties of ς for understanding this simulation of regeneration are: $\varsigma(\alpha_1[\alpha_2/X]) = \varsigma(\alpha_1)[\varsigma(\alpha_2)/X])$ and in particular $\varsigma(\mu X.\gamma) = \mu X.\varsigma(\gamma) = \varsigma(\gamma)[\mu X.\varsigma(\gamma)/X] = \varsigma(\gamma)[\varsigma(\mu X.\gamma)/X] = \varsigma(\gamma[\mu X.\gamma/X])$.

Applying the above construction, an infinitely regenerated least fixpoint in the graph $\mathcal{G}_\times$ gives us an infinitely regenerated least fixpoint in $\mathcal{G}$.

Finally, since $\mathcal{S} \models^{v_c} \varsigma(\alpha)$ then $\langle s_0, \varsigma(\alpha)\rangle \in V$. Consequently by (1), we have $\langle(s_0, \langle s_0, \varsigma(\alpha)\rangle), \alpha\rangle \in V_\times$ that is $\mathcal{S} \times \mathcal{T}_\mathcal{G} \models^{v_\times} \alpha$.

Proof of Theorem 16, part (2)

The second part of Theorem 16 is shown in the inverse sense, based on Proposition 2, part (2): If $\mathcal{S} \not\models^{v_c} \varsigma(\alpha)$ then $\mathcal{S} \models^{v_c} \overline{\varsigma(\alpha)}$. Based on the following Lemma 20 and another application of Proposition 2, we conclude $\mathcal{S} \times \mathcal{C} \not\models^{v_\times} \alpha$.

Lemma 20. *If $\mathcal{S} \models^{v_c} \overline{\varsigma(\alpha)}$ then for all controllers $\mathcal{C}$, $\mathcal{S} \times \mathcal{C} \models^{v_\times} \overline{\alpha}$.*

Proof. Let $\mathcal{S} = (S, s_0, \delta)$ be the controllable state transition system and $\mathcal{C} = (S', s'_0, \delta')$ be a controller for $\mathcal{S}$. Let $\mathcal{S} \times \mathcal{C}$ be the synchronous product $(S \times S', (s_0, s'_0), \delta_\times)$. Since $\mathcal{S} \models^{v_c} \overline{\varsigma(\alpha)}$ there is a derivation graph $\mathcal{G} = (V, \longrightarrow)$ for $\mathcal{S}$, $\overline{\varsigma(\alpha)}$ and v_c with $\langle s_0, \overline{\varsigma(\alpha)}\rangle \in V$. From $\mathcal{G}$ we construct a derivation graph $\mathcal{G}_\times = (V_\times, \longrightarrow_\times)$ for $\mathcal{S} \times \mathcal{C}$, $\overline{\alpha}$ and $v_\times$ as it follows:

1. $V_\times = \{\langle(s, s'), \overline{\beta}\rangle \in S \times S' \times FL(\overline{\alpha}) \mid \langle s, \overline{\varsigma(\beta)}\rangle \in V\}$,
2. $\langle(s, s'), \overline{\beta}\rangle \xrightarrow{\varepsilon}_\times \langle(s, s'), \overline{\beta'}\rangle$ iff $\langle s, \overline{\varsigma(\beta)}\rangle \xrightarrow{\varepsilon} \langle s, \overline{\varsigma(\beta')}\rangle$,
3. $\langle(s, s'), \overline{\rightarrow \Phi}\rangle \xrightarrow{a}_\times \langle(s_1, s'_1), \overline{\beta'}\rangle$ iff $\langle s, \overline{\varsigma(\rightarrow \Phi)}\rangle \overset{a}{\rightsquigarrow} \langle s_1, \overline{\varsigma(\beta')}\rangle$ and $(s', a, s'_1) \in \delta'$.

We prove now that $\mathcal{G}_\times$ is really a derivation graph for $\mathcal{S} \times \mathcal{C}$, $\overline{\alpha}$ and $v_\times$. Let $\langle(s, s'), \overline{\beta}\rangle$ be a node of $V_\times$.

Condition (v): $\overline{\beta}$ is a proposition p. Then $\overline{\varsigma(\beta)} = p$ and by (1) $\langle s, p\rangle \in V$. Since $\mathcal{G}$ is a derivation graph $s \in v(p)$. It follows that $(s, s') \in v_\times(p)$.

Condition ($\vee$): $\overline{\beta}$ is of the form $\bigvee \Phi$. Then $\overline{\varsigma(\beta)} = \bigvee\{\overline{\varsigma(\beta')} \mid \overline{\beta'} \in \Phi\}$ and by (1) $\langle s, \overline{\varsigma(\beta)}\rangle \in V$. Since $\mathcal{G}$ is a derivation graph there exists a unique $\overline{\beta'} \in \Phi$ such that $\langle s, \overline{\varsigma(\beta')}\rangle \in V$ and $\langle s, \overline{\varsigma(\beta)}\rangle \xrightarrow{\varepsilon} \langle s, \overline{\varsigma(\beta')}\rangle$. It follows by (1) that $\langle(s, s'), \overline{\beta'}\rangle \in V_\times$ and by (2) that $\langle(s, s'), \overline{\beta}\rangle \xrightarrow{\varepsilon} \langle(s, s'), \overline{\beta'}\rangle$.

Conditions ($\wedge$), (μ) and (ν): the proofs are similar to the previous case ($\vee$). Moreover we have not to deal with the conditions ($\Diamond$), ($\Box$) and ($\rightarrow$) because α is in a disjunctive form and then these operators never appear in formulas of $FL(\overline{\alpha})$. The last case to deal with is the condition ($\Rightarrow$).

Condition ($\Rightarrow$): $\overline{\beta}$ is of the form $\Rightarrow \Phi$. Then $\overline{\varsigma(\beta)} = (\neg c \vee \bigvee\{\Box\overline{\varsigma(\beta')} \mid \overline{\beta'} \in \Phi\}) \wedge (c \vee \Rightarrow \{\overline{\varsigma(\beta')} \mid \overline{\beta'} \in \Phi\})$ and by (1) $\langle s, \overline{\varsigma(\beta)}\rangle \in V$.

If s is an uncontrollable state (i.e. $s \notin C$): Then following the condition ($\Rightarrow$), either there exists a formula $\overline{\beta'} \in \Phi$ such that for all transitions $(s, a, s_1) \in \delta$,

$\langle s_1, \overline{\varsigma(\beta')}\rangle \in V$ and $\langle s, \overline{\varsigma(\beta)}\rangle \overset{a}{\rightsquigarrow} \langle s_1, \overline{\varsigma(\beta')}\rangle$. Then for all $((s,s'),a,(s_1,s_1')) \in \delta_\times$ by (1) $\langle (s_1,s_1'), \overline{\beta'}\rangle \in V_\times$ and by (3) $\langle (s,s'), \overline{\beta}\rangle \xrightarrow{a}_\times \langle (s_1,s_1'), \overline{\beta'}\rangle$. Or there exists a transition $(s,a,s_1) \in \delta$ such that for all formulas $\overline{\beta'} \in \Phi$, $\langle s_1, \overline{\varsigma(\beta')}\rangle \in V$ and $\langle s, \overline{\varsigma(\beta)}\rangle \overset{a}{\rightsquigarrow} \langle s_1, \overline{\varsigma(\beta')}\rangle$. Since s is an uncontrollable state, there exists also a transition $((s,s'),a,(s_1,s_1')) \in \delta_\times$ for some s_1'. Moreover for all formula $\overline{\beta'} \in \Phi$ by (1) $\langle (s_1,s_1'), \overline{\beta'}\rangle \in V_\times$ and by (3) $\langle (s,s'), \overline{\beta}\rangle \xrightarrow{a} \langle (s_1,s_1'), \overline{\beta'}\rangle$.

If s is a controllable state: Then following the condition ($\Box$), there exists a formula $\overline{\beta'} \in \Phi$ such that for all transitions $(s,a,s_1) \in \delta$, $\langle s_1, \overline{\varsigma(\beta')}\rangle \in V$ and $\langle s, \overline{\varsigma(\beta)}\rangle \overset{a}{\rightsquigarrow} \langle s_1, \overline{\varsigma(\beta')}\rangle$. Then for all transitions $((s,s'),a,(s_1,s_1')) \in \delta_\times$ by (1) $\langle (s_1,s_1'), \overline{\beta'}\rangle \in V_\times$ and by (3) $\langle (s,s'), \overline{\beta}\rangle \xrightarrow{a}_\times \langle (s_1,s_1'), \overline{\beta'}\rangle$.

Infinitely regenerated condition. Similarly to the proof of Lemma 19, an infinite regeneration of a least fixpoint $\mu X.\overline{\beta} = \overline{\nu X.\beta}$ in the graph $\mathcal{G}_\times$ can be reduced to an infinite regeneration of $\overline{\varsigma(\nu X.\beta)} = \mu X.\varsigma(\beta)$ in $\mathcal{G}$.

Finally, since $\mathcal{S} \models^{v_c} \overline{\varsigma(\alpha)}$ then $\langle s_0, \overline{\varsigma(\alpha)}\rangle \in V$ and by (1), $\langle (s_0,s_0'), \overline{\alpha}\rangle \in V_\times$ that is $\mathcal{S} \times \mathcal{C} \models^{v_\times} \overline{\alpha}$.

6 Controller Synthesis in an Action Based Setting

Since previous works like [2] present the synthesis problem in an action based setting rather than the state/proposition based setting we have presented so far, we indicate in which manner our results carry over and are fully valid with actions.

The μ-calculus is typically presented with a modality $\langle a\rangle\alpha$ (there is a transition labeled "a" to a state with α rather than any transition as for $\Diamond\alpha$) and $[a]\alpha$ (for all a-labeled transitions). Likewise, the operator $\rightarrow$ can be restricted to a-labeled transitions, written as $\overset{a}{\rightarrow} \Phi$ as originally described in [5]:

$$\overset{a}{\rightarrow} \Phi = \bigwedge\{\langle a\rangle\alpha \mid \alpha \in \Phi\} \wedge [a]\bigvee \Phi \text{ and } \overset{a}{\Rightarrow} \Phi = \bigvee\{[a]\alpha \mid \alpha \in \Phi\} \vee \langle a\rangle \bigwedge \Phi$$

where Φ is a finite set of formulas of μ-calculus. Disjunctivity in this setting may allow conjunctions of formulas $\overset{a}{\rightarrow} \Phi$ and $\overset{b}{\rightarrow} \Phi'$, provided $a \neq b$.

For the control synthesis problem in the action based setting, we now assume a distinction into controllable and uncontrollable actions by labels: $\Sigma = \Sigma_c \uplus \Sigma_u$. A controller is now a transition system that never refuses uncontrollable actions but which may refuse controllable actions.

Based on disjunctive formulas, the definition of the transformation $\varsigma(.)$ changes with respect to the $\overset{a}{\rightarrow}$ operator: (1) $\varsigma(\overset{a}{\rightarrow} \emptyset) = \overset{a}{\rightarrow} \emptyset$ if $a \in \Sigma_u$, (2) $\varsigma(\overset{a}{\rightarrow} \emptyset) = \top$ iff $a \in \Sigma_c$, (3) $\varsigma(\overset{a}{\rightarrow} \Phi) = \overset{a}{\rightarrow} \{\varsigma(\alpha) \mid \alpha \in \Phi\}$ if $\Phi \neq \emptyset$.

The meaning of a control formula is easier to understand when we study this setting. The key is the formulas of the form $\overset{a}{\rightarrow} \emptyset$. Suppose that we want control a state s of a controllable action transition system S in such a way the controlled state s satisfies $\overset{a}{\rightarrow} \emptyset$. There is two cases: Either the state a is controllable in which case it suffices to prevent all transitions labeled by a from s in S. Or a is uncontrollable then no transition a must be possible from s since no controller could prevent a at s from occurring and rendering $\overset{a}{\rightarrow} \emptyset$ invalid.

Example 21. Let S be a transition system over $\Sigma = \{a, b\}$, $\Sigma_c = \{b\}$, $\Sigma_u = \{a\}$. The property *"there is no transition labeled by a in S"* may be expressed the disjunctive formula $\alpha := \nu X.((\overset{a}{\rightarrow} \emptyset \wedge \overset{b}{\rightarrow} \{X\}) \vee (\overset{a}{\rightarrow} \emptyset \wedge \overset{b}{\rightarrow} \emptyset)$. The corresponding control formula is (equivalent to) $\varsigma(\alpha) := \nu X.((\overset{a}{\rightarrow} \emptyset \wedge \overset{b}{\rightarrow} \{X\}) \vee (\overset{a}{\rightarrow} \emptyset \wedge \top)$, after simplification this is easily seen to be equivalent to $\overset{a}{\rightarrow} \emptyset$, i.e. the property is controllable for systems that do not propose a at the initial state (by cutting all b-transitions).

The construction of a controller from a derivation graph remains unchanged as well as the essential structure of all proofs.

The reason why the action based setting becomes seemingly simpler than the elaborated state based setting is due to the more severe structure of disjunctive formulas. Typically, expressing state based properties in the action based setting leads to an explosion of the disjunctive representation.

Acknowledgments

We thank Rémi Morin for many discussions on the topic. A lot of thanks go to Luigi Santocanale for advice with normal forms of the μ-calculus. This work was supported by the IST project AMETIST, contract IST-2001-35304, `http://ametist.cs.utwente.nl`.

References

1. H.R. Andersen and G. Winskel. Compositional checking of satisfaction. *Formal Methods in System Design*, 1(4), 1992.
2. A. Arnold, A. Vincent, and I. Walukiewicz. Games for synthesis of controllers with partial observation. *Theor. Comput. Sci.*, 303(1):7–34, 2003.
3. A. Browne, E. M. Clarke, S. Jha, D. E. Long, and W. Marrero. An improved algorithm for the evaluation of fixpoint expressions. *Theor. Comput. Sci.*, 178(1-2):237–255, 1997.
4. Rance Cleaveland, Marion Klein, and Bernhard Steffen. Faster model checking for the modal mu-calculus. In *Proceedings of the Fourth International Workshop on Computer Aided Verification*, pages 410–422. Springer-Verlag, 1993.
5. David Janin and Igor Walukiewicz. Automata for the modal mu-calculus and related results. In *Proceedings of the 20th International Symposium on Mathematical Foundations of Computer Science*, pages 552–562. Springer-Verlag, 1995.
6. O. H. Jensen, J. T. Lang, C. Jeppesen, and K. G. Larsen. Model construction for implicit specifications in modal logic. In *CONCUR'93: Proc. of the 4th International Conference on Concurrency Theory*, volume 715 of *Lecture Notes in Computer Science*, pages 247–261. Springer, 1993.
7. D. Kozen. Results on the propositional μ-calculus. *Theoretical Computer Science*, 27:333–354, 1983.
8. D. Park. Fixpoint induction and proofs of program properties. *Machine Intelligence*, 5, 1970.
9. P.J.G. Ramadge and W.M. Wonham. The control of discrete event systems. *IEEE Proceedings: Special issue on Discrete Event Systems*, 77:81–98, 1989.

10. R.S. Streett and E.A. Emerson. An automata theoretic decision procedure for the propositional mu-calculus. *Inf. Comput.*, 81(3):249–264, 1989.
11. L. Tan and R. Cleaveland. Evidence-based model checking. In *Computer Aided Verification (CAV)*, volume 2404 of *Lecture Notes in Computer Science*, pages 455–470, 2002.