Diploma Thesis (Final Version)

Extensions of an Algorithm for Generalised Fair Model Checking

Matthias Schmalz

Last Changes: March 25, 2008

Supervisors:

Prof. Dr. K. Rüdiger Reischuk

and

Dr. Hagen Völzer

Institute for Theoretical Computer Science
University of Lübeck

Erklärung:
Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig und unter ausschließlicher Verwendung der angegebenen Literatur angefertigt habe.

Lübeck, den

To my fiancé Yang:
Thank you very much
for your patience
and your great support
during the last six months!

Zusammenfassung

Wir untersuchen einen Algorithmus für eine Variante von Modelchecking. Das Problem des *verallgemeinert fairen Modelcheckings* besteht darin, für ein endliches Modell eines verteilten oder reaktiven Systems und eine temporale Spezifikation zu überprüfen, ob eine Fairnessannahme (in einem wohldefinierten Sinn) existiert, so dass das Modell unter dieser Fairnessannahme die Spezifikation erfüllt. Der von uns betrachtete Algorithmus zur Lösung dieses Problems (vorgeschlagen von Courcoubetis und Yannakakis [1995]) erlaubt Spezifikationen in propositionaler linearer temporaler Logik (ohne Vergangenheitsoperatoren).

Der bisher bekannte Korrektheitsbeweis basiert auf wahrscheinlichkeitstheoretischen Überlegungen. Wir erarbeiten einen alternativen Beweis, der auf einer spieltheoretischen Sichtweise aufbaut. Unser Korrektheitsbeweis hat eine modulare Struktur, die es erleichtert, den Algorithmus zu verbessern sowie Grenzen des gewählten Ansatzes aufzuzeigen. Wir nutzen dies, um den Algorithmus zu erweitern, so dass er auch Spezifikationen in temporaler Logik mit Vergangenheitsoperatoren erlaubt.

Wir definieren eine Teilmenge temporaler Spezifikationen, die Sprache der *Muller Formeln*, und erweitern den Algorithmus, so dass er das verallgemeinert faire Modelchecking Problem in Linearzeit löst, falls die Spezifikation eine Muller Formel ist. Zuvor benötigte der Algorithmus im schlechtesten Fall exponentielle Zeit in der Größe der Spezifikation, selbst wenn man sich auf Muller Formeln als Spezifikationen beschränkte. Dieses Ergebnis ist überraschend, da wir auf der anderen Seite zeigen werden, dass das gewöhnliche Modelchecking Problem für Spezifikationen, die sich durch Muller Formeln ausdrücken lassen, co-NP-vollständig ist. Dies ist das erste uns bekannte Beispiel für eine natürliche Menge von Spezifikationen, für die sich die Komplexität von gewöhnlichem und verallgemeinert fairem Modelchecking unterscheiden.

Falls ein System seine Spezifikation nicht erfüllt, so erzeugen viele Algorithmen für traditionelles Modelchecking einen so genannten *Beleg*; das ist ein Lauf des Systems, der die Spezifikation nicht erfüllt. Belege unterstützen die Entwickler darin, Fehler zu lokalisieren und zu beheben. Wir schlagen eine Definition für Belege im Kontext des verallgemeinert fairen Modelcheckings vor und erweitern den Algorithmus, so dass er seine Ausgabe durch einen Beleg begründet.

Abstract

We examine an algorithm for a variation of model checking. The task of *generalised fair model checking* is to check, for a finite model of a concurrent or reactive system and a temporal specification, whether there exists some fairness assumption (in a well-defined sense) such that the model satisfies the specification under this fairness assumption. The algorithm for this problem which we consider (proposed by Courcoubetis and Yannakakis [1995]) allows specifications in propositional linear temporal logic (without past operators).

The original proof of correctness is based on arguments of probability theory. We develop an alternative proof by analysing the algorithm from a game-theoretic point of view. Our proof of correctness has a modularised structure, which makes it easier to find extensions that improve the algorithm and to figure out limitations of the approach the algorithm follows. We take advantage of this, when we develop techniques that allow the algorithm to handle specifications in temporal logic with past operators.

We define a subset of temporal specifications, the language of *Muller formulae*, and extend the algorithm such that the generalised fair model checking problem is solved in linear time, provided the specification is a Muller formula. Before, the algorithm required, in the worst case, exponential time in the size of the specification, even if we restrict us to Muller formulae as specifications. This result is surprising, because, on the other hand, we show that the traditional model checking problem for Muller formulae is co-NP-complete. This is the first example we are aware of where the complexity of traditional, and generalised fair model checking differ.

If the system violates its specification, many algorithms for traditional model checking generate a *witness*; that is a run of the system that violates the specification. Witnesses support the developers in finding and eliminating errors in the model. We propose a definition of witnesses in the context of generalised fair model checking and extend the algorithm such that its output is supplemented by a witness.

Contents

1. Introduction

Today, in many applications failure of hardware or software is unacceptable: electronic commerce, highway and air traffic control systems, medical instruments, nuclear power plants, rocket control computers and other examples, too numerous to list. Simulation and testing are very powerful in early stages of debugging, when there are still many bugs. But specially in the case of concurrent or reactive systems, it may be rarely possible to find all the more subtle bugs of a design with these techniques. A very attractive alternative to simulation and testing is the approach of formal verification. While simulation and testing only checks if *some* runs of a system meet a desired specification, leaving open the question of whether some of the remaining runs may contain the fatal bug, formal verification methods perform an exhaustive examination of *all* possible runs. During the last three decades, a lot of effort has been put into the investigation of model checking. Traditional model checking is the task of checking whether, for a linear-time temporal specification X, each run of a finite system satisfies X. Model checking is now becoming widely used in industry as a practical verification technique.

Sometimes the system does not satisfy the desired specification, but the runs that violate it seem to be artificial. As an example, suppose that a specification for Dijkstra's dining philosophers requires starvation freedom; however, a philosopher may starve because at each moment one of his neighbours is eating such that he never can pick up his two forks at a time. Although such runs exist, it seems that they can be neglected in many applications. There exist solutions for this example that meet starvation freedom, but it has been shown that for several problems this is not the case, because a system satisfying its specification is impossible, too expensive to obtain or inefficient (cf. Fich and Ruppert [2003]). In these cases, we would be content with a system where "almost" all runs satisfy the specification.

One possibility of formalising "almost" is to require that, for a certain *fairness assumption* F and a desired specification X, each run satisfies $F \Rightarrow X$. This problem is called *model checking constrained by some fairness assumption* and has been discussed, e. g., by Lichtenstein and Pnueli [1985]. However, in practice it may be difficult or expensive to find and specify a suitable fairness assumption.

Most fairness assumptions demand that some choice is taken sufficiently often if it is possible sufficiently often. They differ in the definition of "choice", "sufficiently often" and "possible". Völzer et al. [2005] propose a game-theoretic characterisation of all fairness assumptions based on the idea that a property is a fairness property if it can be realised by a scheduler that infinitely often gets control over the system.

All popular fairness notions satisfy the definition, the proposed family of fairness assumptions is closed under superset and countable intersection, and it is, in a sense, the largest family having this property. Thus, an alternative way of formalising that "almost" all runs of a system satisfy a desired specification X is to require that there exists a fairness assumption F according to Völzer et al. [2005] such that each run satisfies $F \Rightarrow X$. We call this the problem of *generalised fair model checking*.

There is a third approach of formalising "almost" which requires that the set of runs that satisfy the specification has probability one. This is the *qualitative probabilistic model checking*. One might object that, for instance in the example of Dijkstra's dining philosophers, it might be hard to justify a concrete probability measure in practice. However, it is a well-known fact that, under weak assumptions, the concrete probability measure does not influence whether a specification is met with probability one. Varacca and Völzer [2006] show that, under weak assumptions, the specification is satisfied with probability one if and only if it is satisfied under some fairness assumption according to Völzer et al. [2005]. Thus, generalised fair model checking and qualitative probabilistic model checking coincide.

Vardi [1985] shows that generalised fair model checking with *automata specifications*, i. e. specifications given by *Büchi automata* (cf. Thomas [1990]), is PSPACE-complete in the size of the specification and can be solved in polylogarithmic space in the size of the system. As a rule, systems tend to be large and specifications to be short. Thus, the high complexity might still be acceptable for short specifications.

There are algorithms of the generalised fair model checking problem for various fragments of *propositional linear temporal logic* (LTL) (cf. Emerson [1990]). For the fragment that allows only the temporal operators X (next) and $\Diamond$ (eventually), Vardi and Wolper [1986] give an algorithm that runs in exponential time in the size of the specification and linear time in the size of the system. Pnueli and Zuck [1986] propose an algorithm for the fragment of LTL allowing the temporal operators X (next), strict eventually, Y (yesterday) and S (since) that runs in exponential time in the size of the specification and quadratic time in the size of the system. Varacca and Völzer [2006] develop a method for *reactivity formulae* (cf. Manna and Pnueli [1990]) requiring polynomial space. If the specification is a *state reactivity formula*, their method runs in linear time.

Courcoubetis and Yannakakis [1995] show that generalised fair model checking can be solved in exponential time in the size of the specification and linear time in the size of the system, or, alternatively, in polynomial space. They develop three algorithms having the same complexity, one for LTL specifications, one for automata specifications and one for specifications in an *extended temporal logic* (cf. Wolper [1983] and Wolper et al. [1983]). As Vardi [1985] shows PSPACE-hardness for each of these specification formalisms, their algorithms are optimal.

Couvreur et al. [2003] develop a translation that maps an LTL specification to an exponentially larger special type of automata specification, which can be computed

in exponential time in the size of the LTL specification. They then show that the generalised fair model checking problem with this special type of automata specification can be solved in linear time. Thus, they construct an automata based algorithm for LTL specifications meeting the same time bounds as the related algorithm of Courcoubetis and Yannakakis [1995].

The opposite way, i. e. translating automata specifications to LTL specifications, is impossible, as automata specifications are strictly more expressive than LTL specifications (cf. Wolper [1983]).

As an alternative specification formalism, Bustan et al. [2004] examine the so called *alternating Büchi infinite-word automata* (ABWs). As the motivation for using ABWs they mention that ABWs have the full expressive power of automata specifications. Fortunately, an LTL specification can be translated to an equivalent ABW in linear time. They give an algorithm for the generalised fair model checking problem with ABW specifications running in exponential time in the size of the specification and linear time in the size of the system, or, alternatively, polynomial space. Thus, the additional expressive power comes at no penalty.

The discussion if LTL specifications or specifications in *computation tree logic* (CTL specifications) (cf. Emerson [1990]) are better suited for model checking goes back to the early 1980s. We do not consider CTL specifications, as we believe, following Vardi [2001], that this logic has several disadvantages for the purpose of model checking. However, the branching time logic CTL* (cf. Emerson [1990]), an extension of CTL and LTL, is an interesting candidate for a specification language. Varacca and Völzer [2006] give a definition for generalised fair model checking with specifications in CTL* and illustrate how an algorithm for LTL specifications can easily be extended to an algorithm for CTL* specifications.

As far as we know, none of the algorithms above, apart from the techniques of Couvreur et al. [2003], has been implemented, yet. The question arises, which one can be realised with a good relation of benefit and cost. We exclude the algorithms of Pnueli and Zuck [1986], Vardi and Wolper [1986], and Varacca and Völzer [2006] from further considerations, as they do not allow arbitrary LTL specifications. Among the remaining approaches, the algorithms of Courcoubetis and Yannakakis [1995] for LTL and automata specifications are the simplest ones. Because propositional linear temporal logic is more popular and more intuitive than Büchi automata, we have decided to examine the algorithm of Courcoubetis and Yannakakis [1995] for LTL specifications.

When Courcoubetis and Yannakakis [1995] developed their algorithms, it was not known that qualitative probabilistic, and generalised fair model checking coincide. Therefore, their proof of correctness is based on Markov chains. We provide an alternative proof based on *path games*, an important notion in the characterisation of fairness assumptions of Völzer et al. [2005]. We leave it to the reader to decide which proof is more intuitive and easier to understand.

As a consequence, we show that there is one algorithm for both generalised fair, and qualitative probabilistic model checking. On the one hand, the algorithm accepts a system and a specification iff a run of the system meets the specification with probability one. On the other hand, the algorithm accepts a system and a specification X iff there is a fairness assumption F according to Völzer et al. [2005] such that each run of the system satisfies $F \Rightarrow X$. This shows independently from Varacca and Völzer [2006] that generalised fair, and qualitative probabilistic model checking coincide. (Nevertheless, we also give a detailed proof of this fact based on the arguments of Varacca and Völzer [2006].)

Moreover, our proof of correctness has a modularised structure. This makes it easier, on the one hand, to find extensions that improve the algorithm and, on the other hand, to figure out which aspects cannot be improved anymore.

We consider the question of how to improve running time for a restricted set of specifications. For this sake, we define the language $\overline{L}(\Box\Diamond)$ of *Muller formulae* and prove that the generalised fair model checking problem with such specifications can be solved in linear time in the size of the system *and* the specification. This is mainly based on a characterisation of a fundamental semantical property of Muller formulae. We show that, in contrast, checking whether *each* run of a system satisfies a Muller formula is co-NP-complete. This is surprising, as it is the first example we are aware of where the complexity for generalised fair, and traditional model checking differ.

We extend the algorithm of Courcoubetis and Yannakakis [1995] by the techniques for Muller formulae and show that if the specification contains Muller formulae then the upper bound of running time is basically improved by an exponential factor in the summarised size of these Muller formulae. Moreover, we prove formally that, roughly speaking, the running time of the new version, is bounded by the running time of the old version of the algorithm.

We also tried to enlarge the set of specifications for which the generalised fair model checking problem can be solved in linear time. A natural superset of $\overline{L}(\Box\Diamond)$ is $L(\Box)$, the fragment of propositional linear temporal logic that only allows the temporal operators $\Box$ (always) and $\Diamond$ (eventually). However, we could only prove a negative result: generalised fair model checking for specifications in $L(\Box)$ is co-NP-complete.

If a model checking algorithm figures out that a system does not satisfy its specification, the question arises why this is the case. It is desirable that the algorithm explains its result in a way that humans can understand. Therefore, in the case of a negative result, a lot of algorithms for traditional model checking create a run of the system that violates the specification. Such a run is called a *witness*. In the case of generalised fair model checking, it would also be convenient to compute such a witness. However, as long as the problem is examined with the methods of probability theory, it is hard to find a suitable definition of a witness. In the negative case, we just know that the set of runs that violate the specification has a positive probability. It remains difficult to find a suitable representation of this set. With the

game-theoretic characterisation developed by Varacca and Völzer [2006], the situation becomes different. We use this characterisation to propose a formal definition of a witness in the generalised fair model checking problem. Moreover, we develop an extension of the algorithm that allows to compute such a witness.

As specifications with past operators in many cases are more intuitive than equivalent specifications without, we introduce methods for handling past operators. (The original algorithm of Courcoubetis and Yannakakis [1995] only allows future operators.)

Several times we use the fact that *state fairness* is complete for $L(\Box)$; i. e., if a specification in $L(\Box)$ holds under some fairness assumption, then it holds under state fairness. Zuck et al. [2002] illustrate how this can be derived from a theorem of Pnueli and Zuck [1993]. In this document, we provide an alternative, direct and detailed proof.

This document is organised as follows. In Chapter 2, we introduce preliminary notions. Our version of the model checking algorithm is described in Chapter 3. The computation of witnesses is discussed in Chapter 4. In Chapter 5, we reflect about future work. In the Appendix, we sketch our thoughts on this in a more technical way.

2. Preliminaries

Throughout this document, we use the following conventions:

- The set of natural numbers inclusive 0 is denoted by $\mathbb{N}$, and $\mathbb{N}^+ := \mathbb{N} \setminus \{0\}$.
- We fix a set AP of *atomic propositions*.
- The complement of a set A w. r. t. a universe which becomes clear from the context is denoted by A^c.
- The operator $\times$ stands for the *flat* Cartesian product. Therefore,

  $$(A \times B) \times C = A \times B \times C = A \times (B \times C),$$

  where A, B, C are arbitrary sets. For ease of notation, we sometimes write (A, B) instead of $A \times B$. If $B = \{b\}$, we abbreviate (A, B) by (A, b) and (B, A) by (b, A).
- We use standard notation from computational complexity, such as $\mathcal{O}$, P, NP. See, for example, the textbook of Reischuk [1999].

2.1. Sequences and Regular Expressions

Let Q be some finite set. A *path over* Q is an infinite sequence over Q. Finite, possibly empty, sequences over Q are called *path fragments over* Q. The empty sequence is denoted by λ. If not mentioned otherwise, α, β denote path fragments, and x, y, paths. We call a set containing paths over Q a *linear-time property (over Q)*.

Let α be a path fragment, and z an arbitrary sequence over Q. Moreover, let B be a set of path fragments, and C a set of arbitrary sequences over Q. The *concatenation* of α and z is denoted by αz. Furthermore, $BC := \{\beta y \mid \beta \in B,\ y \in C\}$, $\alpha C := \{\alpha\}C$ and $Bz := B\{z\}$.

We define $A^0 := \{\lambda\}$ and, recursively, $A^{i+1} := A^i A$, for $i \in \mathbb{N}$. The *Kleene closure* of A is defined as $A^* := \bigcup_{i \in \mathbb{N}} A^i$. Moreover, $A^+ := \bigcup_{i \in \mathbb{N}^+} A^i$. For ease of notation, we let α^i be the unique element of $\{\alpha\}^i$, where $i \in \mathbb{N}$. Thus, $\alpha^3 = \alpha\alpha\alpha$, $\{\alpha\}^* = \{\lambda, \alpha, \alpha\alpha, \dots\}$ and $\{\alpha\}^+ = \{\alpha, \alpha\alpha, \dots\}$.

We say that α is a *prefix* of z and write $\alpha \sqsubseteq z$ iff $z = \alpha x$, for some arbitrary sequence x over Q. If $\alpha \neq z$ and $\alpha \sqsubseteq z$, then α is a *proper prefix* of z, written as

$\alpha \sqsubset z$. We define

$$\begin{aligned} \alpha\uparrow &:= \{y \mid \alpha \sqsubseteq y,\ y \text{ is a path over } Q\}, \\ z\downarrow &:= \{\beta \mid \beta \sqsubseteq z,\ \beta \text{ is a path fragment over } Q\}. \end{aligned}$$

The set $\alpha\uparrow$ is called the *shadow* of α; the set $z\downarrow$ contains the finite prefixes of z. Similarly, $B\uparrow := \bigcup_{\beta\in B} \beta\uparrow$ and $C\downarrow := \bigcup_{y\in C} y\downarrow$. Two sequences are called *compatible* iff one of them is a prefix of the other one.

We define $B^\omega := \bigcap_{i\in\mathbb{N}}(B^i\uparrow)$ and α^ω as the unique element of $\{\alpha\}^\omega$. Hence, $\alpha^\omega = \alpha\alpha\alpha\ldots$

Let $i, j \in \mathbb{N}$ with $i \leq j$. The *i-th element* of z is denoted by z_i. (We start counting at zero.) Therefore, $z = z_0 z_1 \ldots$ Moreover, $z[i,j] := z_i z_{i+1} \ldots z_j$, provided that z_j exists. If z is an infinite sequence, then $z[i,\infty] := z_i z_{i+1} \ldots$ We define $|\alpha| \in \mathbb{N}$ as the *length* of α; that means $\alpha = \alpha_0 \ldots \alpha_{|\alpha|-1}$.

2.2. Graph Theoretic Notions

Let $G = (V, E)$ be a directed graph. Here and later, we abuse notation and do not distinguish between $V' \subseteq V$ and the *subgraph induced by V'*; i. e. $(V', E \cap (V', V'))$.

The *strongly connected components (s. c. c.s)* of G constitute a partition of V. Two vertices p, q belong to the same s. c. c. iff there are sequences $(p, r_1), (r_1, r_2), \ldots, (r_n, q)$ and $(q, s_1), (s_1, s_2), \ldots, (s_m, p)$ over E. A b. s. c. c. (*bottom strongly connected component*) K of G is an s. c. c. of G with no outgoing edges; i. e. there is no edge $(p, q) \in E$ such that $p \in K$ and $q \in V \setminus K$.

Two graphs $G = (V, E)$ and $G' = (V', E')$ are *isomorphic* iff there is a bijection $\sigma : V \to V'$ with $(p, q) \in E$ iff $(\sigma(p), \sigma(q)) \in E'$.

2.3. Systems

Let $Q \subseteq AP$ be a nonempty set of *states* and $\to \subseteq (Q, Q)$ a *state relation* on Q which is *total*; i. e., for each $p \in Q$, there exists a $q \in Q$ such that $p \to q$. A *valuation function* v maps each $q \in Q$ to $v(q) \subseteq AP$ such that $v(q) \cap Q = \{q\}$. Given a valuation function v and a nonempty set $S \subseteq Q$ of *starting states*, $\Sigma := (Q, S, \to, v)$ denotes a *system*. Sometimes we need to modify the starting states of a system; therefore, we define, given $\Sigma = (Q, S, \to, v)$ and $q \in Q$, $\Sigma_q := (Q, \{q\}, \to, v)$.

If $p \to q$, we say that p is a *predecessor* of q and q is a *successor* of p. A *path* x of Σ is a path over Q such that $x_0 \in S$ and $x_i \to x_{i+1}$, for each $i \in \mathbb{N}$. We define $path(\Sigma)$ as the set of all paths of Σ. The elements of $path(\Sigma)\downarrow$ are called *path fragments of* Σ. We abbreviate $v(z_0)v(z_1)\ldots$ by $v(z)$, where z is an arbitrary sequence over Q.

Sometimes, we talk of a system in terms of a directed graph, meaning the graph $(Q, \to)$.

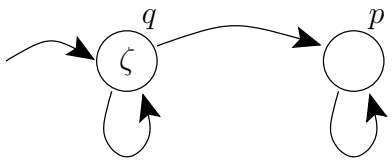

Figure 2.1.: A system $\Sigma = (Q, S, \to, v)$. The states are represented by circles, which are connected by arrows according to the state relation. Next to the circle related to the state $s \in Q$, we put the state name s, and, inside the circle, we write the elements of $v(s) \setminus \{s\}$. A starting state is marked by an additional incoming arrow.

An example for a system is $\Sigma = (Q, S, \to, v)$ with $Q := \{p, q\}$, $S := \{q\}$, $\to := \{(q,q), (q,p), (p,p)\}$ and $v(q) := \{q, \zeta\}$, $v(p) := \{p\}$, as described in Figure 2.1. The paths q^ω, qp^ω, qqp^ω are paths of Σ, and λ, qq, qp, $qqqpp$ are path fragments of Σ. The path pq^ω is not a path of Σ, as p is not a starting state and $p \not\to q$. If we define $\to := \{(q,q), (q,p)\}$, then Σ is not a valid system, as $\to$ is not total.

2.4. Temporal Logic

We introduce propositional linear temporal logic, basically as described by Emerson [1990]. The language of *state formulae* over AP is the smallest set such that the following statements S1 and S2 hold:

S1: Each atomic proposition is a state formula.

S2: If ξ and ψ are state formulae, then so are $\xi \vee \psi$ and $\neg\xi$.

The *boolean operators* $\vee$ and $\neg$ are called *disjunction* and *negation*, respectively.

There are the following types of *basic temporal operators*: the *future* operators X (next) and U (until), and the *past* operators Y (yesterday) and S (since). The operators X and Y are unary operators, and U and S are binary operators.

The language of *path formulae* is the smallest set which satisfies the following conditions P1 and P2:

P1: Each state formula is a path formula.

P2: If Ξ, Ψ are path formulae, then so are $\neg\Xi$, $\Xi \vee \Psi$, $\mathsf{X}\,\Xi$, $\mathsf{Y}\,\Xi$, $\Xi\ \mathsf{U}\ \Psi$ and $\Xi\ \mathsf{S}\ \Psi$.

Let $\Sigma = (Q, S, \to, v)$ be a system. The semantics of state formulae is as follows. Suppose $q \in Q$ and ϕ is a state formula.

- If ϕ is an atomic proposition, we have $q \vDash \phi$, i. e. q *satisfies* ϕ, iff $\phi \in v(q)$.
- If $\phi = \xi \vee \psi$, we have $q \vDash \phi$ iff $q \vDash \xi$ or $q \vDash \psi$.
- If $\phi = \neg\xi$, we have $q \vDash \phi$ iff $q \nvDash \xi$.

Note that, as it depends on the underlying system Σ whether $q \vDash \phi$, we might write $\Sigma, q \vDash \phi$ instead of $q \vDash \phi$; however, throughout this document, the underlying systems are clear from the context, and thus we can safely omit Σ.

The semantics of path formulae is as follows. Let $x \in Q^\omega$, $i \in \mathbb{N}$ and Φ a path formula.

- If Φ is a state formula, we have $x, i \vDash \Phi$, i. e. x *satisfies* Φ *at position* i, iff $x_i \vDash \Phi$.
- If $\Phi = \Xi \vee \Psi$, then $x, i \vDash \Phi$ iff $x, i \vDash \Xi$ or $x, i \vDash \Psi$.
- If $\Phi = \neg\Xi$, then $x, i \vDash \Phi$ iff $x, i \nvDash \Xi$.
- If $\Phi = \Xi \,\mathsf{U}\, \Psi$, then $x, i \vDash \Phi$ iff there is a $j \geq i$ such that $x, j \vDash \Psi$ and, for all k with $i \leq k < j$, we have $x, k \vDash \Xi$.
- If $\Phi = \mathsf{X}\Xi$, then $x, i \vDash \Phi$ iff $x, i+1 \vDash \Xi$.
- If $\Phi = \Xi \,\mathsf{S}\, \Psi$, then $x, i \vDash \Phi$ iff there is a j with $0 \leq j \leq i$ such that $x, j \vDash \Psi$ and, for all k with $j < k \leq i$, we have $x, k \vDash \Xi$.
- If $\Phi = \mathsf{Y}\Xi$, then $x, i \vDash \Phi$ iff $i > 0$ and $x, i-1 \vDash \Xi$.

Again, we do not write $\Sigma, x, i \vDash \Phi$, as, throughout this document, Σ is clear from the context.

There are some common abbreviations. Let Ξ, Ψ be path formulae and ξ a state formula. We define the *conjunction* $\Xi \wedge \Psi := \neg(\neg\Xi \vee \neg\Psi)$, the *boolean constants* $\mathit{true} := \xi \vee \neg\xi$, $\mathit{false} := \neg\mathit{true}$, the *implication* $\Xi \Rightarrow \Psi := \Psi \vee \neg\Xi$ and the *equivalence* $\Xi \Leftrightarrow \Psi := (\Xi \Rightarrow \Psi) \wedge (\Psi \Rightarrow \Xi)$. The unary *derived temporal operators* $\Diamond$ (eventually), $\Box$ (always) are defined by $\Diamond\Xi := \mathit{true}\ \mathsf{U}\ \Xi$ and $\Box\Xi := \neg(\Diamond\neg\Xi)$. For convenience, we also introduce the unary derived temporal operators $\Box\Diamond$ (infinitely often) and $\Diamond\Box$ (almost everywhere) by $\Box\Diamond\Xi := \Box(\Diamond(\Xi))$ and $\Diamond\Box\Xi := \Diamond(\Box(\Xi))$. For a more elegant representation, we use the convention $\neg\neg\Xi = \Xi$.

Let Φ, Ψ be path formulae and ϕ a state formula. We write $x \vDash \Phi$, i. e. x *satisfies* Φ, iff $x, 0 \vDash \Phi$. The set $\mathit{sat}(\Sigma, \phi)$ contains each $q \in Q$ with $q \vDash \phi$. Analogously, $\mathit{Sat}(\Sigma, \Phi)$ contains each path over Q that satisfies Φ. If Σ is clear from the context, we write $\mathit{sat}(\phi)$ and $\mathit{Sat}(\Phi)$ instead of $\mathit{sat}(\Sigma, \phi)$ and $\mathit{Sat}(\Sigma, \Phi)$, respectively. As usual, we say that Φ and Ψ are *equivalent* iff $x \vDash \Phi \Leftrightarrow x \vDash \Psi$, for an arbitrary underlying system $\Sigma = (Q, S, \rightarrow, v)$ and $x \in Q^\omega$.

Path formulae are very useful for specifying the behaviour of systems. For the moment, we say that Σ satisfies a *specification* $\Phi \in L(\mathsf{X}, \mathsf{Y}, \mathsf{U}, \mathsf{S})$ iff $x \vDash \Phi$, for each path x of Σ. We list some specifications and explain, in each case, which properties of a system are necessary such that the system satisfies the specification. Suppose *hungry* and *eating* are atomic propositions. The specification $\Box(\neg \mathit{hungry})$ requires a system never to become hungry. With $\Box((\mathsf{Y}\, \mathit{eating}) \Rightarrow \neg \mathit{hungry})$, we say that today the system is never hungry if it has eaten yesterday. Sometimes, it makes sense to demand that whenever the system is hungry then it remains hungry until eating: $\Box(\mathit{hungry} \Rightarrow (\mathit{hungry}\ \mathsf{U}\ \mathit{eating}))$. Note that this specification also requires that whenever the system becomes hungry then it is eventually eating. With $\Box\Diamond\, \mathit{eating}$, we say that the system is eating infinitely often, and, with $(\Diamond\Box\, \mathit{hungry}) \Rightarrow (\Box\Diamond\, \mathit{eating})$, that it is eating infinitely often, provided it is hungry at all but finitely many times.

In some occasions, we want to consider formulae over a restricted set of temporal operators. Therefore, we use a notation we found in Sistla and Clarke [1985], and Schnoebelen [2003]. Let $\mathsf{Un}_1, \ldots, \mathsf{Un}_k$ be some unary, and $\mathsf{Bin}_1, \ldots, \mathsf{Bin}_\ell$ some binary temporal operators (basic or derived). The language $L(\mathsf{Un}_1, \ldots, \mathsf{Un}_k, \mathsf{Bin}_1, \ldots, \mathsf{Bin}_\ell)$ is the smallest set which satisfies the following conditions P1* and P2*:

P1*: Each state formula belongs to $L(\mathsf{Un}_1, \ldots, \mathsf{Un}_k, \mathsf{Bin}_1, \ldots, \mathsf{Bin}_\ell)$.

P2*: If Ξ, Ψ belong to $L(\mathsf{Un}_1, \ldots, \mathsf{Un}_k, \mathsf{Bin}_1, \ldots, \mathsf{Bin}_\ell)$, then so do $\Xi \vee \Psi$, $\neg\Xi$, $\mathsf{Un}_i\, \Xi$ and $\Xi\ \mathsf{Bin}_j\ \Psi$, for $1 \leq i \leq k$ and $1 \leq j \leq \ell$.

The language $L(\mathsf{X}, \mathsf{Y}, \mathsf{U}, \mathsf{S})$ is the set of path formulae, that is also referred to as *LTL+past (linear temporal logic with past)*. Another important language is $L(\mathsf{X}, \mathsf{U})$, called *LTL (linear temporal logic)*. The members of $L(\Box)$ are the *RLTL (restricted linear temporal logic)* formulae.

Note that

$$\begin{aligned} L(\Box) &= L(\Box, \Diamond) \qquad [\Diamond\, \Xi = \neg\, \Box\, \neg \Xi] && (2.1)\\ &\subseteq L(\mathsf{X}, \mathsf{U}, \Box, \Diamond) && (2.2)\\ &= L(\mathsf{X}, \mathsf{U}) \qquad [\Diamond\, \Xi = \mathit{true}\ \mathsf{U}\ \Xi, \Box\, \Xi = \neg\, \Diamond\, \neg \Xi] && (2.3)\\ &\subseteq L(\mathsf{X}, \mathsf{Y}, \mathsf{U}, \mathsf{S}). && (2.4) \end{aligned}$$

Suppose that p, q are atomic propositions. As examples for RLTL formulae, we mention $\Box\, p$ and $\Box(p \Rightarrow \Diamond\, q)$. The formulae $\mathsf{X}\, q$, $\Box\, p \wedge \mathsf{X}\, q$ and $p\ \mathsf{U}\ q$ belong to LTL, LTL+past but not to RLTL, as X and U cannot be derived from $\Box$ and $\Diamond$. It is even impossible to find RLTL formulae which are equivalent to $\mathsf{X}\, q$, $\Box\, p \wedge \mathsf{X}\, q$ or $p\ \mathsf{U}\ q$. (For the cases of $\mathsf{X}\, q$ and $p\ \mathsf{U}\ q$, we give a formal proof of this at the end of Section 3.6.2.) Furthermore, $\Box(p\ \mathsf{S}\ q)$ is an LTL+past formula, but it is neither an LTL nor an RLTL formula. Thus, the inclusions of equations (2.2) and (2.4) are

strict; however, for each LTL+past formula, there exists an equivalent LTL formula (cf. Emerson [1990]).

Finally, we introduce the *subformula relation* $\preceq$ and the *substitution*. The subformula relation is defined recursively. Let Θ, Φ be path formulae. We have $\Theta \preceq \Phi$ iff one of the following holds:

- $\Theta = \Phi$,
- $\Phi = \mathsf{Un}\,\Xi$ and $\Theta \preceq \Xi$, or
- $\Phi = \Xi\ \mathsf{Bin}\ \Psi$, and $\Theta \preceq \Xi$ or $\Theta \preceq \Psi$.

Here, Un ranges over the unary temporal operators and $\neg$, and Bin, over the binary temporal operators and $\vee$. If we write $\Theta \prec \Phi$, then $\Theta \preceq \Phi$ and $\Theta \neq \Phi$. Thus, $\psi \preceq \psi$, $\psi \prec \phi\ \mathsf{U}\ \psi$ and $\phi \wedge \psi \not\preceq (\Box\,\phi) \wedge \psi$, where ϕ, ψ are state formulae.

Let Θ_1, Θ_2 and Φ be path formulae. The *substitution* is defined as follows:

- If $\Theta_1 \not\preceq \Phi$, then $[\Phi]^{\Theta_2}_{\Theta_1} := \Phi$.
- If $\Theta_1 = \Phi$, then $[\Phi]^{\Theta_2}_{\Theta_1} := \Theta_2$.
- If $\Theta_1 \prec \Phi = \mathsf{Un}\,\Xi$, then $[\Phi]^{\Theta_2}_{\Theta_1} := \mathsf{Un}[\Xi]^{\Theta_2}_{\Theta_1}$.
- If $\Theta_1 \prec \Phi = \Xi\,\mathsf{Bin}\,\Psi$, then $[\Phi]^{\Theta_2}_{\Theta_1} := [\Xi]^{\Theta_2}_{\Theta_1}\,\mathsf{Bin}[\Psi]^{\Theta_2}_{\Theta_1}$.

Again, Un ranges over the unary temporal operators and $\neg$, and Bin, over the binary temporal operators and $\vee$. Hence, $[\zeta\ \mathsf{U}\ (\zeta \wedge \eta)]^{\Phi}_{\zeta\wedge\eta} = \zeta\ \mathsf{U}\ \Phi$ and $[\zeta]^{\Phi}_{\zeta\wedge\eta} = \zeta$, where $\zeta, \eta \in AP$ and Φ is a path formula.

2.5. Path Games

So far, we introduced the concepts we need to formalise systems and specifications. Now, we spend our attention to fairness. For this, we define *path games*, which are similar to the *Banach-Mazur game* in the literature (see Grädel [2004] or Oxtoby [1971]). We follow the definition of Varacca and Völzer [2006].

A *path game* $\mathcal{G}$ is a tuple (κ, Σ, X), where $\kappa \in \{\overline{AE}, \overline{EA}\}$, $\Sigma = (Q, S, \rightarrow, v)$ is a system and $X \subseteq Q^\omega$ is a *winning condition*.

There are two players, namely Ego (E) and Alter (A). A *play* starts with $\alpha_0 = \lambda$. In the i-th move ($i \in \mathbb{N}$), one of the players extends α_i to α_{i+1} such that $\alpha_i \sqsubseteq \alpha_{i+1}$, and α_{i+1} is a path fragment of Σ. Ego and Alter by turns choose the new path fragment α_{i+1}. In a game with $\kappa = \overline{AE}$, Alter chooses α_{i+1} iff i is even. If $\kappa = \overline{EA}$, then Alter chooses α_{i+1} iff i is odd. In fact, $\overline{AE}$ and $\overline{EA}$ are abbreviations for $AEAEAE\ldots$ and $EAEAEA\ldots$, respectively, indicating, for each move, which player chooses the new path fragment.

The *result* of the play is the set $Z := \bigcap_{i \in \mathbb{N}} \alpha_i\uparrow \cap path(\Sigma)$ containing the paths of Σ that are compatible with α_i, for each $i \in \mathbb{N}$. *Ego (E) wins the play* if $Z \subseteq X$; otherwise, *Alter (A) wins the play*.

A *strategy* (in Σ) is a mapping $f : path(\Sigma)\downarrow \rightarrow path(\Sigma)\downarrow$ such that $\alpha \sqsubseteq f(\alpha)$, for each $\alpha \in path(\Sigma)\downarrow$. Player $P \in \{E, A\}$ plays *according to the strategy* f iff, in each move i in which he chooses α_{i+1}, we have $\alpha_{i+1} = f(\alpha_i)$. We write $Z_\kappa^{P,f}$ for the set of all paths that are the result of some play of $\mathcal{G} = (\kappa, \Sigma, X)$ in which player P plays according to f. A strategy f is *winning* for a player (in $\mathcal{G}$) iff he wins every play in which he plays according to f.

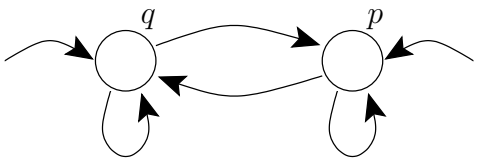

Figure 2.2.: A system Σ.

Consider the system Σ, as described in Figure 2.2. We define the strategy f with $f(\alpha) = \alpha q$ if $\alpha \in \{p\}^*$ and $f(\alpha) = \alpha$ otherwise. Then, $Z_{\overline{AE}}^{E,f} = \{p\}^* q\uparrow$ and $Z_{\overline{EA}}^{E,f} = q\uparrow$. Thus, f is a winning strategy for Ego in $(\kappa, \Sigma, Sat(\Diamond q))$, regardless of κ. There are also winning strategies for Ego in $(\overline{AE}, \Sigma, Sat(\Box\Diamond q))$, $(\overline{AE}, \Sigma, Sat(\Box(p \Rightarrow \Diamond(q \wedge \mathsf{X} q))))$ and $(\overline{EA}, \Sigma, Sat(q))$. Ego does not have a winning strategy in $(\overline{EA}, \Sigma, Sat(\Diamond\Box q))$, $(\overline{EA}, \Sigma, Sat(\Box(p \Rightarrow \mathsf{X} q)))$ or $(\overline{AE}, \Sigma, Sat(q))$. But, there are winning strategies for Alter in those games.

It is well known that linear-time properties expressible by path formulae are *determinate* (see Berwanger et al. [2003]). This means that there is a winning strategy either for Ego or for Alter if the winning condition is expressible by a path formula.

If $\alpha \sqsubset f(\alpha)$, for each $\alpha \in path(\Sigma)\downarrow$, we call f a *progressive* strategy. The following Proposition is due to Völzer et al. [2005].

Proposition 2.1. *Let $\mathcal{G} = (\kappa, \Sigma, X)$ be a path game, $P \in \{A, E\}$, and suppose that f is a winning strategy for P in $\mathcal{G}$. Then, there is a progressive winning strategy $\hat{f}$ for P in $\mathcal{G}$.*

Proof. If $\alpha \sqsubset f(\alpha)$, then $\hat{f}(\alpha) := f(\alpha)$. Otherwise, $\hat{f}(\alpha) := \alpha\beta$ for some $\beta \neq \lambda$ such that $\alpha\beta \in path(\Sigma)\downarrow$. Hence, $Z_\kappa^{P,\hat{f}} \subseteq Z_\kappa^{P,f}$. The assertion follows. □

If there is an $\ell \in \mathbb{N}$ such that, for each $\alpha \in Q^*$, we have $|f(\alpha)| \leq |\alpha| + \ell$, we say that f is *bounded*. If $f(\alpha p) = \alpha\beta$ and β does not depend on α, where $\alpha p\beta \in path(\Sigma)\downarrow$ and $p \in Q$, we call f a *positional* strategy. In this case, $f(p)$ is defined by $f(\alpha p) =: \alpha f(p)$. The following theorem is due to Berwanger et al. [2003].

Theorem 2.2. *Let Φ be a path formula and $\mathcal{G} = (\kappa, \Sigma, Sat(\Phi))$ a path game. Then, there is a positional winning strategy either for Ego or for Alter in $\mathcal{G}$.*

2.6. Large, Medium-Sized and Small Linear-Time Properties

Oxtoby [1971] illustrates that, given a system $\Sigma = (Q, S, \rightarrow, v)$ and a linear-time property X over Q, Ego has a winning strategy in $(\overline{AE}, \Sigma, X)$ iff X is a co-meager set in the Cantor topology relative to $path(\Sigma)$. He also motivates that co-meagerness is a natural notion of largeness, which we adopt in the following:

Definition 2.3. Let $\Sigma = (Q, S, \rightarrow, v)$ and X a linear-time property over Q. We say that

- X is *large* (in Σ) iff there is a winning strategy for Ego in the game $(\overline{AE}, \Sigma, X)$,
- X is *small* (in Σ) iff X^c is large (in Σ).
- Otherwise, if X is neither large nor small in Σ, it is *medium-sized* (in Σ).

Note that X is small iff there is a winning strategy for Alter in the game $(\overline{EA}, \Sigma, X)$. Furthermore, a determinate linear-time property is medium-sized iff there are a winning strategies both for Ego in the game $(\overline{EA}, \Sigma, X)$ and for Alter in the game $(\overline{AE}, \Sigma, X)$.

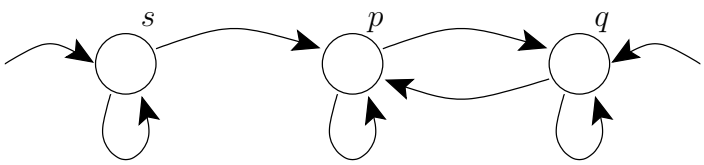

Figure 2.3.: A system Σ.

Consider the system Σ of Figure 2.3. Then $Sat(\Box\Diamond q)$, $Sat(\Diamond\Box(p \vee q))$ are large, $Sat(s)$, $Sat(s \wedge \mathsf{X} p)$ are medium-sized, and $Sat(\Diamond\Box q)$, $Sat(\Box\Diamond s)$ are small.

In Section 2.8, we see that, under weak assumptions, a linear-time property X is large in a system Σ iff it has probability one. Völzer et al. [2005] point out that the large linear-time properties usually are considered as fairness properties in the literature and therefore propose largeness as a characterisation of all fairness properties. For this reason, we say that a linear-time property X is a *fairness property* (or a *fairness assumption*) w. r. t. Σ iff X is large in Σ. A *fairness notion* maps each system Σ to a fairness property w. r. t. Σ.

Summarising, the following statements are equivalent:

- X is large in Σ.
- Ego has a winning strategy in $(\overline{AE}, \Sigma, X)$.
- X is a co-meager set in the Cantor topology relative to $path(\Sigma)$.
- X is a fairness property (or a fairness assumption) w. r. t. Σ.

For later use, we prove some closure properties of largeness and smallness. Proposition 2.4 and 2.5 are due to Völzer et al. [2005].

Proposition 2.4. *Let $\Sigma = (Q, S, \rightarrow, v)$ a system and Y, Z linear-time properties over Q.*

- *If Y is large in Σ and $Y \subseteq Z$, then Z is large in Σ.*
- *If Y is small in Σ and $Z \subseteq Y$, then Z is small in Σ.*

Proof. Both assertions follow immediately from the definition of largeness and smallness, respectively. □

Now, we show that largeness is closed under countable intersection and smallness, under countable union. We assume that finite sets are countable.

Proposition 2.5. *Let $I \neq \emptyset$ be a countable set, Σ a system and $X_i \subseteq path(\Sigma)$, for each $i \in I$. Then, the following statements hold:*

1. *If X_i is large, for each $i \in I$, then so is $X := \bigcap_{i \in I} X_i$.*
2. *If X_i is small, for each $i \in I$, then so is $X := \bigcup_{i \in I} X_i$.*

Proof. It suffices to prove 1. in the case that I is infinite. For finite I, 1. can be derived with Proposition 2.4. Then, 2. follows by duality.

Without loss of generality, we assume $I = \mathbb{N}$. For each $i \in \mathbb{N}$, using Proposition 2.1, let f_i be a progressive winning strategy for Ego in $(\overline{AE}, \Sigma, X_i)$. For $\alpha \in path(\Sigma)\downarrow$, we define $f(\alpha) := f_{|\alpha|}(\dots(f_1(f_0(\alpha)))\dots)$. Then, $Z^{E,f}_{\overline{AE}} \subseteq Z^{E,f_i}_{\overline{AE}}$, for each $i \in \mathbb{N}$. Therefore, f is a winning strategy for Ego in $(\overline{AE}, \Sigma, X)$. We conclude that X is large in Σ. □

We show that, as a consequence, the intersection of a medium-sized and a large, and the union of a medium-sized and a small linear-time property remain medium-sized.

Proposition 2.6. *Let $\Sigma = (Q, S, \rightarrow, v)$ be a system, X a medium-sized, and Y some arbitrary linear-time property over Q. Then, the following statements hold:*

1. *If Y is large, then $X \cap Y$ is medium-sized.*

2. *If Y is small, then $X \cup Y$ is medium-sized.*

Proof. Again, it suffices to prove 1. Part 2 of the assertion follows by duality.

Due to Proposition 2.4, it cannot be the case that $X \cap Y$ is large, as $X \cap Y \subseteq X$ and X is medium-sized. Note that $X \cap Y^c$ is small, due to similar reasons. It is impossible that $X \cap Y$ is small, because in this case $X = (X \cap Y) \cup (X \cap Y^c)$ would be small, due to Proposition 2.5. Thus, $X \cap Y$ is medium-sized. □

2.7. Word Fairness and State Fairness

We consider two special fairness notions: *word fairness* and *state fairness*. Let x be a path of Σ. We say that a path fragment $\alpha \in Q^+$ is *enabled at some state* $q \in Q$ iff $q\alpha$ is a path fragment of Σ_q. The path fragment α is *enabled at some position i of x* iff α is enabled at x_i. We say that α is *taken* at position i of x iff $\alpha \sqsubseteq x[i, \infty]$. The path x is *fair w. r. t. α* iff α is taken at infinitely many positions of x or it is enabled at only finitely many positions of x. The path x is *word fair w. r. t. Σ* iff it is fair w. r. t. each word $\alpha \in Q^+$. *Word fairness* maps Σ to the linear-time property WF_Σ containing the word fair paths w. r. t. Σ. As WF_Σ is the intersection of the fairness properties $\{x \mid x \text{ is fair w. r. t. } \alpha\}$, by Proposition 2.5, word fairness is a fairness notion.

The path x is *state fair* w. r. t. Σ iff each state q of Σ which is enabled at infinitely many positions of x is also taken at infinitely many positions of x. *State fairness* maps Σ to the linear-time property SF_Σ containing the state fair paths w. r. t. Σ. As $WF_\Sigma \subseteq SF_\Sigma$, SF_Σ is a fairness property. Hence, state fairness is a fairness notion.

A path fragment $\alpha \in Q^+$ is *reachable from a state* $q \in Q$ iff $\beta\alpha$ is a path fragment of Σ_q for some $\beta \in Q^*$. The path fragment α is *reachable from position i of x* iff α is reachable from x_i. For a path z over Q, we define

$$inf(z) := \{q \in Q \mid z \vDash \Box\Diamond q\}.$$

(We omit Q on the left hand side, as, throughout this document, it is clear from the context.)

We alternatively characterise state fairness in terms of reachability or b. s. c. c.s of Σ:

Proposition 2.7. *Let $\Sigma = (Q, S, \rightarrow, v)$ a system and $x \in path(\Sigma)$. Then, the following statements are equivalent:*

1. *The path x is state fair w. r. t. Σ.*

2. *Each state $q \in Q$ which is reachable from infinitely many position of x is also taken at infinitely many positions of x.*

3. We have $x = \alpha y$, where $y \in K^\omega$ and $inf(y) = K$, for some b. s. c. c. K of Σ.

Proof. We show that 1. implies 2. Suppose that $x \in SF_\Sigma$ and $q \in Q$ is reachable from infinitely many positions of x. As the number of states is finite, there is a $p \in Q$ that is taken infinitely often by x and from which q is reachable. Thus, we can choose $\beta \in Q^*$ such that βq is a path fragment of Σ_p. We show by induction over $|\beta|$ that, for each $\beta \in Q^*$ and $s \in Q$, if βs is a path fragment of Σ_p, then s is taken infinitely often by x. Part 2 of the assertion follows from this. If $|\beta| = 0$, then $s = p$. As p is taken infinitely often by x, s is also taken infinitely often by x. If $|\beta| > 0$, then, by the induction hypothesis, $s := \beta_{|\beta|-1}$ is taken infinitely often by x, as $\beta[0, |\beta| - 2]s = \beta$ is a path fragment of Σ_p. Because $s \to q$, q is enabled at infinitely many positions of x. Due to the state fairness of x, q is taken infinitely often by x.

We show that 2. implies 3. Suppose that x satisfies 2. We choose $\alpha \in Q^*$ and $y \in Q^\omega$ such that $x = \alpha y$ and $y \in (inf(x))^\omega$. Note that the states in $inf(x)$ belong to the same s. c. c. K of Σ. Thus, $y \in K^\omega$. If K is not a *b. s. c. c.*, then K has an outgoing edge to a state which is reachable from each position of x but never taken, a contradiction. Hence, K is a b. s. c. c. of Σ. As each state of K is reachable from each position of x, each state of K is taken infinitely often by x. We conclude that $inf(y) = K$.

We show that 3. implies 1. If x satisfies 3., then x is state fair w. r. t. Σ, as each state outside of K is enabled at only finitely many positions of x. □

Consider again the system Σ of Figure 2.3. Then s^ω, sp^ω, $spqpq^\omega$ are neither state, nor word fair w. r. t. Σ. The path $s(pq)^\omega$ is state, but not word fair w. r. t. Σ, as pp is enabled at infinitely many positions but never taken. In fact it is absolutely not straightforward to define a word fair path explicitly. Nevertheless, such paths exist, as word fairness is a fairness notion.

2.8. Probabilistic Largeness

As an alternative approach, we define largeness by means of probability theory. (For an introduction in the basic notions of probability theory, see, for example, the textbook of Krengel [2002].) Given a system $\Sigma = (Q, S, \to, v)$, let $\mathcal{B}$ be the σ-algebra generated by the shadows $\alpha\uparrow$, where $\alpha \in Q^*$. The elements of $\mathcal{B}$ are called *measurable* linear-time properties. Now, let $(Q^\omega, \mathcal{B}, \mu)$ be a probability space. For consistency, we require that $\mu(\alpha\uparrow) > 0$ iff $\alpha \in path(\Sigma)\downarrow$. The (Borel) measure μ is a *Markov measure* iff $\mu(\alpha qp\uparrow \mid \alpha q\uparrow)$ does not depend on α, where $\alpha q \in path(\Sigma)\downarrow$ and $p \in Q$. It is *bounded* iff there exists a constant $c > 0$ such that $\mu(\alpha q\uparrow \mid \alpha\uparrow) \geq c$, for each path fragment αq of Σ. As Q is finite, every Markov measure is bounded.

A measurable linear-time property X is *P-large* (in Σ w. r. t. μ) iff $\mu(X) = 1$. It is *P-medium-sized* (in Σ w. r. t. μ) iff $\mu(X) \in (0, 1)$. It is *P-small* (in Σ w. r. t. μ) iff $\mu(X) = 0$.

Proposition 2.8 and Theorem 2.9 are due to Varacca and Völzer [2006]. We give detailed proofs, which are based on their ideas.

Proposition 2.8. *Let X be a measurable linear-time property and $\mathcal{G} := (\overline{AE}, \Sigma, X)$. Suppose that μ is bounded and there is a bounded winning strategy either for Ego or for Alter in $\mathcal{G}$. Then, X is large iff X is P-large.*

Proof. First suppose that X is large. Then, there is a bounded winning strategy f for Ego in $\mathcal{G}$. We define $\ell \in \mathbb{N}$ such that, for each $\alpha \in \mathit{path}(\Sigma)\downarrow$, we have $|f(\alpha)| \leq |\alpha| + \ell$. Let $U_\alpha := (\alpha\uparrow \setminus f(\alpha)\uparrow) \cap \mathit{path}(\Sigma)$, for each $\alpha \in \mathit{path}(\Sigma)\downarrow$. Given $k \in \mathbb{N}$, we write Path_k for the set of all path fragments $\alpha \in \mathit{path}(\Sigma)\downarrow$ of length $k \cdot \ell$. We define W_k as the set of all paths of Σ that are not compatible with any path fragment $f(\alpha)$, where $\alpha \in \mathit{Path}_k$. That is

$$W_k := \bigcup_{\alpha \in \mathit{Path}_k} U_\alpha. \tag{2.5}$$

It is not too hard to see that

$$\bigcap_{n=0}^{\infty} \bigcup_{k=n}^{\infty} (\mathit{path}(\Sigma) \setminus W_k) \subseteq Z^{E,f}_{\overline{AE}}; \tag{2.6}$$

i. e. a path of Σ that, for infinitely many $k \in \mathbb{N}$, does not belong to W_k is a path of a play in which Ego plays according to f. Thus, it is sufficient to show that the left hand side of (2.6) has probability one. To accomplish this, we prove that, for each $n \in \mathbb{N}$, $Y_n := \bigcap_{k=n}^{\infty} W_k$ has probability zero.

For ease of notation, we set $Y_n^m := \bigcap_{k=n}^{m-1} W_k$, where $0 \leq n \leq m$. With (2.5), it can easily be shown that

$$Y_n^m = \bigcup \{\alpha\uparrow \cap \mathit{path}(\Sigma) \mid \alpha \in \mathit{Path}_m \cap Y_n^m\downarrow\} \tag{2.7}$$

$$= \bigcup \{U_\alpha \mid \alpha \in \mathit{Path}_{m-1} \cap Y_n^{m-1}\downarrow\} \qquad (0 \leq n < m). \tag{2.8}$$

As μ is bounded, there is a $c > 0$ such that $\mu(\alpha q\uparrow \mid \alpha\uparrow) \geq c$, for each path fragment αq of Σ. Therefore, we have $\mu(f(\alpha)\uparrow \mid \alpha\uparrow) \geq c^\ell =: d > 0$, for each path fragment α of Σ. Hence,

$$\mu(U_\alpha \mid \alpha\uparrow) = 1 - \mu(f(\alpha)\uparrow \mid \alpha\uparrow) \leq 1 - d \qquad (\alpha \in \mathit{path}(\Sigma)\downarrow). \tag{2.9}$$

We prove by induction that $\mu(Y_n^m) \leq (1-d)^{m-n}$, for each $m \geq n$. For $m = n$, there

is nothing to show. For $m \to m+1$, we have

$$\begin{aligned}
\mu(Y_n^{m+1}) &= \sum_{\alpha \in Path_m \cap Y_n^{m\downarrow}} \mu(U_\alpha) \\
&= \sum_{\alpha \in Path_m \cap Y_n^{m\downarrow}} \mu(U_\alpha | \alpha\uparrow) \cdot \mu(\alpha\uparrow) \\
&\leq (1-d) \cdot \sum_{\alpha \in Path_m \cap Y_n^{m\downarrow}} \mu(\alpha\uparrow) \\
&= (1-d) \cdot \mu(Y_n^m) \\
&\leq (1-d)^{(m+1-n)},
\end{aligned}$$

where the first line is a consequence of (2.8), the second is trivial, the third follows from (2.9), the fourth, from (2.7) and the fifth, from the induction hypothesis. Now, we let $m \to \infty$, which yields $\mu(Y_n) = 0$. (As $\{Y_n^m\}_m$ is a non-increasing sequence, we have $\lim_{m\to\infty} \mu(Y_n^m) = \mu(Y_n)$.)

On the other hand, suppose X is not large. Then, there exists a bounded winning strategy g for Alter in $\mathcal{G}$. Thus, g is winning for Ego in $(\overline{EA}, \Sigma, X^c)$. If we enlarge the winning condition by the paths that are not compatible with $g(\lambda)$, Ego does not need to do the first move; i.e. g is a bounded winning strategy for Ego in $(\overline{AE}, \Sigma, X^c \cup (g(\lambda)\uparrow)^c)$. By the part above,

$$1 = \mu(X^c \cup (g(\lambda)\uparrow)^c) \leq \mu(X^c) + \mu((g(\lambda)\uparrow)^c).$$

As $\mu(g(\lambda)\uparrow) > 0$, we have $\mu((g(\lambda)\uparrow)^c) < 1$. Thus, $\mu(X^c) > 0$, and therefore $\mu(X) < 1$. We conclude that X is not P-large. □

Now, we are ready to show that the two notions of largeness coincide.

Theorem 2.9. *Let Φ be a path formula, and suppose that μ is bounded. Then, the following statements hold:*

1. *$Sat(\Phi)$ is large iff $Sat(\Phi)$ is P-large w. r. t. μ.*
2. *$Sat(\Phi)$ is medium-sized iff $Sat(\Phi)$ is P-medium-sized w. r. t. μ.*
3. *$Sat(\Phi)$ is small iff $Sat(\Phi)$ is P-small w. r. t. μ.*

Proof. By Theorem 2.2, there is a positional winning strategy f either for Ego or for Alter in $(\overline{AE}, \Sigma, Sat(\Phi))$. As Q is finite, the strategy f is bounded. Moreover, it has been shown by Vardi [1985] that $Sat(\Phi)$ is measurable. Hence, by Proposition 2.8, 1. holds.

Part 3 follows from 1., as $Sat(\Phi)$ is (P-)small iff $Sat(\neg\Phi)$ is (P-)large. Part 2 is a consequence of 1. and 3., because $Sat(\Phi)$ is (P-)medium-sized iff $Sat(\Phi)$ is neither (P-)large nor (P-)small. □

2.9. Complete Fairness

Let Σ be a system. By Proposition 2.4, each fairness property X w. r. t. Σ is *sound*, in the sense that if $X \subseteq Y$ then Y is large in Σ. Thus, if we show that $X \subseteq Y$, then we have a proof for the fact that Y is large in Σ; however, there may be large linear-time properties Y with $X \not\subseteq Y$. Therefore, the largeness of Y in Σ cannot be shown by a proof based on $X \subseteq Y$.

A natural question is whether there is also a *complete* fairness property X w. r. t. Σ; i. e., if Y is large in Σ, then $X \subseteq Y$. Hence, if Y is large in Σ, there exists a proof for this fact based on $X \subseteq Y$. Such a complete fairness property w. r. t. Σ has to be a subset of the intersection over all fairness properties w. r. t. Σ. But, in general, this intersection is not large in Σ, and therefore a complete fairness property w. r. t. Σ does not exist:

Let $\Sigma = (Q, S, (Q, Q), v)$ be a system with at least two states. For $x \in Q^\omega$, $\{x\}^c$ is a fairness property w. r. t. Σ. But $\bigcap_{x \in Q^\omega} \{x\}^c = \emptyset$ is small in Σ. (This example is due to Völzer et al. [2005].)

Nevertheless, if we restrict to a limited family of linear-time properties, such a complete fairness property possibly exists. Let Ω be a family of linear-time properties, Σ a system and X a fairness property in Σ. We say that X is Ω-*complete w. r. t.* Σ iff $X \subseteq Y$, for each $Y \in \Omega$ that is large in Σ.

As an example of a complete fairness property, we mention a result of Varacca and Völzer [2006]:

Theorem 2.10. *Let Σ be a system. Then WF_Σ is $Sat(L(\mathsf{X}, \mathsf{Y}, \mathsf{U}, \mathsf{S}))$-complete w. r. t. Σ.*

The proof depends on the fact that each path which is word fair w. r. t. Σ belongs to $Z^{E,f}_{\overline{AE}}$, where f is a positional strategy in Σ.

3. Fair Checking of an LTL Formula

Let Σ_0 be a system and $\Phi_0 \in L(\mathsf{X}, \mathsf{Y}, \mathsf{U}, \mathsf{S})$ a *specification*. For technical reasons, we assume, without loss of generality, that Σ_0 is *reduced*; i. e. each state of Σ_0 is reachable from some starting state of Σ_0. The task is to check whether $Sat(\Phi_0)$ is large, medium-sized or small in Σ_0. We henceforth refer to this problem as the problem of *generalised fair model checking*. (Due to Proposition 2.4, $Sat(\Phi_0)$ is large in Σ_0 iff there exists some fairness assumption F w. r. t. Σ_0 such that $F \subseteq Sat(\Phi_0)$.)

We examine an algorithm due to Courcoubetis and Yannakakis [1995], which is based on several transformations of the system and the specification each of which preserve the degree of satisfaction of the specification (large, medium-sized or small). We write Σ_i and Φ_i for the system and specification after the i-th transformation step. The total number of transformation steps is n. Each transformation step reduces the number of temporal operators in the specification such that the final specification Φ_n is a state formula, while retaining how much the system satisfies the specification (large, medium-sized or small). Thus, the original problem has been reduced to checking whether a linear-time property expressible by a state formula is large, medium-sized or small in a system. But this can be accomplished easily: the linear-time property $Sat(\Phi_n)$ is large in a system iff it is satisfied by each starting state, small iff it is satisfied by none of them and medium-sized iff it is neither large nor small.

The original algorithm of Courcoubetis and Yannakakis [1995] only allows specifications in $L(\mathsf{X}, \mathsf{U})$ and does not create a witness. They proved correctness in a probabilistic framework using the theory of Markov chains. In contrast, our proof of correctness is based on path games; thus, we hope that it will be more understandable for the reader. We develop additional transformations such that specifications containing past operators can be checked and a special transformation that improves performance. In Chapter 4, we explain how to compute witnesses.

As the algorithm of Courcoubetis and Yannakakis [1995] contains only two transformations, it was convenient to explain one of them in detail and give a sketch of the other one. Unfortunately, in our case this is not possible, as we develop five transformations which are too different from each other. On the other hand, it would be lengthy to examine each of them from scratch, as they are too similar to each other. Therefore, in Section 3.1), we give a formal definition of transformations and point out properties of transformations that are necessary and sufficient for correctness. We exploit these techniques in the Sections 3.2 up to 3.6 to develop several transformations which take part of our version of the algorithm. The methods of Section 3.1

also facilitate the research for completely new transformations. Finally, in Section 3.7, we analyse the complexity of the algorithm.

3.1. Common Properties of the Transformations

The original algorithm of Courcoubetis and Yannakakis [1995] chooses in each step a subformula Θ of the current specification (later called *feasible* formula) which is of the form $\xi \mathsf{U} \psi$ or $\mathsf{X}\xi$, where ξ, ψ are state formulae. The subformula Θ is replaced by a fresh atomic proposition in the specification, reducing its number of temporal operators. The system is blown up while retaining the degree of satisfaction of the specification (large, medium-sized or small). We generalise this as follows:

Each *transformation* of systems and specifications has a set of *feasible* formulae. We say that a formula is *feasible* if it is feasible for any of the transformations defined in the Sections 3.2 up to 3.6. A transformation $\mathcal{T}$ maps a reduced system $\Sigma = (Q, S, \rightarrow, v)$, a specification Φ and a feasible formula Θ to a new system $\Sigma' := (Q', S', \rightarrow', v') := \mathcal{T}(\Sigma, \Theta)$ and a new specification $\Phi' := \mathcal{T}(\Phi, \Theta)$. Note that we do not require that Θ is a subformula of Φ.

Let θ be a *fresh* atomic proposition; i. e. $\theta \not\leq \Phi, \Theta$ and, for each $q \in Q$, $\theta \notin v(q)$. Then, $\Phi' := [\Phi]^{\theta}_{\Theta}$. As a rule, Greek capital letters denote feasible formulae, and Greek small letters, the corresponding fresh atomic propositions. The new system Σ' satisfies the following basic conditions T1, T2 and T3:

(We recall that $(Q, \Theta) = Q \times \{\Theta\}$.)

T1: We have $Q' \subseteq (Q, \Theta) \cup (Q, \neg\Theta)$.

T2: The new system Σ' is reduced.

T3: If $(q, \Theta) \in Q'$, then $v'(q, \Theta) = v(q) \cup \{\theta, (q, \Theta)\}$.
If $(q, \neg\Theta) \in Q'$, then $v'(q, \neg\Theta) = v(q) \cup \{(q, \neg\Theta)\}$.

We assume, without loss of generality, that each state q' of Σ' is a fresh atomic proposition; i. e. $q' \not\leq \Phi, \Theta$ and, for each $q \in Q$, $q' \notin v(q)$.

In Figure 3.1, there is a system Σ, and, in Figure 3.2, there is a possible new system Σ', where $\Theta = \Diamond(\eta \wedge \rho)$ is the feasible formula. Note that T1, T2 and T3 hold. If $\Phi = \zeta \wedge \Diamond(\eta \wedge \rho)$, then $\Phi' = \zeta \wedge \theta$.

The *projection* π maps a sequence $(x_0, \Theta_0)(x_1, \Theta_1)\ldots$ over Q' to the sequence $x_0 x_1 \ldots$ over Q; thus, π eliminates the last component from each state in a sequence. If no confusion arises, we write x instead of $\pi(x')$, where x' is an arbitrary sequence over Q'.

The transformation $\mathcal{T}$ is *correct* iff, for an arbitrary reduced system Σ, specification Φ and feasible formula Θ, the following conditions are satisfied:

Let $\Sigma' := \mathcal{T}(\Sigma, \Theta)$ and $\Phi' := \mathcal{T}(\Phi, \Theta)$.

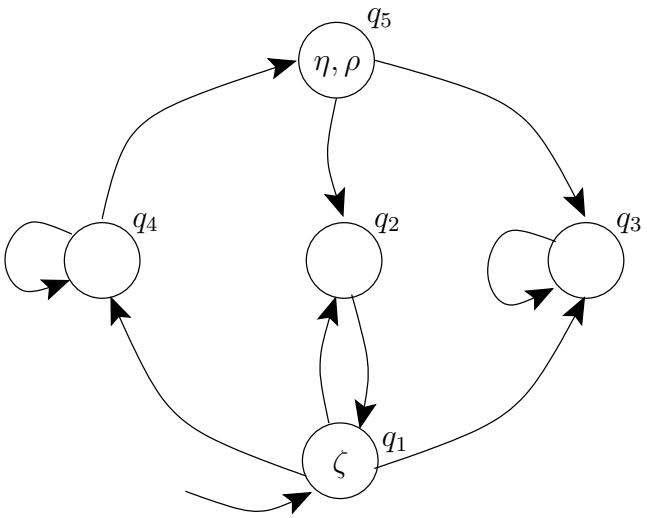

Figure 3.1.: A system Σ.

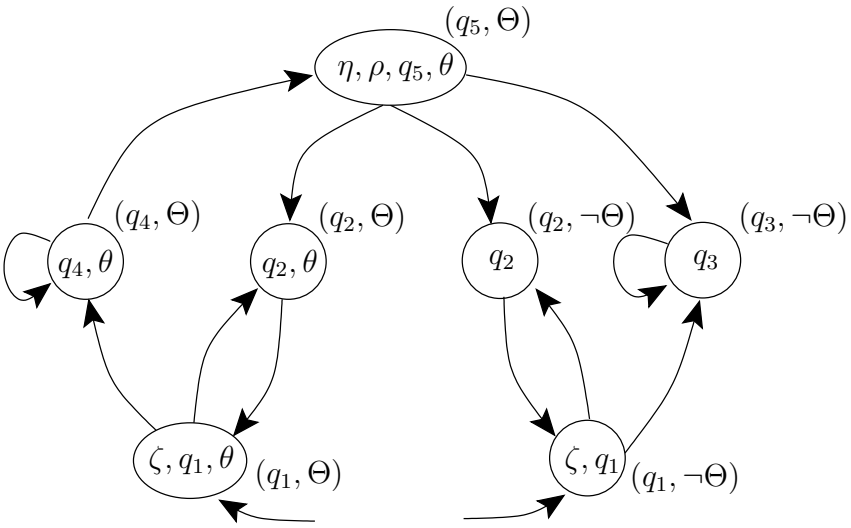

Figure 3.2.: A possible new system $\Sigma' := \mathcal{T}(\Sigma, \Theta)$, where Σ is as in Figure 3.1 and $\Theta = \Diamond(\eta \wedge \rho)$. (We suppose that Θ is feasible for $\mathcal{T}$.)

1. The linear-time property $Sat(\Phi)$ is large in Σ iff $Sat(\Phi')$ is large in Σ'.
2. The linear-time property $Sat(\Phi)$ is medium-sized in Σ iff $Sat(\Phi')$ is medium-sized in Σ'.
3. The linear-time property $Sat(\Phi)$ is small in Σ iff $Sat(\Phi')$ is small in Σ'.

Note that condition 2 and 3 can be derived from condition 1; i. e. condition 2 and 3 are redundant.

Consider the systems Σ and Σ' of Figure 3.1 and 3.2. Again, $\Phi := \zeta \wedge \Diamond(\eta \wedge \rho)$, $\Theta := \Diamond(\eta \wedge \rho)$ and therefore $\Phi' := \zeta \wedge \theta$. As only one of the starting states of Σ' satisfies Φ', $Sat(\Phi')$ is medium-sized in Σ'. Indeed, $Sat(\Phi)$ is medium-sized in Σ, too: Ego wins $(\overline{EA}, \Sigma, Sat(\Phi))$ if he starts with $q_1q_4q_5$. Alter wins $(\overline{AE}, \Sigma, Sat(\Phi))$ if he starts with q_1q_3. Thus, it may be the case that $\Sigma' = \mathcal{T}(\Sigma, \Theta)$ and $\Phi' = \mathcal{T}(\Phi, \Theta)$, where $\mathcal{T}$ is a correct transformation for which Θ is feasible.

In a transformation step, the algorithm chooses a (correct) transformation and a subformula of the current specification that is feasible for this transformation and contains at least one temporal operator. With this transformation, the algorithm eliminates at least one temporal operator from the specification. Thus, it suffices to define correct transformations such that each of $\xi \,\mathsf{U}\, \psi$, $\mathsf{X}\xi$, $\xi \,\mathsf{S}\, \psi$ and $\mathsf{Y}\xi$ is feasible, where ξ and ψ are arbitrary state formulae. But before we do so, we provide some tools which facilitate the development of transformations and the proof of correctness.

Lemma 3.1. *A transformation $\mathcal{T}$ is correct iff the following conditions A1 and A2 hold for each reduced system Σ and formula Θ that is feasible for $\mathcal{T}$.*

A1: *The linear-time property $Sat(\Sigma', \Box(\Theta \Leftrightarrow \theta))$ is large in Σ'.*

A2: *Let Υ be a path formula with $\theta, q' \not\preceq \Upsilon$, for each $q' \in Q'$.*
Then, $Sat(\Sigma, \Upsilon)$ is large in Σ iff $Sat(\Sigma', \Upsilon)$ is large in Σ'.

In both cases, $\Sigma' := (Q', S', \rightarrow', v') := \mathcal{T}(\Sigma, \Theta)$ and $\theta := \mathcal{T}(\Theta, \Theta)$ is the fresh atomic proposition.

Proof. Let $\Sigma = (Q, S, \rightarrow, v)$ be a reduced system, Φ a specification and Θ a formula that is feasible for $\mathcal{T}$. Furthermore, suppose that A1 and A2 hold.

First, we prove that $Sat(\Sigma, \Phi)$ is large in Σ iff $Sat(\Sigma', \Phi')$ is large in Σ', where $\Phi' := \mathcal{T}(\Phi, \Theta)$. Suppose $Sat(\Sigma, \Phi)$ is large in Σ. With A2, $Sat(\Sigma', \Phi)$ is large in Σ'. Thus, with A1 and Proposition 2.5, $Sat(\Sigma', \Phi) \cap Sat(\Sigma', \Box(\Theta \Leftrightarrow \theta))$ is large in Σ'. As $\Phi' = [\Phi]^{\theta}_{\Theta}$, we have $Sat(\Sigma', \Phi) \cap Sat(\Sigma', \Box(\Theta \Leftrightarrow \theta)) \subseteq Sat(\Sigma', \Phi')$. Applying Proposition 2.4, $Sat(\Sigma', \Phi')$ is large in Σ'. It can be shown in a similar way that if $Sat(\Sigma, \Phi)$ is not large in Σ then $Sat(\Sigma', \Phi')$ is not large in Σ'.

Note that it can be derived only from A1 that $Sat(\Sigma', \Phi)$ is large in Σ' iff $Sat(\Sigma', \Phi')$ is large in Σ'. We use this fact below.

The linear-time property $Sat(\Sigma, \Phi)$ is small in Σ iff $Sat(\Sigma, \neg\Phi)$ is large in Σ. By the part above, the latter is equivalent to $Sat(\Sigma', \mathcal{T}(\neg\Phi, \Theta))$ being large in Σ'. As $\mathcal{T}(\neg\Phi, \Theta) = \neg\Phi'$, this is equivalent to $Sat(\Sigma', \Phi')$ being small in Σ'.

Because a linear-time property is medium-sized iff it is neither large nor small, we conclude that $Sat(\Sigma, \Phi)$ is medium-sized in Σ iff $Sat(\Sigma', \Phi')$ is medium-sized in Σ'. Thus, $\mathcal{T}$ is correct.

Now, suppose A1 does not hold for some reduced system $\Sigma = (Q, S, \rightarrow, v)$ and some formula Θ which is feasible for $\mathcal{T}$. First, we consider the case that Θ is not a single atomic proposition. We define $\hat{\Theta}$ by replacing each atomic proposition ζ in Θ by $\zeta \vee (\eta \vee \neg\eta)$, where $\eta \neq \theta$ is an atomic proposition with $\eta \not\preceq \Theta$. Clearly, $\Phi := \Box(\hat{\Theta} \Leftrightarrow \Theta)$ is satisfied by each path of Σ. Therefore, $Sat(\Sigma, \Phi)$ is large in Σ. As Θ is not an atomic proposition, $\Theta \not\preceq \hat{\Theta}$. Hence, $\Phi' := \mathcal{T}(\Phi, \Theta) = \Box(\hat{\Theta} \Leftrightarrow \theta)$. The new formula Φ' is satisfied by exactly the paths of Σ' that satisfy $\Box(\Theta \Leftrightarrow \theta)$. As A1 does not hold, $Sat(\Sigma', \Phi')$ is not large in Σ'. We conclude that $\mathcal{T}$ is not correct in this case.

We consider the case that Θ is a single atomic proposition. As $Sat(\Sigma', \Box(\Theta \Leftrightarrow \theta))$ is not large in Σ', its complement $Sat(\Sigma', \Diamond(\Theta \not\Leftrightarrow \theta))$ is not small in Σ'. Thus, there is a state q' of Σ' with $q' \vDash \Theta \not\Leftrightarrow \theta$. Suppose $q' \vDash \Theta \wedge \neg\theta$. (The other case is similar.) Hence, $q \vDash \Theta$. Let $\Phi := \Diamond(q \wedge \neg\Theta)$. As no path of Σ satisfies Φ, $Sat(\Sigma, \Phi)$ is small in Σ. Let $\Phi' := \mathcal{T}(\Phi, \Theta) = \Diamond(q \wedge \neg\theta)$. Then, $Sat(\Sigma', \Phi')$ is not small in Σ', as $q' \vDash \neg\theta$. We conclude that $\mathcal{T}$ is not correct.

Finally, suppose A1 holds but A2 is not true for a reduced system $\Sigma = (Q, S, \rightarrow, v)$ and some formula Θ that is feasible for $\mathcal{T}$. Suppose there is a path formula Υ with $\theta, q' \not\preceq \Upsilon$, for each $q' \in Q'$, such that $Sat(\Sigma, \Upsilon)$ is not large in Σ iff $Sat(\Sigma', \Upsilon)$ is large in Σ'. From above, we know that $Sat(\Sigma', \Upsilon)$ is large in Σ' iff $Sat(\Sigma', \Upsilon')$ is large in Σ'. Thus, $Sat(\Sigma, \Upsilon)$ is not large in Σ iff $Sat(\Sigma', \Upsilon')$ is large in Σ'. We conclude that $\mathcal{T}$ is not correct. □

Note that, for Σ and Σ' as described in Figure 3.1 and 3.2 and $\Theta = \Diamond(\eta \wedge \rho)$, A1 of Lemma 3.1 holds. As far as A2 is concerned, the reader may figure out that, even after a moment of serious pondering, he will not be able to give a counterexample. In fact, $\Sigma' = \mathcal{T}_{\mathsf{U}}(\Sigma, \Theta)$, where $\mathcal{T}_{\mathsf{U}}$ is the transformation described in Section 3.2. In Section 3.2, we show that $\mathcal{T}_{\mathsf{U}}$ is correct and hence A2 is true.

So far, the question arises if there are distinct correct transformations $\mathcal{T}$ and $\hat{\mathcal{T}}$ with the same set of feasible formulae. The answer is no, as stated in Lemma 3.3 below. In other words, if there is a transformation $\mathcal{T}$ such that a certain path formula Θ is feasible for $\mathcal{T}$, there is no need to search for a somehow better transformation $\hat{\mathcal{T}}$ such that Θ is also feasible for $\hat{\mathcal{T}}$. Under the assumption that $\mathcal{T}$ is correct, Lemma 3.3 also provides an explicit definition of the new set of states Q'. Thus, when specifying a concrete transformation $\mathcal{T}$, it suffices to define the new state relation $\rightarrow'$ and the new set of starting states S', and to show that $\mathcal{T}$ is correct and satisfies T2.

First of all, we need some notations. We partition the set of states Q into

- $Q^L_\Theta := \{q \in Q \mid Sat(\Box(q \Rightarrow \Theta))$ is large in $\Sigma\}$,
- $Q^S_\Theta := \{q \in Q \mid Sat(\Box(q \Rightarrow \neg\Theta))$ is large in $\Sigma\}$,
- $Q^M_\Theta := Q \setminus (Q^L_\Theta \cup Q^S_\Theta)$.

If Θ does not contain past operators, there is an alternative characterisation of this partition:

Lemma 3.2. *Let $\Sigma = (Q, S, \rightarrow, v)$ be as defined above and $\Theta \in L(\mathsf{X}, \mathsf{U})$. Then, we have:*

- $Q^L_\Theta = \{q \mid Sat(\Theta)$ *is large in* $\Sigma_q\}$,
- $Q^M_\Theta = \{q \mid Sat(\Theta)$ *is medium-sized in* $\Sigma_q\}$,
- $Q^S_\Theta = \{q \mid Sat(\Theta)$ *is small in* $\Sigma_q\}$.

Proof. We prove the assertion in the case of Q^L_Θ. Suppose that $Sat(\Theta)$ is large in Σ_q. We show that $WF_\Sigma \subseteq Sat(\Box(q \Rightarrow \Theta))$. Let $x \in WF_\Sigma$, and suppose $x, i \vDash q$. As $x[i, \infty] \in WF_{\Sigma_q}$, we conclude, applying Theorem 2.10, that $x[i, \infty] \vDash \Theta$. Because $\Theta \in L(\mathsf{X}, \mathsf{U})$, we have $x, i \vDash \Theta$. Therefore, $x \vDash \Box(q \Rightarrow \Theta)$.

Now, suppose that $Sat(\Box(q \Rightarrow \Theta))$ is large in Σ. We show that $WF_{\Sigma_q} \subseteq Sat(\Theta)$. Let $x \in WF_{\Sigma_q}$, and choose $\alpha \in Q^*$ such that $\alpha x \in WF_\Sigma$. Due to Theorem 2.10, we have $\alpha x \vDash \Box(q \Rightarrow \Theta)$. As $\Box(q \Rightarrow \Theta) \in L(\mathsf{X}, \mathsf{U})$, $x \vDash q \Rightarrow \Theta$. Hence, $x \vDash \Theta$.

Now, we examine the case of Q^S_Θ. Note that $Sat(\Theta)$ is small in Σ_q iff $Sat(\neg\Theta)$ is large in Σ_q. As the assertion holds in the case of Q^L_Θ, $Sat(\neg\Theta)$ is large in Σ_q iff $Sat(\Box(q \Rightarrow \neg\Theta))$ is large in Σ.

In the case of Q^M_Θ, the assertion follows from the fact that $Sat(\Theta)$ is medium-sized in Σ_q iff it is neither large nor small in Σ_q. □

In Figure 3.3, we illustrate the partition of Q. We consider the system Σ of Figure 3.1 and $\Theta = \Diamond(\eta \wedge \rho)$.

Lemma 3.3. *Let $\mathcal{T}, \hat{\mathcal{T}}$ be correct transformations, Θ a formula which is feasible for $\mathcal{T}$ and $\hat{\mathcal{T}}$ and $\Sigma = (Q, S, \rightarrow, v)$ a reduced system. We define $\Sigma' := (Q', S', \rightarrow', v') := \mathcal{T}(\Sigma, \Theta)$ and $\hat{\Sigma} := (\hat{Q}, \hat{S}, \hat{\rightarrow}, \hat{v}) := \hat{\mathcal{T}}(\Sigma, \Theta)$. Then, Q' is the smallest set such that*

- $(q, \Theta) \in Q'$ *if* $q \in Q^L_\Theta$,
- $(q, \neg\Theta) \in Q'$ *if* $q \in Q^S_\Theta$ *and*
- $(q, \Theta), (q, \neg\Theta) \in Q'$ *if* $q \in Q^M_\Theta$.

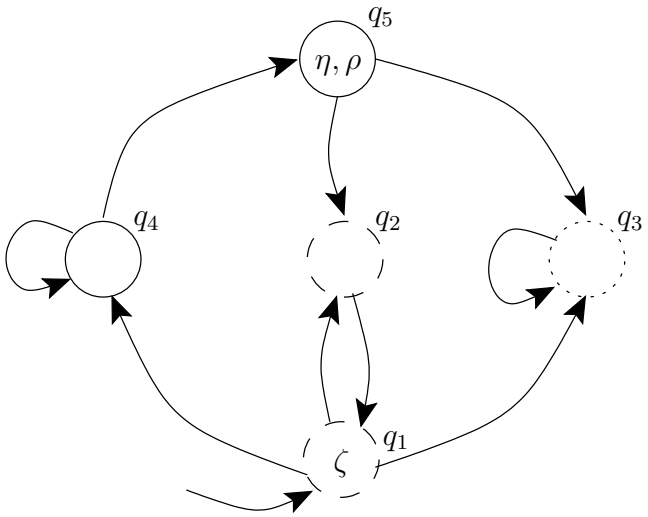

Figure 3.3.: The system Σ of Figure 3.1. We illustrate the partition of Q into Q^L_Θ, Q^M_Θ and Q^S_Θ for $\Theta = \Diamond(\eta \wedge \rho)$. The states in Q^L_Θ have continuous, the states in Q^M_Θ, dashed, and the states in Q^S_Θ, dotted borders.

Furthermore, $\Sigma' = \hat{\Sigma}$.

Proof. Let $q \notin Q^S_\Theta$. We show that $(q, \Theta) \in Q'$. As $q \notin Q^S_\Theta$, $Sat(\Box(q \Rightarrow \neg\Theta))$ is not large in Σ. Thus, $Sat(\Diamond(q \wedge \Theta))$ is not small in Σ. Because $\mathcal{T}$ is correct, $Sat(\Diamond(q \wedge \theta))$ is not small in Σ'. Therefore, Σ' has a state that satisfies $q \wedge \theta$. By the construction of v' (condition T3), we have $(q, \Theta) \in Q'$.

Now, suppose $q \in Q^S_\Theta$. We show that $(q, \Theta) \notin Q'$. As $q \in Q^S_\Theta$, $Sat(\Box(q \Rightarrow \neg\Theta))$ is large in Σ. Thus, $Sat(\Diamond(q \wedge \Theta))$ is small in Σ. Because $\mathcal{T}$ is correct, $Sat(\Diamond(q \wedge \theta))$ is small in Σ'. Therefore, Σ' does not have a state that is reachable from some state in S' and satisfies $q \wedge \theta$. Because of the construction of v' (condition T3) and as Σ' is reduced (T2), we have $(q, \Theta) \notin Q'$.

It can be shown in a similar way that $q \in Q^L_\Theta$ iff $(q, \neg\Theta) \notin Q'$. The first part of the assertion follows.

As this characterisation of Q' does not depend on $\mathcal{T}$, we have $Q' = \hat{Q}$. We show that $S' = \hat{S}$. It suffices to show $S' \subseteq \hat{S}$. Suppose $(q, \Theta) \in S'$. Then, $Sat(q \wedge \theta)$ is not small in Σ'. As $\mathcal{T}$ is correct, $Sat(q \wedge \Theta)$ is not small in Σ. Because $\hat{\mathcal{T}}$ is correct, $Sat(q \wedge \theta)$ is not small in $\hat{\Sigma}$. Hence, $(q, \Theta) \in \hat{S}$. Analogously, $(q, \neg\Theta) \in S'$ implies $(q, \neg\Theta) \in \hat{S}$.

We show that $\rightarrow' = \hat{\rightarrow}$. Again, it suffices to show $\rightarrow' \subseteq \hat{\rightarrow}$. Suppose $(p, \Theta_1) \rightarrow' (q, \Theta_2)$. Then, $Sat(\Diamond(p \wedge \Theta_1 \wedge \mathsf{X}(q \wedge \Theta_2)))$ is not small in Σ. Hence, $(p, \Theta_1) \hat{\rightarrow} (q, \Theta_2)$.

Because of $Q' = \hat{Q}$ and T3, we have $v' = \hat{v}$. Putting all together, we conclude that $\Sigma' = \hat{\Sigma}$. □

Let $\Sigma = (Q, S, \rightarrow, v)$ and $\Sigma' = (Q', S', \rightarrow', v')$ be the systems of Figure 3.1 and 3.2, and $\Theta = \Diamond(\eta \wedge \rho)$. Then, Q' consists of the states announced in Lemma 3.3. Note that Q' and v' can immediately be derived from Lemma 3.3 and T3, but it is, in general, not straightforward to find appropriate S' and $\rightarrow'$.

3.2. The Transformation $\mathcal{T}_{\mathsf{U}}$

For the transformation $\mathcal{T}_{\mathsf{U}}$, each formula $\xi \mathsf{U} \psi$ is feasible, where ξ, ψ are state formulae. Let $\Sigma = (Q, S, \rightarrow, v)$ be a reduced system, Φ a specification, $\Theta = \xi \mathsf{U} \psi$ a feasible formula, $\Sigma' := (Q', S', \rightarrow', v') := \mathcal{T}_{\mathsf{U}}(\Sigma, \Theta)$ and $\Phi' := \mathcal{T}_{\mathsf{U}}(\Phi, \Theta)$. According to Section 3.1, it suffices to define $\rightarrow'$ and S'.

The new state relation $\rightarrow'$ is the smallest relation which satisfies the conditions U1 and U2 below. Let $p, q \in Q$ with $p \rightarrow q$ and $\Theta_1, \Theta_2 \in \{\Theta, \neg\Theta\}$ such that $(p, \Theta_1), (q, \Theta_2) \in Q'$.

U1: If $p \notin Q^M_\Theta$, then $(p, \Theta_1) \rightarrow' (q, \Theta_2)$.

U2: If $p \in Q^M_\Theta$, then $(p, \Theta_2) \rightarrow' (q, \Theta_2)$.

The new set of starting states is $S' := Q' \cap (S, \{\Theta, \neg\Theta\})$. Note that $\mathcal{T}_{\mathsf{U}}$ does not produce unreachable states. Hence, T2 is satisfied.

The state relations of Σ and Σ' can be visualised by *block graphs*. A *block graph* of Σ consists of some *blocks* $R_1, \ldots, R_m \subseteq Q$ and a relation $\bullet\!\!\longrightarrow$ over those blocks. The blocks $R_1, \ldots, R_m$ are drawn as areas, and the relation $\bullet\!\!\longrightarrow$ is visualised by arrows between the corresponding areas. The area for R_i encloses the area for R_j, where $1 \leq i, j \leq m$, iff $R_j \subseteq R_i$. The meaning of $\bullet\!\!\longrightarrow$ is as follows. If $R_i \bullet\!\!\longrightarrow R_j$, then, for each $q \in R_i$, we have $path(\Sigma_q) \cap R_i^+ R_j\!\uparrow \neq \emptyset$. In other words, if $R_i \bullet\!\!\longrightarrow R_j$, then, from each state $q \in R_i$, there is a path of Σ_q starting in q and leaving R_i over an edge ending in a state in R_j. Moreover, if $p \rightarrow q$, for some $p, q \in Q$, then there exist i, j, where $1 \leq i, j \leq m$, such that $p \in R_i$, $q \in R_j$, and $R_i \bullet\!\!\longrightarrow R_j$ or $R_i = R_j$. That means, if an edge of Σ runs between two different blocks R_i, R_j, then $R_i \bullet\!\!\longrightarrow R_j$.

Figure 3.4 shows a block graph of Σ. It can be verified by trivial arguments. As an example, we show that, for each $q \in Q^M_\Theta$, we have $path(\Sigma_q) \cap (Q^M_\Theta)^+ Q^L_\Theta\!\uparrow \neq \emptyset$. Let $q \in Q^M_\Theta$. As Ego has a winning strategy for $(\overline{EA}, \Sigma_q, Sat(\Theta))$, there is a path x of Σ_q with $x[0, i] \in sat(\xi \wedge \neg\psi)^+ sat(\psi)$. By the definition of Q^S_Θ, for $0 \leq j \leq i$, we have $x_j \notin Q^S_\Theta$. Moreover, $x_i \in sat(\psi) \subseteq Q^L_\Theta$. Thus, $x \in (Q^M_\Theta)^+ Q^L_\Theta\!\uparrow$. We conclude that $path(\Sigma_q) \cap (Q^M_\Theta)^+ Q^L_\Theta\!\uparrow \neq \emptyset$.

Figure 3.5 shows a block graph of Σ'. It can be easily derived from the block graph of Σ above.

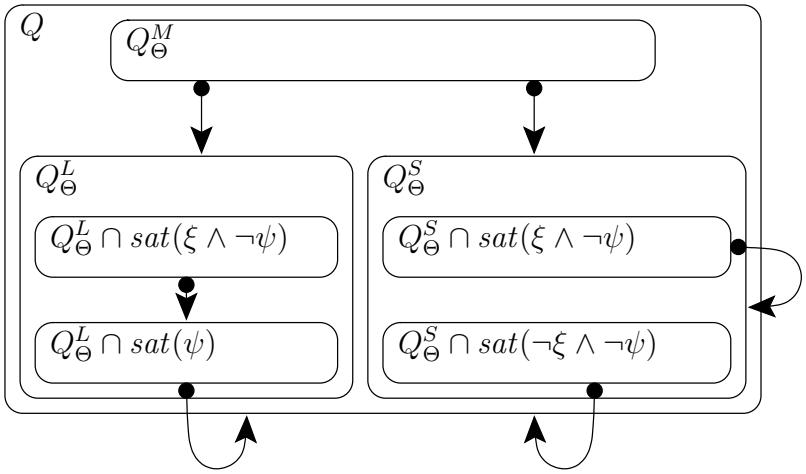

Figure 3.4.: A block graph of Σ, where $\Theta = \xi \,\mathsf{U}\, \psi$.

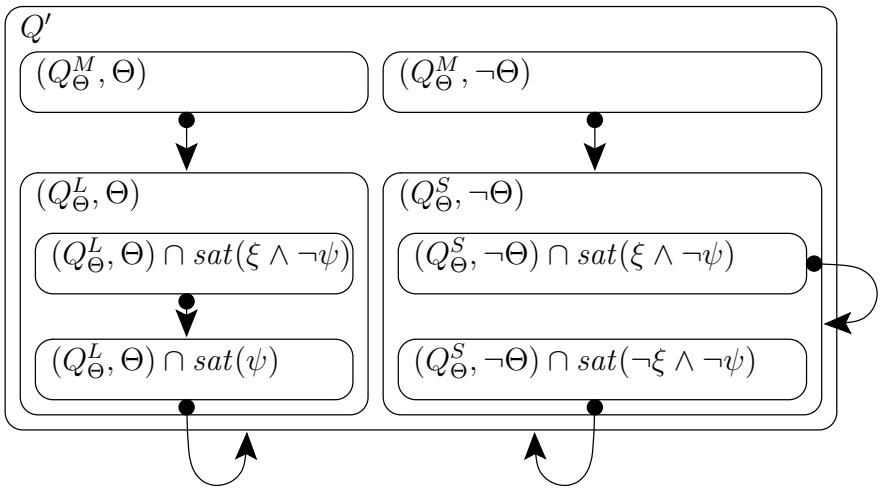

Figure 3.5.: A block graph of Σ', where $\Theta = \xi \,\mathsf{U}\, \psi$.

Now, we explain how the transformation $\mathcal{T}_{\mathsf{U}}$ can be computed efficiently. The main difficulty is to compute the partition of Q into Q^L_Θ, Q^M_Θ and Q^S_Θ. The transformations afterwards are straightforward. Firstly, the algorithm evaluates ξ and ψ at each state of Q. Clearly, all states $q \in Q$ with $q \vDash \psi$ belong to Q^L_Θ, and all states $q \in Q$ with $q \vDash \neg\xi \wedge \neg\psi$ belong to Q^S_Θ. To classify the remaining states the algorithm computes the s. c. c.s of $sat(\xi \wedge \neg\psi)$, the subgraph of Σ induced by the (remaining) states $q \in Q$ with $q \vDash \xi \wedge \neg\psi$. The algorithm travels through the s. c. c.s in reversed topological order; i. e., if there is an edge from the s. c. c. K_1 to the s. c. c. K_2, K_2 is visited first. The states of Σ are classified according to the following rules:

- If all outgoing edges of an s. c. c. go to states of Q^S_Θ, then the states of this s. c. c. are added to Q^S_Θ.
- If an s. c. c. has outgoing edges and all of them go to states of Q^L_Θ, then the states of this s. c. c. are added to Q^L_Θ.
- Otherwise, if the states of an s. c. c. are neither added to Q^L_Θ nor to Q^S_Θ, they are added to Q^M_Θ.

As the algorithm travels through the s. c. c.s in reversed topological order, each considered outgoing edge ends in an already classified state.

We show by induction over the number of s. c. c.s that have already been processed that the classification is correct for those s. c. c.s. Let K be the current s. c. c., and suppose the algorithm decides $K \subseteq Q^L_\Theta$. (The remaining cases can be treated in a similar way.)

For $q \in K$, we show $WF_\Sigma \subseteq Sat(\Box(q \Rightarrow \Theta))$. Thus, $Sat(\Box(q \Rightarrow \Theta))$ is large, and q has been classified correctly. Let $x \in WF_\Sigma$, and suppose $x_i = q$. As x is word fair (and therefore state fair) and K has outgoing edges, there is a minimal $j > i$ such that $x_j \notin K$. By the induction hypothesis, $x_j \in Q^L_\Theta$. With Theorem 2.10 and $x \in WF_\Sigma$, $x, j \vDash \Theta$. As $x[i, j-1] \in sat(\xi \wedge \neg\psi)^*$, we conclude $x, i \vDash \Theta$, and therefore $x \vDash \Box(q \Rightarrow \Theta)$.

It remains to show that $\mathcal{T}_{\mathsf{U}}$ is correct. For this, we need the next lemma.

Lemma 3.4. *For the systems $\Sigma = (Q, S, \to, v)$, $\Sigma' = (Q', S', \to', v')$ as defined above, the following statements hold:*

1. *If $x' \in path(\Sigma') \cup path(\Sigma')\downarrow$, then $x \in path(\Sigma) \cup path(\Sigma)\downarrow$.*
2. *For each $\alpha \in path(\Sigma)\downarrow$, there is an $\alpha' \in path(\Sigma')\downarrow$ such that $\pi(\alpha') = \alpha$.*
3. *If the last states of $\alpha'_1, \alpha'_2 \in path(\Sigma')\downarrow$, $\alpha'_1, \alpha'_2 \neq \lambda$ do not differ and $\alpha_1 = \alpha_2$, then $\alpha'_1 = \alpha'_2$.*

Proof. Part 1 is an immediate consequence of the definition of $\rightarrow'$ and S'. Part 2 and 3 of the assertion follow from these facts:

If $(s, \Theta_1) \in S'$, then $s \in S$. For each $(q, \Theta_2) \in Q'$ and each $p \in Q$ with $p \rightarrow q$, there is a unique $\Theta_1 \in \{\Theta, \neg\Theta\}$ such that $(p, \Theta_1) \rightarrow' (q, \Theta_2)$. □

Note that it is *not* the case that, for each $(p, \Theta_1) \in Q'$ and $q \in Q$ such that $p \rightarrow q$, there is a $\Theta_2 \in \{\Theta, \neg\Theta\}$ such that $(p, \Theta_1) \rightarrow' (q, \Theta_2)$. Consider, e. g., $(q_1, \neg\Theta)$ and q_4 in Figure 3.2.

We show that $\mathcal{T}_{\mathsf{U}}$ is correct.

Lemma 3.5. *Let $\Sigma = (Q, S, \rightarrow, v)$, $\Sigma' = (Q', S', \rightarrow', v')$, $\Theta = \xi \,\mathsf{U}\, \psi$ as defined above and $\theta = \mathcal{T}_{\mathsf{U}}(\Theta, \Theta)$ the fresh atomic proposition. Then, the following conditions hold:*

1. *The linear-time property $Sat(\Box(\theta \Rightarrow \Diamond\psi))$ is large in Σ'.*
2. *If $x' \vDash \Box(\theta \Rightarrow \Diamond\psi)$, then $x' \vDash \Box(\Theta \Leftrightarrow \theta)$, for each $x' \in path(\Sigma')$.*
3. *The transformation $\mathcal{T}_{\mathsf{U}}$ is correct.*

Proof. A path x' of Σ' that does not satisfy $\Box(\theta \Rightarrow \Diamond\psi)$ belongs to

$$Q'^*[(Q^M_\Theta, \Theta) \cup ((Q^L_\Theta, \Theta) \cap sat(\xi \wedge \neg\psi))]^\omega$$

(cf. Figure 3.5). But such a path is not state fair, as some state satisfying ψ is reachable infinitely often but taken only finitely often. Thus,

$$SF_{\Sigma'} \subseteq Sat(\Box(\theta \Rightarrow \Diamond\psi)),$$

which yields assertion 1.

Let $x' \in path(\Sigma')$ such that $x' \vDash \Box(\theta \Rightarrow \Diamond\psi)$. Suppose $x', i \vDash \theta$. With the block graph of Σ' above, we conclude that

$$x'[i, \infty] \in (Q^M_\Theta, \Theta)^*((Q^L_\Theta, \Theta) \cap sat(\xi \wedge \neg\psi))^* sat(\psi)\uparrow .$$

Because each state in (Q^M_Θ, Θ) satisfies ξ, we conclude $x', i \vDash \Theta$. If $x', i \nvDash \theta$, it can be proved in a similar way that $x', i \nvDash \Theta$. Therefore, $x' \vDash \Box(\Theta \Leftrightarrow \theta)$, which yields 2.

We show 3. with Lemma 3.1. Condition A1 follows immediately from 1. and 2. For A2, let Υ be a path formula with $\theta, q' \not\preceq \Upsilon$, for each $q' \in Q'$. We show that the same player has a winning strategy in $(\overline{AE}, \Sigma, Sat(\Upsilon))$ and a winning strategy in $(\overline{AE}, \Sigma', Sat(\Upsilon))$. Suppose that $P \in \{E, A\}$ and f is a winning strategy for P in $(\overline{AE}, \Sigma, Sat(\Upsilon))$. We construct a strategy f' in Σ' such that $\pi(f'(\lambda)) = f(\lambda)$ and, for each $\alpha' \in path(\Sigma')\downarrow$ with $\alpha' \neq \lambda$, we have $f'(\alpha') = \alpha'\beta'\gamma'$ with $f(\alpha\beta) = \alpha\beta\gamma$. Hence, $\pi(Z^{P,f'}_{\overline{AE}}) \subseteq Z^{P,f}_{\overline{AE}}$. As $\theta, q' \not\preceq \Upsilon$, for each $q' \in Q'$, we conclude that f' is winning for P in $(\overline{AE}, \Sigma', Sat(\Upsilon))$.

We define $f'(\lambda)$ using Lemma 3.4. Let $\alpha' \in path(\Sigma')\downarrow$, $\alpha' \neq \lambda$. If α' ends in (Q^M_Θ, Θ), we choose β' such that $\alpha'\beta' \in path(\Sigma')\downarrow$ and $\alpha'\beta'$ ends in (Q^L_Θ, Θ). If α' ends in $(Q^M_\Theta, \neg\Theta)$, we choose β' such that $\alpha'\beta' \in path(\Sigma')\downarrow$ and $\alpha'\beta'$ ends in $(Q^S_\Theta, \neg\Theta)$. We define γ with $f(\alpha\beta) =: \alpha\beta\gamma$. Now, we choose, using Lemma 3.4, $\tilde{\alpha}'$, $\tilde{\beta}', \gamma'$ such that $\pi(\tilde{\alpha}'\tilde{\beta}') = \alpha\beta$, $\pi(\gamma') = \gamma$ and $\tilde{\alpha}'\tilde{\beta}'\gamma' \in path(\Sigma')\downarrow$. As the last state of $\alpha\beta$ does not belong to Q^M_Θ, we conclude with 3. of Lemma 3.4 that $\tilde{\alpha}'\tilde{\beta}' = \alpha'\beta'$. Hence, $\alpha'\beta'\gamma' \in path(\Sigma')\downarrow$. We set $f'(\alpha') := \alpha'\beta'\gamma'$. □

3.3. The Transformation $\mathcal{T}_{\mathsf{X}}$

For the transformation $\mathcal{T}_{\mathsf{X}}$, each formula $X\xi$ is feasible, where ξ is a state formula. Let $\Sigma = (Q, S, \rightarrow, v)$ be a reduced system, Φ a specification, $\Theta = \mathsf{X}\xi$ a feasible formula, $\Sigma' := (Q', S', \rightarrow', v') := \mathcal{T}_{\mathsf{X}}(\Sigma, \Theta)$ and $\Phi' := \mathcal{T}_{\mathsf{X}}(\Phi, \Theta)$. According to Section 3.1, it suffices to define $\rightarrow'$ and S'.

The new state relation $\rightarrow'$ is the smallest relation such that the following statements X1 and X2 hold. Let $p, q \in Q$ with $p \rightarrow q$ and $\Theta_2 \in \{\Theta, \neg\Theta\}$ such that $(q, \Theta_2) \in Q'$.

X1: If $q \vDash \xi$, then $(p, \Theta) \rightarrow' (q, \Theta_2)$. (In this case, $p \notin Q^S_\Theta$.)

X2: If $q \nvDash \xi$, then $(p, \neg\Theta) \rightarrow' (q, \Theta_2)$. (In this case, $p \notin Q^L_\Theta$.)

The new set of starting states is $S' := Q' \cap (S, \{\Theta, \neg\Theta\})$. Note that $\mathcal{T}_{\mathsf{X}}$ does not produce unreachable states. Hence, T2 is satisfied.

The computation of $\mathcal{T}_{\mathsf{X}}$ is as follows. First, the algorithm evaluates ξ at each state of Q. For the partition of Q we use these facts:

- A state q of Σ belongs to Q^L_Θ iff each successor of q satisfies ξ.
- A state q of Σ belongs to Q^S_Θ iff no successor of q satisfies ξ.
- Otherwise, if a state q of Σ does not belong to $Q^L_\Theta \cup Q^S_\Theta$, we have $q \in Q^M_\Theta$.

After that, the construction of Σ' and Φ' is straightforward.

Using the arguments of the proof of Lemma 3.4, we show:

Lemma 3.6. *For the systems Σ, Σ' as defined above, the following statements hold:*

1. *If $x' \in path(\Sigma') \cup path(\Sigma')\downarrow$, then $x \in path(\Sigma) \cup path(\Sigma)\downarrow$.*
2. *For each $\alpha \in path(\Sigma)\downarrow$, there is an $\alpha' \in path(\Sigma')\downarrow$ such that $\pi(\alpha') = \alpha$.*
3. *If the last states of $\alpha'_1, \alpha'_2 \in path(\Sigma')\downarrow$, $\alpha'_1, \alpha'_2 \neq \lambda$ do not differ and $\alpha_1 = \alpha_2$, then $\alpha'_1 = \alpha'_2$.*

Now, we are ready to prove that $\mathcal{T}_{\mathsf{X}}$ is correct.

Lemma 3.7. *Let $\Sigma = (Q, S, \rightarrow, v)$, $\Sigma' = (Q', S', \rightarrow', v')$, $\Theta = \mathsf{X}\xi$ as defined above and $\theta = \mathcal{T}_{\mathsf{X}}(\Theta, \Theta)$ the new atomic proposition. Then, the following conditions hold:*

1. *Each path of Σ' satisfies $\Box(\Theta \Leftrightarrow \theta)$.*
2. *The transformation $\mathcal{T}_{\mathsf{X}}$ is correct.*

Proof. By the construction of $\rightarrow'$, each successor of a state in (Q, Θ) satisfies ξ, and each successor of a state in $(Q, \neg\Theta)$ does not satisfy ξ. Thus, each path of Σ' satisfies $\Box(\Theta \Leftrightarrow \theta)$.

We show 2. using Lemma 3.1. Condition A1 follows from 1. For A2, let Υ be a path formula with $\theta, q' \not\preceq \Upsilon$, for each $q' \in Q'$. We show that the same player has a winning strategy in $(\overline{AE}, \Sigma, \mathit{Sat}(\Upsilon))$ and a winning strategy in $(\overline{AE}, \Sigma', \mathit{Sat}(\Upsilon))$. Suppose that $P \in \{E, A\}$ and f is a winning strategy for P in $(\overline{AE}, \Sigma, \mathit{Sat}(\Upsilon))$. We construct a strategy f' in Σ' such that $\pi(f'(\lambda)) = f(\lambda)$ and, for each $\alpha' \in \mathit{path}(\Sigma')\downarrow$ with $\alpha' \neq \lambda$, we have $f'(\alpha') = \alpha' q' \gamma'$ with $f(\alpha q) = \alpha q \gamma$. Hence, $\pi(Z^{P,f'}_{\overline{AE}}) \subseteq Z^{P,f}_{\overline{AE}}$. As $\theta, q' \not\preceq \Upsilon$, for each $q' \in Q'$, we conclude that f' is winning for P in $(\overline{AE}, \Sigma', \mathit{Sat}(\Upsilon))$.

We define $f'(\lambda)$ using Lemma 3.6. For $\alpha' \in \mathit{path}(\Sigma')\downarrow$, $\alpha' \neq \lambda$, let $q \in Q$ be the projection of some successor of α' in Σ'; i. e. $\alpha'(q, \Theta_1) \in \mathit{path}(\Sigma')\downarrow$, for some $\Theta_1 \in \{\Theta, \neg\Theta\}$. Note that each state of Σ' with projection q is a successor of α'. We define γ with $f(\alpha q) =: \alpha q \gamma$. Now, we choose, using Lemma 3.6, $\tilde{\alpha}'$, q', γ' such that $\pi(\tilde{\alpha}' q') = \alpha q$, $\pi(\gamma') = \gamma$ and $\tilde{\alpha}' q' \gamma' \in \mathit{path}(\Sigma')\downarrow$. As $\tilde{\alpha}' q' \in \mathit{path}(\Sigma')\downarrow$, we conclude with 3. of Lemma 3.6 that $\alpha' q' = \tilde{\alpha}' q'$. Hence, $\alpha' q' \gamma' \in \mathit{path}(\Sigma')\downarrow$, too. We define $f'(\alpha') := \alpha' q' \gamma'$. □

3.4. The Transformation $\mathcal{T}_{\mathsf{S}}$

For the transformation $\mathcal{T}_{\mathsf{S}}$, each formula $\xi \ \mathsf{S} \ \psi$ is feasible, where ξ, ψ are state formulae. Let $\Sigma = (Q, S, \rightarrow, v)$ be a reduced system, Φ a specification, $\Theta = \xi \ \mathsf{S} \ \psi$ a feasible formula, $\Sigma' := (Q', S', \rightarrow', v') := \mathcal{T}_{\mathsf{S}}(\Sigma, \Theta)$ and $\Phi' := \mathcal{T}_{\mathsf{S}}(\Phi, \Theta)$. According to Section 3.1, it suffices to define $\rightarrow'$ and S'.

The new state relation $\rightarrow'$ is the smallest relation which satisfies the conditions S1 and S2 below. Let $p, q \in Q$ with $p \rightarrow q$ and $\Theta_1, \Theta_2 \in \{\Theta, \neg\Theta\}$ such that $(p, \Theta_1), (q, \Theta_2) \in Q'$.

S1: If $q \notin Q^M_\Theta$, then $(p, \Theta_1) \rightarrow' (q, \Theta_2)$.

S2: If $q \in Q^M_\Theta$, then $(p, \Theta_1) \rightarrow' (q, \Theta_1)$.

The new set of starting states S' is the smallest set such that $(s, \neg\Theta) \in S'$ if $s \in S \backslash Q^L_\Theta$ and $(s, \Theta) \in S'$ if $s \in S \cap Q^L_\Theta$. Note that $S \cap Q^L_\Theta \subseteq \mathit{sat}(\psi)$. In Lemma 3.9 we show that

Σ' is reduced (T2). (The steps before are independent of whether $\mathcal{T}_{\mathsf{S}}$ is a well-defined transformation.)

For the following steps, a modified version of the block graphs introduced in Section 3.2 is useful. A block graph of Σ consists of the *blocks* $R_1, \ldots, R_m \subseteq Q$ and the relation $\dashrightarrow$ over these blocks. The blocks $R_1, \ldots, R_m$ are drawn as areas, and the relation $\dashrightarrow$ is visualised by arrows between the corresponding areas. The area for R_i encloses the area for R_j, where $1 \leq i, j \leq m$, iff $R_j \subseteq R_i$. The meaning of $\dashrightarrow$ is as follows. If $p \rightarrow q$ for some $p, q \in Q$, then there exist i, j, where $1 \leq i, j \leq m$, such that $p \in R_i$, $q \in R_j$, and $R_i = R_j$ or $R_i \dashrightarrow R_j$.

It is easy to verify the block graph of Σ as described in Figure 3.6. From Figure

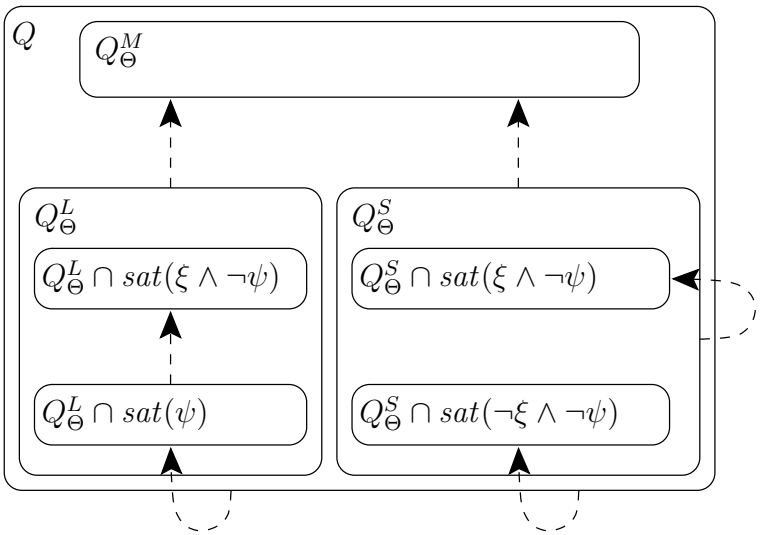

Figure 3.6.: A block graph of Σ, where $\Theta = \xi \mathsf{S} \psi$.

3.6, the block graph of Σ' in Figure 3.7 can be derived.

The transformation $\mathcal{T}_{\mathsf{S}}$ is computed as follows. First, the algorithm evaluates ξ and ψ at each state of Σ. Clearly, a state which satisfies ψ belongs to Q^L_Θ, and a state which satisfies $\neg\xi \wedge \neg\psi$ belongs to Q^S_Θ. The algorithm computes the s. c. c.s of $sat(\xi \wedge \neg\psi)$ and visits them in topological order. That is, if the s. c. c. K_1 has an outgoing edge to the s. c. c. K_2, then K_1 is visited first. The states of Σ are classified according to the following rules:

Let K be an s. c. c. of $sat(\xi \wedge \neg\psi)$.

- The states of K are added to Q^L_Θ if no starting state belongs to K and all incoming edges start in Q^L_Θ.
- The states of K are added to Q^S_Θ if all incoming edges start in Q^S_Θ.

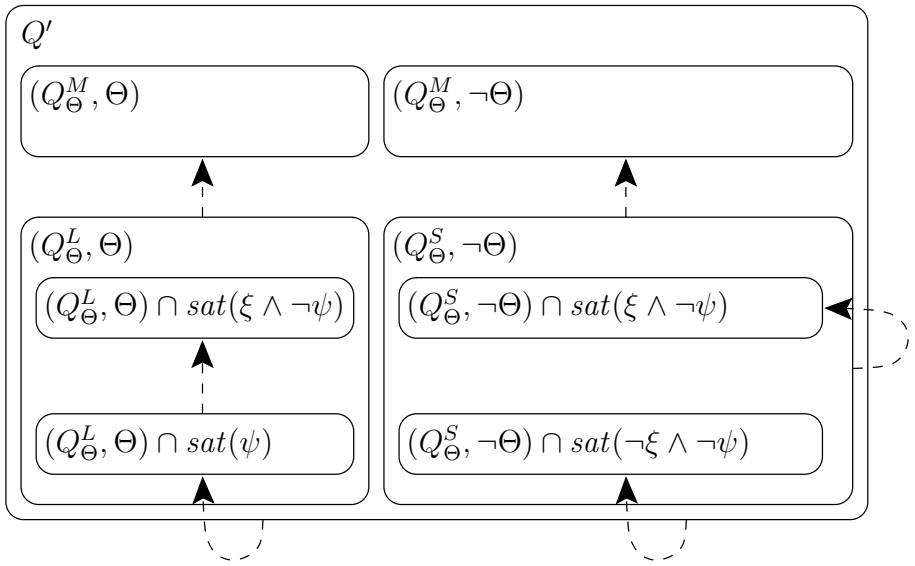

Figure 3.7.: A block graph of Σ', where $\Theta = \xi \,\mathsf{S}\, \psi$.

- Otherwise, if the states of K are neither added to Q^L_Θ nor to Q^S_Θ, they are added to Q^M_Θ.

We show by induction over the number of b. s. c. c.s that have already been processed that the classification is correct for those b. s. c. c.s. Let K be the current b. s. c. c., and suppose the algorithm decides $K \subseteq Q^L_\Theta$. (The remaining cases can be treated in a similar way.)

For $q \in K$, we show $WF_\Sigma \subseteq Sat(\Box(q \Rightarrow \Theta))$. Thus, $Sat(\Box(q \Rightarrow \Theta))$ is large, and q has been classified correctly. Let $x \in WF_\Sigma$, and suppose $x_i = q$. As the algorithm has decided $K \subseteq Q^L_\Theta$, K does not contain any starting state. Hence, there is a maximal j with $0 \leq j < i$ such that $x_j \notin K$. By the induction hypothesis, $x_j \in Q^L_\Theta$. With Theorem 2.10 and $x \in WF_\Sigma$, $x, j \vDash \Theta$. As $x[j+1, i] \in sat(\xi \wedge \neg\psi)^*$, we conclude $x, i \vDash \Theta$, and therefore $x \vDash \Box(q \Rightarrow \Theta)$.

It remains to show that $\mathcal{T}_{\mathsf{S}}$ is correct. For this, we need the next lemma.

Lemma 3.8. *For $\Sigma = (Q, S, \to, v)$, $\Sigma' = (Q', S', \to', v')$ as defined above, the following statements hold:*

- *If $x' \in path(\Sigma') \cup path(\Sigma')\downarrow$, then $x \in path(\Sigma) \cup path(\Sigma)\downarrow$.*
- *For each $x \in path(\Sigma) \cup path(\Sigma)\downarrow$, there is a unique $x' \in path(\Sigma') \cup path(\Sigma')\downarrow$ such that $\pi(x') = x$.*

Proof. Part 1 of the assertion immediately follows from the construction of $\rightarrow'$ and S'. Part 2 follows from these facts:

For each $s \in S$, there is a unique $\Theta_1 \in \{\Theta, \neg\Theta\}$ such that $(s, \Theta_1) \in S'$. For each $(p, \Theta_1) \in Q'$ and $q \in Q$ such that $p \rightarrow q$, there is a unique $\Theta_2 \in \{\Theta, \neg\Theta\}$ such that $(p, \Theta_1) \rightarrow' (q, \Theta_2)$. □

We show that $\mathcal{T}_{\mathsf{S}}$ is correct.

Lemma 3.9. *Let $\Sigma = (Q, S, \rightarrow, v)$, $\Sigma' = (Q', S', \rightarrow', v')$, $\Theta = \xi \,\mathsf{S}\, \psi$ as defined above and $\theta = \mathcal{T}_{\mathsf{S}}(\Theta, \Theta)$ the fresh atomic proposition. Then, the following conditions hold:*

1. *Each path of Σ' satisfies $\Box(\Theta \Leftrightarrow \theta)$.*
2. *Condition T2 holds; i. e. Σ' is reduced.*
3. *$\mathcal{T}_{\mathsf{S}}$ is correct.*

Proof. We show 1. Let $x' \in path(\Sigma')$, and suppose $x', i \vDash \theta$. From the block graph of Σ', we know that

$$x'[0, i] \in Q'^* sat(\psi)((Q^L_\Theta, \Theta) \cap sat(\xi \wedge \neg\psi))^*(Q^M_\Theta, \Theta)^*.$$

(Note that there is no starting state in $(Q^L_\Theta, \Theta) \cap sat(\xi \wedge \neg\psi)$ or (Q^M_Θ, Θ).) As each state in (Q^M_Θ, Θ) satisfies ξ, we have $x', i \vDash \Theta$. If $x', i \nvDash \theta$, it can be proved in a similar way that $x', i \nvDash \Theta$. Thus, $x' \vDash \Box(\Theta \Leftrightarrow \theta)$.

We show 2. Let $(q, \Theta_1) \in Q'$. Due to the definition of Q^L_Θ, Q^M_Θ and Q^S_Θ, there is a path x of Σ and some position i such that $x, i \vDash q \wedge \Theta_1$. We define, with Lemma 3.8, x' as the unique path of Σ' with $\pi(x') = x$. As, for each $q' \in Q'$, $\theta, q' \not\leq q \wedge \Theta_1$, we have $x', i \vDash q \wedge \Theta_1$. From 1., we know that $x' \vDash \Box(\Theta \Leftrightarrow \theta)$. Thus, $x'_i = (q, \Theta_1)$. Because (q, Θ_1) is arbitrary chosen, we conclude that T2 holds.

We prove 3. with Lemma 3.1. Condition A1 is a direct consequence of 1. For A2, let Υ be a path formula with $\theta, q' \not\leq \Upsilon$, for each $q' \in Q'$. We show that the same player has a winning strategy in $(\overline{AE}, \Sigma, Sat(\Upsilon))$ and a winning strategy in $(\overline{AE}, \Sigma', Sat(\Upsilon))$. Suppose that $P \in \{E, A\}$ and f is a winning strategy for P in $(\overline{AE}, \Sigma, Sat(\Upsilon))$. Given $\alpha' \in path(\Sigma')\downarrow$, we define β with $f(\alpha) =: \alpha\beta$. Due to Lemma 3.8, there is a β' such that $\pi(\beta') = \beta$ and $\alpha'\beta' \in path(\Sigma')\downarrow$. We set $f'(\alpha') := \alpha'\beta'$. Thus, $\pi(Z^{P,f'}_{\overline{AE}}) \subseteq Z^{P,f}_{\overline{AE}}$. As $\theta, q' \not\leq \Upsilon$, for each $q' \in Q'$, we conclude that f' is winning for P in $(\overline{AE}, \Sigma', Sat(\Upsilon))$. □

3.5. The Transformation $\mathcal{T}_{\mathsf{Y}}$

For the transformation $\mathcal{T}_{\mathsf{Y}}$, each formula $\mathsf{Y}\,\xi$ is feasible, where ξ is a state formula. Let $\Sigma = (Q, S, \rightarrow, v)$ be a reduced system, Φ a specification, $\Theta = \mathsf{Y}\,\xi$ a feasible

formula, $\Sigma' := (Q', S', \to', v') := \mathcal{T}_{\mathsf{Y}}(\Sigma, \Theta)$ and $\Phi' := \mathcal{T}_{\mathsf{Y}}(\Phi, \Theta)$. According to Section 3.1, it suffices to define $\to'$ and S'.

The new state relation $\to'$ is the smallest relation which satisfies the conditions Y1 and Y2 below. Let $p, q \in Q$ with $p \to q$ and $\Theta_1 \in \{\Theta, \neg\Theta\}$ such that $(p, \Theta_1) \in Q'$.

Y1: If $p \vDash \xi$, then $(p, \Theta_1) \to' (q, \Theta)$. (In this case, $q \notin Q^S_\Theta$.)

Y2: If $p \nvDash \xi$, then $(p, \Theta_1) \to' (q, \neg\Theta)$. (In this case, $q \notin Q^L_\Theta$.)

The new set of starting states is $S' := Q' \cap (S, \neg\Theta)$. As $S \cap Q^L_\Theta = \emptyset$, we have $S' \neq \emptyset$. In Lemma 3.11 we see that Σ' is reduced (T2). (The steps before do not depend on whether T2 is satisfied.)

The computation of $\mathcal{T}_{\mathsf{Y}}$ is as follows. First, the algorithm evaluates ξ at each state of Q. For the partition of Q, we use the following facts:

- The state $q \in Q$ belongs to Q^L_Θ iff $q \notin S$ and each predecessor of q satisfies ξ.
- The state $q \in Q$ belongs to Q^S_Θ iff no predecessor of q satisfies ξ.
- Otherwise, if $q \in Q \setminus (Q^L_\Theta \cup Q^S_\Theta)$, we have $q \in Q^M_\Theta$.

After that, the construction of Σ' and Φ' is straightforward.

Using the arguments of the proof of Lemma 3.8, we show:

Lemma 3.10. *For Σ, Σ' as defined above, the following statements hold:*

- *If $x' \in path(\Sigma') \cup path(\Sigma')\downarrow$, then $x \in path(\Sigma) \cup path(\Sigma)\downarrow$.*
- *For each $x \in path(\Sigma) \cup path(\Sigma)\downarrow$, there is a unique $x' \in path(\Sigma') \cup path(\Sigma')\downarrow$ such that $\pi(x') = x$.*

Now, we are ready to show that $\mathcal{T}_{\mathsf{Y}}$ is correct.

Lemma 3.11. *Let $\Sigma = (Q, S, \to, v)$, $\Sigma' = (Q', S', \to', v')$, $\Theta = \mathsf{Y}\,\xi$ as defined above and $\theta = \mathcal{T}_{\mathsf{Y}}(\Theta, \Theta)$ the fresh atomic proposition. Then, the following conditions hold:*

1. *Each path of Σ' satisfies $\Box(\Theta \Leftrightarrow \theta)$.*
2. *Condition T2 holds; i. e. Σ' is reduced.*
3. *The transformation $\mathcal{T}_{\mathsf{Y}}$ is correct.*

Proof. By the construction of $\to'$, if $q' \in (Q, \Theta)$, then each predecessor of q' satisfies ξ, and $q' \notin S'$. Moreover, each predecessor of a state in $(Q, \neg\Theta)$ does not satisfy ξ. Thus, each path of Σ' satisfies $\Box(\Theta \Leftrightarrow \theta)$.

Part 2 of the assertion can be shown exactly as in the proof of Lemma 3.9.

We prove that $\mathcal{T}_{\mathsf{Y}}$ is correct using Lemma 3.1. Condition A1 follows from 1. For A2, let Υ be a path formula with $\theta, q' \not\preceq \Upsilon$, for each $q' \in Q'$. We show that the same player has a winning strategy in $(\overline{AE}, \Sigma, \mathit{Sat}(\Upsilon))$ and a winning strategy in $(\overline{AE}, \Sigma', \mathit{Sat}(\Upsilon))$. Suppose that $P \in \{E, A\}$ and f is a winning strategy for P in $(\overline{AE}, \Sigma, \mathit{Sat}(\Upsilon))$. Given $\alpha' \in \mathit{path}(\Sigma')\downarrow$, we define β with $f(\alpha) =: \alpha\beta$. Due to Lemma 3.10, there is a β' such that $\pi(\beta') = \beta$ and $\alpha'\beta' \in \mathit{path}(\Sigma')\downarrow$. We set $f'(\alpha') := \alpha'\beta'$. Thus, $\pi(Z^{P,f'}_{\overline{AE}}) \subseteq Z^{P,f}_{\overline{AE}}$. As $\theta, q' \not\preceq \Upsilon$, for each $q' \in Q'$, we conclude that f' is winning for P in $(\overline{AE}, \Sigma', \mathit{Sat}(\Upsilon))$. □

3.6. The Transformation $\mathcal{T}_{\overline{L(\Box\Diamond)}}$

3.6.1. Motivation

Before we define the transformation $\mathcal{T}_{\overline{L(\Box\Diamond)}}$, we explain with an example why such a transformation is useful. Consider a system $\Sigma_0 = (Q_0, S_0, \rightarrow_0, v_0)$ as described in Figure 3.8 with the specification $\Phi_0 := \bigvee_{i=1}^{m} \Box\Diamond\zeta_i$. Note that $\mathit{Sat}(\Phi_0)$ is small in Σ_0. The algorithm described so far repeatedly applies the transformation $\mathcal{T}_{\mathsf{U}}$, where the feasible path formula in the i-th transformation step is $\Theta_i := \Diamond\zeta_i$, for $1 \leq i \leq m$, and $\Theta_i := \Box\theta_i$, for $m < i \leq 2m$. The fresh atomic propositions are denoted $\theta_i := \mathcal{T}_{\mathsf{U}}(\Theta_i, \Theta_i)$, where $1 \leq i \leq 2m$. Finally, we have the system Σ_{2m} as described in Figure 3.9 and 3.10 and the specification $\Phi_{2m} = \bigvee_{i=m+1}^{2m} \theta_i$. Unfortunately, Σ_{2m} is exponentially larger (in the size of Φ_0) than Σ_0.

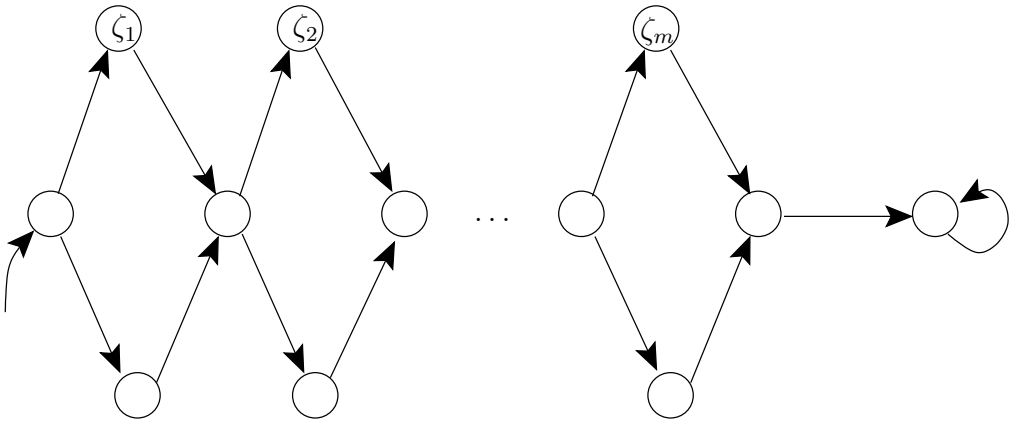

Figure 3.8.: A system Σ_0. We omit state names and related atomic propositions.

To bypass this, we develop a transformation $\mathcal{T}_{\overline{L(\Box\Diamond)}}$ for which $\Theta := \Phi_0$ is feasible. Looking at Σ_0, we conclude that $(Q_0)^S_\Theta = Q_0$. Due to Theorem 3.3, we do not have

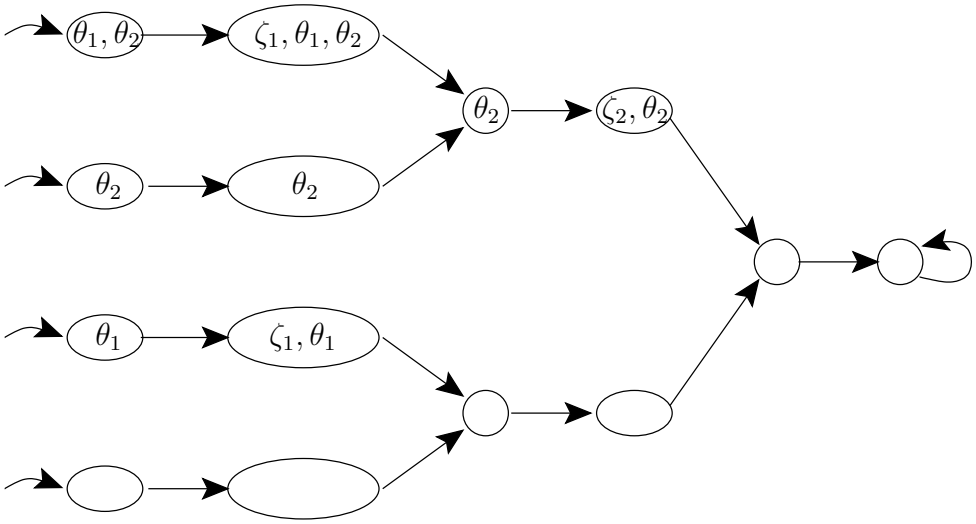

Figure 3.9.: The system Σ_4, in the case $m = 2$. We omit state names and related atomic propositions.

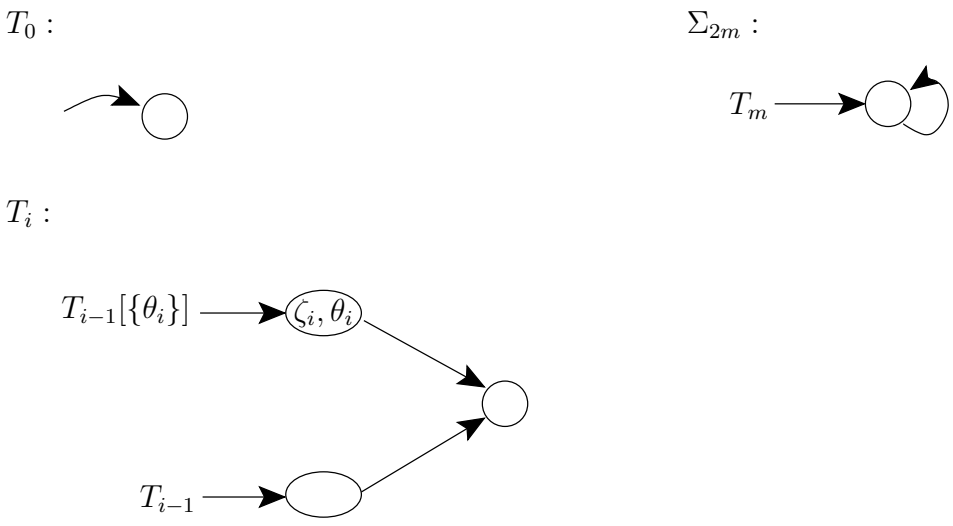

Figure 3.10.: The system Σ_{2m}. For ease of representation, we describe Σ_{2m} recursively. For a labelled graph G, $G[\chi]$ denotes the graph obtained by adding the elements of χ as labels to each node of G. We omit state names and related atomic propositions.

to duplicate any state of Σ_0. We define $\mathcal{T}_{\overline{L}(\Box\Diamond)}(\Sigma_0, \Theta) := \Sigma_1 := (Q_1, S_1, \rightarrow_1, v_1)$ such that the following conditions are satisfied:

- $Q_1 = (Q, \neg\Theta)$,
- $S_1 = (S, \neg\Theta)$,
- $(p, \neg\Theta) \rightarrow_1 (q, \neg\Theta)$ iff $p \rightarrow_0 q$, for $p, q \in Q_0$, and
- v_1 is chosen according to T3 of Section 3.1.

Note that T1-T3 of Section 3.1 hold. The new formula is $\Phi_1 := \theta$, where θ is a fresh atomic proposition. Then $Sat(\Phi_1)$ is small in Σ_1, as desired. But, as Σ_1 and Σ_0 are isomorphic, we avoid the exponential blowup. The problem is even solved in only one transformation step.

3.6.2. Muller Formulae

In this section, we define the language $\overline{L}(\Box\Diamond)$ and give a semantical characterisation of the formulae in $\overline{L}(\Box\Diamond)$. More precisely, we show that whether a path satisfies a formula in $\overline{L}(\Box\Diamond)$ depends only on the states (modulo a valuation function) that are taken infinitely often by that path. Moreover, if the membership of a path to a linear-time property depends only on which states (modulo the valuation function) are taken infinitely often by that path, then this linear-time property can be expressed as a formula in $\overline{L}(\Box\Diamond)$. Thus, in analogy to Muller acceptance conditions of ω-automata (see Thomas [1990]), we call a formula in $\overline{L}(\Box\Diamond)$ a *Muller formula*. Finally, we show related semantical properties of the formulae in $L(\Box)$.

Definition 3.12. The language $\overline{L}(\Box\Diamond)$ of *Muller formulae* is the smallest set that satisfies the following conditions M1 and M2:

M1: If $\Xi \in L(\Box\Diamond)$, then $\Box\Diamond\,\Xi \in \overline{L}(\Box\Diamond)$.

M2: If $\Xi, \Psi \in \overline{L}(\Box\Diamond)$, then $\Xi \vee \Psi,\ \neg\Xi \in \overline{L}(\Box\Diamond)$.

Suppose ϕ, ψ are state formulae. Examples for Muller formulae are $\Box\Diamond\,\phi \Rightarrow \Box\Diamond\,\psi$, $\Diamond\Box\,\phi \Rightarrow \Box\Diamond\,\psi$ and $\Box\Diamond(\phi \wedge \Box\Diamond\,\psi) \vee \Diamond\Box\,\phi$. Among the Muller formulae are the *Street constraints*; see, e. g., Alur and Henzinger [1995], or Varacca and Völzer [2006]. Each member of $L(\mathsf{X}, \mathsf{Y}, \mathsf{U}, \mathsf{S}) \setminus L(\Box\Diamond)$ is obviously not a Muller formula. Furthermore, $\phi \Rightarrow \Box\Diamond\,\psi$ is not a Muller formula, too.

Before we state the semantical characterisation of Muller formulae, we need to introduce some notations. Let $\Sigma = (Q, S, \rightarrow, v)$ be a system, x, y paths over Q and Φ a path formula. We define some alternative notions of satisfaction:

- We write $x \vDash^{i.o.} \Phi$ (x *satisfies* Φ *infinitely often*) iff, for infinitely many $i \in \mathbb{N}$, $x, i \vDash \Phi$.
- Similarly, we write $x \vDash^{a.e.} \Phi$ (x *satisfies* Φ *almost everywhere*) iff, for all but finitely many $i \in \mathbb{N}$, $x, i \vDash \Phi$.

Recall that

$$inf(x) = \{q \in Q \mid x \overset{i.o.}{\vDash} q\}.$$

Suppose that $x = s(pq)^\omega$, where s, p, q are pairwise different states of Σ. Then $inf(x) = \{p, q\}$. Note that $x \vDash^{i.o.} p$; however, it is not the case that $x \vDash^{a.e.} p$, as $x \vDash^{i.o.} q$.

We formalise a notion of equivalence:

- We write $x \sim_\Phi y$ (x, y are *indistinguishable by* Φ) iff $x \vDash \Phi \Leftrightarrow y \vDash \Phi$.
- For a linear-time property X over Q, we write $x \sim_X y$ (x, y are *indistinguishable by* X) iff $x \in X \Leftrightarrow y \in X$.
- For a set Ω of path formulae, we write $x \sim_\Omega y$ (x, y are *indistinguishable by* Ω) iff $x \sim_\Phi y$, for each $\Phi \in \Omega$.

For some state $q \in Q$, we call $v(q) \subseteq AP$ an *assignment*. We need assignments in our presentation, as it does not directly depend on x whether x satisfies a path formula but, on $v(x)$, the related sequence of assignments.

Suppose that r, s, p, q are pairwise different states of Σ. Then,

$$sr(pq)^\omega \sim_{\Box\Diamond p \wedge \Box\Diamond q} rs(pqp)^\omega.$$

In Proposition 3.14 below, we even show that

$$sr(pq)^\omega \sim_{\overline{L}(\Box\Diamond)} rs(pqp)^\omega;$$

however,

$$sr(pq)^\omega \nsim_{L(\Box)} rs(pqp)^\omega, \text{ as } \quad sr(pq)^\omega \nsim_\phi rs(pqp)^\omega, \text{ for } \phi = s.$$

Definition 3.13. Let $\Sigma = (Q, S, \to, v)$ be a system. A linear-time property X over Q is a *Muller property* iff, for all paths x, y over Q with $v(inf(x)) = v(inf(y))$, we have $x \sim_X y$.

Now, we give the semantical characterisation of Muller formulae.

Proposition 3.14. *Let $\Sigma = (Q, S, \to, v)$ be a system. Then, the following conditions hold:*

1. *Let x, y be paths over Q. Then, $x \sim_{\overline{L}(\Box\Diamond)} y$ iff $v(inf(x)) = v(inf(y))$.*

2. *We have*

$$\{Sat(\Phi) \mid \Phi \text{ is a Muller formula}\} = \{X \mid X \text{ is a Muller property}\}.$$

Proof. We prove 1.

Suppose $v(inf(x)) = v(inf(y))$. First, we show by structural induction over Φ that

$$x \overset{i.o.}{\vDash} \Phi \quad \Leftrightarrow \quad y \overset{i.o.}{\vDash} \Phi \quad \text{and} \tag{3.1}$$

$$x \overset{a.e.}{\vDash} \Phi \quad \Leftrightarrow \quad y \overset{a.e.}{\vDash} \Phi, \tag{3.2}$$

for each $\Phi \in L(\Box\Diamond)$. Furthermore, we show by structural induction over Φ that

$$x \vDash \Phi \Leftrightarrow y \vDash \Phi, \tag{3.3}$$

for each $\Phi \in \overline{L}(\Box\Diamond)$. Therefore, $x \sim_{\overline{L}(\Box\Diamond)} y$. Note that, in each case, it suffices to show the "only if" part. The other direction follows from symmetry arguments.

Let Φ be a state formula. If we assume that the left hand side of (3.1) (resp. (3.2)) is true, then, as Q is finite, $inf(x)$ contains some state satisfying Φ (resp. all states in $inf(x)$ satisfy Φ). Hence, with $v(inf(x)) = v(inf(y))$, the right hand side of (3.1) (resp. (3.2)) holds.

Let $\Xi, \Psi \in L(\Box\Diamond)$ and $\Phi = \Xi \vee \Psi$, and suppose the left hand side of (3.1) is true. Thus, we have $x \vDash^{i.o.} \Xi$ or $x \vDash^{i.o.} \Psi$. By the induction hypothesis, this implies $y \vDash^{i.o.} \Xi$ or $y \vDash^{i.o.} \Psi$. Hence the right hand side of (3.1) is true. Equation (3.2) can be shown in a similar way.

Let $\Xi \in L(\Box\Diamond)$ and $\Phi = \neg\Xi$, and suppose the left hand side of (3.1) is satisfied. Then it is not the case that $x \vDash^{a.e.} \Xi$. By the induction hypothesis, $y \vDash^{a.e.} \Xi$ does not hold. Therefore, the right hand side of (3.1) is satisfied. Again, the proof of (3.2) is similar.

Let $\Xi \in L(\Box\Diamond)$ and $\Phi = \Box\Diamond\Xi$. If we assume that the left hand side of (3.3) is true, then $x \vDash^{i.o.} \Xi$. Therefore, by the induction hypothesis, $y \vDash^{i.o.} \Xi$. Hence, the right hand side of (3.3) is true.

Note that Φ is either satisfied at each position of x or, at none. As (3.3) is true, (3.1) and (3.2) hold, too.

The cases $\Phi = \Xi \vee \Psi$ and $\Phi = \neg\Xi$, where $\Xi, \Psi \in \overline{L}(\Box\Diamond)$, can be treated analogously to the related cases, where $\Xi, \Psi \in L(\Box\Diamond)$.

Now, suppose $v(inf(x)) \neq v(inf(y))$. First, note that, for an assignment $\varphi \in v(Q)$, there is a state formula χ_φ that is satisfied by a state $q \in Q$ iff $v(q) = \varphi$. Without loss of generality, let $\varphi \in v(inf(x))$ such that $\varphi \notin v(inf(y))$. Then $x \vDash \Box\Diamond\chi_\varphi$ and $y \nvDash \Box\Diamond\chi_\varphi$. Thus, $x \nsim_{\overline{L}(\Box\Diamond)} y$.

We prove 2. As a consequence of 1.,

$$\{Sat(\Phi) \mid \Phi \text{ is a Muller formula}\} \subseteq \{X \mid X \text{ is a Muller property}\}.$$

The converse can be shown as follows:

Let X be a Muller property. For a set of assignments $\tau \subseteq v(Q)$, we define a Muller formula

$$\Phi_\tau := \bigwedge_{\varphi \in \tau} \Box\Diamond \chi_\varphi \wedge \bigwedge_{\varphi \in v(Q) \setminus \tau} \neg \Box\Diamond \chi_\varphi.$$

A path x over Q satisfies Φ_τ iff $v(inf(x)) = \tau$. As X is a Muller property, we can choose $\tau_1, \ldots, \tau_k \subseteq v(Q)$ such that a path x over Q belongs to X iff $v(inf(x)) = \tau_i$, for some $i \in \{1, \ldots, k\}$. Then, for $\Phi := \bigvee_{i=1}^{k} \Phi_{\tau_i} \in \overline{L}(\Box\Diamond)$, we have $Sat(\Phi) = X$. To see this, consider some path x over Q. By assumption, $x \in X$ iff $v(inf(x)) = \tau_i$, for some $i \in \{1, \ldots, k\}$. By construction of the Φ_{τ_i}, the latter is equivalent to $x \vDash \Phi_{\tau_i}$, for some $i \in \{1, \ldots, k\}$. This statement can be rewritten to $x \in Sat(\Phi)$. $\Box$

As an immediate consequence, Muller formulae are either satisfied at each point of a sequence of assignments, or, at none. Moreover, Proposition 3.14 says that, for each Muller formula Φ, there is a normal form; that is the Muller formula Φ_N defined in the proof such that $Sat(\Phi_N) = X$, where $X := Sat(\Phi)$.

Proposition 3.14 can be seen as a syntactical characterisation of $\sim_{\overline{L}(\Box\Diamond)}$. Now, we give a syntactical characterisation of $\sim_{L(\Box)}$. For this, we define

$$R(T) = \{x \in T^\omega \mid inf(x) = T\},$$

where $T \subseteq Q$.

Proposition 3.15. *Let $\Sigma = (Q, S, \rightarrow, v)$ be a system and x, y be paths over Q. Then $x \sim_{L(\Box)} y$ iff*

1. *$v(x_0) = v(y_0)$,*
2. *$v(inf(x)) = v(inf(y))$ and*
3. *there are $n \in \mathbb{N}$, $\alpha \in Q^n$ such that*

$$v(x), v(y) \in \{v(\alpha_0)\}^+ \ldots \{v(\alpha_{n-1})\}^+ v(R(inf(x))).$$

According to Proposition 3.15,

- $sr(pq)^\omega \sim_{L(\Box)} ssrrr(pqp)^\omega \qquad [\alpha = sr]$,
- $(pq)^\omega \sim_{L(\Box)} (pqp)^\omega \qquad [\alpha = \lambda]$,
- $(pq)^\omega \nsim_{L(\Box)} (qp)^\omega$,

- $p^\omega \nsim_{L(\Box)} (pq)^\omega$,
- $spsp^\omega \nsim_{L(\Box)} ssp^\omega$.

Note that, in the proof, regular expressions range over the set of assignments 2^{AP}, not over AP. Thus, if p, q are atomic propositions, then

$$\{p,q\}\{\{p\},\{q\}\} = \{\{p,q\}\{p\},\{p,q\}\{q\}\},$$

and

$$\{p,q\}\{\{p\},\{q\}\} \neq \{p\{p\}, p\{q\}, q\{p\}, q\{q\}\}.$$

Proof. First, we show by structural induction over Φ that the right hand side of the assertion implies

$$x \vDash \Phi \Leftrightarrow y \vDash \Phi, \tag{3.4}$$

for each $\Phi \in L(\Box)$. Note that it suffices to show the "only if" part of (3.4). The other direction follows from symmetry arguments.

Let Φ be a state formula. The assertion (3.4) follows from $v(x_0) = v(y_0)$.

Let $\Phi = \Box\Xi$ for some $\Xi \in L(\Box)$, and suppose the left hand side of (3.4) holds. Hence, $x[j,\infty] \vDash \Xi$, for each $j \in \mathbb{N}$. Let $i \in \mathbb{N}$. By assumption, we have

$$v(y[i,\infty]) \in \{v(\alpha_\ell)\}^+ \ldots \{v(\alpha_{n-1})\}^+ v(R(inf(x))),$$

for some ℓ with $0 \leq \ell \leq n$. We choose $j \in \mathbb{N}$ such that $v(x_j) = v(y_i)$ and

$$v(x[j,\infty]) \in \{v(\alpha_\ell)\}^+ \ldots \{v(\alpha_{n-1})\}^+ v(R(inf(x))), \text{ too.}$$

Thus, replacing x by $x[j,\infty]$ and y by $y[i,\infty]$ in the right hand side of the assertion and applying the induction hypothesis yields $y[i,\infty] \vDash \Xi$. As i was arbitrary chosen, the right hand side of (3.4) holds.

In the cases $\Phi = \neg\Xi$ and $\Phi = \Xi \vee \Psi$, for some $\Xi, \Psi \in L(\Box)$, Equation (3.4) follows immediately from the induction hypothesis.

Now, suppose the right hand side of the assertion does not hold. If $v(x_0) \neq v(y_0)$, it is easy to find a state formula ϕ such that $x \nsim_\phi y$. In the case $v(inf(x)) \neq v(inf(y))$, with Proposition 3.14, there is a Muller formula Φ such that $x \nsim_\Phi y$.

For the remaining case, suppose that 1. and 2. hold but 3. does not. Let $\alpha \in Q^n, \beta \in Q^k$ of minimal length such that

$$v(x) \in \{v(\alpha_0)\}^+ \ldots \{v(\alpha_{n-1})\}^+ v(R(inf(x)))$$

and

$$v(y) \in \{v(\beta_0)\}^+ \ldots \{v(\beta_{k-1})\}^+ v(R(inf(x))).$$

As 3. does not hold, we have $v(\alpha) \neq v(\beta)$. Without loss of generality, we assume $n \geq k$. It cannot be the case that $n = k = 0$, because in this case $\alpha = \lambda = \beta$, a contradiction. Thus, α_{n-1} exists. As α has minimal length, $v(\alpha_{n-1}) \notin v(R(inf(x)))$ and $v(\alpha_i) \neq v(\alpha_{i+1})$ $(0 \leq i < n-1)$. For an assignment $\varphi \in v(Q)$, let χ_φ be a state formula that is satisfied by a state $q \in Q$ iff $v(q) = \varphi$. We define

$$\Phi := \chi_{v(\alpha_0)} \wedge \Diamond(\chi_{v(\alpha_1)} \wedge \Diamond(\cdots \wedge \Diamond(\chi_{v(\alpha_{n-1})}) \dots)).$$

For $z \in Q^\omega$, we have

$$z \vDash \Phi \text{ iff } v(z) \in v(\alpha_0)v(Q)^*v(\alpha_1) \dots v(Q)^*v(\alpha_{n-1})\uparrow .$$

Clearly, $x \vDash \Phi$. Suppose $y \vDash \Phi$. As $v(y) \in \{v(\beta_0)\}^+ \dots \{v(\beta_{k-1})\}^+ v(R(inf(x)))$ and $v(\alpha_{n-1}) \notin v(R(inf(x)))$, there are $i_0, \dots, i_{n-1} \in \mathbb{N}$ with

$$0 = i_0 < i_1 < \cdots < i_{n-1} \leq k-1$$

such that $v(\beta_{i_j}) = v(\alpha_j)$, for $0 \leq j \leq n-1$. Because $n \geq k$, this implies

$$i_0 = 0,\ i_1 = 1,\ \dots,\ i_{n-1} = n-1.$$

Hence, $v(\alpha) = v(\beta)$, a contradiction. We conclude that $y \nvDash \Phi$ and therefore $x \nsim_{L(\Box)} y$. □

Note that Proposition 3.15 can alternatively be derived from results of Sistla and Zuck [1993].

Proposition 3.15 is a useful tool, which enables us to show that for certain LTL formulae there are no equivalent RLTL formulae. Let Σ be a system such that p, q, r are pairwise different states of Σ.

First, suppose there is an RLTL formula Φ which is equivalent to $p \mathsf{U} q$. With Proposition 3.15, we have $p(qr)^\omega \sim_\Phi p(rq)^\omega$. As Φ and $p \mathsf{U} q$ are equivalent, $p(qr)^\omega \sim_{p\mathsf{U}q} p(rq)^\omega$. But, $p(qr)^\omega \vDash p \mathsf{U} q$ and $p(rq)^\omega \nvDash p \mathsf{U} q$, a contradiction. Therefore, there is no RLTL formula which is equivalent to $p \mathsf{U} q$.

As a second example, suppose there is an RLTL formula which is equivalent to $\mathsf{X}q$. Thus, there is also an RLTL formula Φ which is equivalent to $\mathsf{X}q$. With Proposition 3.15, we have $(pq)^\omega \sim_\Phi p(pq)^\omega$. As Φ and $\mathsf{X}q$ are equivalent, we have $(pq)^\omega \sim_{\mathsf{X}q} p(pq)^\omega$, a contradiction. We conclude that there is no RLTL formula which is equivalent to $\mathsf{X}q$.

3.6.3. State Fairness is Complete for RLTL.

For a system Σ, we show that SF_Σ is $Sat(L(\Box))$-complete w. r. t. Σ. It has been illustrated by Zuck et al. [2002] how this can be derived from a theorem of Pnueli and Zuck [1993]. Here, we provide an alternative, direct and detailed proof.

Theorem 3.16. *Let $\Sigma = (Q, S, \to, v)$ be a system. Then SF_Σ is $Sat(L(\Box))$-complete w. r. t. Σ.*

Proof. Let $\Phi \in L(\Box)$ such that $Sat(\Phi)$ is large in Σ and x be a state fair path of Σ. We have to show that $x \vDash \Phi$.

With Proposition 2.7, we choose $\alpha \in Q^*, q \in Q, y \in Q^\omega$ such that $x = \alpha q y$, $qy \in K^\omega$ and $inf(qy) = K$, where K is a b. s. c. c. of Σ. Note that $\alpha q\uparrow$ is medium-sized and SF_Σ is large. With Proposition 2.6, $\alpha q\uparrow \cap SF_\Sigma$ is medium-sized. Hence, $(\alpha q\uparrow \cap SF_\Sigma)^c$ is medium-sized, too. Let g be a winning strategy for Alter in $(\overline{AE}, \Sigma, (\alpha q\uparrow \cap SF_\Sigma)^c)$. Moreover, f is a winning strategy for Ego in $(\overline{AE}, \Sigma, Sat(\Phi))$. If Ego plays according to f and Alter according to g, then they create a path $u = \alpha q z$ of Σ which is state fair *and* satisfies Φ. Clearly, $x_0 = u_0$. As both x and u are state fair and eventually take a state of K, we have $inf(x) = K = inf(u)$. Moreover, $x, u \in \alpha R(K)$. With Proposition 3.15, $x \sim_{L(\Box)} u$. As $\Phi \in L(\Box)$ and $u \vDash \Phi$, we conclude that $x \vDash \Phi$. □

3.6.4. Description of the Transformation $\mathcal{T}_{\overline{L}(\Box\Diamond)}$

For the transformation $\mathcal{T}_{\overline{L}(\Box\Diamond)}$, each Muller formula is feasible. Let $\Sigma = (Q, S, \to, v)$ be a reduced system, Φ a specification, $\Theta \in \overline{L}(\Box\Diamond)$ a feasible formula, $\Sigma' := (Q', S', \to', v') := \mathcal{T}_{\overline{L}(\Box\Diamond)}(\Sigma, \Theta)$ and $\Phi' := \mathcal{T}_{\overline{L}(\Box\Diamond)}(\Phi, \Theta)$. According to Section 3.1, it suffices to define $\to'$ and S'.

The new state relation $\to'$ and the new set of starting states S' are exactly as in the case of $\mathcal{T}_{\mathsf{U}}$. Note that $\mathcal{T}_{\overline{L}(\Box\Diamond)}$ does not produce unreachable states. Hence, T2 is satisfied.

We want to visualise the state relations of Σ and Σ' with block graphs as introduced in Section 3.2. For this, we need the following lemma:

Lemma 3.17. *Let $\Sigma = (Q, S, \to, v)$ and Θ be as defined above, and q a state of Σ.*

1. *If $q \in Q^L_\Theta$, then each successor of q belongs to Q^L_Θ.*
2. *If $q \in Q^S_\Theta$, then each successor of q belongs to Q^S_Θ.*
3. *If $q \in Q^M_\Theta$, then there exists a path fragment αs of Σ_q such that $s \in Q^L_\Theta$.*
4. *If $q \in Q^M_\Theta$, then there exists a path fragment αs of Σ_q such that $s \in Q^S_\Theta$.*

Proof. To see 1., suppose $q \in Q^L_\Theta$, and let s be a successor of q. We show that $s \in Q^L_\Theta$. Let x be a state fair path of Σ_s. Then, qx is a path of Σ_q. Note that qx is state fair w. r. t. Σ_q. With Theorem 3.16, $qx \vDash \Theta$. Applying Proposition 3.14, we have $qx \sim_{\overline{L}(\Box\Diamond)} x$, and thus $x \vDash \Theta$. We conclude that $SF_{\Sigma_s} \subseteq Sat(\Theta)$, which implies $s \in Q^L_\Theta$.

For 2., suppose $q \in Q^S_\Theta$, and let s be a successor of q. As $q \in Q^L_{\neg\Theta}$, we conclude, with the paragraph above, that $s \in Q^L_{\neg\Theta}$. Hence, $s \in Q^S_\Theta$.

For the proof of 3., suppose $q \in Q^M_\Theta$. As $Sat(\neg\Theta)$ is not large in Σ_q, we have $SF_{\Sigma_q} \cap Sat(\Theta) \neq \emptyset$. Let $x \in SF_{\Sigma_q} \cap Sat(\Theta)$. We choose i such that each state of $x[i,\infty]$ belongs to a b. s. c. c. of Σ. Now, let $y \in SF_{\Sigma_{x_i}}$. With Proposition 3.14, $x \sim_{\overline{L}(\Box\Diamond)} y$, and therefore $y \vDash \Theta$. We conclude $SF_{\Sigma_{x_i}} \subseteq Sat(\Theta)$, which implies $x_i \in Q^L_\Theta$.

Part 4 can be shown by exchanging Θ and $\neg\Theta$. □

As a direct consequence of Lemma 3.17, we can describe Σ and Σ' by the block graphs, as introduced in Section 3.2, in Figure 3.11 and 3.12.

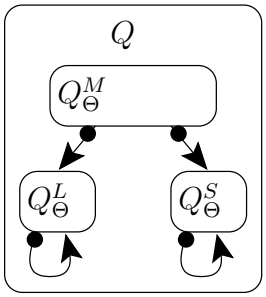

Figure 3.11.: A block graph of Σ, where Θ is a Muller formula.

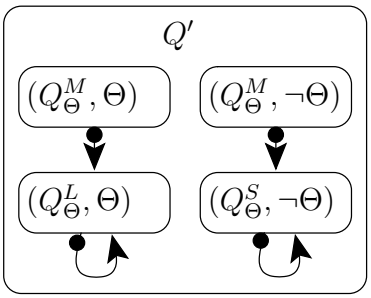

Figure 3.12.: A block graph of Σ', where Θ is a Muller formula.

Now, we explain how the transformation $\mathcal{T}_{\overline{L}(\Box\Diamond)}$ can be computed efficiently. The main difficulty is to compute the partition of Q into Q^L_Θ, Q^M_Θ and Q^S_Θ. The transformations afterwards are straightforward. First, the algorithm computes the b. s. c. c.s of Σ. For each b. s. c. c. K and each $\Upsilon \preceq \Theta$, we define $K^L_\Upsilon := K \cap Q^L_\Upsilon$ and $K^S_\Upsilon := K \cap Q^S_\Upsilon$. (Although, in general, Υ is no Muller formula, it turns out that

no state of a b. s. c. c. belongs to Q^M_Υ.) The algorithm determines with the following recursive procedure which states of K belong to K^L_Υ and which, to K^S_Υ:

In each case, we set $K^S_\Upsilon := K \setminus K^L_\Upsilon$.

1. If Υ is a state formula, then exactly the states of K which satisfy Υ belong to K^L_Υ.
2. If $\Upsilon = \Xi \vee \Psi$, then $K^L_\Upsilon := K^L_\Xi \cup K^L_\Psi$.
3. If $\Upsilon = \neg\Xi$, then $K^L_\Upsilon := K^S_\Xi$.
4. If $\Upsilon = \Box\Diamond\,\Xi$, then $K^L_\Upsilon := K$ if $K^L_\Xi \neq \emptyset$; otherwise, $K^L_\Upsilon := \emptyset$.

We explain why the procedure above correctly computes the partition of the b. s. c. c. K into K^L_Υ and K^S_Υ. In case 1, the partition is correct, as it depends only on the first state of a path if that path satisfies Υ. In case 2, this is true, as largeness is upward closed and smallness is closed under countable union. In case 3, K^L_Υ and K^S_Υ are computed properly, as the complement of a large linear-time property is small and vice versa.

For case 4, suppose that $K^L_\Xi \neq \emptyset$, and choose $q \in K$ and $p \in K^L_\Xi$. We show that $SF_\Sigma \subseteq Sat(\Box(q \Rightarrow \Upsilon))$. Let $x \in SF_\Sigma$. With Theorem 3.16, x satisfies $\Box(p \Rightarrow \Xi)$. As p and q belong to the same b. s. c. c., x also satisfies $\Box(q \Rightarrow \Box\Diamond p)$. Putting this together, we have $x \vDash \Box(q \Rightarrow \Box\Diamond\,\Xi) = \Box(q \Rightarrow \Upsilon)$.

On the other hand, suppose $K^L_\Xi = \emptyset$, and choose $q \in K$. We show that $SF_\Sigma \subseteq Sat(\Box(q \Rightarrow \neg\Upsilon))$. Let $x \in SF_\Sigma$. With Theorem 3.16, $x \vDash \Box(p \Rightarrow \neg\Xi)$, for each $p \in K$. As K cannot be left, $x \vDash \Box(q \Rightarrow \neg\Diamond\,\Xi)$, and therefore $x \vDash \Box(q \Rightarrow \neg\Upsilon)$.

A state q of Σ that does not belong to a b. s. c. c. is classified using the following facts:

- If each successor of q belongs to Q^L_Θ, then $q \in Q^L_\Theta$, too.
- If each successor of q belongs to Q^S_Θ, then $q \in Q^S_\Theta$, too.
- Otherwise, if q belongs neither to Q^L_Θ nor to Q^S_Θ, then $q \in Q^M_\Theta$.

The correctness follows from Lemma 3.17.

Using the arguments of the proof of Lemma 3.4, we show:

Lemma 3.18. *For the systems Σ, Σ' as defined above, the following statements hold:*

1. *If $x' \in path(\Sigma') \cup path(\Sigma')\downarrow$, then $x \in path(\Sigma) \cup path(\Sigma)\downarrow$.*
2. *For each $\alpha \in path(\Sigma)\downarrow$, there is an $\alpha' \in path(\Sigma')\downarrow$ such that $\pi(\alpha') = \alpha$.*
3. *If the last states of $\alpha'_1, \alpha'_2 \in path(\Sigma')\downarrow$, $\alpha'_1, \alpha'_2 \neq \lambda$ do not differ and $\alpha_1 = \alpha_2$, then $\alpha'_1 = \alpha'_2$.*

Now, we are ready to show that $\mathcal{T}_{\overline{L(\Box\Diamond)}}$ is correct.

Lemma 3.19. *Let $\Sigma = (Q, S, \rightarrow, v)$, $\Sigma' = (Q', S', \rightarrow', v')$, Θ as defined above and $\theta = \mathcal{T}_{\overline{L(\Box\Diamond)}}(\Theta, \Theta)$ the fresh atomic proposition. Then, the following conditions hold:*

1. *Each path in $SF_{\Sigma'}$ satisfies $\Box(\Theta \Leftrightarrow \theta)$.*
2. *The transformation $\mathcal{T}_{\overline{L(\Box\Diamond)}}$ is correct.*

Proof. We show 1. Let $x' \in SF_{\Sigma'}$, and suppose $x', i \vDash \theta$. According to Figure 3.12, x' belongs either to $(Q^M_\Theta, \Theta)^\omega$ or to $(Q^M_\Theta, \Theta)^*(Q^L_\Theta, \Theta)^\omega$. As x' is state fair, we conclude that $x' \in (Q^M_\Theta, \Theta)^*(Q^L_\Theta, \Theta)^\omega$. Because (Q^L_Θ, Θ) and Q^L_Θ are isomorphic and $x \in (Q^M_\Theta)^*(Q^L_\Theta)^\omega$, x is state fair w. r. t. Σ, too. We choose $j \geq i$ such that $x_j \in Q^L_\Theta$. Due to Theorem 3.16, we have $x \vDash \Box(x_j \Rightarrow \Theta)$, and hence $x, j \vDash \Theta$. As Θ does not contain θ, $x', j \vDash \Theta$. We conclude, applying Proposition 3.14, that $x', i \vDash \Theta$. If $x', i \nvDash \theta$, it can be proved in a similar way that $x', i \nvDash \Theta$. Therefore, $x' \vDash \Box(\Theta \Leftrightarrow \theta)$.

We use Lemma 3.1 to show 2. A1 follows from 1. For A2, let Υ be a path formula with $\theta, q' \not\trianglelefteq \Upsilon$, for each $q' \in Q'$. We show that the same player has a winning strategy in $(\overline{AE}, \Sigma, \mathit{Sat}(\Upsilon))$ and a winning strategy in $(\overline{AE}, \Sigma', \mathit{Sat}(\Upsilon))$. Suppose that $P \in \{E, A\}$ and f is a winning strategy for P in $(\overline{AE}, \Sigma, \mathit{Sat}(\Upsilon))$. We construct a strategy f' in Σ' such that $\pi(f'(\lambda)) = f(\lambda)$ and, for each $\alpha' \in \mathit{path}(\Sigma')\downarrow$ with $\alpha' \neq \lambda$, we have $f'(\alpha') = \alpha'\beta'\gamma'$ with $f(\alpha\beta) = \alpha\beta\gamma$. Hence, $\pi(Z^{P,f'}_{\overline{AE}}) \subseteq Z^{P,f}_{\overline{AE}}$. As $\theta, q' \not\trianglelefteq \Upsilon$, for each $q' \in Q'$, we conclude that f' is winning for P in $(\overline{AE}, \Sigma', \mathit{Sat}(\Upsilon))$.

We define $f'(\lambda)$ using Lemma 3.18. Let $\alpha' \in \mathit{path}(\Sigma')\downarrow$, $\alpha' \neq \lambda$. As illustrated in Figure 3.12, we can choose β' such that $\alpha'\beta' \in \mathit{path}(\Sigma)\downarrow$ and $\alpha\beta$ ends in $Q^S_\Theta \cup Q^L_\Theta$. We define γ with $f(\alpha\beta) =: \alpha\beta\gamma$. As the block graph of Σ shows, γ belongs to $(Q^L_\Theta)^*$ or to $(Q^S_\Theta)^*$, depending on the last state of $\alpha\beta$. Thus, we can choose γ' such that $\pi(\gamma') = \gamma$ and $\alpha'\beta'\gamma' \in \mathit{path}(\Sigma')\downarrow$. □

3.6.5. There is no Transformation $\mathcal{T}_{L(\Box)}$.

After all, we might try to develop a transformation $\mathcal{T}_{L(\Box)}$ for which the formulae in $L(\Box)$ are feasible. To carry out such a transformation $\mathcal{T}_{L(\Box)}$, the generalised fair model checking problem with specifications in $L(\Box)$ has to be solved efficiently. That is necessary to determine Q^L_Θ. For convenience, we write GFMC(L) for the generalised fair model checking problem with specifications in L. Unfortunately, we can prove that GFMC($L(\Box)$) is co-NP-complete. Below, we even show a stronger result. Thus, it is impossible to find an efficient transformation $\mathcal{T}_{L(\Box)}$, provided that P $\neq$ NP.

First, we define the language $L^1(\Box)$ as the smallest set which satisfies the conditions RP1 and RP2 below:

RP1: If ξ is a state formulae, then ξ, $\Box\,\xi \in L^1(\Box)$.

RP2: If Ξ, Ψ belong to $L^1(\Box)$, then so do $\Xi \vee \Psi$, $\neg\Xi$.

In contrast to $L(\Box)$, the language $L^1(\Box)$ does not allow nested temporal operators. Thus, $\Box(\psi \Rightarrow \Diamond\xi) \in L(\Box)$, but $\Box(\psi \Rightarrow \Diamond\xi) \notin L^1(\Box)$, where ψ, ξ are atomic propositions.

Now, we are ready to state the completeness result.

Theorem 3.20. *The problem* GFMC($L^1(\Box)$) *is co-NP-hard. Moreover, the problem* GFMC($L(\Box)$) *belongs to co-NP.*

Proof. We prove hardness, adapting an idea of Sistla and Clarke [1985], with a reduction from TAUTOLOGY to GFMC($L^1(\Box)$). TAUTOLOGY contains the state formulae ϕ that are *tautologies*; i. e., for each system Σ and each state q of Σ, we have $q \vDash \phi$. It is well-known that TAUTOLOGY is co-NP-complete. Let ϕ be a state formula over the atomic propositions $\zeta_1, \ldots, \zeta_m$. The specification $\hat{\Phi} \in L^1(\Box)$ is obtained by replacing each atomic proposition ζ_i in ϕ by $\Diamond\zeta_i$. If, e. g., $\phi = (\zeta_1 \vee \neg\zeta_2) \wedge \zeta_3$, then $\hat{\Phi} = (\Diamond\zeta_1 \vee \neg\Diamond\zeta_2) \wedge \Diamond\zeta_3$. The reduction maps ϕ to $\hat{\Phi}$ and the system Σ as described in Figure 3.8. Clearly, the reduction can be computed in polynomial time. Ego has a winning strategy in $(\overline{AE}, \Sigma, \mathit{Sat}(\hat{\Phi}))$ iff ϕ is a tautology. Thus, $\mathit{Sat}(\hat{\Phi})$ is large in Σ iff $\phi \in$ TAUTOLOGY.

Now, we prove the membership of GFMC($L(\Box)$) in co-NP. Let $\Sigma = (Q, S, \rightarrow, v)$ a system. First, we show that SF_Σ can be expressed as a formula in $L(\Box)$. As Q is finite, for each $q \in Q$, there is a state formula $\mathit{enabled}(q)$ that is exactly satisfied by the states at which q is enabled. Thus, SF_Σ can be expressed by

$$\Psi(\Sigma) := \bigwedge_{q \in Q} (\Box\Diamond\, \mathit{enabled}(q) \Rightarrow \Box\Diamond\, q),$$

which belongs to $L(\Box)$.

It is sufficient to define a reduction from GFMC($L(\Box)$) to (traditional) model checking with specifications in $L(\Box)$. The latter problem belongs to co-NP, as has been shown by Sistla and Clarke [1985]. Given a system Σ and a specification $\Phi \in L(\Box)$, the reduction maps (Φ, Σ) to $(\hat{\Phi}, \Sigma)$, where $\hat{\Phi} := (\Psi(Q) \Rightarrow \Phi) \in L(\Box)$. Clearly, the reduction can be computed in polynomial time. By Theorem 3.16, $\mathit{Sat}(\Phi)$ is large in Σ iff each path of Σ satisfies $\hat{\Phi}$. □

3.7. Complexity Analysis of the Algorithm

In the following, we measure the size $|\Sigma|$ of a system Σ by the number of its nodes and edges. The size $|\Phi|$ of a specification Φ is the number of its temporal and boolean operators. The number of transformation steps carried out by the algorithm on input (Σ, Φ) is n. In the complexity analysis, we apply the uniform cost criterion, under

which a single node, edge or atomic proposition can be read in a constant amount of time.

Theorem 3.21 states that our version of the algorithm (with or without $\mathcal{T}_{\overline{L(\Box\Diamond)}}$) runs in exponential time in the size of the specification and linear time in the size of the system. This is the time bound which Courcoubetis and Yannakakis [1995] also prove for the original algorithm.

Theorem 3.21. *Checking whether $Sat(\Phi)$ is large, medium-sized or small in a system Σ, where $\Phi \in L(\mathsf{X}, \mathsf{Y}, \mathsf{U}, \mathsf{S})$, can be accomplished in time*

$$\mathcal{O}((\max_{i=1,\ldots,n} |\Theta_i| + |\Phi_n|) \cdot 2^n |\Sigma|) \leq \mathcal{O}(2^{|\Phi|}|\Sigma|).$$

Here, Θ_i is the feasible subformula the algorithm chooses in the i-th transformation step, $\Phi_0 := \Phi$ and Φ_i is the specification after the i-th transformation step ($1 \leq i \leq n$).

Proof. The transformation of Σ in a reduced system can be accomplished in time $\mathcal{O}(|\Sigma|)$. Thus, we assume, without loss of generality, that Σ is reduced and apply the algorithm developed above.

Let $\Sigma_0 := \Sigma$ and Σ_i be the system after the i-th transformation step ($1 \leq i \leq n$). The operations the algorithm possibly performs in the i-th transformation step are listed below:

1. in the case of $\mathcal{T}_{\overline{L(\Box\Diamond)}}$: partition of K into K^L_Υ and K^S_Υ, for each $\Upsilon \preceq \Theta_i$ and each b. s. c. c. K of Σ_{i-1},
2. in the cases of $\mathcal{T}_{\mathsf{U}}, \mathcal{T}_{\mathsf{S}}$: evaluation of ξ_i and ψ_i at each state of Σ_{i-1}, where $\Theta_i = \xi_i \,\mathsf{U}\, \psi_i$ or $\Theta_i = \xi_i \,\mathsf{S}\, \psi_i$,
3. in the cases of $\mathcal{T}_{\mathsf{X}}, \mathcal{T}_{\mathsf{Y}}$: evaluation of ξ_i at each state of Σ_{i-1}, where $\Theta_i = \mathsf{X}\xi_i$ or $\Theta_i = \mathsf{Y}\xi_i$,
4. computation of the s. c. c.s of Σ_{i-1} or a subgraph of Σ_{i-1},
5. travelling through the s. c. c.s in (possibly reversed) topological order,
6. construction of the new system and the new specification.

Each of these operations can be computed in time

$$\mathcal{O}(|\Sigma_{i-1}| \cdot |\Theta_i|).$$

For the 1. operation, this is true, as Θ_i has $\mathcal{O}(|\Theta_i|)$ subformulae and the computation time for each subformula is in $\mathcal{O}(|\Sigma_{i-1}|)$. In the remaining cases, the operations can be performed in $\mathcal{O}(|\Sigma_{i-1}| \cdot |\Theta_i|)$ by using standard techniques.

As the postprocessing (i. e. checking whether $Sat(\Phi_n)$ is large, medium-sized or small in Σ_n) needs at most $\mathcal{O}(|\Sigma_n| \cdot |\Phi_n|)$ steps, the total execution time belongs to

$$\mathcal{O}(\sum_{i=1}^{n} |\Sigma_{i-1}| \cdot |\Theta_i| + |\Sigma_n| \cdot |\Phi_n|).$$

Because the size of the system at most doubles after each transformation step, the number of computation steps is in

$$\begin{aligned} & \mathcal{O}(\max_{i=1,\ldots,n} |\Theta_i| \cdot \sum_{i=0}^{n-1} 2^i |\Sigma| + 2^n |\Sigma| \cdot |\Phi_n|) \\ = \; & \mathcal{O}((\max_{i=1,\ldots,n} |\Theta_i| + |\Phi_n|) \cdot 2^n |\Sigma|). \end{aligned}$$

The other upper bound can be derived as follows:

$$\begin{aligned} \mathcal{O}((\max_{i=1,\ldots,n} |\Theta_i| + |\Phi_n|) \cdot 2^n |\Sigma|) &\leq \mathcal{O}((2^{\max_{i=1,\ldots,n} |\Theta_i|} + 2^{|\Phi_n|}) \cdot 2^n |\Sigma|) \\ &\leq \mathcal{O}((2^{\max_{i=1,\ldots,n} |\Theta_i| + n} + 2^{|\Phi_n| + n}) \cdot |\Sigma|) \\ &\leq \mathcal{O}((2^{|\Phi|+1} + 2^{|\Phi|}) \cdot |\Sigma|) \\ &= \mathcal{O}(2^{|\Phi|} \cdot |\Sigma|). \end{aligned}$$

For the third line, note that in each transformation step at least one temporal operator is removed from the specification. Thus, $n \leq |\Phi| - |\Phi_n|$, and $n - 1 \leq |\Phi| - \max_{i=1,\ldots,n} |\Theta_i|$. □

The first upper bound of Theorem 3.21 is mainly influenced by the number of transformation steps. Thus, it seems to be the case that, whenever possible, the transformation $\mathcal{T}_{\overline{L}(\Box\Diamond)}$ should be applied instead of repeatedly applying $\mathcal{T}_{\mathsf{U}}$, because the resulting time bound is better. Indeed, we can show that, in some sense, applying $\mathcal{T}_{\overline{L}(\Box\Diamond)}$ as often as possible needs, in the worst case, only negligible more running time than never applying $\mathcal{T}_{\overline{L}(\Box\Diamond)}$. Consider Appendix A for the sketch of a proof.

If the specification is a Muller formula, the number of transformation steps is one. Thus, the running time is linear in the size of the specification and the system:

Corollary 3.22. *Checking whether $Sat(\Phi)$ is large, medium-sized or small in a system Σ, where $\Phi \in \overline{L}(\Box\Diamond)$, can be accomplished in time $\mathcal{O}(|\Phi||\Sigma|)$.*

This is interesting, as checking whether *each* path of a system Σ satisfies a specification $\Phi \in \overline{L}(\Box\Diamond)$ is co-NP-complete.

Theorem 3.23. *Let Σ be a system and $\Phi \in \overline{L}(\Box\Diamond)$ a specification. Checking whether each path of Σ satisfies Φ is co-NP-complete.*

Proof. To develop the proof, a hint of Schnoebelen [2003] was helpful. Let L be a set of specifications. For shortness, the problem of checking, given a system and a specification $\Phi \in L$, whether each path of the system satisfies Φ is denoted by MC(L). The membership of MC($\overline{L}(\Box\Diamond)$) to co-NP follows immediately from the fact that MC($L(\Box)$) belongs to co-NP (cf. Sistla and Clarke [1985]).

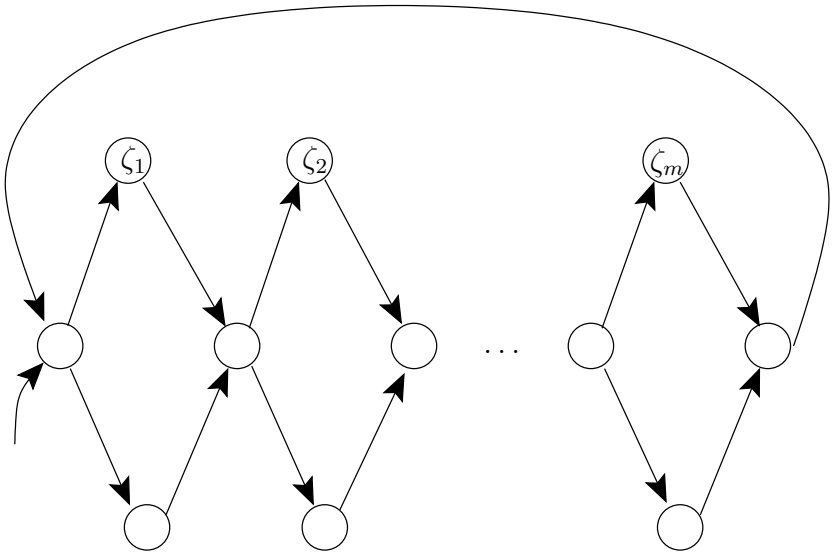

Figure 3.13.: A system Σ. We omit state names and related atomic propositions.

We prove hardness with a reduction from TAUTOLOGY to MC($\overline{L}(\Box\Diamond)$). Let ϕ be a state formula over the atomic propositions $\zeta_1, \ldots, \zeta_m$. The specification $\hat{\Phi} \in \overline{L}(\Box\Diamond)$ is obtained by replacing each atomic proposition ζ_i in ϕ by $\Box\Diamond\zeta_i$. The reduction maps ϕ to $\hat{\Phi}$ and the system Σ as described in Figure 3.13. Clearly, the reduction can be computed in polynomial time. Then, each path of Σ satisfies $\hat{\Phi}$ iff $\phi \in$ TAUTOLOGY. □

4. Witnesses

Besides the information how much a system Σ satisfies its specification Φ (large, medium-sized or small), it is useful to obtain some *witness* for this fact. Suppose, for instance, that $Sat(\Phi)$ is not large in Σ. Then, Alter has a winning strategy f in $(\overline{AE}, \Sigma, Sat(\Phi))$. We do not try to compute the entire strategy f, as we are not aware of a finite representation of f that supports the developer in finding errors in the system. But, we develop a method that allows to determine $f(\alpha)$, given a path fragment α of Σ. Thus, the developer can *interactively play* (and lose) the game $(\overline{AE}, \Sigma, Sat(\Phi))$ against Alter. We hope that, in this way, he gains insight into why $Sat(\Phi)$ is not large in Σ. It still remains to figure out by case studies how useful this interactive play is in practice.

The notion of a witness motivated above is formalised as follows:

Definition 4.1. Let Σ be a reduced system and Φ a specification.

- If $Sat(\Phi)$ is large in Σ, a *witness for* Σ *and* Φ is a winning strategy for Ego in $(\overline{AE}, \Sigma, Sat(\Phi))$.
- If $Sat(\Phi)$ is medium-sized in Σ, a *witness for* Σ *and* Φ is (f, g), where f is a winning strategy for Ego in $(\overline{EA}, \Sigma, Sat(\Phi))$ and g, for Alter in $(\overline{AE}, \Sigma, Sat(\Phi))$.
- Otherwise, if $Sat(\Phi)$ is small in Σ, a *witness for* Σ *and* Φ is a winning strategy for Alter in $(\overline{EA}, \Sigma, Sat(\Phi))$.

We first describe the general approach in Section 4.1. In Section 4.2, we illustrate the main difficulties of constructing witnesses by examples. The details of our techniques are discussed in the Sections 4.3 up to 4.7. Finally, in Section 4.8, we analyse the complexity of computing $f(\alpha)$, where f is part of a witness and $\alpha \in path(\Sigma)\downarrow$.

4.1. The General Approach

Let $\Sigma = (Q, S, \rightarrow, v)$ be a reduced system and Φ a specification. We first explain how a witness for Σ and Φ can be defined and then how it can be computed.

If Φ is a state formula and the player $P \in \{A, E\}$ has a winning strategy in $(\kappa, \Sigma, Sat(\Phi))$, where $\kappa \in \{\overline{AE}, \overline{EA}\}$, then it is easy to define and compute a positional winning strategy for P in $(\kappa, \Sigma, Sat(\Phi))$.

Otherwise, if Φ is not a state formula, suppose the algorithm chooses Θ as feasible formula. Let $\Sigma' := (Q', S', \rightarrow', v') := \mathcal{T}(\Sigma, \Theta)$, $\Phi' := \mathcal{T}(\Phi, \Theta)$ and $\theta := \mathcal{T}(\Theta, \Theta)$,

where $\mathcal{T}$ is one of $\mathcal{T}_{\mathsf{U}}, \mathcal{T}_{\mathsf{X}}, \mathcal{T}_{\mathsf{S}}, \mathcal{T}_{\mathsf{Y}}, \mathcal{T}_{\overline{L(\Box\Diamond)}}$. A witness for Σ and Φ is defined recursively using a witness for Σ' and Φ':

Let $P \in \{E, A\}$ and $\kappa \in \{\overline{AE}, \overline{EA}\}$ such that P has a winning strategy f' in $\mathcal{G}' := (\kappa, \Sigma', \mathit{Sat}(\Phi'))$. We define a strategy f in Σ such that

$$Z_\kappa^{P,f} \subseteq \pi(Z_\kappa^{P,f'} \cap \mathit{Sat}(\Box(\Theta \Leftrightarrow \theta))). \tag{4.1}$$

In other words, each path $x \in Z_\kappa^{P,f}$ corresponds to some path $x' \in Z_\kappa^{P,f'}$ such that $\pi(x') = x$ and $x' \vDash \Box(\Theta \Leftrightarrow \theta)$. Then, f is a winning strategy for P in $\mathcal{G} := (\kappa, \Sigma, \mathit{Sat}(\Phi))$.

In the case $P = E$, this can be seen as follows. Let $x \in Z_\kappa^{E,f}$. With (4.1), there is an $x' \in Z_\kappa^{E,f'} \cap \mathit{Sat}(\Box(\Theta \Leftrightarrow \theta))$ with $\pi(x') = x$. As f' is winning for Ego in $\mathcal{G}'$, $x' \vDash \Phi'$. Moreover, $x' \vDash \Box(\Theta \Leftrightarrow \theta)$. Because $\Phi' = [\Phi]_\Theta^\theta$, we have $x' \vDash \Phi$. As, for each $q' \in Q'$, $\theta, q' \not\vDash \Phi$, this implies $x \vDash \Phi$. The case $P = A$ is similar.

So far, we explained how to define the strategy f based on f'. Given a path fragment α of Σ, we will see that $f(\alpha)$ is computed as follows. First, the algorithm chooses $\alpha' \in \mathit{path}(\Sigma')\downarrow$ such that $\pi(\alpha') = \alpha$. Then, the algorithm computes $f'(\alpha')$ and appends some $\delta' \in Q'^*$ (which highly depends on the transformation $\mathcal{T}$) to $f'(\alpha')$ such that $f'(\alpha')\delta' \in \mathit{path}(\Sigma')\downarrow$. Finally,

$$f(\alpha) := \pi(f'(\alpha')\delta')q,$$

where $q \in Q \cup \{\lambda\}$. In the Sections 4.3 to 4.7, we basically explain, for each transformation $\mathcal{T}$, how δ' and q are computed and why Equation (4.1) holds.

If f' is a positional strategy and $\mathcal{T}$ is one of $\mathcal{T}_{\mathsf{U}}, \mathcal{T}_{\mathsf{X}}, \mathcal{T}_{\overline{L(\Box\Diamond)}}$, we will see that f is a positional strategy, too. Thus, if $\Phi \in L(\mathsf{X}, \mathsf{U})$, our methods allow to compute a witness that consists of one or two positional strategies.

4.2. Examples

Before we explain the details, consider the systems Σ and Σ' as described in Figure 3.3 and 3.2. As in Section 3.1, the old and new specification are $\Phi = \zeta \wedge \Diamond(\eta \wedge \rho)$, $\Phi' = \zeta \wedge \theta$, the feasible formula is $\Theta = \Diamond(\eta \wedge \rho)$, and the fresh atomic proposition is θ. As Φ' is satisfied in only one starting state of Σ', $\mathit{Sat}(\Phi')$ is medium-sized in Σ'. Hence, $\mathit{Sat}(\Phi)$ is medium-sized in Σ. A witness for Σ and Φ consists of a positional winning strategy f for Ego in $(\overline{EA}, \Sigma, \mathit{Sat}(\Phi))$ and a positional winning strategy g for Alter in $(\overline{AE}, \Sigma, \mathit{Sat}(\Phi))$. For convenience, we do not exactly follow the general procedure of Section 4.3.

First, we construct g. A positional winning strategy g_1' for Alter in the game $(\overline{AE}, \Sigma', \mathit{Sat}(\Phi'))$ is given by

$$g_1'(\lambda) := (q_1, \neg\Theta) \quad \text{and} \quad g_1'(\alpha') := \alpha' \quad (\alpha' \in \mathit{path}(\Sigma')\downarrow, \alpha' \neq \lambda).$$

We define g_1 with

$$g_1(\lambda) := \pi(g_1'(\lambda)) = q_1 \quad \text{and} \quad g_1(\alpha) := \alpha \quad (\alpha \in \mathit{path}(\Sigma)\downarrow, \alpha \neq \lambda).$$

One might hope that g_1 is winning for Alter in $(\overline{AE}, \Sigma, \mathit{Sat}(\Phi))$. But, there is an $x := q_1q_4(q_5)^\omega \in Z_{\overline{AE}}^{A,g_1}$ with $x \vDash \Phi$. Thus, g_1 is not winning for Alter in $(\overline{AE}, \Sigma, \mathit{Sat}(\Phi))$.

An important observation is that there is no $x' \in Z_{\overline{AE}}^{A,g_1'}$ such that $\pi(x') = x$. This is due to the fact that $(q_1, \neg\Theta) \nrightarrow' (q_4, \Theta)$ but $q_1 \rightarrow q_4$; thus, in Σ', Ego cannot append $(q_4, \Theta)(q_5, \Theta)$ to $(q_1, \neg\Theta)$, but, in Σ, he can append q_4q_5 to q_1. More general, the problem is that Alter generates a path fragment α' ending in $(Q_\Theta^M, \neg\Theta)$. Unfortunately, there is some $\beta \in (Q_\Theta^M)^*Q_\Theta^L$ such that $\alpha\beta$ is a valid path fragment of Σ, although there is no β' with $\pi(\beta') = \beta$ such that $\alpha'\beta'$ is a valid path fragment of Σ'. This can be fixed by requiring that Alter never creates a path fragment that ends in $(Q_\Theta^M, \neg\Theta)$ or (Q_Θ^M, Θ). Therefore, we define

$$g_2'(\lambda) := (q_1, \neg\Theta)(q_3, \neg\Theta) \quad \text{and} \quad g_2(\alpha') := \alpha' \quad (\alpha' \in \mathit{path}(\Sigma')\downarrow, \alpha' \neq \lambda).$$

Moreover,

$$g_2(\lambda) := q_1q_3 \quad \text{and} \quad g_2(\alpha) := \alpha \quad (\alpha \in \mathit{path}(\Sigma)\downarrow, \alpha \neq \lambda).$$

Then, $g := g_2$ is a positional winning strategy for Alter in $(\overline{AE}, \Sigma, \mathit{Sat}(\Phi))$.

Second, we construct f. A positional winning strategy f_1' for Ego in the game $(\overline{EA}, \Sigma', \mathit{Sat}(\Phi'))$ is given by

$$f_1'(\lambda) := (q_1, \Theta)(q_4, \Theta) \quad \text{and} \quad f_1'(\alpha') := \alpha' \quad (\alpha' \in \mathit{path}(\Sigma')\downarrow, \alpha' \neq \lambda).$$

Note that, for convenience, f_1' does not match the criterion elaborated in the last paragraph. Nevertheless, this time a different problem arises. We define

$$f_1(\lambda) := q_1q_4 \quad \text{and} \quad f_1(\alpha) := \alpha \quad (\alpha \in \mathit{path}(\Sigma)\downarrow, \alpha \neq \lambda).$$

However, f_1 is not winning for Ego in $(\overline{EA}, \Sigma, \mathit{Sat}(\Phi))$, as $x := q_1(q_4)^\omega \in Z_{\overline{EA}}^{E,f_1}$. Now, there is a (unique) path $x' := (q_1, \Theta)(q_4, \Theta)^\omega \in Z_{\overline{EA}}^{E,f_1'}$ such that $\pi(x') = x$.

The problem is that $x' \nvDash \Box(\Theta \Leftrightarrow \theta)$. This is solved if we require that $Z_{\overline{EA}}^{E,f_1'} \subseteq \mathit{Sat}(\Box(\Theta \Leftrightarrow \theta))$. In the special case, it even suffices to define f_2' with

$$f_2'(\lambda) := (q_1, \Theta)(q_4, \Theta)(q_5, \Theta) \quad \text{and} \quad f_2'(\alpha') := \alpha' \quad (\alpha' \in \mathit{path}(\Sigma')\downarrow, \alpha' \neq \lambda).$$

Now, we set

$$f_2(\lambda) := q_1q_4q_5 \quad \text{and} \quad f_2(\alpha) := \alpha \quad (\alpha \in \mathit{path}(\Sigma)\downarrow, \alpha \neq \lambda).$$

Then, $f := f_2$ is a positional winning strategy for Ego in $(\overline{EA}, \Sigma, \mathit{Sat}(\Phi))$.

4.3. Witnesses in the Case of $\mathcal{T}_{\mathsf{U}}$

In the case $\mathcal{T} = \mathcal{T}_{\mathsf{U}}$ and $\Theta = \xi \,\mathsf{U}\, \psi$, we first define a strategy g' in Σ' as follows. Let $\alpha' \in \mathit{path}(\Sigma')\!\downarrow$, and define β' by $f'(\alpha') =: \alpha'\beta'$. We choose γ' such that $\alpha'\beta'\gamma' \in \mathit{path}(\Sigma')\!\downarrow$ and the following conditions hold:

- If $\alpha\beta$ ends in Q^M_Θ, then $\alpha\beta\gamma$ ends in $Q^L_\Theta \cup Q^S_\Theta$.
- If α ends in $Q^L_\Theta \setminus \mathit{sat}(\psi)$ and $\beta \in (Q^L_\Theta \setminus \mathit{sat}(\psi))^*$, then $\alpha'\beta'\gamma'$ ends in a state satisfying ψ.

We set $g'(\alpha') := \alpha'\beta'\gamma'$. Hence, g' is a strategy in Σ'. If f' is positional, then g' is positional, too. Note that, given $f'(\alpha')$, $g'(\alpha')$ can be computed easily.

A path x' of Σ' with $g'(x'[0,i]) \sqsubseteq x'$ for infinitely many $i \in \mathbb{N}$ satisfies $\Box(\theta \Rightarrow \Diamond\psi)$. This can be seen in the block graph of Σ' in Figure 3.5. Moreover, $Z^{P,g'}_\kappa \subseteq Z^{P,f'}_\kappa$.

For $\alpha \in \mathit{path}(\Sigma)\!\downarrow$, choose $\alpha' \in \mathit{path}(\Sigma')\!\downarrow$ with $\pi(\alpha') = \alpha$. Such an α' exists due to Lemma 3.4. If $\pi(g'(\alpha')) = \alpha$, then let $q \in Q$ such that $\pi(g'(\alpha'))q \in \mathit{path}(\Sigma)\!\downarrow$, and set $f(\alpha) := \pi(g'(\alpha'))q$. Otherwise, we define $f(\alpha) := \pi(g'(\alpha'))$. Thus, f is a progressive strategy in Σ. If f' is positional, then f is positional, too.

We show (4.1). Let $x \in Z^{P,f}_\kappa$. For each $i \in \mathbb{N}$, we define $\alpha_i \in \mathit{path}(\Sigma)\!\downarrow$ and $q_i \in Q \cup \{\lambda\}$ such that

$$\alpha_{2i} \sqsubset f(\alpha_{2i}) = \alpha_{2i+1}q_i \sqsubseteq \alpha_{2i+2} \sqsubseteq x \quad (i \in \mathbb{N}),$$

where $\alpha_0 = \lambda$ if P controls the zeroth move in a play of $\mathcal{G}$ and $q_i = \lambda$ iff $\alpha_{2i} \neq \alpha_{2i+1}$. In the same way as in the construction of f, we choose $\alpha'_i \in \mathit{path}(\Sigma)\!\downarrow$ such that $\pi(\alpha'_i) = \alpha_i$ and $g'(\alpha'_{2i}) = \alpha'_{2i+1}$, for each $i \in \mathbb{N}$.

Let $\tilde{\alpha}'_{2i+1}$ be the prefix of α'_{2i+2} of length $|\alpha'_{2i+1}|$, where $i \in \mathbb{N}$. By construction, $\pi(\tilde{\alpha}'_{2i+1}) = \alpha_{2i+1} = \pi(\alpha'_{2i+1})$. As $\alpha'_{2i+1} = g'(\alpha'_{2i})$, α_{2i+1} ends in Q^L_Θ or Q^S_Θ, or $\alpha_{2i+1} = \lambda$. Thus, $\tilde{\alpha}'_{2i+1}$ and α'_{2i+1} end in the same state, or $\tilde{\alpha}'_{2i+1} = \lambda = \alpha'_{2i+1}$. Due to Lemma 3.4, we have $\tilde{\alpha}'_{2i+1} = \alpha'_{2i+1}$. Hence,

$$g'(\alpha'_{2i}) = \alpha'_{2i+1} \sqsubseteq \alpha'_{2i+2} \quad (i \in \mathbb{N}).$$

Let x' be the path of Σ' which, for each $i \in \mathbb{N}$, is compatible to α'_i. Thus, $\pi(x') = x$. As $x' \in Z^{P,g'}_\kappa$ and, for infinitely many $i \in \mathbb{N}$, $g'(x'[0,i]) \sqsubseteq x'$, we have $x' \in Z^{P,f'}_\kappa$ and $x' \vDash \Box(\theta \Rightarrow \Diamond\psi)$. With Lemma 3.5, Equation (4.1) follows.

4.4. Witnesses in the Case of $\mathcal{T}_{\mathsf{X}}$

In the case $\mathcal{T} = \mathcal{T}_{\mathsf{X}}$, we define f as follows. Let $\alpha \in \mathit{path}(\Sigma)\!\downarrow$, and choose $\alpha' \in \mathit{path}(\Sigma')\!\downarrow$ such that $\pi(\alpha') = \alpha$. By Lemma 3.6, such an α' exists. We set $f(\alpha) := \pi(f'(\alpha')q')$, where $q' \in Q'$ such that $f'(\alpha')q' \in \mathit{path}(\Sigma')\!\downarrow$. Clearly, f is a progressive

strategy in Σ. If f' is positional, then f is positional, too. Note that, given $f'(\alpha')$, $f(\alpha)$ can be computed easily.

We show (4.1). Let $x \in Z_\kappa^{P,f}$. For each $i \in \mathbb{N}$, we define $\alpha_i \in \mathit{path}(\Sigma)\downarrow$ and $q_i \in Q$ such that

$$\alpha_{2i} \sqsubset f(\alpha_{2i}) = \alpha_{2i+1}q_i \sqsubseteq \alpha_{2i+2} \sqsubseteq x \quad (i \in \mathbb{N}),$$

where $\alpha_0 = \lambda$ if P controls the zeroth move in a play of $\mathcal{G}$. In the same way as in the construction of f, we choose $\alpha'_i \in \mathit{path}(\Sigma')\downarrow$ $(i \in \mathbb{N})$ such that, for each $i \in \mathbb{N}$, $\pi(\alpha'_i) = \alpha_i$, $f'(\alpha'_{2i}) = \alpha'_{2i+1}$ and $\alpha'_{2i+1}q'_i \in \mathit{path}(\Sigma')\downarrow$ for some $q'_i \in Q'$ with $\pi(q'_i) = q_i$.

Let $\tilde{\alpha}'_{2i+1}$ be the prefix of α'_{2i+2} of length $|\alpha'_{2i+1}|$, where $i \in \mathbb{N}$. As $\alpha_{2i+1}q_i \sqsubseteq \alpha_{2i+2}$, we have $\tilde{\alpha}'_{2i+1}q'_i \in \mathit{path}(\Sigma')\downarrow$ for some $q'_i \in Q'$ with $\pi(q'_i) = q_i$. By the construction of $\rightarrow'$, both $\alpha'_{2i+1}q'_i, \tilde{\alpha}'_{2i+1}q'_i \in \mathit{path}(\Sigma')\downarrow$ for the same $q'_i \in Q'$ with $\pi(q'_i) = q_i$. Because $\pi(\tilde{\alpha}'_{2i+1}q'_i) = \alpha_{2i+1}q_i = \pi(\alpha'_{2i+1}q'_i)$, we can apply Lemma 3.6 and conclude that $\tilde{\alpha}'_{2i+1} = \alpha'_{2i+1}$. Hence,

$$f'(\alpha'_{2i}) = \alpha'_{2i+1} \sqsubseteq \alpha'_{2i+2} \quad (i \in \mathbb{N}).$$

Let $x' \in \mathit{path}(\Sigma')$ be the sequence that, for each $i \in \mathbb{N}$, is compatible to α'_i. We have $x' \in Z_\kappa^{P,f'}$ and $\pi(x') = x$. Thus, with Lemma 3.7, Equation (4.1) follows.

4.5. Witnesses in the Case of $\mathcal{T}_{\mathsf{S}}$

In the case $\mathcal{T} = \mathcal{T}_{\mathsf{S}}$, we define f as follows. Let $\alpha \in \mathit{path}(\Sigma)\downarrow$, and choose $\alpha' \in \mathit{path}(\Sigma')\downarrow$ such that $\pi(\alpha') = \alpha$. Due to Lemma 3.8, such an α' exists and is uniquely determined. We define $f(\alpha) := \pi(f'(\alpha'))$. Given $f'(\alpha')$, $f(\alpha)$ can be computed easily. Note that, even if f' is positional, f is, in general, not positional, as the last state of α' possibly does not only depend on the last state of α.

We show (4.1). Let $x \in Z_\kappa^{P,f}$. With Lemma 3.8, we define x' as the unique path of Σ' with $\pi(x') = x$. We show that $x' \in Z_\kappa^{P,f'}$. Let $\alpha \in \mathit{path}(\Sigma)\downarrow$ such that $f(\alpha) \sqsubseteq x$. We choose $\alpha' \in \mathit{path}(\Sigma)\downarrow$ such that $\pi(\alpha') = \alpha$ and $\beta' \sqsubseteq x'$ with $|\beta'| = |f'(\alpha')|$. As $\pi(\beta') \sqsubseteq \pi(x') = x$ and $\pi(f'(\alpha')) = f(\alpha) \sqsubseteq x$, we have $\pi(\beta') = \pi(f'(\alpha'))$. Due to Lemma 3.8, $f'(\alpha') = \beta' \sqsubseteq x'$.

Therefore, for each $\alpha \in \mathit{path}(\Sigma)\downarrow$ with $f(\alpha) \sqsubseteq x$, we have $f'(\alpha') \sqsubseteq x'$, where α' is the unique path fragment of Σ' with $\pi(\alpha') = \alpha$. Thus, $x' \in Z_\kappa^{P,f'}$. With Lemma 3.9, we conclude that (4.1) holds.

4.6. Witnesses in the Case of $\mathcal{T}_{\mathsf{Y}}$

In the case $\mathcal{T} = \mathcal{T}_{\mathsf{Y}}$, we define f exactly as in the case $\mathcal{T} = \mathcal{T}_{\mathsf{S}}$. In the proof of (4.1), we just need to replace Lemma 3.8 by Lemma 3.10 and Lemma 3.9 by Lemma 3.11.

4.7. Witnesses in the Case of $\mathcal{T}_{\overline{L}(\Box\Diamond)}$

In the case $\mathcal{T} = \mathcal{T}_{\overline{L}(\Box\Diamond)}$, we first define a strategy g' in Σ' as follows. Let $\alpha' \in path(\Sigma')\downarrow$. We define β' by $f'(\alpha') =: \alpha'\beta'$. We choose γ' such that $\alpha'\beta'\gamma' \in path(\Sigma')\downarrow$ and the following conditions hold:

- If $\alpha'\beta'$ ends in $(Q^M_\Theta, \{\Theta, \neg\Theta\})$, then $\alpha'\beta'\gamma'$ ends in $(Q^L_\Theta, \Theta) \cup (Q^S_\Theta, \neg\Theta)$.
- If α' ends in $(Q^L_\Theta, \Theta) \cup (Q^S_\Theta, \neg\Theta)$ but not in a b. s. c. c. of Σ', then $\alpha'\beta'\gamma'$ ends in a b. s. c. c. of Σ'.
- If α' ends in a b. s. c. c. of Σ', then each state of that b. s. c. c. appears at least once in β', γ' or at the last position of α'.

We set $g'(\alpha') := \alpha'\beta'\gamma'$. Hence, g' is a strategy in Σ'. If f' is positional, then g' is positional, too. Note that, given $f'(\alpha')$, $g'(\alpha')$ can be computed easily.

A path x' with $g'(x'[0,i]) \sqsubseteq x'$ for infinitely many $i \in \mathbb{N}$ is state fair w. r. t. Σ'. This is due to the fact that x' eventually takes some state of a b. s. c. c. of Σ and visits each state of that b. s. c. c. infinitely often. Moreover, $Z^{P,g'}_\kappa \subseteq Z^{P,f'}_\kappa$.

Now, we define f. For $\alpha \in path(\Sigma)\downarrow$, let $\alpha' \in path(\Sigma')\downarrow$ such that $\pi(\alpha') = \alpha$. Such an α' exists due to Lemma 3.18. If $\pi(g'(\alpha')) = \alpha$, we choose $q \in Q$ such that $\pi(g'(\alpha'))q \in path(\Sigma)\downarrow$, and set $f(\alpha) := \pi(g'(\alpha'))q$. Otherwise, we define $f(\alpha) := \pi(g'(\alpha'))$. Clearly, f is a progressive strategy in Σ. If f' is positional, then f is positional, too.

The proof of (4.1) is similar as in the case of $\mathcal{T} = \mathcal{T}_{\mathsf{U}}$.

4.8. Complexity of Computing a Witness

In the following complexity analysis, we assume that the algorithm of Chapter 3 has already been executed and that the system, specification and feasible formula of each transformation step are still available. We analyse the *additional* time the methods described above require to compute a move in the interactive play. Recall that n is the number of transformation steps carried out by the algorithm on input (Σ, Φ).

Theorem 4.2. *Let Σ be a system and Φ a specification. Suppose that f is part of the witness for Σ and Φ as defined above. Then,*

- *given a path fragment α of Σ, the computation of $f(\alpha)$ can be accomplished in $\mathcal{O}(2^n|\Sigma| + n|\alpha|)$ time steps.*
- *If $\Phi \in L(\mathsf{X}, \mathsf{U})$ and f is positional, $f(q)$ can be determined in $\mathcal{O}(2^n|\Sigma|)$ time steps, where q is a state of Σ or $q = \lambda$.*

Proof. Let $\Sigma_0 := \Sigma$, $\Phi_0 := \Phi$, Σ_i the system, and Φ_i the specification after the i-th transformation step ($1 \leq i \leq n$). Moreover, let f_i be a strategy which is part of the witness for Σ_i and Φ_i as defined above, where $0 \leq i \leq n$. We analyse the time that our methods need to compute $f_0(\alpha)$, where $\alpha \in \mathit{path}(\Sigma)\downarrow$. We recall that, for $i \in \{0, \ldots, n-1\}$, the algorithm computes $f_i(\alpha_i)$ from $f_{i+1}(\alpha_{i+1})$, for some $\alpha_i \in \mathit{path}(\Sigma_i)\downarrow$ and $\alpha_{i+1} \in \mathit{path}(\Sigma_{i+1})\downarrow$ with $\pi(\alpha_{i+1}) = \alpha_i$. Here,

$$f_i(\alpha_i) := \pi(f_{i+1}(\alpha_{i+1})\delta_{i+1})q_i,$$

where δ_{i+1} is a finite sequence of states of Σ_{i+1} and q_i is a state of Σ_i or λ. The computation of α_{i+1} needs $\mathcal{O}(|\alpha_i|) = \mathcal{O}(|\alpha|)$ steps, the sequence δ_{i+1} can be determined in $\mathcal{O}(|\Sigma_{i+1}|)$, and q_i is constructed in $\mathcal{O}(1)$. A careful examination yields that the projection can be performed with no additional cost. The preprocessing, i. e. the computation of $f_n(\alpha_n)$, where $\alpha_n \in \mathit{path}(\Sigma_n)\downarrow$, can be done in $\mathcal{O}(|\Sigma_n|)$. (If $\alpha_n \neq \lambda$, the preprocessing can even be accomplished in $\mathcal{O}(1)$.) Thus, the time for the computation of $f_0(\alpha)$ is

$$\mathcal{O}(|\Sigma_n| + \sum_{i=0}^{n-1}(|\alpha| + |\Sigma_{i+1}| + 1)) \leq \mathcal{O}(2^n|\Sigma| + n|\alpha|).$$

If $\Phi \in L(\mathsf{X}, \mathsf{U})$ and f is positional, it can be shown in a similar way that the computation of $f_0(q)$, for some state q of Σ, can be accomplished in $\mathcal{O}(2^n|\Sigma|)$.

□

5. Conclusion

We examined an algorithm, originally due to Courcoubetis and Yannakakis [1995], for generalised fair model checking, a "more permissive" variation of traditional model checking. This algorithm is based on transformations that preserve how much the specification is satisfied in the system (large, medium-sized or small). The modularised structure of our proof of correctness has made it easier to develop new transformations, both for optimisation and for processing specifications with past operators. The game-theoretic view allows to define witnesses, which support the developer in locating and removing errors in the system. A witness consists of one or two strategies in the system which allow to satisfy or violate the specification, depending on how much the specification is satisfied in the system. We extended the algorithm such that the developer can interactively play against a player which plays according such a strategy. It remains to perform some case studies to figure out if such an interactive play is useful in practice.

In our opinion, the algorithm of Courcoubetis and Yannakakis [1995] is suitable for implementation, as it is based on a simple idea, which can be implemented with standard algorithms and data structures. Its running time can be improved by several techniques. One of these techniques is to develop new transformations such that formulae which frequently arise in specifications (e. g. Muller formulae) become feasible.

Another approach for improving running time is based on the observation that the algorithm splits up, but never joins states. Suppose, for instance, that the specification Φ contains $\xi \Rightarrow \Diamond \psi$ as a subformula, where ξ, ψ are state formulae. At some time, the algorithm splits up each state $q \in Q^M_{\Diamond \psi}$, regardless whether $q \vDash \xi$. If $q \nvDash \xi$, then both $(q, \Diamond \psi)$ and $(q, \neg \Diamond \psi)$ satisfy $\xi \Rightarrow \Diamond \psi$. Thus, in some occasions, $(q, \Diamond \psi)$ and $(q, \neg \Diamond \psi)$ may be joined to a single state. In Appendix B, we formalise a notion of *equivalence of states* and make a suggestion on how to join equivalent states; however, it remains to be worked out how to eliminate equivalent states efficiently. There might be related methods in the field of traditional model checking.

Note that, in some occasions, the transformation which the algorithm applies next to the current system and the current specification is not uniquely determined. Unfortunately, the running time depends on these choices. In our opinion, it seems to be convenient to keep the system as long as possible as small as possible, as the running time for a single transformation step mainly depends on the size of the current system and feasible formula and not on their structure. However, it is hard to predict how much the system grows if a certain transformation is applied. Thus,

it remains to develop good heuristics for this problem.

In the literature, there have been discussed other notions of largeness such that the family of large linear-time properties is closed under superset and finite intersection. As these are the fundamental properties of largeness which we need in Section 3.1, it seems to be easy to modify the algorithm for these notions of largeness. However, we can show by simple counterexamples that, for a number of alternative largeness notions, the approach of Section 3.1 cannot be used to check largeness. See Appendix C for a sketch of a proof.

Acknowledgement: I thank Hagen Völzer for numerous helpful discussions and useful comments. I thank Rüdiger Reischuk and Walter Dosch for the willingness to review this thesis.

Bibliography

Proceedings, Symposium on Logic in Computer Science, 16-18, June 1986, Cambridge, Massachusetts, USA, 1986. IEEE Computer Society.

R. Alur and T. A. Henzinger. Local liveness for compositional modeling of fair reactive systems. In P. Wolper, editor, *CAV*, volume 939 of *Lecture Notes in Computer Science*, pages 166–179. Springer, 1995.

D. Berwanger, E. Grädel, and S. Kreutzer. Once upon a time in the west - determinacy, definability, and complexity of path games. In Vardi and Voronkov [2003], pages 229–243.

D. Bustan, S. Rubin, and M. Y. Vardi. Verifying ω-regular properties of Markov chains. In *CAV*, pages 189–201, 2004.

C. Courcoubetis and M. Yannakakis. The complexity of probabilistic verification. *Journal of the ACM*, 42(4):857–907, 1995.

J.-M. Couvreur, N. Saheb, and G. Sutre. An optimal automata approach to LTL model checking of probabilistic systems. In Vardi and Voronkov [2003], pages 361–375.

E. A. Emerson. Temporal and modal logic. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, chapter 16, pages 995–1072. Elsevier Science, 1990.

F. E. Fich and E. Ruppert. Hundreds of impossibility results for distributed computing. *Distributed Computing*, 16(2-3):121–163, 2003.

E. Grädel. Positional determinacy of infinite games. In V. Diekert and M. Habib, editors, *STACS*, volume 2996 of *Lecture Notes in Computer Science*, pages 4–18. Springer, 2004.

U. Krengel. *Einführung in die Wahrscheinlichkeitstheorie und Statistik*. Vieweg, 6. edition, 2002.

O. Lichtenstein and A. Pnueli. Checking that finite state concurrent programs satisfy their linear specification. In *Proceedings of the 12th Annual ACM Symposium on Principles of Programming Languages (New Orleans, La., Jan. 14-16)*, pages 97–107. ACM, New York, 1985.

Z. Manna and A. Pnueli. A hierarchy of temporal properties. In *Proceedings of the 9th Annual ACM Symposium on Principles of Distributed Computing*, pages 377–408. ACM, 1990.

J. C. Oxtoby. *Measure and Category. A Survey of the Analogies between Topological and Measure Spaces*. Springer-Verlag, 1971.

M. Pistore and M. Y. Vardi. The planning spectrum - one, two, three, infinity. In *LICS*, pages 234–243. IEEE Computer Society, 2003.

A. Pnueli and L. D. Zuck. Probabilistic verification by tableaux. In *LICS* [1986], pages 322–331.

A. Pnueli and L. D. Zuck. Probabilistic verification. *Information and Control*, 103 (1):1–29, 1993.

K. R. Reischuk. *Grundlagen: Machinenmodelle, Zeit- und Platzkomplexität, Nichtdeterminismus*, volume 1 of *Komplexitätstheorie*. Teubner, 1999.

Ph. Schnoebelen. The complexity of temporal logic model checking. In *Selected Papers from the 4th Workshop on Advances in Modal Logics (AiML'02)*, pages 393–436, 2003. Invited paper.

A. P. Sistla and E. M. Clarke. The complexity of propositional linear temporal logics. *Journal of the ACM*, 32(3):733–749, 1985.

A. P. Sistla and L. D. Zuck. Reasoning in a restricted temporal logic. *Information and Computation*, 102(2):167–195, 1993.

W. Thomas. Automata on infinite objects. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B: Formal Models and Semantics, chapter 4, pages 133–192. Elsevier, 1990.

D. Varacca and H. Völzer. Temporal logics and model checking for fairly correct systems. In *LICS*, pages 389–398. IEEE Computer Society, 2006.

M. Y. Vardi. Automatic verification of probabilistic concurrent finite-state programs. In *Proceedings of 26th FOCS*, pages 327–338, 1985.

M. Y. Vardi. Branching vs. linear time: Final showdown. In *TACAS*, volume 2031 of *Lecture Notes in Computer Science*, pages 1–22, 2001.

M. Y. Vardi and A. Voronkov, editors. *Logic for Programming, Artificial Intelligence, and Reasoning, 10th International Conference, LPAR 2003, Almaty, Kazakhstan, Proceedings*, volume 2850 of *Lecture Notes in Computer Science*, 2003. Springer.

M. Y. Vardi and P. Wolper. An automata-theoretic approach to automatic program verification. In *LICS* [1986], pages 332–344.

H. Völzer, D. Varacca, and E. Kindler. Defining fairness. In M. Abadi and L. de Alfaro, editors, *CONCUR*, volume 3653 of *Lecture Notes in Computer Science*, pages 458–472. Springer-Verlag, 2005.

P. Wolper. Temporal logic can be more expressive. *Information and Control*, 56 (1/2):72–99, 1983.

P. Wolper, M. Y. Vardi, and A. P. Sistla. Reasoning about infinite computation paths. In *Proceedings of the 24th IEEE Symposium on Foundations of Computer Science*, pages 185–194. IEEE, New York, 1983.

L. D. Zuck, A. Pnueli, and Y. Kesten. Automatic verification of probabilistic free choice. In A. Cortesi, editor, *VMCAI*, volume 2294 of *Lecture Notes in Computer Science*, pages 208–224. Springer, 2002.

A. Can the Application of $\mathcal{T}_{\overline{L}(\Box\Diamond)}$ Increase Running Time?

Let $\Phi \in L(\mathsf{X}, \mathsf{Y}, \mathsf{U}, \mathsf{S})$ be a specification and Σ a reduced system. We choose a correct transformation $\mathcal{T}_i$ and a formula Θ_i which is feasible for $\mathcal{T}_i$, where $1 \leq i \leq n$. For $1 \leq i \leq n$, we define $\Sigma_0 := (Q_0, S_0, \rightarrow_0, v_0) := \Sigma$, $\Sigma_i := (Q_i, S_i, \rightarrow_i, v_i) := \mathcal{T}_i(\Sigma_{i-1}, \Theta_i)$, $\Phi_0 := \Phi$ and $\Phi_i := \mathcal{T}_i(\Phi_{i-1}, \Theta_i)$. The fresh atomic proposition introduced for Θ_i is denoted by θ_i, where $1 \leq i \leq n$. We assume, without loss of generality, that $\theta_1, \ldots, \theta_n$ are pairwise different.

We call $\{\Sigma_i\}_{i=0}^n$ the *system sequence* generated by $\{(\mathcal{T}_i, \Theta_i, \theta_i)\}_{i=1}^n$ on Σ and, $\{\Phi_i\}_{i=0}^n$ the *specification sequence* generated by $\{(\mathcal{T}_i, \Theta_i, \theta_i)\}_{i=1}^n$ on Φ. We say that $\{(\mathcal{T}_i, \Theta_i, \theta_i)\}_{i=1}^n$ is a *transformation sequence for* Σ iff, for each $i \in \{1, \ldots, n\}$ and each $q \in Q_i$,

- $\theta_i \not\preceq \Theta_j \quad (1 \leq j \leq i)$,
- $q \not\preceq \Theta_j \quad (1 \leq j \leq n)$,
- $\theta_i, q \notin v_j(p) \quad (p \in Q_j, 0 \leq j < i)$.

Roughly speaking, the fresh atomic proposition and the new states generated in some transformation step have never been used before. The new states generated in some transformation step may not appear in any feasible formula, even not in those of future transformation steps. Note that states of the original system Σ_0 may appear in each of the feasible formulae.

If $\{(\mathcal{T}_i, \Theta_i, \theta_i)\}_{i=1}^n$ is a transformation sequence for Σ, Φ_n is a state formula and, for each $i \in \{1, \ldots, n\}$ and each $q \in Q_i$,

- $\theta_i \not\preceq \Phi_j \quad (0 \leq j < i)$ and
- $q \not\preceq \Phi_j \quad (0 \leq j \leq n)$,

then $\{(\mathcal{T}_i, \Theta_i, \theta_i)\}_{i=1}^n$ is a *transformation sequence for* Σ *and* Φ. In other words, we do not allow that fresh atomic propositions are used before they are generated. New states may not appear in any of the specifications.

If the algorithm, on input of Σ and Φ, in the i-th step ($1 \leq i \leq n$) chooses the transformation $\mathcal{T}_i$, the feasible formula Θ_i and the fresh atomic proposition θ_i, we say that it *applies* $\{(\mathcal{T}_i, \Theta_i, \theta_i)\}_{i=1}^n$ to Σ and Φ. Roughly speaking, during the application of a transformation sequence, there are no naming conflicts due to fresh atomic propositions or new states. Thus, it is natural to assume that $\{(\mathcal{T}_i, \Theta_i, \theta_i)\}_{i=1}^n$ is a transformation sequence for Σ and Φ.

Definition A.1. Let Σ be a reduced system. We define $\equiv$ as the minimal equivalence relation over transformation sequences for Σ such that the following conditions hold:
Suppose that $\{(\mathcal{T}_i, \Theta_i, \theta_i)\}_{i=1}^n \equiv \{(\hat{\mathcal{T}}_i, \hat{\Theta}_i, \hat{\theta}_i)\}_{i=1}^n$.

1. If both $\{(\mathcal{T}_i, \Theta_i, \theta_i)\}_{i=1}^{n+1}$ and $\{(\hat{\mathcal{T}}_i, \hat{\Theta}_i, \hat{\theta}_i)\}_{i=1}^{n+1}$ are transformation sequences for Σ and
$$(\mathcal{T}_{n+1}, \Theta_{n+1}, \theta_{n+1}) = (\hat{\mathcal{T}}_{n+1}, \hat{\Theta}_{n+1}, \hat{\theta}_{n+1}),$$
then
$$\{(\mathcal{T}_i, \Theta_i, \theta_i)\}_{i=1}^{n+1} \equiv \{(\hat{\mathcal{T}}_i, \hat{\Theta}_i, \hat{\theta}_i)\}_{i=1}^{n+1}.$$

2. If both $\{(\mathcal{T}_i, \Theta_i, \theta_i)\}_{i=1}^{n+1}$ and $\{(\hat{\mathcal{T}}_i, \hat{\Theta}_i, \hat{\theta}_i)\}_{i=1}^{n+1}$ are transformation sequences for Σ and
$$(\mathcal{T}_{n+1}, [\Theta_{n+1}]_{\hat{\theta}_i}^{\hat{\Theta}_i}, \theta_{n+1}) = (\hat{\mathcal{T}}_{n+1}, \hat{\Theta}_{n+1}, \hat{\theta}_{n+1}),$$
where $1 \leq i \leq n$, then
$$\{(\mathcal{T}_i, \Theta_i, \theta_i)\}_{i=1}^{n+1} \equiv \{(\hat{\mathcal{T}}_i, \hat{\Theta}_i, \hat{\theta}_i)\}_{i=1}^{n+1}.$$

3. If both $\{(\mathcal{T}_i, \Theta_i, \theta_i)\}_{i=1}^{n+2}$ and $\{(\hat{\mathcal{T}}_i, \hat{\Theta}_i, \hat{\theta}_i)\}_{i=1}^{n+2}$ are transformation sequences for Σ,
$$(\mathcal{T}_{n+1}, \Theta_{n+1}, \theta_{n+1}) = (\hat{\mathcal{T}}_{n+2}, \hat{\Theta}_{n+2}, \hat{\theta}_{n+2}) \quad \text{and}$$
$$(\mathcal{T}_{n+2}, \Theta_{n+2}, \theta_{n+2}) = (\hat{\mathcal{T}}_{n+1}, \hat{\Theta}_{n+1}, \hat{\theta}_{n+1}),$$
then
$$\{(\mathcal{T}_i, \Theta_i, \theta_i)\}_{i=1}^{n+2} \equiv \{(\hat{\mathcal{T}}_i, \hat{\Theta}_i, \hat{\theta}_i)\}_{i=1}^{n+2}.$$

If $\{(\mathcal{T}_i, \Theta_i, \theta_i)\}_{i=1}^n \equiv \{(\hat{\mathcal{T}}_i, \hat{\Theta}_i, \hat{\theta}_i)\}_{i=1}^n$, then $\{(\mathcal{T}_i, \Theta_i, \theta_i)\}_{i=1}^n$ and $\{(\hat{\mathcal{T}}_i, \hat{\Theta}_i, \hat{\theta}_i)\}_{i=1}^n$ are said to be *similar*.

Similar transformation sequences for a system do essentially the same if they are applied to that system. The relation $\equiv$ allows to reorder the elements of a transformation sequence and to replace, in some feasible formula of that transformation sequence, a fresh atomic proposition that was introduced earlier by the related feasible formula.

Proposition A.2. *Let Σ be a reduced system, $\{(\mathcal{T}_i, \Theta_i, \theta_i)\}_{i=1}^n$ and $\{(\hat{\mathcal{T}}_i, \hat{\Theta}_i, \hat{\theta}_i)\}_{i=1}^n$ similar transformation sequences for Σ, $\{\Sigma_i\}_{i=0}^n$ the system sequence generated by $\{(\mathcal{T}_i, \Theta_i, \theta_i)\}_{i=1}^n$ on Σ and $\{\hat{\Sigma}_i\}_{i=0}^n$ the system sequence generated by $\{(\hat{\mathcal{T}}_i, \hat{\Theta}_i, \hat{\theta}_i)\}_{i=1}^n$ on Σ.*

Then, Σ_n and $\hat{\Sigma}_n$ are isomorphic.

Sketch of Proof. The proof is manly based on arguments of Lemma 3.3 and 3.1. We prove a more general assertion by induction over the structure of $\equiv$:

Let $(Q_i, S_i, \rightarrow_i, v_i) := \Sigma_i$ and $(\hat{Q}_i, \hat{S}_i, \hat{\rightarrow}_i, \hat{v}_i) := \hat{\Sigma}_i$, where $0 \leq i \leq n$.

Isomorphism: there is a bijection $\sigma : Q_n \to \hat{Q}_n$ with $p \to_n q$ iff $\sigma(p) \hat{\to}_n \sigma(q)$, where $p, q \in Q_n$.

Smallness is preserved: let $\Upsilon \in L(\mathsf{X}, \mathsf{Y}, \mathsf{U}, \mathsf{S})$ with $q \not\leq \Upsilon$, for each $q \in Q_i \cup \hat{Q}_i$, where $0 < i < n$. We define $\hat{\Upsilon}$ by replacing q by $\sigma(q)$ in Υ, for each $q \in Q_n$. Then, $Sat(\Upsilon)$ is small in Σ_n iff $Sat(\hat{\Upsilon})$ is small in $\hat{\Sigma}_n$.

Induction base. If $\{(\mathcal{T}_i, \Theta_i, \theta_i)\}_{i=1}^{n} = \{(\hat{\mathcal{T}}_i, \hat{\Theta}_i, \hat{\theta}_i)\}_{i=1}^{n}$, it is clear that the conditions "Isomorphism" and "Smallness is preserved" hold.

Induction step. In the cases that two transformation sequences are similar due to symmetry or transitivity, it is straightforward to prove the induction step. Now, we consider the cases that two transformation sequences are similar due to condition 1, 2 or 3 of the definition of $\equiv$. The case of condition 1 is similar to, and simpler than the case of condition 2 and is therefore omitted. During the remainder of the proof, we assume that the conditions "Isomorphism" and "Smallness is preserved" hold.

Condition 2. We suppose that both $\{(\mathcal{T}_i, \Theta_i, \theta_i)\}_{i=1}^{n+1}$ and $\{(\hat{\mathcal{T}}_i, \hat{\Theta}_i, \hat{\theta}_i)\}_{i=1}^{n+1}$ are transformation sequences for Σ and

$$(\mathcal{T}_{n+1}, [\Theta_{n+1}]_{\hat{\theta}_i}^{\hat{\Theta}_i}, \theta_{n+1}) = (\hat{\mathcal{T}}_{n+1}, \hat{\Theta}_{n+1}, \hat{\theta}_{n+1}),$$

where $1 \leq i \leq n$. Then, due to the definition of $\equiv$,

$$\{(\mathcal{T}_i, \Theta_i, \theta_i)\}_{i=1}^{n+1} \equiv \{(\hat{\mathcal{T}}_i, \hat{\Theta}_i, \hat{\theta}_i)\}_{i=1}^{n+1}.$$

We choose $\{\Sigma_i\}_{i=0}^{n+1}$ as the system sequence generated by $\{(\mathcal{T}_i, \Theta_i, \theta_i)\}_{i=1}^{n+1}$ on Σ and $\{\hat{\Sigma}_i\}_{i=0}^{n+1}$ as the system sequence generated by $\{(\hat{\mathcal{T}}_i, \hat{\Theta}_i, \hat{\theta}_i)\}_{i=1}^{n+1}$ on Σ. Moreover, let $(Q_i, S_i, \to_i, v_i) := \Sigma_i$ and $(\hat{Q}_i, \hat{S}_i, \hat{\to}_i, \hat{v}_i) := \hat{\Sigma}_i$, where $0 \leq i \leq n+1$.

First, we show that there is a bijection $\sigma' : Q_{n+1} \to \hat{Q}_{n+1}$ with $p' \to_{n+1} q'$ iff $\sigma'(p') \hat{\to}_{n+1} \sigma'(q')$, where $p', q' \in Q_{n+1}$. For $(q, \tilde{\Theta}_{n+1}) \in Q_{n+1}$, we set $\sigma'(q, \tilde{\Theta}_{n+1}) := (\sigma(q), \tilde{\hat{\Theta}}_{n+1})$, where $\tilde{\hat{\Theta}}_{n+1} \in \{\hat{\Theta}_{n+1}, \neg\hat{\Theta}_{n+1}\}$ such that $\tilde{\Theta}_{n+1} = \Theta_{n+1}$ iff $\tilde{\hat{\Theta}}_{n+1} = \hat{\Theta}_{n+1}$.

Let $q \in Q_n$, $\tilde{\Theta}_{n+1} \in \{\Theta_{n+1}, \neg\Theta_{n+1}\}$ and $\tilde{\hat{\Theta}}_{n+1} \in \{\hat{\Theta}_{n+1}, \neg\hat{\Theta}_{n+1}\}$ such that $\tilde{\Theta}_{n+1} = \Theta_{n+1}$ iff $\tilde{\hat{\Theta}}_{n+1} = \hat{\Theta}_{n+1}$. Then, the following statements are equivalent:

- $(q, \tilde{\Theta}_{n+1}) \in Q_{n+1}$,
- $Sat(\Diamond(q \wedge \tilde{\Theta}_{n+1}))$ is not small in Σ_{n+1},
- $Sat(\Diamond(q \wedge \tilde{\Theta}_{n+1}))$ is not small in Σ_n,

- $Sat(\Diamond(\sigma(q) \wedge \tilde{\Theta}_{n+1}))$ is not small in $\hat{\Sigma}_n$,
- $Sat(\Diamond(\sigma(q) \wedge \tilde{\hat{\Theta}}_{n+1}))$ is not small in $\hat{\Sigma}_n$,
- $Sat(\Diamond(\sigma(q) \wedge \tilde{\hat{\Theta}}_{n+1}))$ is not small in $\hat{\Sigma}_{n+1}$,
- $(\sigma(q), \tilde{\hat{\Theta}}_{n+1}) \in \hat{Q}_{n+1}$.

Thus, $\sigma' : Q_{n+1} \to \hat{Q}_{n+1}$ is bijective.

Let (p, Θ^1_{n+1}), $(q, \Theta^2_{n+1}) \in Q_{n+1}$, $(\hat{p}, \hat{\Theta}^1_{n+1}) := \sigma'(p, \Theta^1_{n+1})$ and $(\hat{q}, \hat{\Theta}^2_{n+1}) := \sigma(p, \Theta^1_{n+1})$. Then, the following statements are equivalent:

- $(p, \Theta^1_{n+1}) \rightarrow_{n+1} (q, \Theta^2_{n+1})$,
- $Sat(\Diamond(p \wedge \Theta^1_{n+1} \wedge \mathsf{X}(q \wedge \Theta^2_{n+1})))$ is not small in Σ_{n+1},
- $Sat(\Diamond(p \wedge \Theta^1_{n+1} \wedge \mathsf{X}(q \wedge \Theta^2_{n+1})))$ is not small in Σ_n,
- $Sat(\Diamond(\hat{p} \wedge \Theta^1_{n+1} \wedge \mathsf{X}(\hat{q} \wedge \Theta^2_{n+1})))$ is not small in $\hat{\Sigma}_n$,
- $Sat(\Diamond(\hat{p} \wedge \hat{\Theta}^1_{n+1} \wedge \mathsf{X}(\hat{q} \wedge \hat{\Theta}^2_{n+1})))$ is not small in $\hat{\Sigma}_n$,
- $Sat(\Diamond(\hat{p} \wedge \hat{\Theta}^1_{n+1} \wedge \mathsf{X}(\hat{q} \wedge \hat{\Theta}^2_{n+1})))$ is not small in $\hat{\Sigma}_{n+1}$,
- $(\hat{p}, \hat{\Theta}^1_{n+1}) \hat{\rightarrow}_{n+1} (\hat{q}, \hat{\Theta}^2_{n+1})$.

Thus, $p' \rightarrow_{n+1} q'$ iff $\sigma'(p') \hat{\rightarrow}_{n+1} \sigma'(q')$, where $p', q' \in Q_{n+1}$.

Let $\Upsilon' \in L(\mathsf{X}, \mathsf{Y}, \mathsf{U}, \mathsf{S})$ with $q \not\trianglelefteq \Upsilon'$, for each $q \in Q_i \cup \hat{Q}_i$, where $0 < i < n+1$. We define $\hat{\Upsilon}'$ by replacing q by $\sigma'(q)$ in Υ', for each $q \in Q_{n+1}$. We show that $Sat(\Upsilon')$ is small in Σ_{n+1} iff $Sat(\hat{\Upsilon}')$ is small in $\hat{\Sigma}_{n+1}$. For this, we define Υ by replacing θ_{n+1} by Θ_{n+1} and $(q, \tilde{\Theta}_{n+1})$ by $q \wedge \tilde{\Theta}_{n+1}$ in Υ', where $(q, \tilde{\Theta}_{n+1}) \in Q_{n+1}$. In the same way, we define $\hat{\Upsilon}$ by replacing θ_{n+1} by Θ_{n+1} and $(\hat{q}, \tilde{\hat{\Theta}}_{n+1})$ by $\hat{q} \wedge \tilde{\Theta}_{n+1}$ in $\hat{\Upsilon}'$, where $(\hat{q}, \tilde{\hat{\Theta}}_{n+1}) \in \hat{Q}_{n+1}$ and $\tilde{\Theta}_{n+1} \in \{\Theta_{n+1}, \neg\Theta_{n+1}\}$ such that $\tilde{\Theta}_{n+1} = \Theta_{n+1}$ iff $\tilde{\hat{\Theta}}_{n+1} = \hat{\Theta}_{n+1}$. Then, the following statements are equivalent:

- $Sat(\Upsilon')$ is small in Σ_{n+1},
- $Sat(\Upsilon)$ is small in Σ_n,
- $Sat(\hat{\Upsilon})$ is small in $\hat{\Sigma}_n$,
- $Sat(\hat{\Upsilon}')$ is small in $\hat{\Sigma}_{n+1}$.

Condition 3. Suppose that both $\{(\mathcal{T}_i, \Theta_i, \theta_i)\}_{i=1}^{n+2}$ and $\{(\hat{\mathcal{T}}_i, \hat{\Theta}_i, \hat{\theta}_i)\}_{i=1}^{n+2}$ are transformation sequences for Σ,

$$(\mathcal{T}_{n+1}, \Theta_{n+1}, \theta_{n+1}) = (\hat{\mathcal{T}}_{n+2}, \hat{\Theta}_{n+2}, \hat{\theta}_{n+2}) \quad \text{and}$$

$$(\mathcal{T}_{n+2}, \Theta_{n+2}, \theta_{n+2}) = (\hat{\mathcal{T}}_{n+1}, \hat{\Theta}_{n+1}, \hat{\theta}_{n+1}).$$

Then, due to the definition of $\equiv$, we have

$$\{(\mathcal{T}_i, \Theta_i, \theta_i)\}_{i=1}^{n+2} \equiv \{(\hat{\mathcal{T}}_i, \hat{\Theta}_i, \hat{\theta}_i)\}_{i=1}^{n+2}.$$

We choose $\{\Sigma_i\}_{i=0}^{n+2}$ as the system sequence generated by $\{(\mathcal{T}_i, \Theta_i, \theta_i)\}_{i=1}^{n+2}$ on Σ and $\{\hat{\Sigma}_i\}_{i=0}^{n+2}$ as the system sequence generated by $\{(\hat{\mathcal{T}}_i, \hat{\Theta}_i, \hat{\theta}_i)\}_{i=1}^{n+2}$ on Σ. Moreover, let $(Q_i, S_i, \rightarrow_i, v_i) := \Sigma_i$ and $(\hat{Q}_i, \hat{S}_i, \hat{\rightarrow}_i, \hat{v}_i) := \hat{\Sigma}_i$, where $0 \leq i \leq n+2$.

First, we show that there is a bijection $\sigma' : Q_{n+2} \rightarrow \hat{Q}_{n+2}$ with $p' \rightarrow_{n+2} q'$ iff $\sigma'(p') \hat{\rightarrow}_{n+2} \sigma'(q')$, where $p', q' \in Q_{n+2}$. For $(q, \tilde{\Theta}_{n+1}, \tilde{\Theta}_{n+2}) \in Q_{n+2}$, we define $\sigma'(q, \tilde{\Theta}_{n+1}, \tilde{\Theta}_{n+2}) := (\sigma(q), \tilde{\Theta}_{n+2}, \tilde{\Theta}_{n+1})$.

Let $q \in Q_n$, $\tilde{\Theta}_{n+1} \in \{\Theta_{n+1}, \neg\Theta_{n+1}\}$ and $\tilde{\Theta}_{n+2} \in \{\Theta_{n+2}, \neg\Theta_{n+2}\}$. Then, the following statements are equivalent:

- $(q, \tilde{\Theta}_{n+1}, \tilde{\Theta}_{n+2}) \in Q_{n+2}$,
- $Sat(\Diamond(q \wedge \tilde{\Theta}_{n+1} \wedge \tilde{\Theta}_{n+2}))$ is not small in Σ_n,
- $Sat(\Diamond(\sigma(q) \wedge \tilde{\Theta}_{n+2} \wedge \tilde{\Theta}_{n+1}))$ is not small in $\hat{\Sigma}_n$,
- $(\sigma(q), \tilde{\Theta}_{n+2}, \tilde{\Theta}_{n+1}) \in \hat{Q}_{n+2}$.

Thus, $\sigma' : Q_{n+2} \rightarrow \hat{Q}_{n+2}$ is bijective. It can be shown by similar arguments that $p' \rightarrow_{n+2} q'$ iff $\sigma'(p') \hat{\rightarrow}_{n+2} \sigma'(q')$, where $p', q' \in Q_{n+2}$.

Let $\Upsilon' \in L(\mathsf{X}, \mathsf{Y}, \mathsf{U}, \mathsf{S})$ with $q \not\leq \Upsilon'$, for each $q \in Q_i \cup \hat{Q}_i$, where $0 < i < n+2$. We define $\hat{\Upsilon}'$ by replacing q by $\sigma'(q)$ in Υ', for each $q \in Q_{n+2}$. We show that $Sat(\Upsilon')$ is small in Σ_{n+2} iff $Sat(\hat{\Upsilon}')$ is small in $\hat{\Sigma}_{n+2}$. For this, we define Υ by replacing θ_{n+1} by Θ_{n+1}, θ_{n+2} by Θ_{n+2} and $(q, \tilde{\Theta}_{n+1}, \tilde{\Theta}_{n+2})$ by $q \wedge \tilde{\Theta}_{n+1} \wedge \tilde{\Theta}_{n+2}$ in Υ', where $(q, \tilde{\Theta}_{n+1}, \tilde{\Theta}_{n+2}) \in Q_{n+2}$. In the same way, we define $\hat{\Upsilon}$ by replacing θ_{n+1} by Θ_{n+1}, θ_{n+2} by Θ_{n+2} and $(\hat{q}, \tilde{\Theta}_{n+2}, \tilde{\Theta}_{n+1})$ by $\hat{q} \wedge \tilde{\Theta}_{n+2} \wedge \tilde{\Theta}_{n+1}$ in Υ', where $(\hat{q}, \tilde{\Theta}_{n+2}, \tilde{\Theta}_{n+1}) \in \hat{Q}_{n+2}$. Then, the following statements are equivalent:

- $Sat(\Upsilon')$ is small in Σ_{n+2},
- $Sat(\Upsilon)$ is small in Σ_n,
- $Sat(\hat{\Upsilon})$ is small in $\hat{\Sigma}_n$,
- $Sat(\hat{\Upsilon}')$ is small in $\hat{\Sigma}_{n+2}$.

□

Now, we exploit Proposition A.2 to answer the question how much the running time of the algorithm, in the worst case, increases if we use $\mathcal{T}_{\overline{L(\Box\Diamond)}}$ instead of repeatedly applying $\mathcal{T}_{\mathsf{U}}$.

Theorem A.3. *Let Σ be a reduced system, $\Phi \in L(\mathsf{X}, \mathsf{Y}, \mathsf{U}, \mathsf{S})$ a specification and $\{(\mathcal{T}_i, \Theta_i, \theta_i)\}_{i=1}^n$ a transformation sequence for Σ and Φ. Then, there is a transformation sequence $\{(\hat{\mathcal{T}}_i, \hat{\Theta}_i, \hat{\theta}_i)\}_{i=1}^m$ for Σ and Φ with the following properties:*

Let $\{\hat{\Phi}_i\}_{i=0}^m$ be the specification sequence generated by $\{(\hat{\mathcal{T}}_i, \hat{\Theta}_i, \hat{\theta}_i)\}_{i=1}^m$.

1. *For each $k \in \{1, \ldots, m\}$, if $\hat{\theta}_k \not\trianglelefteq \hat{\Phi}_m$, then there is an $i \in \{k+1, \ldots, m\}$ such that $[\hat{\Theta}_i]_{\hat{\theta}_k}^{\hat{\Theta}_k}$ is not feasible.*

2. *Let t be the number of time steps the algorithm needs to apply $\{(\mathcal{T}_i, \Theta_i, \theta_i)\}_{i=1}^n$, and $\hat{t}$ the number of time steps the algorithm needs to apply $\{(\hat{\mathcal{T}}_i, \hat{\Theta}_i, \hat{\theta}_i)\}_{i=1}^m$ on Σ and Φ. Then, $\hat{t} \in \mathcal{O}(|\Phi| \cdot t)$.*

Before the proof, we explain how the theorem answers the original question. Note that $\{(\mathcal{T}_i, \Theta_i, \theta_i)\}_{i=1}^n$ could be the transformation sequence which yields the best running time for the input Σ and Φ; however, the running time of the algorithm when applying $\{(\hat{\mathcal{T}}_i, \hat{\Theta}_i, \hat{\theta}_i)\}_{i=1}^m$ on Σ and Φ is only negligible worse, as Φ tends to be short in practice. Thus, we can still obtain an almost optimal running time if we restrict us to transformations that satisfy condition 1 of the assertion. This is closely connected to the question whether $\mathcal{T}_{\overline{L(\Box\Diamond)}}$ can increase running time, as, roughly speaking, condition 1 allows exactly the transformation sequences that apply $\mathcal{T}_{\overline{L(\Box\Diamond)}}$ as often as possible instead of repeatedly applying $\mathcal{T}_{\mathsf{U}}$.

Sketch of Proof. We show by induction over n that there exists a transformation sequence $\{(\hat{\mathcal{T}}_i, \hat{\Theta}_i, \hat{\theta}_i)\}_{i=1}^m$ for Σ and Φ such that 1. holds and $\sum_{i=0}^m |\hat{\Sigma}_i| \leq \sum_{i=0}^n |\Sigma_i|$. Here, $\{\hat{\Sigma}_i\}_{i=0}^m$ is the system sequence, and $\{\hat{\Phi}_i\}_{i=0}^m$ the specification sequence generated by $\{(\hat{\mathcal{T}}_i, \hat{\Theta}_i, \hat{\theta}_i)\}_{i=1}^m$ on Σ and Φ, respectively.

First, we show that the assertion of the theorem follows from this claim. From the proof of Theorem 3.21, we know that

$$\hat{t} \in \mathcal{O}(\sum_{i=0}^{m} |\hat{\Sigma}_i| \cdot |\Phi|).$$

With the claim above,

$$\hat{t} \in \mathcal{O}(|\Phi| \cdot \sum_{i=0}^{n} |\Sigma_i|).$$

As the application of $\{(\mathcal{T}_i, \Theta_i, \theta_i)\}_{i=1}^{n}$ on Σ and Φ takes at least $\Omega(\sum_{i=0}^{n} |\Sigma_i|)$ steps, we have

$$\hat{t} \in \mathcal{O}(|\Phi| \cdot t).$$

Thus, the assertion follows from the claim above.

Now, we prove the claim. For $n = 0$, there is nothing to show. For the induction step $n \to n+1$, let $\{(\mathcal{T}_i, \Theta_i, \theta_i)\}_{i=1}^{n+1}$ be a transformation sequence for Σ and Φ. We define $\{\Sigma_i\}_{i=0}^{n+1}$ as the system sequence and $\{\Phi_i\}_{i=0}^{n+1}$ as the specification sequence generated by $\{(\mathcal{T}_i, \Theta_i, \theta_i)\}_{i=1}^{n+1}$ on Σ and Φ, respectively. If $\{(\hat{\mathcal{T}}_i, \hat{\Theta}_i, \hat{\theta}_i)\}_{i=1}^{n+1} := \{(\mathcal{T}_i, \Theta_i, \theta_i)\}_{i=1}^{n+1}$ satisfies 1., there is nothing to show.

Otherwise, suppose there is some $k \in \{1, \ldots, n+1\}$ such that $\theta_k \not\preceq \Phi_{n+1}$ and $[\Theta_i]_{\theta_k}^{\Theta_k}$ is feasible for each $i \in \{k+1, \ldots, n+1\}$. For $1 \leq i < k$, we define $\tilde{\mathcal{T}}_i := \mathcal{T}_i$, $\tilde{\Theta}_i := \Theta_i$ and $\tilde{\theta}_i := \theta_i$, and, for $k \leq i \leq n$, $\tilde{\Theta}_i := [\Theta_{i+1}]_{\theta_k}^{\Theta_k}$, $\tilde{\theta}_i := \theta_{i+1}$ and $\tilde{\mathcal{T}}_i$ such that $\tilde{\Theta}_i$ is feasible for $\tilde{\mathcal{T}}_i$. By assumption, $\{(\tilde{\mathcal{T}}_i, \tilde{\Theta}_i, \tilde{\theta}_i)\}_{i=1}^{n}$ is a transformation sequence for Σ and Φ.

Let $\{\tilde{\Sigma}_i\}_{i=0}^{n}$ be the system sequence generated by $\{(\tilde{\mathcal{T}}_i, \tilde{\Theta}_i, \tilde{\theta}_i)\}_{i=1}^{n}$ on Σ. Clearly, $\tilde{\Sigma}_i = \Sigma_i$, for $0 \leq i < k$. Let $i \in \{k, \ldots, n\}$. Recall that

$$\Sigma_{i+1} = \mathcal{T}_{i+1}(\ldots \mathcal{T}_1(\Sigma_0, \Theta_1) \ldots, \Theta_{i+1}).$$

Because of 2. of the definition of $\equiv$, we have

$$\{(\mathcal{T}_j, \Theta_j, \theta_j)\}_{j=1}^{i+1} \quad \equiv$$

$$(\mathcal{T}_1, \Theta_1, \theta_1) \ldots (\mathcal{T}_{k-1}, \Theta_{k-1}, \theta_{k-1})(\mathcal{T}_k, \Theta_k, \theta_k)(\tilde{\mathcal{T}}_k, \tilde{\Theta}_k, \tilde{\theta}_k) \ldots (\tilde{\mathcal{T}}_i, \tilde{\Theta}_i, \tilde{\theta}_i).$$

With 3. of the definition of $\equiv$, we conclude that

$$\{(\mathcal{T}_j, \Theta_j, \theta_j)\}_{j=1}^{i+1} \quad \equiv$$

$$(\mathcal{T}_1, \Theta_1, \theta_1) \ldots (\mathcal{T}_{k-1}, \Theta_{k-1}, \theta_{k-1})(\tilde{\mathcal{T}}_k, \tilde{\Theta}_k, \tilde{\theta}_k) \ldots (\tilde{\mathcal{T}}_i, \tilde{\Theta}_i, \tilde{\theta}_i)(\mathcal{T}_k, \Theta_k, \theta_k) \quad =$$

$$(\tilde{\mathcal{T}}_1, \tilde{\Theta}_1, \tilde{\theta}_1) \ldots (\tilde{\mathcal{T}}_i, \tilde{\Theta}_i, \tilde{\theta}_i)(\mathcal{T}_k, \Theta_k, \theta_k).$$

Because of Proposition A.2,

$$|\Sigma_{i+1}| = |\mathcal{T}_k(\tilde{\mathcal{T}}_i(\ldots \tilde{\mathcal{T}}_1(\Sigma_0, \tilde{\Theta}_1) \ldots, \tilde{\Theta}_i), \Theta_k)|.$$

Thus,

$$|\Sigma_{i+1}| = |\mathcal{T}_k(\tilde{\Sigma}_i, \Theta_k)|.$$

Therefore, $|\tilde{\Sigma}_i| \leq |\Sigma_{i+1}|$. We conclude that $\sum_{i=0}^{n} |\tilde{\Sigma}_i| \leq \sum_{i=0}^{n+1} |\Sigma_i|$. With the induction hypothesis, we construct a transformation sequence $\{(\hat{\mathcal{T}}_i, \hat{\Theta}_i, \hat{\theta}_i)\}_{i=1}^{m}$ such that 1. holds and $\sum_{i=0}^{m} |\hat{\Sigma}_i| \leq \sum_{i=0}^{n} |\tilde{\Sigma}_i|$. The claim follows. □

B. Elimination of Equivalent States

Let $\Sigma = (Q, S, \to, v)$ be a reduced system and Φ a specification. Without loss of generality, we assume that each state formula ξ with $\xi \preceq \Phi$ is an atomic proposition. This can be achieved by replacing the state formulae in Φ by fresh atomic propositions. Furthermore, we assume that atomic propositions that do not occur in the specification have been removed from the valuation function v. (In fact, this is not covered by our definition of systems, as it requires $q \in v(q)$, for each $q \in Q$; however, the necessary modifications are not too difficult.) Then, the states p, q are *equivalent* iff

- $v(p) = v(q)$,
- for each $r \in Q$ we have $r \to p \Leftrightarrow r \to q$,
- $p \in S \Leftrightarrow q \in S$.

Thus, equivalent states satisfy the same atomic propositions, have the same predecessors, and either both or none of them is a starting state. In Figure B.1, there is a system with equivalent states.

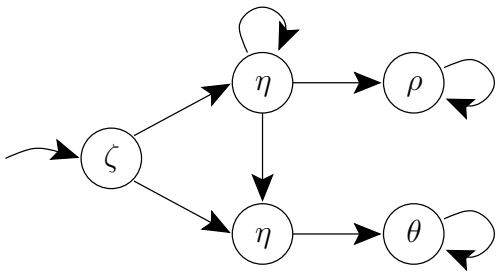

Figure B.1.: A system Σ. The states with the atomic proposition η are equivalent.

If p, q are equivalent, then they can be *joined* to a new state (p, q). This results in a new system $\Sigma' = (Q', S', \to', v')$ such that the following conditions hold:

- $Q' = Q \setminus \{p, q\} \cup \{(p, q)\}$.
- $S' = S$ if $p, q \notin S$; otherwise $S' = S \setminus \{p, q\} \cup \{(p, q)\}$.
- For $r, s \in Q \setminus \{p, q\}$, we have $r \to' s$ iff $r \to s$.

- $r \rightarrow' (p,q)$ iff $r \rightarrow p$ and $r \rightarrow q$, for each $r \in Q \setminus \{p,q\}$.
- $(p,q) \rightarrow' r$ iff $p \rightarrow r$ or $q \rightarrow r$, for each $r \in Q \setminus \{p,q\}$.
- $(p,q) \rightarrow' (p,q)$ iff $p \rightarrow p$ or $q \rightarrow q$.
- $v(p,q) = v(p) = v(q)$.

In Figure B.2, there is the system Σ', where Σ is as in Figure B.1.

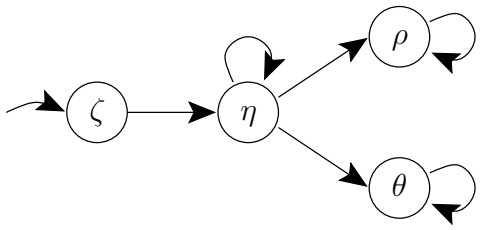

Figure B.2.: The system Σ', where Σ is as in Figure B.1.

It is not too hard to see that joining equivalent states does not affect how much $Sat(\Phi)$ is satisfied. It remains to be worked out how to efficiently join equivalent states during the execution of the algorithm; however, there might already exist related methods in the field of traditional model checking.

C. Alternative Notions of Largeness

Let $\Sigma = (Q, S, \to, v)$ be a reduced system and X a linear-time property over Q. We consider the following notions of largeness:

- The linear-time property X is *AEA-large* in Σ iff Ego has a winning strategy in (AEA, Σ, X) (cf. Pistore and Vardi [2003], Varacca and Völzer [2006]). In a play of the game (AEA, Σ, X), Ego controls only move number one. (We start counting at zero.) See the paper of Berwanger et al. [2003] for the complete lattice of path games.

- The linear-time property X is *A-large* in Σ iff $path(\Sigma) \subseteq X$. The term *A*-large is motivated by the game (A, Σ, X). Checking *A*-largeness is the same as traditional model checking.

- The linear-time property X is *SF-large* in Σ iff $SF_\Sigma \subseteq X$ (cf. Lichtenstein and Pnueli [1985]). Checking *SF*-largeness is the same as model checking constraint by state fairness.

The related terms of medium-sizedness and smallness are defined as in Section 2.6. For instance, X is *SF*-small iff X^c is *SF*-large, and X is *SF*-medium-sized iff it is neither *SF*-large nor *SF*-small.

We try to adapt the methods of Section 3.1 by simply replacing "large", "medium-sized" and "small" by one of the related terms defined above. Unfortunately, for any of these largeness notions and even for simple formulae Θ, it is not possible to develop correct transformations such that Θ becomes feasible. This can be shown as follows. First note that Lemma 3.3 holds for any largeness notion defined above. Suppose $\Sigma = (Q, S, \to, v)$ is the system described in Figure 2.2 on page 23. (That means $Q = \{p, q\}$, $S = Q$ and $\to = (Q, Q)$.)

We start with the case of *AEA*-largeness. Consider the specification $\Phi := \Diamond\Box\, p$. As Ego has neither a winning strategy for $(AEA, \Sigma, Sat(\Phi))$ nor, for $(AEA, \Sigma, Sat(\neg\Phi))$, $Sat(\Phi)$ is *AEA*-medium-sized in Σ. Now, suppose $\Theta := \Box\, p$ is feasible for some correct transformation $\mathcal{T}$ and $\Sigma' := (Q', S', \to', v') := \mathcal{T}(\Sigma, \Theta)$. With arguments from the proof of Lemma 3.3, it is not too hard to see that $Q' = \{(p, \Box\, p), (p, \neg\Box\, p), (q, \neg\Box\, p)\}$, $(p, \neg\Box\, p) \to' (q, \neg\Box\, p)$ and $(q, \neg\Box\, p) \to' (p, \Box\, p)$. As $(p, \Box\, p) = (p, \Theta)$ is reachable from each state of Q', $Sat(\mathcal{T}(\Phi, \Theta)) = Sat(\Diamond\,\theta)$ is *AEA*-large in Σ', a contradiction to the correctness of $\mathcal{T}$.

In the case of *A*-largeness, a counterexample can be constructed with the same system Σ, $\Phi := \Box\neg q \wedge \neg\Box\, p$ and $\Theta := \Box\, p$. Note that $Sat(\Phi)$ is *A*-small in Σ. Then,

$(p, \neg \Box p) \in S'$ and $(p, \neg \Box p) \rightarrow' (p, \neg \Box p)$. As $(p, \neg \Box p)^\omega \in path(\Sigma')$, $Sat(\Box \neg q \wedge \neg \theta)$ is not A-small in Σ'.

In the case of SF-largeness, we consider the same system Σ, $\Phi := \Diamond(p \wedge \mathsf{X} p)$ and $\Theta := \mathsf{X} p$. Although $Sat(\Phi)$ is not SF-large in Σ, $Sat(\Diamond(p \wedge \theta))$ is SF-large in Σ'. This is due to the fact that $(p, \mathsf{X} p)$ is a successor of $(p, \mathsf{X} p)$ and $(q, \mathsf{X} p)$, and $(q, \mathsf{X} p)$, of $(p, \mathsf{X} q)$ and $(q, \mathsf{X} q)$.

We conclude that the approach illustrated in Section 3.1 cannot be used for checking AEA-largeness, A-largeness or SF-largeness; however, Lichtenstein and Pnueli [1985] developed algorithms for checking A-largeness and SF-largeness in exponential time in the size of the specification and linear time in the size of the system. The complexity of checking AEA-largeness is open (cf. Pistore and Vardi [2003], Varacca and Völzer [2006]).