Contents

Chapter 10

Temporal Liveness Requirements

Not all liveness requirements of a reactive module can be formulated as a response verification problem. In Chapter 5, we studied temporal logics over observation structures to specify logical safety requirements of a module. Along the same lines, we now consider temporal logics over observation structures with fairness constraints to specify logical liveness requirements of a fair module.

10.1 Fair Structures

10.1.1 ω-Traces

A trace of an observation structure is obtained by executing the underlying transition graph for finitely many steps, and mapping each state to its observation. Similarly, an ω-trace of an observation structure is obtained by considering an ω-trajectory of the underlying transition graph, and mapping each state to its observation.

ω-TRACES

Let $K = (G, A, \langle\langle \cdot \rangle\rangle)$ be an observation structure. An *ω-trace* of K is an ω-word $\underline{a}$ over the alphabet A of observations such that there is an ω-trajectory $\underline{s}$ of G with $\underline{a} = \langle\langle \underline{s} \rangle\rangle$. The ω-word $\underline{a}$ is an *initialized w-trace* of K if there is an initialized ω-trajectory $\underline{s}$ of G with $\underline{a} = \langle\langle \underline{s} \rangle\rangle$. The ω-language $\mathcal{L}_K$ of the observation structure K is the set of initialized ω-traces of K.

Remark 10.1 [Fusion-closure] The ω-language $\mathcal{L}_K$ of an observation structure K is not necessarily fusion-closed. ■

We know that the ω-language of a transition graph is safe. What about the ω-language of an observation structure? Clearly, if $\underline{a}$ is an ω-trace of an observation structure K, then every prefix of $\underline{a}$ is a trace of K. However, to establish that the set $\mathcal{L}_K$ contains every limit of L_K, we use the fact that the structure is finitely-branching over observations: for every observation a, there are only finitely many initial states with observation a, and every state has only finitely many successors with observation a.

Proposition 10.1 [Limit closure of ω-traces] *Let K be an observation structure. Then, $\mathcal{L}_K = safe(L_K)$.*

Proof. Let $K = (\Sigma, \sigma^I, \rightarrow, A, \langle\!\langle\cdot\rangle\!\rangle)$ be an observation structure. If $\underline{a}$ is an ω-trace then, by definition, for all $i \geq 0$, the prefix $\overline{a}_{0\ldots i}$ is a trace of K. This establishes $\mathcal{L}_K \subseteq safe(L_K)$.

We wish to prove that every limit of L_K is an ω-trace of K. Consider an ω-word $\underline{a}$ over the set of observations of K, and suppose for all $i \geq 0$, $\overline{a}_{0\ldots i}$ is an initialized trace of K. Let us define a transition graph H. The states of H are pairs of the form (s, i), for a state s of K and a natural number i, such that $\langle\!\langle s\rangle\!\rangle = a_i$. The state (s, i) is an initial state of H if s is an initial state of K and $i = 0$. The transition graph H has a transition from the state (s, i) to the state (t, j) if there is a transition from s to t in K and $j = i + 1$. The graph H has finitely many initial states, and each state has finitely many successors. For every $i \geq 0$, $\underline{a}_{0\ldots i}$ is a trace of K, and hence, there exists a trajectory $\overline{s}_{0\ldots i}$ of K with $\langle\!\langle s_j\rangle\!\rangle = a_j$ for $0 \leq j \leq i$. Hence, for every $i \geq 0$, there is a trajectory of H of length i. From König's lemma, the graph H has an infinite path: $(s_0, 0)(s_1, 1)(s_2, 2)\cdots$ The corresponding ω-word $\underline{s}$ is an ω-trajectory of K, and $\langle\!\langle \underline{s}\rangle\!\rangle = \underline{a}$ is an ω-trace of K. ■

Since a safe language is completely characterized by the set of its prefixes, the next theorem follows.

Theorem 10.1 [Language inclusion] *For two observation structures K_1 and K_2, $L_{K_1} \subseteq L_{K_2}$ iff $\mathcal{L}_{K_1} \subseteq \mathcal{L}_{K_2}$. For two reactive modules P_1 and P_2, $L_{P_1} \subseteq L_{P_2}$ iff $\mathcal{L}_{P_1} \subseteq \mathcal{L}_{P_2}$.*

10.1.2 Fair traces

A fairness constraint and a fairness assumption for an observation structure is a fairness constraint and a fairness assumption for the underlying transition graph. Fair structures are obtained from observation structures by adding fairness assumptions.

FAIR STRUCTURE
A *fair structure* $\mathcal{K} = (K, F)$ consists of an observation structure K and a fairness assumption F for K.

If all the fairness constraints in F are weak, then (K, F) is a *weakly-fair* structure.

Fair traces of a fair structure are obtained from fair trajectories by projecting states to observations.

FAIR TRACE Let $\mathcal{K}$ be a fair structure with observations A and observation function $\langle\!\langle\cdot\rangle\!\rangle$. An ω-word $\underline{a}$ is a fair trace of $\mathcal{K}$ if there exists a fair trajectory $\underline{s}$ of $\mathcal{K}$ such that $\underline{a} = \langle\!\langle \underline{s} \rangle\!\rangle$. An ω-word $\underline{a}$ is an initialized fair trace of $\mathcal{K}$ if there exists an initialized fair trajectory $\underline{s}$ of $\mathcal{K}$ such that $\underline{a} = \langle\!\langle \underline{s} \rangle\!\rangle$. The *fair language* $\mathcal{L}_{\mathcal{K}}$ of $\mathcal{K}$ the set of initialized fair traces of $\mathcal{K}$.

Exercise 10.1 {T3} [Fair traces] Show that the set of fair initialized traces of a fair structure is not necessarily the intersection of the set of fair traces and the set of initialized ω-traces. ■

In Chapter 11, we will study fair structures as a specification formalism for fair languages.

The fair structure of a fair module

Every reactive module P defines the observation structure K_P. The fair structure of a fair module $(P, \mathit{WeakF}_P, \mathit{StrongF}_P)$ is obtained from the observation structure K_P by adding all the fairness constraints corresponding to the declaration of weak and strong fair update choices.

FAIR STRUCTURE OF A FAIR MODULE The fair module $\mathcal{P}$ defines the fair structure $\mathcal{K}_{\mathcal{P}} = (K_P, F_{\mathcal{P}})$.

Observe that if $\mathcal{P}$ has only weak-fairness constraints, then the corresponding fair structure $\mathcal{K}_{\mathcal{P}}$ is also weakly fair.

10.2 The Temporal Logic CTL

We specify requirements of fair modules using fair state logics. The formulas of fair state logics are interpreted over the states of fair structures, and may refer to the infinite behavior of fair structures. The satisfaction relation of a fair state logic defines, for each formula ϕ and each fair ϕ-structure $\mathcal{K}$, the characteristic region $[\![\phi]\!]_{\mathcal{K}}$. We start by extending STL to a fair state logic called CTL.

Remark 10.2 [State logics vs. Fair state logics] Every state logic is a fair state logic. For every formula ϕ of a state logic and a fair structure $\mathcal{K} = (K, F)$, the characteristic regions $[\![\phi]\!]_{\mathcal{K}}$ and $[\![\phi]\!]_K$ coincide. Every fair state logic is also a state logic: for a formula ϕ of a fair state logic, an observation structure K, and a state s of K, $s \models_K \phi$ iff $s \models_{(K,\emptyset)} \phi$. ■

10.2.1 Syntax and Semantics of CTL

The fair state logic CTL is obtained from the state logic STL by adding the unary temporal connective *possibly-always*, written $\exists\Box$. Consider a state s of a fair structure $\mathcal{K}$, and let p be an observation predicate of $\mathcal{K}$. The state s satisfies the formula $\exists\Box p$ if there is a source-s fair trajectory all of whose states satisfy p. In other words, the formula $\exists\Box p$ asserts that it is possible to execute infinitely many rounds in a fair fashion so that p is satisfied at every step.

COMPUTATION TREE LOGIC

The formulas of CTL are defined inductively by the grammar

$$\phi \ ::= \ p \mid \phi \vee \phi \mid \neg\phi \mid \exists\bigcirc\, \phi \mid \exists\Box\phi \mid \phi\exists\mathcal{U}\phi$$

for atomic formulas p. For a CTL formula ϕ, if $\mathcal{K} = (K, F)$ is a fair ϕ-structure, and s is a state of K, then

$$s \models_{\mathcal{K}} \exists\Box\phi \quad \text{iff} \quad \begin{array}{l}\text{there is an source-}s\text{ fair trajectory } \underline{s} \text{ of } \mathcal{K} \\ \text{such that for all } i \geq 0,\ s_i \models_{\mathcal{K}} \phi.\end{array}$$

The interpretation of the temporal connectives $\exists\bigcirc$ and $\exists\mathcal{U}$ is the same as in STL: a state s satisfies the possibly-next formula $\exists\bigcirc\, p$ if some successor of s satisfies p; a state s satisfies the possibly-until formula $p\,\exists\mathcal{U}\,q$ if there exists a source-s trajectory $\overline{s}_{0\ldots m}$ such that s_m satisfies q and s_i satisfies p for all $0 \leq i < m$. The temporal connectives $\forall\bigcirc$, $\exists\Diamond$, $\forall\Box$, and $\forall\mathcal{W}$ are defined from $\exists\bigcirc$ and $\exists\mathcal{U}$ as in STL. In addition, we define the following temporal connectives in CTL:

Inevitably	$\forall\Diamond\phi$	for	$\neg\exists\Box\neg\phi$;
Inevitably-until	$\psi\forall\mathcal{U}\phi$	for	$\psi\forall\mathcal{W}\phi \wedge \forall\Diamond\phi$;
Possibly-waiting-for	$\psi\exists\mathcal{W}\phi$	for	$\psi\exists\mathcal{U}\phi \vee \exists\Box\psi$.

The modality $\forall\Diamond$ is the dual of $\exists\Box$: a state s of the fair structure $\mathcal{K}$ satisfies the CTL formula $\forall\Diamond\, p$ if every source-s fair ω-trajectory contains a p-state.

Exercise 10.2 {T2} [Inevitably-until connective] Let ϕ and ψ be two CTL formulas, let $\mathcal{K}$ be a (ϕ, ψ)-structure, and let s be a state of K. Show that $s \models_K \phi\forall\mathcal{U}\psi$ iff for all source-s fair ω-trajectories $\underline{s}$ of $\mathcal{K}$, there exists a position $m \geq 0$ such that (1) $s_m \models_{\mathcal{K}} \psi$ and (2) for all $0 \leq i < m$, $s_i \models_{\mathcal{K}} \phi$. ■

Remark 10.3 [Fair emptiness] The fair-emptiness problem is a special case of CTL model checking: for a fair structure $\mathcal{K}$, the answer to the fair emptiness problem $\mathcal{K}$ is YES iff $s \models_{\mathcal{K}} \exists\Box\mathit{true}$ for some initial state s of $\mathcal{K}$. ■

Response verification problem is also a special case of CTL verification:

Proposition 10.2 [Response verification in CTL] *Let $\mathcal{P}$ be a fair module and let p and q be two observation predicates of $\mathcal{P}$. Then, $p \leadsto_{\mathcal{P}} q$ iff $\mathcal{P} \models \forall\Box(p \rightarrow \forall\Diamond\, q)$.*

Remark 10.4 [Recurrence verification] Recurrence verification is also a special case of CTL verification: The observation predicate p is a recurrent of a fair structure $\mathcal{K}$ if $\mathcal{K} \models \forall\Box\,\forall\Diamond\, p$. ■

Example 10.1 [Mutual exclusion] For a mutual-exclusion protocol with weak fairness, the *deadlock-freedom requirement* asserts that if a process requests the critical section, then some process is eventually in the critical section:

$$\phi_{df} : \forall\Box((pc_1 = \mathit{reqC} \vee pc_2 = \mathit{reqC}) \rightarrow \forall\Diamond(pc_1 = \mathit{inC} \vee pc_2 = \mathit{inC}))$$

The *starvation-freedom requirement* asserts that if a process requests the critical section, then that process eventually enters the critical section:

$$\phi_{sf} : \forall\Box((pc_1 = \mathit{reqC} \rightarrow \forall\Diamond pc_1 = \mathit{inC}) \wedge (pc_2 = \mathit{reqC} \rightarrow \forall\Diamond pc_2 = \mathit{inC})).$$

The fair module *FairPete* satisfies both ϕ_{df} and ϕ_{sf}. It also satisfies the stronger until-requirement:

$$\forall\Box\left(\begin{array}{c} (pc_1 = \mathit{reqC} \rightarrow (pc_1 = \mathit{reqC})\forall\mathcal{U}(pc_1 = \mathit{inC})) \\ \wedge \\ (pc_2 = \mathit{reqC} \rightarrow (pc_2 = \mathit{reqC})\forall\mathcal{U}(pc_2 = \mathit{inC})) \end{array}\right).$$

■

Exercise 10.3 {T2} [CTL connectives] The CTL formula ϕ *implies* the CTL formula ψ if $[\![\phi]\!]_{\mathcal{G}} \subseteq [\![\psi]\!]_{\mathcal{G}}$ for all fair (ϕ, ψ)-structures $\mathcal{G}$ (i.e., the CTL formula $\phi \rightarrow \psi$ is valid). Let p be an atomic state formula. Group the 16 CTL formulas of the form $Q_1 T_1 Q_2 T_2\, p$, where $Q_1, Q_2 \in \{\forall, \exists\}$ and $T_1, T_2 \in \{\Box, \Diamond\}$, into eight pairs (ϕ, ψ) (any such grouping is fine). Prove or disprove that ϕ implies ψ for each of your pairs. ■

Exercise 10.4 {T3} [Interdefinability of temporal connectives] Assuming that a fair state logic contains the temporal connective $\exists\mathcal{U}$, show that each of the connectives $\exists\Box$, $\forall\Diamond$, $\forall\mathcal{U}$, and $\exists\mathcal{W}$ can be used to define the remaining three. ■

Every STL formula is also a CTL formula. For a STL formula ϕ and a fair structure $\mathcal{K} = (K, F)$, $[\![\phi]\!]_{\mathcal{K}} = [\![\phi]\!]_K$. This implies that to check STL specifications of fair structures we can ignore the fairness constraints.

Remark 10.5 [Fair semantics of STL] Note that while interpreting STL formulas over states of fair structures, we have retained the original semantics of STL over observation structures. To account for the fairness constraints, suppose we redefine the semantics of STL over fair structures the following way. The state s of a fair structure $\mathcal{K}$ satisfies the possibly-until formula $\phi \, \exists\mathcal{U} \, \psi$ if there exists a source-s fair trajectory $\underline{s}$ of $\mathcal{K}$ such that for some $m \geq 0$, $s_m \models_{\mathcal{K}} \psi$ and $s_i \models_{\mathcal{K}} \phi$ for $0 \leq i < m$. For machine-closed fair structures $\mathcal{K}$, since every finite trajectory is a prefix of some fair ω-trajectory, this new definition of $\exists\mathcal{U}$ coincides with the old definition. ■

10.2.2 CTL Model Checking

In the model-checking problem for CTL, we are given a CTL formula ϕ and a fair ϕ-structure $\mathcal{K}$. To compute the characteristic region $[\![\phi]\!]_K$, we proceed inductively on the structure of the formula ϕ, by first finding the characteristic regions for the subformulas of ϕ. For this purpose, we first compute the the set $Sub(\phi)$ of subformulas of ϕ. The function Sub is extended to include the new connective $\exists\Box$:

$$Sub(\exists\Box\psi) \;=\; \{\exists\Box\psi\} \cup Sub(\psi).$$

The function *OrderedSub* is also redefined so that it accepts a CTL formula ϕ as input, and returns a queue with the formulas in $Sub(\phi)$ such that if $\psi \in Sub(\chi)$ and $\chi \in Sub(\phi)$, then ψ precedes χ in $OrderedSub(\phi)$. As in case of STL, the CTL formula ϕ has at most $|\phi|$ subformulas.

For the enumerative algorithm, assume that the atomic formulas of ϕ are propositions, and the fair structure $\mathcal{K}$ is finite. The algorithm computes, for each state s of $\mathcal{K}$, the set $\lambda(s) \subseteq Sub(\phi)$ of subformulas of ϕ that are satisfied by the state s. Initially, $\lambda(s)$ is empty for each state s. The algorithm considers each subformula ψ, in the order given by $OrderedSub(\phi)$, and decides, for every state s, whether s satisfies ψ, and updates $\lambda(s)$ accordingly (see Algorithm 5.1 for enumerative STL model checking). The structure of ψ leads to various cases. The cases corresponding to propositions, logical connectives, and the temporal connectives $\exists\bigcirc$ and $\exists\mathcal{U}$ are handled as in the case of STL. The case when $\psi = \exists\Box\chi$ is reduced to the fair-region problem.

The fair-region problem

The fair-region problem is to determine which states belong to the fair ω-trajectories of a fair graph. Let G be a transition graph, and let F be a fairness assumption for G. The F-fair region σ^F of G consists of precisely the states s such that there is a source-s F-fair ω-trajectory of G.

An instance of the *fair-region* problem is a fair graph (G, F). The answer to the fair-region problem (G, F) is the F-fair region σ^F of G.

Remark 10.6 [Fair-region problem vs. fair-emptiness problem] For a fair graph $\mathcal{G}$, the answer to the fair-emptiness problem $\mathcal{G}$ is YES iff $\sigma^I \cap \sigma^F$ is nonempty. ■

For CTL model checking, we need to construct the characteristic region $[\![\exists\Box\chi]\!]_{\mathcal{K}}$ from the characteristic region $[\![\chi]\!]_{\mathcal{K}}$ for the fair structure $\mathcal{K} = (\Sigma, \sigma^I, \rightarrow, A, \langle\!\langle\cdot\rangle\!\rangle, F)$. Let G_χ be the transition graph with the state space $[\![\chi]\!]_{\mathcal{K}}$ and the transition relation $\rightarrow$ restricted to $[\![\chi]\!]_{\mathcal{K}}$. The region $[\![\exists\Box\chi]\!]_{\mathcal{K}}$ is precisely the answer to the fair-region problem (G_χ, F).

To solve the fair-region problem (G, F), observe that a state s belongs to the fair region σ^F iff there exists a F-fair strongly connected component σ of G such that $post^*(s) \cap \sigma$ is nonempty. Thus,

$$\sigma^F \;=\; \bigcup\{\sigma \mid \sigma \text{ is a } F\text{-fair component of } G\}.\, pre^*(\sigma).$$

Hence, σ^F can be by first computing the F-fair strongly connected components of G using Algorithm 9.2, and then computing the region σ^F by a depth-first search. If G has n states and m transitions, and F contains ℓ fairness constraints, then the overall time-complexity is $O((n+m)\cdot \ell^2)$.

Exercise 10.5 {P3} [Fair-region problem] Write an enumerative algorithm to solve the fair-region problem $\mathcal{G}$ using an on-the-fly representation of the fair graph. ■

Theorem 10.2 [Model checking of CTL] *Let $\mathcal{K} = (K, F)$ be a fair structure, and let ϕ be an* CTL *formula. Suppose K has n states and m transitions, and F has ℓ fairness constraints. The model-checking problem $(\mathcal{K}, \phi)$ can be solved in $O((n+m)\cdot \ell^2 \cdot |\phi|)$ time.*

The algorithms of Section 9.3 that employ nested depth-first search can be used to solve the fair-region problem when the fairness assumption is of a restricted form. When the fairness assumption F contains only weak constraints, the CTL model-checking problem $((K, F), \phi)$ can be solved in in time linear in the number of fairness constraints.

Theorem 10.3 [Model checking of CTL for weak-fair structures] *Let $\mathcal{K} = (K, F)$ be a weak-fair structure, and let ϕ be an* CTL *formula. Suppose K has n states and m transitions, and F has ℓ weak-fairness constraints. The model-checking problem $(\mathcal{K}, \phi)$ can be solved in $O((n+m)\cdot \ell \cdot |\phi|)$ time.*

In particular, the CTL model-checking problem (K, ϕ) can be solved in $O((n+m)\cdot|\phi|)$ time. Thus, the additional complexity of CTL model checking over STL model checking is not due to the introduction of $\exists\Box$ connective in the logic, but due to the introduction of fairness constraints in the model.

To solve the CTL-verification problem $(\mathcal{P}, \phi)$, for a finite fair module $\mathcal{P}$ and a CTL specification ϕ, we can first construct the fair structure $\mathcal{K}_{\mathcal{P}}$, and then employ the model checking algorithm. As usual, since the structure $\mathcal{K}_{\mathcal{P}}$ may be exponentially larger than the module description, this results in an exponential algorithm. As in case of STL, the CTL verification problem of determining whether a fair module satisfies a CTL-formula is PSPACE-complete.

10.2.3 Compositionality and CTL

As in STL, satisfaction of existential CTL-formulas is not preserved under parallel composition.

Exercise 10.6 {T3} [Non-compositionality of CTL] Give an example of a fair module $\mathcal{P} \| \mathcal{Q}$ and an observation predicate p such that the answer to the verification problem $(\mathcal{P}, \exists\Box p)$ is YES, while the answer to $(\mathcal{P} \| \mathcal{Q}, \exists\Box p)$ is NO. ■

As in case on STL, if we restrict ourselves only to the universal formulas, then the compositionality principle holds. Let $\forall$CTL be the fragment of CTL generated by the grammar

$$\phi \ ::= \ p \mid \neg p \mid \phi \wedge \phi \mid \phi \vee \phi \mid \forall\bigcirc \phi \mid \phi \forall\mathcal{U} \phi \mid \forall\Box\phi$$

The logic $\forall$CTL is not closed under negation. The parallel composition operation on fair modules ensures that the projection of a fair trajectory of a compound module onto the variables of a component is a fair trajectory of that component. This implies that the compositionality principle holds for $\forall$CTL.

Proposition 10.3 [Compositionality for $\forall$CTL] *If the fair module $\mathcal{P}$ satisfies the $\forall$CTL-formula ϕ, then for every fair module $\mathcal{Q}$ that is compatible with $\mathcal{P}$, the compound fair module $\mathcal{P} \| \mathcal{Q}$ satisfies ϕ.*

Exercise 10.7 {T3} [Compositionality of $\forall$CTL] Prove Proposition 10.3. ■

10.3 The μ-Calculus

We now introduce a state logic, called *μ-calculus*, that is more expressive than CTL. Before we present syntax and semantics of μ-calculus, two points must be noted. First, comprehending μ-calculus formulas requires considerable expertise, and hence, it is not a convenient specification language for writing requirements. On the other hand, its semantics immediately suggests a symbolic procedure for model checking. The role of of μ-calculus, then, is as an intermediate language which can be analyzed by symbolic algorithms. Second, the syntax of μ-calculus is expressive enough to specify fairness constraints. Consequently, we consider μ-calculus as a state logic, and interpret its formulas over states of observation structures.

10.3.1 Syntax and semantics

In μ-calculus, properties are expressed as fixpoints of functions that map regions to regions. As an example, consider the STL-formula $\exists\Diamond p$. The characteristic region $[\![\exists\Diamond p]\!]_K$ consists of all states of the observation structure K from which a state satisfying p is reachable. Consider the function $\mathcal{F}_{\exists\Diamond p}$ that maps regions of K to regions of K:

$$\mathcal{F}_{\exists\Diamond p}(\sigma) \;=\; [\![p]\!]_K \cup \mathit{pre}_K(\sigma).$$

Then, the region $[\![\exists\Diamond\, p]\!]_K$ is the least fixpoint of the function $\mathcal{F}_{\exists\Diamond p}$: it is the smallest region σ that contains $[\![p]\!]_K$ as well as $\mathit{pre}_K(\sigma)$. The μ-calculus formula corresponding to $\exists\Diamond p$ is $\mu\mathbf{x}.\,(p \vee \exists\bigcirc\mathbf{x})$. Here, the variable $\mathbf{x}$ ranges over regions, $\mu\mathbf{x}.$ is called the *least fixpoint operator*, and given a region σ, $\exists\bigcirc\,\sigma$ denotes the region containing states that have at least one successor in σ.

The dual of the least fixpoint operator is the *greatest fixpoint operator* $\nu\mathbf{x}$. As an example, the characteristic region $[\![\forall\Box p]\!]_K$ is the greatest fixpoint of the function $\mathcal{F}_{\forall\Box p}$ that maps regions of K to regions of K:

$$\mathcal{F}_{\forall\Box p}(\sigma) \;=\; [\![p]\!]_K \cap \{s \mid \mathit{post}_K(s) \subseteq \sigma\}.$$

The μ-calculus formula corresponding to $\forall\Box p$ is $\nu\mathbf{x}.\,(p \wedge \forall\bigcirc\,\mathbf{x})$.

μ-CALCULUS SYNTAX

Let **Var** be a set of *region variables*. The formulas of the *μ-calculus* (CTμ) are defined inductively by the grammar

$$\phi \;::=\; p \mid \neg p \mid \phi_1 \wedge \phi_2 \mid \phi_1 \vee \phi_2 \mid \exists\bigcirc\,\phi \mid \forall\bigcirc\,\phi \mid \mu\mathbf{x}.\,\phi \mid \nu\mathbf{x}.\,\phi \mid \mathbf{x},$$

where p is an atomic formula and $\mathbf{x} \in \mathbf{Var}$ is a region variable.

A CTμ formula of the form $\mu\mathbf{x}.\,\phi$ is called a *μ-formula*, and a CTμ formula of the form $\nu\mathbf{x}.\,\phi$ is called a *ν-formula*. A μ-formula or a ν-formula is also called a *fixpoint-formula*. The fixpoint operator is like a quantifier in first-order logic. Every occurrence of a region variable $\mathbf{x}$ in a formula is either free or bound, and if bound, has a unique fixpoint operator that binds it. The CTμ formula ϕ is *closed* if for all region variables $\mathbf{x} \in \mathbf{Var}$, each occurrence of $\mathbf{x}$ in ϕ is bound by a fixpoint operator. The CTμ formula ϕ is *open* if it contains a free occurrence of a region variable.

The logic CTμ is a state logic, and its formulas are interpreted over states of observation structures. As in state logics, for a formula ϕ of CTμ, a ϕ-structure is an observation structure whose observations give interpretation to the atomic formulas appearing in ϕ.

μ-CALCULUS SEMANTICS

Let $K = (\Sigma, \sigma^I, \rightarrow, A, \langle\!\langle\cdot\rangle\!\rangle)$ be an observation structure. A *region environment* $\mathbf{E}$ assigns to each region variable $\mathbf{x} \in \mathbf{Var}$ a region $\sigma \subseteq \Sigma$. Given a state $s \in \Sigma$ and a region environment $\mathbf{E}$,

$$
\begin{array}{lll}
s \models_{K,\mathbf{E}} p & \text{iff} & \langle\!\langle s\rangle\!\rangle \models p; \\
s \models_{K,\mathbf{E}} \neg p & \text{iff} & \langle\!\langle s\rangle\!\rangle \models \neg p; \\
s \models_{K,\mathbf{E}} \phi_1 \wedge \phi_2 & \text{iff} & s \models_{K,\mathbf{E}} \phi_1 \text{ and } s \models_{K,\mathbf{E}} \phi_2; \\
s \models_{K,\mathbf{E}} \phi_1 \vee \phi_2 & \text{iff} & s \models_{K,\mathbf{E}} \phi_1 \text{ or } s \models_{K,\mathbf{E}} \phi_2; \\
s \models_{K,\mathbf{E}} \exists\bigcirc\, \phi & \text{iff} & \text{for some state } t \in \mathit{post}_K(s),\ t \models_{K,\mathbf{E}} \phi; \\
s \models_{K,\mathbf{E}} \forall\bigcirc\, \phi & \text{iff} & \text{for all states } t \in \mathit{post}_K(s),\ t \models_{K,\mathbf{E}} \phi; \\
s \models_{K,\mathbf{E}} \mu\mathbf{x}.\, \phi & \text{iff} & \text{for all fixpoints } \sigma \text{ of } \mathcal{F}^{\mathbf{x},\phi}_{K,\mathbf{E}},\ s \in \sigma \\
s \models_{K,\mathbf{E}} \nu\mathbf{x}.\, \phi & \text{iff} & \text{for some fixpoint } \sigma \text{ of } \mathcal{F}^{\mathbf{x},\phi}_{K,\mathbf{E}},\ s \in \sigma \\
s \models_{K,\mathbf{E}} \mathbf{x} & \text{iff} & s \in \mathbf{E}(\mathbf{x}).
\end{array}
$$

The function $\mathcal{F}^{\mathbf{x},\phi}_{K,\mathbf{E}}$ maps regions to regions: for all regions $\sigma \subseteq \Sigma$ and all states $s \in \Sigma$,

$$s \in \mathcal{F}^{\mathbf{x},\phi}_{K,\mathbf{E}}(\sigma) \text{ iff } s \models_{K,\mathbf{E}[\mathbf{x}:=\sigma]} \phi.$$

From the following proposition it follows by the Knaster-Tarski fixpoint theorem that the function $\mathcal{F}^{\mathbf{x},\phi}_{K,\mathbf{E}}$ has a least fixpoint as well as a greatest fixpoint.

Proposition 10.4 [Monotonicity in μ-calculus] *Let ϕ be a* CTμ *formula and let $\mathbf{E}$ be a region environment. The function $\mathcal{F}^{\mathbf{x},\phi}_{K,\mathbf{E}}$ is monotonic; that is, $\sigma \subseteq \tau$ implies $\mathcal{F}^{\mathbf{x},\phi}_{K,\mathbf{E}}(\sigma) \subseteq \mathcal{F}^{\mathbf{x},\phi}_{K,\mathbf{E}}(\tau)$.*

Exercise 10.8 {T3} [Monotonicity in μ-calculus] Prove Proposition 10.4. ∎

Remark 10.7 [Region environments in μ-calculus] Let ϕ be a CTμ formula. If two region environments $\mathbf{E}$ and $\mathbf{E}'$ agree on the values of the region variables that are free in ϕ, then $s \models_{K,\mathbf{E}} \phi$ iff $s \models_{K,\mathbf{E}'} \phi$. In particular, for a closed formula ϕ, in the definition of the satisfaction relation $\models_{K,\mathbf{E}}$, the value of $\mathbf{E}$ is not important. ∎

A state $s \in \Sigma$ *satisfies* the closed CTμ formula ϕ, written $s \models_K \phi$, if $s \models_{K,\mathbf{E}} \phi$ for all region environments $\mathbf{E}$. For notational convenience, we admit regions as formulas of state logics: for all regions $\sigma \subseteq \Sigma$ and all states $s \in \Sigma$, $s \models_K \sigma$ iff $s \in \sigma$. Given a region environment $\mathbf{E}$, a CTμ formula of the form $\mu\mathbf{x}.\,\phi$, then, defines the least fixpoint of the function $\mathcal{F}^{\phi}_{K,\mathbf{E}} \colon \mathbf{2}^\Sigma \rightarrow \mathbf{2}^\Sigma$ that maps each region $\sigma \subseteq \Sigma$ to the region $[\![\phi[\mathbf{x} := \sigma]]\!]_{K,\mathbf{E}}$; that is,

$$[\![\mu\mathbf{x}.\,\phi]\!]_{K,\mathbf{E}} = \bigcup_{\kappa \in \mathbb{O}} (\mathcal{F}^{\phi}_{K,\mathbf{E}})^{\kappa}(\emptyset).$$

Exercise 10.9 {T4} [Continuity in μ-calculus] A transition relation $\rightarrow$ is finitely branching iff every state has finitely many successors. Let K be an observation structure. (1) Prove that the function pre_K that maps regions of K to regions of K is $\bigcap$-continuous iff the transition relation of K is finitely branching. (2) Prove that the function $\mathcal{F}^{\mathbf{X},\phi}_{K,\mathbf{E}}$ is both $\bigcup$-continuous and $\bigcap$-continuous if the function pre_K is $\bigcap$-continuous. It follows that finite branching of the transition relation is both a sufficient and necessary condition for continuity of the functions $\mathcal{F}^{\mathbf{X},\phi}_{K,\mathbf{E}}$. ■

If the transition relation of K is finitely branching, then the function $\mathcal{F}^{\phi}_{K,\mathbf{E}}$ is $\bigcup$-continuous (Exercise 10.9) and, by the Kleene fixpoint theorem,

$$[\![\mu\mathbf{x}.\,\phi]\!]_{K,\mathbf{E}} \;=\; \bigcup_{i\in\mathbb{N}} (\mathcal{F}^{\phi}_{K,\mathbf{E}})^{i}(\emptyset);$$

that is, the characteristic region $[\![\mu\mathbf{x}.\,\phi]\!]_{K,\mathbf{E}}$ is the limit of the infinite approximation sequence $\emptyset$, $\mathcal{F}^{\phi}_{K,\mathbf{E}}(\emptyset)$, $\mathcal{F}^{\phi}_{K,\mathbf{E}}(\mathcal{F}^{\phi}_{K,\mathbf{E}}(\emptyset))$, etc. We will use this observation to compute the characteristic regions of Ctμ formulas. For example,

$$[\![\mu\mathbf{x}.\,(p \,\vee\, \exists\bigcirc\,\mathbf{x})]\!] \;=\; [\![\mathit{false}]\!] \cup [\![p]\!] \cup [\![\exists\bigcirc\, p]\!] \cup [\![\exists\bigcirc\,\exists\bigcirc\, p]\!] \cup \cdots$$

Similarly, a Ctμ formula of the form $\nu\mathbf{x}.\,\phi$, then, defines the greatest fixpoint of the function $\mathcal{F}^{\phi}_{K,\mathbf{E}}$:

$$[\![\nu\mathbf{x}.\,\phi]\!]_{K,\mathbf{E}} \;=\; \bigcap_{\kappa\in\mathbb{O}} (\mathcal{F}^{\phi}_{K,\mathbf{E}})^{\kappa}(\Sigma).$$

If the transition relation of K is finitely branching, then the function $\mathcal{F}^{\phi}_{K,\mathbf{E}}$ is $\bigcap$-continuous (Exercise 10.9) and the characteristic region $[\![\nu\mathbf{x}.\,\phi]\!]_{K,\mathbf{E}}$ is the limit of the infinite approximation sequence Σ, $\mathcal{F}^{\phi}_{K,\mathbf{E}}(\Sigma)$, $\mathcal{F}^{\phi}_{K,\mathbf{E}}(\mathcal{F}^{\phi}_{K,\mathbf{E}}(\Sigma))$, etc. For example,

$$[\![\nu\mathbf{x}.\,(p \,\wedge\, \forall\bigcirc\,\mathbf{x})]\!] \;=\; [\![\mathit{true}]\!] \cap [\![p]\!] \cap [\![\forall\bigcirc\, p]\!] \cap [\![\forall\bigcirc\,\forall\bigcirc\, p]\!] \cap \cdots$$

10.3.2 Expressive Power

Alternation depth

For a CTμ formula ϕ, its *nesting depth* is the the length of the longest chain of fixpoint-subformulas of ϕ that are nested in one another. The *alternation depth*, on the other hand, is computed by counting the number of alternations between μ-formulas and ν-formulas along chains of nested fixpoint-subformulas. The alternation depth is a better measure of the complexity of CTμ formulas.

ALTERNATION DEPTH

The *alternation depth* $ad(\phi)$ of a CTμ formula ϕ is defined inductively: If ϕ is not a fixpoint-formula then,

$$ad(\phi) = max\{ad(\psi) \mid \psi \text{ is a fixpoint-subformula of } \phi\};$$

else if $\phi = \mu\mathbf{x}.\,\psi$ then

$$ad(\phi) = max\{1, ad(\psi), 1 + max\{ad(\chi) \mid \chi \text{ is open } \nu\text{-subformula of } \psi\}\};$$

else if $\phi = \nu\mathbf{x}.\,\psi$ then

$$ad(\phi) = max\{1, ad(\psi), 1 + max\{ad(\chi) \mid \chi \text{ is open } \mu\text{-subformula of } \psi\}\}.$$

For every integer $k \geq 0$, the logic CTμ^k consists of all CTμ formulas ϕ with $ad(\phi) \leq k$. The CTμ formula ϕ is said to be *alternation-free* if $ad(\phi) \leq 1$, and the logic CTμ^1 is called alternation-free μ-calculus.

Remark 10.8 [Alternation depth] Alternation depth of a CTμ formula ϕ is the maximum integer $k \geq 0$ such that there exists a sequence $\phi_1\phi_2\ldots\phi_k$ of fixpoint-formulas such that (1) ϕ_1 is a subformula of ϕ, (2) for each $1 \leq j < k$, the formula ϕ_{j+1} is a subformula of ϕ_j, (3) for $2 \leq j \leq k$, the fixpoint-formula ϕ_j is open, and (4) for each $1 \leq j < k$, the types of ϕ_j and ϕ_{j+1} are different: ϕ_j is a μ-formula iff ϕ_{j+1} is a ν-formula. ∎

Example 10.2 [Alternation depth] The definition of the alternation-depth is illustrated by the following examples

$$\begin{array}{lcl}
ad(\mu\mathbf{x}.\,p \vee \exists\bigcirc\,\mathbf{x}) & = & 1 \\
ad(\mu\mathbf{x}.\,((\nu\mathbf{y}.\,p \wedge \forall\bigcirc\,\mathbf{y}) \vee \exists\bigcirc\,\mathbf{x})) & = & 1 \\
ad(\nu\mathbf{x}.\,(p \wedge \exists\bigcirc\,\nu\mathbf{y}.\,(q \wedge \forall\bigcirc\,\mathbf{y} \vee \exists\bigcirc\,\mathbf{x})) & = & 1 \\
ad(\nu\mathbf{x}.\,\mu\mathbf{y}.\,((p \wedge \mathbf{x}) \vee \exists\bigcirc\,\mathbf{y})) & = & 2
\end{array}$$

Note that the nesting depth of the first formula is 1, but for all the rest, the nesting depth is 2. ∎

Closure under negation

While the syntax of the logic CTμ does not admit negation, it is effectively closed under negation because every operator has its dual within the logic.

Exercise 10.10 {T3} [Duality of least and greatest fixpoint operators] Let B be a boolean algebra and let $\mathcal{F} : B \rightarrow B$ be a monotonic function. We write $\mu\mathcal{F}$ for the least fixpoint of $\mathcal{F}$ and $\mathcal{F}^{\neg\neg}$ for the function that maps each $x \in B$ to $\neg\mathcal{F}(\neg x)$. Prove that $\neg\mu\mathcal{F}^{\neg\neg}$ is the greatest fixpoint of $\mathcal{F}$. ∎

Proposition 10.5 [Closure under negation] *Let K be an observation structure, and let ϕ be a closed* CTμ *formula. Then, there exists a* CTμ *formula ψ such that for every state s of K, $s \models \phi$ iff $s \not\models \psi$. Furthermore, $ad(\phi) = ad(\psi)$.*

Proof. The proof uses the fact that the logical connectives $\wedge$ and $\vee$ are duals of each other, the temporal connectives $\exists\bigcirc$ and $\forall\bigcirc$ are duals of each other, and the fixpoint operators μ and ν are duals of each other. Specifically, define the function f that maps every CTμ formula to another CTμ formula. The function f is defined inductively:

$f(p) = \neg p$; $f(\neg p) = p$;
$f(\phi_1 \wedge \phi_2) = f(\phi_1) \vee f(\phi_2)$; $f(\phi_1 \vee \phi_2) = f(\phi_1) \wedge f(\phi_2)$;
$f(\exists\bigcirc\, \phi) = \forall\bigcirc\, f(\phi)$; $f(\forall\bigcirc\, \phi) = \exists\bigcirc\, f(\phi)$;
$f(\mu\mathbf{x}.\, \phi) = \nu\mathbf{x}.\, f(\phi)$; $f(\nu\mathbf{x}.\, \phi) = \mu\mathbf{x}.\, f(\phi)$; $f(\mathbf{x}) = \mathbf{x}$.

We prove that for every state s of an observation structure K, and a region environment $\mathbf{E}$, $s \models_{K,\mathbf{E}} \phi$ iff $s \not\models_{K,\mathbf{E}} f(\phi)$. This is proved by induction on the structure of ϕ. ■

Remark 10.9 [Alternative definition of CTμ syntax] The syntax of CTμ can alternatively be defined by the following clauses: (1) every atomic formula is a CTμ formula, (2) every region variable is a CTμ formula, (3) if ϕ is a CTμ formula, then so are $\neg\phi$ and $\exists\bigcirc\, \phi$, (4) if ϕ_1 and ϕ_2 are CTμ formulas then so is $\phi_1 \vee \phi_2$, and (5) if ϕ is a CTμ formula, and $\mathbf{x}$ is a region variable that is within the scope of an even number of negations in ϕ, then $\mu\mathbf{x}.\, \phi$ is a CTμ formula. ■

Alternation-free μ-calculus is as expressive as CTL

We establish that every CTL formula is equivalent to an alternation-free CTμ formula over observation structures.

Proposition 10.6 [Fixpoint characterization of $\exists\Diamond$] *Let K be an observation structure, and let p be an observation predicate of K. Then, the characteristic regions $[\![\exists\Diamond p]\!]_K$ and $[\![\mu\mathbf{x}.\, (p \vee \exists\bigcirc\, \mathbf{x})]\!]_K$ are identical.*

Proof. Consider the function $\mathcal{F}_{\exists\Diamond p}$ that maps regions of K to regions of K:

$$\mathcal{F}_{\exists\Diamond p}(\sigma) \;=\; [\![p]\!]_K \cup pre_K(\sigma).$$

Observe that the operator $\exists\bigcirc$ of CTμ is same as the function pre, and hence, $[\![\mu\mathbf{x}.\, (p \vee \exists\bigcirc\, \mathbf{x})]\!]_K$ is the least fixpoint of the function $\mathcal{F}_{\exists\Diamond p}$. First, we show that the characteristic region $[\![\exists\Diamond p]\!]_K$ is a fixpoint of the function $\mathcal{F}_{\exists\Diamond p}$:

$$[\![\exists\Diamond p]\!] \;\Leftrightarrow\; [\![p]\!] \cup pre([\![\exists\Diamond p]\!]).$$

This is established from the definition of the CTL operator $\exists\Diamond$. Second, we show that the region $[\![\exists\Diamond\sigma]\!]$ is contained in all fixpoints of $\mathcal{F}_{\exists\Diamond p}$: for all regions $\sigma \subseteq \Sigma$ and all states $s \in \Sigma$, if $\sigma = [\![p]\!] \cup pre(\sigma)$ and $s \models \exists\Diamond p$, then $s \in \sigma$.

So assume that $\sigma = [\![p]\!] \cup pre(\sigma)$ and that there is a source-s trajectory $\overline{s}_{0..n}$ of K such that $s_n \models p$. Then $s_n \in \sigma$, and by backward induction on $\overline{s}_{0..n}$, $s_i \in \sigma$ for all $0 \leq i \leq n$. ■

Exercise 10.11 {T2} [μ-calculus vs. CTL] Which CTL formula is equivalent to the CTμ formula $\mu\mathbf{x}.\,\exists\bigcirc\,(\mathbf{x} \vee p)$? ■

Remark 10.10 [Fixpoint characterization of $\exists\Diamond$] Let ϕ be a CTμ formula and ψ be a CTL formula. If ϕ and ψ are equivalent, then so are the formulas $\mu\mathbf{x}.\,(\phi \vee \exists\bigcirc\,\mathbf{x})$ and $\exists\Diamond\psi$. ■

To obtain fixpoint characterization of the possibly-until connective $\exists\mathcal{U}$, observe the following equivalence:

$$(\phi\,\exists\mathcal{U}\,\psi) \;\;\leftrightarrow\;\; \psi \vee (\phi \wedge (\phi\,\exists\mathcal{U}\,\psi).$$

A state satisfies $(\phi\,\exists\mathcal{U}\,\psi)$ if either it satisfies ψ, or it satisfies ϕ and has a successor that is already known to satisfy $(\phi\,\exists\mathcal{U}\,\psi)$. This suggests that $\exists\mathcal{U}$ can be defined as a μ-formula:

Proposition 10.7 [Fixpoint characterization of $\exists\mathcal{U}$] *Let ϕ_1 and ϕ_2 be a* CTμ *formulas, and let ψ_1 and ψ_2 be* CTL *formulas. If the formulas ϕ_1 and ψ_1 are equivalent, and the formulas ϕ_2 and ψ_2 are equivalent, then so are the formulas $\mu\mathbf{x}.\,(\phi_2 \vee (\phi_1 \wedge \exists\bigcirc\,\mathbf{x}))$ and $\psi_1\exists\mathcal{U}\psi_2$.*

Finally, let us consider the possibly-always connective $\exists\Box$. A state all of whose successors do not satisfy p cannot satisfy $\exists\Box p$. A state all of whose successors are known not to satisfy $\exists\Box p$ cannot satisfy $\exists\Box p$. This suggests a characterization of $\exists\Box p$ as a greatest fixpoint: $[\![\exists\Box p]\!]$ is the maximal region each of whose states satisfies p and has at least one successor satisfying p.

Proposition 10.8 [Fixpoint characterization of $\exists\Box$] *Let ϕ be a* CTμ *formula and ψ be a* CTL *formula. If ϕ and ψ are equivalent, then so are the formulas $\nu\mathbf{x}.\,(\phi \wedge \exists\bigcirc\,\mathbf{x})$ and $\exists\Box\psi$.*

Theorem 10.4 [From CTL to CTμ] *Every* CTL *formula ϕ is equivalent to an alternation-free* CTμ *formula of length $O(|\phi|)$.*

Exercise 10.12 {T4} [Correctness of translation from CTL to CTμ] Prove Propositions 10.7 and 10.8, and then, prove Theorem 10.4 using Propositions 10.5, 10.7 and 10.8. ■

We can define temporal operators in CTμ:

$$
\begin{array}{lll}
\exists\Diamond\phi & \text{for} & \mu\mathbf{x}.\,(\phi \vee \exists\bigcirc\mathbf{x});\\
\phi_1\exists\mathcal{U}\phi_2 & \text{for} & \mu\mathbf{x}.\,(\phi_2 \vee (\phi_1 \wedge \exists\bigcirc\mathbf{x}));\\
\exists\Box\phi & \text{for} & \nu\mathbf{x}.\,(\phi \wedge \exists\bigcirc\mathbf{x});\\
\forall\Diamond\phi & \text{for} & \mu\mathbf{x}.\,(\phi \vee \forall\bigcirc\mathbf{x});\\
\forall\Box\phi & \text{for} & \nu\mathbf{x}.\,(\phi \wedge \forall\bigcirc\mathbf{x});\\
\phi_1\forall\mathcal{U}\phi_2 & \text{for} & \mu\mathbf{x}.\,(\phi_2 \vee (\phi_1 \wedge \forall\bigcirc\mathbf{x})).
\end{array}
$$

Notice that the CTμ formula $(\nu\mathbf{x}.\,\phi \vee \exists\bigcirc\mathbf{x})$ is equivalent to *true*, and the CTμ formula $(\mu\mathbf{x}.\,\phi \wedge \exists\bigcirc\mathbf{x})$ is equivalent to *false*.

Distinguishing power of CTμ

In Section 5.4 we established that bisimilarity is a fully abstract semantics for STL; that is, two bisimilar states satisfy the same set of STL formulas, and if two states are not bisimilar then some STL formula distinguishes between them. Since (alternation-free) μ-calculus is as expressive as STL, it follows that it can distinguish between states that are not bisimilar. Furthermore, μ-calculus cannot distinguish between bisimilar states.

Proposition 10.9 [CTμ abstraction] *Bisimilarity is an abstract semantics for* CTμ*.*

Exercise 10.13 {T4} [CTμ abstraction] Prove Proposition 10.9. ■

Thus, the distinguishing powers of a variety of state logics, such as STL$^{\bigcirc}$, STL, CTL, CTμ, CTμ^1, coincide, and all these logics are more distinguishing than the structure logic SAL.

Alternation-free CTμ is more expressive than CTL

The alternation-free μ-calculus is more expressive than CTL. There are at least two types of properties that can be specified in CTμ^1, but not in CTL. The first type concerns the inability of CTL to count, while the second one concerns inability of CTL to specify game-like properties.

Proposition 10.10 [CTL vs. CTμ^1] *Let p be a proposition. No* CTL *formula is equivalent to the* CTμ^1 *formula* $\nu\mathbf{x}.\,(p \wedge \forall\bigcirc\forall\bigcirc\mathbf{x})$.

Proof. The formula $\nu\mathbf{x}.\,(p \wedge \forall\bigcirc\forall\bigcirc\mathbf{x})$ is satisfied by a state s of an observation structure K iff for every source-s ω-trajectory $\underline{s}$, $s_i \models p$ for all even numbers i. Thus, the formula $\nu\mathbf{x}.\,(p \wedge \forall\bigcirc\forall\bigcirc\mathbf{x})$ is equivalent to the SAL formula ϕ_{even} (see proof of Theorem 6.2). We already know that no STL formula is equivalent to ϕ_{even}. The same proof can be extended to establish that no CTL formula is equivalent to ϕ_{even}. ■

Consider an observation structure K with three observations a, b, and c. Consider the following two-player game between a protagonist and an adversary. The positions of the game is described by a state of K. If the current position s has observation c, then the protagonist wins the game. Otherwise, the position of the game is updated to some successor of s. If the observation of s is a, then the protagonist chooses the successor position, and if the observation of s is b, then the adversary chooses the successor position. Given an initial position, the protagonist wins if it has a strategy to force the game to a state with observation c. Thus, the described game is a standard AND-OR game, where states with observations c are winning positions, states with observation a are OR-positions and states with observation b are AND-positions. Let σ be the set of winning initial positions for the protagonist. To get a fixpoint characterization of σ, observe that (1) all states with observation c belong to σ, (2) for a state s with observation a, if some successor of s is already known to be winning, then the protagonist can win from s also, and (3) for a state s with observation b, if all successors of s are already known to be winning, then the protagonist can win from s also. Thus, σ is the smallest region that contains $[\![c]\!]$, $[\![a \wedge \exists\bigcirc \sigma]\!]$, and $[\![b \wedge \forall\bigcirc \sigma]\!]$. Thus, the set of winning positions is described by the alternation-free formula $\mu\mathbf{x}.\,(c \vee (a \wedge \exists\bigcirc \mathbf{x}) \vee (b \wedge \forall\bigcirc \mathbf{x}))$. It turns out that the set of winning positions cannot be characterized using a CTL formula.

Proposition 10.11 [CTL vs. CTμ^1] *Let p and q be propositions. No* CTL *formula is equivalent to the* CTμ^1 *formula* $\mu\mathbf{x}.\,(q \vee (p \wedge \exists\bigcirc \mathbf{x}) \vee (\neg p \wedge \forall\bigcirc \mathbf{x}))$.

Fair region for a single Büchi

We turn our attention to characterization of fair regions using μ-calculus. Let (K, F) be a fair structure. The fair region σ^F of K consists of states from which there exists a F-fair ω-trajectory. For now, let us assume that F contains a single Büchi constraint specified by the state predicate p. Thus, a state s of K belongs to σ^F iff there exits a source-s ω-trajectory that contains infinitely many states that satisfy p. We use the operator $\Box\Diamond$ to denote infinite repetition:

> $s \models_K \exists\Box\Diamond p$ iff there exists a source-s ω-trajectory $\underline{s}$ of K such that $s_i \models_K p$ for infinitely many positions i.

The formula $\exists\Box\Diamond p$ can be expressed in CTμ using nested fixpoints: it is equivalent to the formula $\nu\mathbf{x}.\,\mu\mathbf{y}.\,\exists\bigcirc\,((\mathbf{x} \wedge p) \vee \mathbf{y})$, which can also be written as $\nu\mathbf{x}.\,\exists\Diamond^+(\mathbf{x} \wedge p)$. That is, $[\![\exists\Box\Diamond p]\!]$ is the maximal region σ such that from every state in σ, some state in $\sigma \cap [\![p]\!]$ is reachable in one or more steps. The i-th approximation in the computation of $\nu\mathbf{x}.\,\exists\Diamond^+(\mathbf{x} \wedge p)$ contains all states from which there exists a trajectory containing i states satisfying p:

$$[\![\nu\mathbf{x}.\,\exists\Diamond^+(\mathbf{x} \wedge p)]\!] \;=\; [\![\mathit{true}]\!] \cap [\![\exists\Diamond^+ p]\!] \cap [\![\exists\Diamond^+(p \wedge \exists\Diamond^+ p)]\!] \cap \cdots$$

Proposition 10.12 [Fixpoint characterization of $\exists\Box\Diamond$] *The* Ctμ *formula* $\nu\mathbf{x}.\,\mu\mathbf{y}.\,\exists\bigcirc((\mathbf{x} \wedge p) \vee \mathbf{y})$ *is equivalent to* $\exists\Box\Diamond p$.

Proof. Let K be an observation structure. Consider the function $\mathcal{F}_{\exists\Box\Diamond p}$ that maps regions of K to regions of K:

$$\mathcal{F}_{\exists\Box\Diamond p}(\sigma) \;=\; pre^{+}(\sigma \cap [\![p]\!]).$$

It suffices to show that the region $[\![\exists\Box\Diamond p]\!]$ is the maximal fixpoint of the function $\mathcal{F}_{\exists\Box\Diamond p}$. First, we show that $[\![\exists\Box\Diamond p]\!]$ is a fixpoint of $\mathcal{F}_{\exists\Box\Diamond p}$:

$$[\![\exists\Box\Diamond p]\!] \;\Leftrightarrow\; pre^{+}([\![\exists\Box\Diamond p]\!] \cap [\![p]\!]).$$

To establish this, for all states s, there is a source-s p-fair trajectory iff there exists a state t such that (i) t is reachable from s in one or more steps (i.e. $s \in pre^{+}(t)$), (ii) t satisfies p, and (iii) there is a source-t p-fair trajectory. Second, we need to establish that every fixpoint of $\mathcal{F}_{\exists\Box\Diamond p}$ is contained in $[\![\exists\Box\Diamond p]\!]$: for all regions σ and all states s,

$$\text{if } \sigma = pre^{+}(\sigma \cap [\![p]\!]) \text{ and } s \in \sigma \text{ then } s \models \exists\Box\Diamond p.$$

So assume that $\sigma = pre^{+}(\sigma \cap [\![p]\!])$ and $s \in \sigma$. We construct an infinite sequence of states $s_0 s_1 \ldots$ as follows. Let $s_0 = s$. Given $s_i \in \sigma$, choose s_{i+1} such that $s_{i+1} \in \sigma$ and $s_{i+1} \models p$ and $s_{i+1} \in post^{+}(s_i)$ (such a state exists since $\sigma = pre^{+}(\sigma \cap [\![p]\!])$). It follows that there exists a source-s ω-trajectory containing infinitely many states satisfying p. ■

Exercise 10.14 {T2} [$\exists\Box\Diamond$ in μ-calculus] Is the formula $\nu\mathbf{x}.\,\exists\Diamond(p \wedge \mathbf{x})$ equivalent to $\exists\Box\Diamond p$? Is the formula $\nu\mathbf{x}.\,\exists\Diamond(p \wedge \exists\bigcirc\,\mathbf{x})$ equivalent to $\exists\Box\Diamond p$? ■

Exercise 10.15 {T2} [Fixpoint characterization of $\exists\Box p$ in Büchi structures] Let $\mathcal{K} = (K, F)$ be a fair structure such that F contains a single Büchi constraint specified by the predicate q. Write a Ctμ formula ϕ such that $[\![\phi]\!]_K$ equals $[\![\exists\Box p]\!]_{\mathcal{K}}$. That is, $s \models_K \phi$ iff there is a source-s F-fair ω-trajectory $\underline{s}$ with $s_i \models p$ for all $i \geq 0$. ■

Exercise 10.16 {T3} [$\exists\Diamond\Box$ in μ-calculus] Given a state s of an observation structure K, and a state predicate p, define $s \models_K \exists\Diamond\Box p$ iff there exist a source-s ω-trajectory $\underline{s}$ and an integer $i \geq 0$ such that $s_j \models_K p$ for all $j \geq i$. Write Ctμ formula that is equivalent to $\exists\Diamond\Box p$. ■

Now let us consider the case when the fairness assumption contains a single weak-fairness constraint specified by an action α. Suppose the action α is specified by the action predicate $p \wedge q'$; that is, $s \xrightarrow{\alpha} t$ iff $s \models p$ and $t \models q$. We wish to characterize the fair region by a Ctμ formula. A state s of K satisfies the Ctμ fromula $\nu\mathbf{x}.\,\exists\Diamond(p \wedge \exists\bigcirc\,(q \wedge \mathbf{x}))$ iff there is a source-s ω-trajectory $\underline{s}$ such that for infinitely many positions i, $s_i \models p$ and $s_{i+1} \models q$, that is, iff there is a source-s α-fair trajectory. This leads to the characterization of fair regions when fairness contains a single weak constraint. In general, the action α will be specified using a disjunction $\vee 0 \leq i \leq k.\, p_i \wedge q_i'$: $s \xrightarrow{\alpha} t$ iff for some $0 \leq i \leq k$, $s \models p_i$ and $t \models q_i$.

Proposition 10.13 [Single weak constraint in $\mathrm{CT}\mu$] *Let $\mathcal{K} = (K, F)$ be a fair structure where F contains a single weak constraint α. Let $p_0, \ldots p_k$ and $q_0 \ldots q_k$ be state predicates of K such that $\alpha = [\![\vee 0 \leq i \leq k.\, p_i \wedge q_i']\!]_K$. Then, the fair region of $\mathcal{K}$ equals*

$$[\![\nu \mathbf{x}.\, \exists\Diamond \vee 0 \leq i \leq k.\, (p_i \wedge \exists\bigcirc (q_i \wedge \mathbf{x}))]\!]_K.$$

Exercise 10.17 {T3} [Single weak constraint in μ-calculus] Prove Proposition 10.13.
■

Exercise 10.18 {T3} [Multiple Büchi constraints] Consider a Büchi structure (K, F) where F contains k Büchi constraints specified by predicates $p_1, \ldots p_k$. Show that the fair region is characterized by the $\mathrm{CT}\mu$ formula

$$\nu \mathbf{x}.\, \exists\Diamond^+(p_1 \wedge \exists\Diamond^+(p_2 \wedge \cdots \wedge \exists\Diamond^+(p_k \wedge \mathbf{x}) \cdots)).$$

■

Consider a module P, and let a be an update choice of an atom U of P. The availability action $avail_a$ of the choice a is be described by a predicate q_{avail_a} over $\mathsf{read}X_U \cup \mathsf{await}X_U'$. The execution action $exec_a$ of the choice a is described by a predicate q_{exec_a} over $\mathsf{read}X_U \cup \mathsf{await}X_U' \cup \mathsf{ctr}X_U'$. The weak-fairness constraint of α is, then, described by the predicate $q_{exec_a} \vee \neg q_{avail_a}$. This predicate can be rewritten to a form required by Proposition 10.13.

Example 10.3 [Fairness constraints for mutual exclusion] Recall the fair module *FairPete* from Figure 8.5. The module has four weak-fairness constraints specified by the choices α_1, β_1, α_2, and β_2. Let us just consider the choice α_1 The weak-fairness constraint corresponding to the update choice α_1 is specified by the action $exec_{\alpha_1} \cup (\rightarrow \backslash avail_{\alpha_1})$. The execution action $exec_{\alpha_1}$ is specified by the predicate $pc_1 = inC \wedge pc_1' = outC$, and the availability action $avail_{\alpha_1}$ is specified by the predicate $pc_1 = inC$. It follows that the fairness constraint corresponding to the choice α_1 is the disjunction

$$(pc_1 = inC \wedge pc_1' = outC) \vee (pc_1 \neq inC).$$

The corresponding fair region, then, is expressed by the $\mathrm{CT}\mu$ formula

$$\nu \mathbf{x}.\, \exists\Diamond\, [\, (pc_1 = inC \wedge \exists\bigcirc (pc_1 = outC \wedge \mathbf{x})) \vee (pc_1 \neq inC \wedge \exists\bigcirc \mathbf{x})\,].$$

■

While the operator $\exists\Box\Diamond$ is specifiable in $\text{CT}\mu^2$, it is not specifiable in CTL.

Proposition 10.14 [CTL cannot express $\exists\Box\Diamond$] *There is no* CTL *formula that is equivalent to* $\exists\Box\Diamond p$.

Proof. Suppose there is a CTL formula ϕ such that for every structure K, $[\![\exists\Box\Diamond p]\!]_K$ equals $[\![\phi]\!]_K$. Suppose the length of ϕ is k. Consider the observation structure of Figure 10.1. States that satisfy the atomic formula p are labeled with p. We first prove the following lemma.

Lemma A. For every CTL formula ψ, for all integers $|\psi| - 1 \leq i \leq j \leq k$, $s_i \models \psi$ iff $s_j \models \psi$ and $t_i \models \psi$ iff $t_j \models \psi$.

Proof of Lemma A. The proof is by induction on the structure of the formula ψ. For $0 \leq i \leq k$, all the states s_i satisfy the same atomic formulas, and so do all the states t_i. Hence, the lemma holds if ψ is an atomic formula. When $\psi = \neg\chi$, or when $\psi = \chi_1 \vee \chi_2$, the lemma follows from induction.

Case $\psi = \exists\bigcirc\chi$. For $1 \leq i \leq k$, $s_i \models \psi$ iff $t_{i-1} \models \chi$, and $t_i \models \psi$ iff $t_i \models \chi$ or $s_{i-1} \models \chi$. For $|\psi| - 1 \leq i \leq j \leq k$, $i \geq 1$ and $|\chi| \leq i - 1 \leq j - 1 \leq k$. By induction, $t_{i-1} \models \chi$ iff $t_{j-1} \models \chi$; $s_{i-1} \models \chi$ iff $s_{j-1} \models \chi$; and $t_i \models \chi$ iff $t_j \models \chi$.

Case $\psi = \exists\Box\chi$. For $1 \leq i \leq k$, $t_i \models \psi$ iff $t_i \models \chi$, and $s_i \models \psi$ iff $s_i \models \chi$ and $t_{i-1} \models \chi$. Now we can proceed as in the previous case.

Case $\psi = \chi_1 \exists\mathcal{U} \chi_2$. Left as an exercise.

Corollary B. For every subformula ψ of ϕ, $s_k \models \psi$ iff $s_{k-1} \models \psi$.

The next lemma implies that $s_k \models \phi$ iff $u \models \phi$. This yields a contradiction, because $s_k \not\models \exists\Box\Diamond p$, but $u \models \exists\Box\Diamond p$.

Lemma C. For every subformula ψ of ϕ, $s_k \models \psi$ iff $u \models \psi$, and $t_k \models \psi$ iff $v \models \psi$.

Proof of Lemma C. The proof is by induction on the structure of the formula ψ. When ψ is an atomic formula, the lemma is immediate as the states s_k and u, and states t_k and v have identical observations. When $\psi = \neg\chi$, or when $\psi = \chi_1 \vee \chi_2$, the lemma follows from induction.

Case $\psi = \forall\bigcirc\chi$. $s_k \models \psi$ iff $t_k \models \chi$ iff, by induction, $v \models \chi$ iff $u \models \psi$. $t_k \models \psi$ iff both t_k and s_{k-1} satisfy χ iff, by Corollary B, all of s_k, t_k, and s_{k-1} satisfy χ iff, by induction, all of u, v, and s_{k-1} satisfy χ iff $v \models \psi$.

Case $\psi = \exists\Box\chi$. $s_k \models \psi$ iff both s_k and t_k satisfy χ iff, by induction, both u and v satisfy χ iff $u \models \psi$. $t_k \models \psi$ iff $t_k \models \chi$ iff, by induction, $v \models \chi$ iff $v \models \psi$.

Case $\psi = \chi_1 \exists\mathcal{U} \chi_2$. Left as an exercise. ■

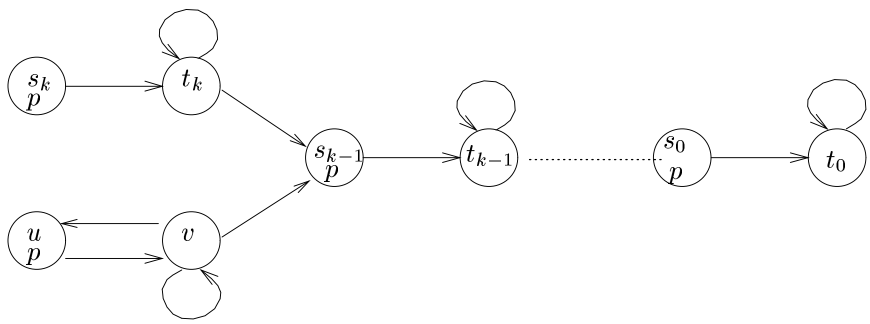

Figure 10.1: $\exists\Box\Diamond$ is not expressible in CTL

Exercise 10.19 {T4} [Alternation-free μ-calculus cannot express $\exists\Box\Diamond$] Prove that no formula of CTμ^1 is equivalent to $\exists\Box\Diamond p$. ■

Remark 10.11 [Hierarchy of expressiveness] For every integer $i \geq 0$, the fragment CTμ^{i+1} is more expressive than the fragment CTμ^i. Thus, the expressiveness of CTμ strictly increases with increasing alternation depth. ■

Specifying fair regions

Now we turn our attention to strong fairness constraints. Let F be a fairness assumption for an observation structure K. Suppose each fairness constraint $f \in F$ is a Streett constraint defined by state predicates p and q: an ω-trajectory $\underline{s}$ is f-fair iff if it is q-fair or not p-fair.

Exercise 10.20 {T3} [Fixpoint characterization of single Streett constraint] Consider an observation structure K and two state predicates p and q of K. Show that a state s of K satisfies the CTμ formula $\exists\Diamond(\exists\Box\neg p \vee \exists\Box\Diamond q)$ iff there exists a source-s (p,q)-fair trajectory of K. ■

Exercise 10.20 suggests characterization of fair regions when the fairness assumption has a single fairness constraint. It can be generalized to multiple Streett constraints. Let F be a Streett assumption for an observation structure K. Then, a state s belongs to the fair region of K iff there exists a state $t \in \mathit{post}^*(s)$, a subset F' of F, and a source-t ω-trajectory (1) that is q-fair for every $(p,q) \in F'$, and (2) all of whose states satisfy $\neg p$ for every $(p,q) \in F \backslash F'$. This suggests a CT$\mu$ formula whose length is exponential in the number of Streett constraints in F. However, a polynomial translation is possible.

Proposition 10.15 [Emerson-Lei Fixpoint characterization of Streett assumption] *Let K be an observation structure, and let F be a Streett assumption for K. Then, the fair region of (K,F) is the characteristic region of the formula*

$$\exists\Diamond\, \nu\mathbf{x}.\ \bigwedge(p,q) \in F.\, [\, \exists\bigcirc\, (\mathbf{x}\exists\mathcal{U}(q \wedge \mathbf{x})) \vee (\neg p \wedge \exists\bigcirc\, \mathbf{x})\,].$$

Proof. Let K be an observation structure. Let $F = \{(\neg p_1, q_1), \ldots (\neg p_k, q_k)\}$ be a Streett assumption with k Streett constraints. An ω-trajectory $\underline{s}$ is F-fair iff for $1 \leq i \leq k$, either $\underline{s}$ is q_i-fair or it has a suffix containing only p_i-states. This requirement on the ω-trajectory is expressed by the formula

$$\phi = \bigwedge 1 \leq i \leq k. (\Diamond\Box p_i \vee \Box\Diamond q_i).$$

The fair region is characterized by the formula $\exists\phi$. Define the formula

$$\phi' = \bigwedge 1 \leq i \leq k. (\Box p_i \vee \Box\Diamond q_i).$$

An ω-trajectory $\underline{s}$ satisfies ϕ' iff for $1 \leq i \leq k$, either $\underline{s}$ is q_i-fair or contains only p_i-states. A state s satisfies $\exists\phi'$ iff there is source-s ω-trajectory satisfying ϕ'. The next two lemmas follow from the definitions of the formulas ϕ and ϕ'.

Lemma A. $[\![\exists\Diamond\exists\phi']\!] = [\![\exists\phi]\!]$.

Lemma B. $[\![\exists\Diamond\exists\phi]\!] = [\![\exists\phi]\!]$.

Now consider the function $\mathcal{F}$ that maps regions of K to regions of K:

$$\mathcal{F}(\sigma) = \bigwedge 1 \leq i \leq k. [\, \exists\bigcirc (\sigma\, \exists\mathcal{U}(q_i \wedge \sigma)) \vee (p_i \wedge \exists\bigcirc \sigma)\,].$$

Lemma C. If σ is a fixpoint of $\mathcal{F}$ then $\sigma \subseteq [\![\exists\phi]\!]$.

Proof of Lemma C. Let σ be a fixpoint of $\mathcal{F}$. Consider $s \in \sigma$. We will construct a source-s ω-trajectory that satisfies ϕ. For every $j \geq 0$, we define a state s_j, and a finite trajectory from s_j to s_{j+1} containing only σ-states. Let $s_0 = s \in \sigma$. Consider s_j in σ. Let i be j *mod* k. Since $\sigma = \mathcal{F}(\sigma)$, s_j satisfies $\exists\bigcirc (\sigma\, \exists\mathcal{U}(q_i \wedge \sigma))$ or $p_i \wedge \exists\bigcirc \sigma$. If s_j satisfies $\exists\bigcirc (\sigma\, \exists\mathcal{U}(q_i \wedge \sigma))$, then there exists a source-s_j trajectory $\overline{t}_{0\ldots n}$ with $n > 0$ containing only σ-states such that $t_n \models q_i$. Choose $s_{j+1} = t_n$. If s_j does not satisfy $\exists\bigcirc (\sigma\, \exists\mathcal{U}(q_i \wedge \sigma))$, then it must satisfy $p_i \wedge \exists\bigcirc \sigma$, and choose s_{j+1} to be a successor of s_j in σ.

Let $\underline{t}$ be the source-s ω-trajectory obtained by concatenating the finite trajectories from s_j to s_{j+1} defined above. Every state in $\underline{s}$ belongs to σ. We wish to establish that $\underline{t}$ satisfies ϕ. Consider $1 \leq i \leq k$. For every $n \geq 0$, if s_{i+kn} satisfies $\exists\bigcirc (\sigma\, \exists\mathcal{U}(q_i \wedge \sigma))$ then s_{i+kn+1} satisfies q_i. Suppose that there are infinitely many n such that s_{i+kn} satisfies $\exists\bigcirc (\sigma\, \exists\mathcal{U}(q_i \wedge \sigma))$. Then, by construction, $\underline{t}$ is q_i-fair. Otherwise, there exists $n \geq 0$ such that $s_{i+kn'}$ does not satisfy $\exists\bigcirc (\sigma\, \exists\mathcal{U}(q_i \wedge \sigma))$ for $n' \geq n$. Since every state in $\underline{t}$ satisfies σ, it follows that there exists $n \geq 0$ such that $t_{n'}$ does not satisfy $\exists\bigcirc (\sigma\, \exists\mathcal{U}(q_i \wedge \sigma))$ for $n' \geq n$. Since σ is a fixpoint of $\mathcal{F}$, it follows that $t_{n'}$ satisfies $p_i \wedge \exists\bigcirc \sigma$, and hence, $\underline{t}$ satisfies $\Diamond\Box p_i$.

Lemma D. $[\![\exists\phi']\!] \subseteq \mathcal{F}([\![\exists\phi']\!])$.

Proof of Lemma D. Consider a state $s \in [\![\exists\phi']\!]$. There exists a source-$s$ ω-trajectory $\underline{s}$ such that for $1 \leq i \leq k$, either $\underline{s}$ is q_i-fair or contains only p_i-states.

Every suffix of $\underline{s}$ satisfies ϕ', and hence, $s_j \models \exists\phi'$ for all $j \geq 0$. We wish to establish that s satisfies $\mathcal{F}([\![\exists\phi']\!])$. Consider $1 \leq i \leq k$. We need to prove that s satisfies either $\exists\bigcirc([\![\exists\phi']\!]\exists\mathcal{U}(q_i \wedge [\![\exists\phi']\!]))$ or $p_i \wedge \exists\bigcirc[\![\exists\phi']\!]$. If $\underline{s}$ is q_i-fair, then s_1 satisfies $[\![\exists\phi']\!]\exists\mathcal{U}(q_i \wedge [\![\exists\phi']\!])$; otherwise $\underline{s}$ contains only p_i-states, and s satisfies $p_i \wedge \exists\bigcirc[\![\exists\phi']\!]$.

Now we proceed to show that $\exists\phi$ is equivalent to $\exists\Diamond\,\nu\mathbf{x}.\,\mathcal{F}(\mathbf{x})$. Suppose $s \models \exists\Diamond\nu\mathbf{x}.\,\mathcal{F}(\mathbf{x})$. By Lemma C, if a state satisfies $\nu\mathbf{x}.\,\mathcal{F}(\mathbf{x})$ then it also satisfies $\exists\phi$. Hence, $s \models \exists\Diamond\exists\phi$. By Lemma A, $s \models \exists\phi$. Conversely, suppose $s \models \exists\phi$. By Lemma B, $s \models \exists\Diamond\exists\phi'$. By Lemma D, $[\![\exists\phi']\!]$ is contained in the maximla fixpoint of $\mathcal{F}$. Hence, $s \models \exists\Diamond\nu\mathbf{x}.\,\mathcal{F}(\mathbf{x})$. ■

Exercise 10.21 {T3} [Fixpoint characterization of fairness assumption] Consider a fair graph (K, F). Every constraint is F is a pair of actions, and suppose every action α is represented by state predicates $p_0, \ldots p_k$ and $q_0 \ldots q_k$ of K such that $\alpha = [\![\vee 0 \leq i \leq k.\, p_i \wedge q_i']\!]_K$. Given this representation of actions, write a Ctμ formula that characterizes the faire region of (K, F). ■

Thus, the fair region of a fair graph can be characterized in μ-calculus using formulas of alternation depth 2. To characterize the region $[\![\exists\Box p]\!]_{\mathcal{K}}$ of fair structure, only a slight modification is required. For instance, for a Streett assumption F, the characteristic region $[\![\exists\Box p]\!]$ equals

$$p\,\exists\mathcal{U}\,\nu\mathbf{x}.\,p \wedge \bigwedge(q,r) \in F.\,[\,\exists\bigcirc(\mathbf{x}\exists\mathcal{U}(r \wedge \mathbf{x})) \vee (\neg q \wedge \exists\bigcirc\mathbf{x})\,].$$

Theorem 10.5 [From CTL over fair structures to CTμ] *For every* CTL *formula ϕ and a fair structure $\mathcal{K} = (K, F)$, there exists a formula ψ of* CTμ^2 *such that $[\![\phi]\!]_{\mathcal{K}} = [\![\psi]\!]_K$ and $|\psi| = O(|\phi| \cdot |F|)$.*

Let a be an update choice of a module P. The strong-fairness constraint of a is the pair $(avail_a, exec_a)$ of actions. After writing the two actions $avail_a$ and $exec_a$ in the form stipulated by Exercise 10.21, we can write a CTμ formula that characterizes the fair region of the fair module.

10.3.3 Model checking

We are given a closed CTμ formula ϕ and ϕ-structure K, we are required to check if all the initial states of K satisfy ϕ. For this purpose, we compute the characteristic region $[\![\phi]\!]_K$. Assume that the fomula ϕ has no name-conflicts in the use of region variables: every variable $\mathbf{x}$ is quantified by a unique fixpoint operator.

The characteristic region $[\![\phi]\!]_K$ can be computed using a recursive function *Eval*. The table $\mathbf{E}$ stores, for every region variable $\mathbf{x}$, a region $\mathbf{E}(\mathbf{x})$ of K. The function *Eval* takes a formula ψ as an argument, and returns the set of states satisfying

ψ using the table $\mathbf{E}$ to evaluate free variables. If ψ is an atomic formula, the computation of *Eval*(ψ) is immediate. If ψ is a conjunction of formulas, then *Eval* calls itself recursively on the conjuncts, and returns the intersection of the results. The case of disjunction is similar. When ψ equals $\exists\bigcirc\,\chi$, *Eval* calls itself recursively on χ, and returns the set of predecessors of the result. The evaluation of $\forall\bigcirc\,\chi$ uses the fact that $\exists\bigcirc$ and $\forall\bigcirc$ are duals of each other: $\forall\bigcirc = \neg\exists\bigcirc\,\neg$.

To evaluate a subformula $\mu\mathbf{x}.\,\chi$, the minimal fixpoint is computed by evaluating χ repeatedly. In the first iteration, $\mathbf{E}(\mathbf{x})$ is chosen to be the empty set, and in each successive iteration, $\mathbf{E}(\mathbf{x})$ is chosen to be the value of *Eval*(χ) from the previous iteration. The fixpoint is reached when two consecutive iterations yield the same result. The number of iterations is bounded by the number of states in the observation structure. The evaluation of $\nu\mathbf{x}.\,\chi$ is similar, but in this case, in the first iteration, $\mathbf{E}(\mathbf{x})$ is chosen to be the set of all states. A naive implementation of this recursive scheme would make the depth of recursion equal to the nesting depth of the formula, resulting in an algorithm with time complexity $O(n^k)$, where k is the nesting depth of the formula. Two improvements are possible.

First, every closed formula needs to be evaluated just once. For example, consider the formula $\mu\mathbf{x}.\,\psi$, where χ is a closed fixpoint subformula of ψ. The invocation *Eval*$(\mu\mathbf{x}.\,\psi)$ results in repeated calls to *Eval*(ψ), and hence to *Eval*(χ), each time with a different value of $\mathbf{E}(\mathbf{x})$. However, χ is a closed formula, and its value does not depend on $\mathbf{E}(\mathbf{x})$. Consequently, it needs to be evaluated only once. For this purpose, we use a hash-table *Done* that stores the results of evaluating closed formulas. Upon invocation, *Eval* checks if its input formula is closed, and if so, whether it has already been evaluated by consulting the hash-table.

Second, consider the formula $\mu\mathbf{x}.\,\phi$, where $\psi = \mu\mathbf{y}.\,\chi$ is a disjunct of ϕ. Let σ_0 be the empty set. The first iteration in *Eval*$(\mu\mathbf{x}.\,\phi)$ calls *Eval*(ϕ) with $\mathbf{E}(\mathbf{x}) = \sigma_0$. This involves evaluation of the fixpoint formula ψ, which itself involves an iterative computation of χ during which the region $\mathbf{E}(\mathbf{y})$ keeps growing. Let $\tau_0 = [\![\psi]\!]$ and $\sigma_1 = [\![\phi]\!]$ with $\mathbf{E}(\mathbf{x}) = \sigma_0$. If σ_0 is a strict subset of σ_1, the second iteration in *Eval*(ϕ) calls *Eval*(ψ) with $\mathbf{E}(\mathbf{x}) = \sigma_1$. This would result in repeated evaluation of χ starting with $\mathbf{E}(\mathbf{y})$ to be the empty set until the value of $\mathbf{E}(\mathbf{y})$ becomes stable. Let $\tau_1 = [\![\psi]\!]$ with $\mathbf{E}(\mathbf{x}) = \sigma_1$. However, due to the monotonicity property, $\tau_0 \subseteq \tau_1$. This implies that, instead of computing τ_1 as a fixpoint starting with $\mathbf{E}(\mathbf{y})$ as empty set, we can speed up the convergence by choosing $\mathbf{E}(\mathbf{y})$ to be τ_0 in the first iteration. That is, there is no need to reinitialize $\mathbf{E}(\mathbf{y})$ from τ_0 to the empty set when $\mathbf{E}(\mathbf{x})$ is updated from σ_0 to σ_1. With this improved policy, *Eval*(χ) is called only n times, rather than n^2 times. The validity of this optimization is captured by the following proposition.

Proposition 10.16 [Optimization in Ctμ model checking] *Let K be an observation structure with finitely branching transition relation, and ϕ be a* CTμ *formula. Let* $\mathbf{E}$ *and* $\mathbf{E}'$ *be region environments such that for every region variable* $\mathbf{y}$ *that is free in* $\mu\mathbf{x}.\,\phi$, $\mathbf{E}(\mathbf{y}) \subseteq \mathbf{E}'(\mathbf{y})$. *Then,*

$$[\![\mu\mathbf{x}.\,\phi]\!]_{\mathbf{E}'} \;=\; \bigcup i \in \mathbb{N}.(\mathcal{F}^{\phi}_{\mathbf{E}'})^i([\![\mu\mathbf{x}.\,\phi]\!]_{\mathbf{E}}),$$

and

$$[\![\nu\mathbf{x}.\,\phi]\!]_{\mathbf{E}} \;=\; \bigcap i \in \mathbb{N}.(\mathcal{F}^{\phi}_{\mathbf{E}})^i([\![\nu\mathbf{x}.\,\phi]\!]_{\mathbf{E}'}).$$

Proof. We consider the case corresponding to the least fixpoints. Whenever a function $\mathcal{F}$ is $\bigcup$-continuous, by Kleene fixpoint theorem, its least fixpoint can be computed by repeatedly applying $\mathcal{F}$ to the minimal element–the empty set: $\mu\mathcal{F} = \bigcup i \in \mathbb{N}.\,\mathcal{F}^i(\emptyset)$. A slight generalization of the Kleene fixpoint theorem states that the least fixpoint of $\mathcal{F}$ can be computed by repeatedly applying $\mathcal{F}$ to any element that is smaller than the least fixpoint; that is, for any $\sigma \subseteq \mu\mathcal{F}$, $\mu\mathcal{F} = \bigcup i \in \mathbb{N}.\,\mathcal{F}^i(\sigma)$.

If K has a finitely branching transition relation, $\mathcal{F}^{\phi}_{\mathbf{E}'}$ is $\bigcup$-continuous. hence, $[\![\mu\mathbf{x}.\,\phi]\!]_{\mathbf{E}'}$ equals $\bigcup i \in \mathbb{N}.\,(\mathcal{F}^{\phi}_{\mathbf{E}'})^i(\sigma)$ for any region $\sigma \subseteq [\![\mu\mathbf{x}.\,\phi]\!]_{\mathbf{E}'}$. It suffices to show that $[\![\mu\mathbf{x}.\,\phi]\!]_{\mathbf{E}} \subseteq [\![\mu\mathbf{x}.\,\phi]\!]_{\mathbf{E}'}$. This can be proved, by induction on the structure of ϕ, using the assumption that for every region variable $\mathbf{y}$ that is free in $\mu\mathbf{x}.\,\phi$, $\mathbf{E}(\mathbf{y}) \subseteq \mathbf{E}'(\mathbf{y})$. ■

The reinitialization is necessary only when there is a switch in the fixpoint quantifiers. The resulting algorithm is shown in Figure 10.2. When *Eval* is invoked on a fixpoint subformula $\mu\mathbf{x}.\,\phi$, the if the enclosing fixpoint subformula is a ν-formula, then $\mathbf{E}(\mathbf{x})$, together with the variables corresponding to μ-subformulas of ϕ that have no enclosing ν-subformula within ϕ, are initialized to the empty set. Otherwise, $\mathbf{E}(\mathbf{x})$ is left unchanged, and equals the value returned by the previous invocation of *Eval*(ϕ).

The algorithm uses the following new operations:

Closed?: **form** $\mapsto \mathbb{B}$. Given a CTμ formula ψ, *Closed?*(ψ) returns *true* if ψ is closed.

Switch?: **form** $\times$ **form** $\mapsto \mathbb{B}$. For CTμ formulas ψ and ϕ, *Switch?*(ψ, ϕ) returns *true* iff there exists a formula χ different from ψ such that (1) ψ is a fixpoint subformula of χ, (2) χ is a fixpoint subformula of ϕ, (3) there is no formula χ' such that χ' is a fixpoint subformula of χ and ψ is a subformula of χ', and (4) the fixpoint-types of ψ and χ are different.

AtomEval. Given a atomic formula p and an observation structure K, *AtomEval*(p, K) returns the characteristic region $[\![p]\!]_K$.

Algorithm 10.1 [Symbolic C$\textsc{t}\mu$ model checking]

Input: a closed C$\textsc{t}\mu$ formula ϕ, and a ϕ-structure K with a finitely-branching transition relation.
Output: the answer to the model-checking problem (K, ϕ).

local *Done* : **table of form** $\times$ **region**; $\mathbf{E}$: **table of var** $\times$ **region**
$\Sigma := \mathit{AtomEval}(\mathit{true}, K)$;
$\mathit{Done} := \mathit{EmptyTable}$; $\mathbf{E} := \mathit{EmptyTable}$;
if $\mathit{InitReg}(K) \subseteq \mathit{Eval}(\phi)$ **then return** YES **else return** NO.

function *Eval*
input ψ: **form**
  if $\mathit{Closed?}(\psi)$ **and** $\mathit{Done}[\psi] \neq \perp$ **then return** $\mathit{Done}[\psi]$ **fi**;
  case $\psi = p$ for an atomic formula p: $\sigma := \mathit{AtomEval}(p, K)$
  case $\psi = \neg p$ for an atomic formula p: $\sigma := \Sigma \setminus \mathit{AtomEval}(p, K)$
  case $\psi = \chi_1 \vee \chi_2$: $\sigma := \mathit{Eval}(\chi_1) \cup \mathit{Eval}(\chi_2)$
  case $\psi = \chi_1 \wedge \chi_2$: $\sigma := \mathit{Eval}(\chi_1) \cap \mathit{Eval}(\chi_2)$
  case $\psi = \exists\bigcirc\, \chi$: $\sigma := \mathit{PreReg}(\mathit{Eval}(\chi), K)$
  case $\psi = \forall\bigcirc\, \chi$: $\sigma := \Sigma \setminus \mathit{PreReg}(\Sigma \setminus \mathit{Eval}(\chi), K)$
  case $\psi = \mu\mathbf{x}.\, \chi$:
    if $\mathit{Switch?}(\psi, \phi)$ **or** $\mathit{Closed?}(\psi)$ **then** $\mathit{Initialize}(\psi, mu)$ **fi**;
    repeat $\sigma := \mathbf{E}(\mathbf{x})$; $\mathbf{E}(\mathbf{x}) := \mathit{Eval}(\chi)$ **until** $\sigma = \mathbf{E}(\mathbf{x})$;
  case $\psi = \nu\mathbf{x}.\, \chi$:
    if $\mathit{Switch?}(\psi, \phi)$ **or** $\mathit{Closed?}(\psi)$ **then** $\mathit{Initialize}(\psi, nu)$ **fi**;
    repeat $\sigma := \mathbf{E}(\mathbf{x})$; $\mathbf{E}(\mathbf{x}) := \mathit{Eval}(\chi)$ **until** $\sigma = \mathbf{E}(\mathbf{x})$;
  case $\psi = \mathbf{x}$: $\sigma := \mathbf{E}(\mathbf{x})$;
  end case
  if $\mathit{Closed?}(\psi)$ **then** $\mathit{Done}[\psi] := \sigma$;
  return σ
  end.

function *Initialize*
input ψ: **form**; m : $\{mu, nu\}$
  case $\psi = p$ for an atomic formula p:
  case $\psi = \neg p$ for an atomic formula p:
  case $\psi = \chi_1 \vee \chi_2$: $\mathit{Initialize}(\chi_1, m)$; $\mathit{Initialize}(\chi_2, m)$
  case $\psi = \chi_1 \wedge \chi_2$: $\mathit{Initialize}(\chi_1, m)$; $\mathit{Initialize}(\chi_2, m)$
  case $\psi = \exists\bigcirc\, \chi$: $\mathit{Initialize}(\chi, m)$
  case $\psi = \forall\bigcirc\, \chi$: $\mathit{Initialize}(\chi, m)$
  case $\psi = \mu\mathbf{x}.\, \chi$:
    if $m = mu$ **then** $\mathbf{E}(\mathbf{x}) := \mathit{EmptySet}$; $\mathit{Initialize}(\chi, m)$ **fi**
  case $\psi = \nu\mathbf{x}.\, \chi$:
    if $m = nu$ **then** $\mathbf{E}(\mathbf{x}) := \Sigma$; $\mathit{Initialize}(\chi, m)$ **fi**
  case $\psi = \mathbf{x}$:
  end case
  end.

Figure 10.2: Symbolic C$\textsc{t}\mu$ model checking

Theorem 10.6 [Correctness of $\text{C}\textsc{t}\mu$ model checking] *Given an observation structure K with finite bisimulation, and a closed $\text{C}\textsc{t}\mu$ formula ϕ, Algorithm 10.1 terminates with the correct answer to the model checking problem (K, ϕ).*

Theorem 10.7 [Complexity of $\text{C}\textsc{t}\mu$ model checking] *Let K be a finite observation structure with n states and m transitions, and let ϕ be a closed $\text{C}\textsc{t}\mu$ formula with length ℓ and alternation-depth k. Algorithm 10.1 solves the model checking problem (K, ϕ) in time $O((\ell \cdot (m+n))^{k+1})$.*

If the input structure for Algorithm 10.1 is finite, then all state predicates that are computed by the algorithm can be viewed as propositional formulas. An implementation of symbolic $\text{C}\textsc{t}\mu$ model checking for finite observation structures, then, may use BDDs. By Theorem 10.5, we can reduce the verification problem for CTL over fair modules to the $\text{C}\textsc{t}\mu$ verification problem. Consequently, we have symbolic procedure for CTL verification.