CSC410

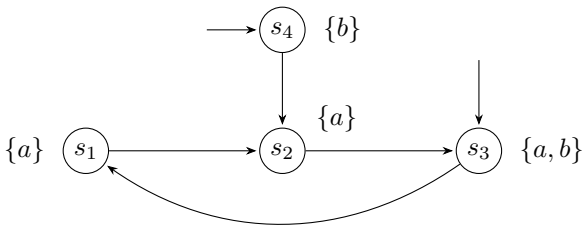
LTL, CTL, AND FIXPOINTS

Solutions

Common assumptions to all problems: transition systems are finite, have a nonempty set of initial states, and have no terminal states. In the LTL fixpoint problems, unknowns range over arbitrary subsets of $(2^{AP})^\omega$, the set of all infinite paths.

Problem 1 Warm-up: LTL Model Checking

For the following transition system,

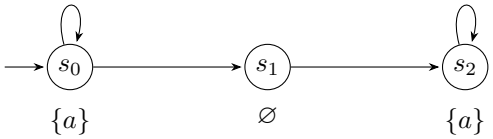


determine which states satisfy each given LTL formula below:

(a)	$\bigcirc a$	(d)	$\Box\Diamond a$
(b)	$\bigcirc\bigcirc\bigcirc a$	(e)	$\Box(b \mathbin{\mathsf{U}} a)$
(c)	$\Box b$	(f)	$\Diamond(a \mathbin{\mathsf{U}} b)$

Solution. (a) all; (b) all; (c) none (s_1 is on every path and lacks b); (d) all; (e) all; (f) all.

Problem 2 LTL vs CTL



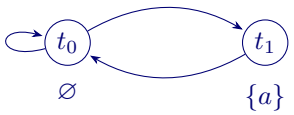
(a) Which states satisfy the LTL formula $\Diamond\Box a$? Which satisfy the CTL formula $\forall\Diamond\forall\Box a$?

Solution. $\Diamond\Box a$: all of s_0, s_1, s_2 (every path either stays in s_0 forever or ends in s_2^ω). $\forall\Box a$ holds only at s_2, so $\forall\Diamond\forall\Box a$ holds at $\{s_1, s_2\}$; it fails at s_0 because of the path s_0^ω, which never reaches a state where $\forall\Box a$ holds (s_0 itself can still escape to s_1).

Observe that the example demonstrates that a transition system satisfying $\Diamond\Box a$ does not necessarily satisfy $\forall\Diamond\forall\Box a$, because the latter is a strictly stronger condition than necessary to describe "stability", as discussed during the CTL inverted lecture.

(b) Design a transition system with at most two states in which every state satisfies $\forall\Box\exists\Diamond a$ but some state violates $\Box\Diamond a$. What does this tell you?

Solution.



From anywhere an a-state is reachable, so $\forall\Box\exists\Diamond a$ holds at both; the path t_0^ω violates $\Box\Diamond a$. $\forall\Box\exists\Diamond a$ has no LTL equivalent.

Problem 3 Mind Your Negations

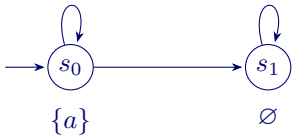
(a) Fill in *yes* or *no*: *does there exist* a transition system TS and a formula φ such that ...

	$TS \models \varphi$ *and* $TS \models \neg\varphi$		$TS \not\models \varphi$ *and* $TS \not\models \neg\varphi$	
	one initial state	several	one initial state	several
φ in LTL	no	no	yes	yes
φ in CTL	no	no	no	yes

Solution. The "no" answers are expected. So, let's focus on the right half of the table. LTL quantifies over *paths*, so even with one initial state, one can have one path satisfying φ and another violating it. A CTL formula is a *state* formula, so with a single initial state s_0, exactly one of $s_0 \models \Phi$, $s_0 \models \neg\Phi$ holds. With several initial states, take one satisfying Φ and one violating it. Hence, "$TS \models \neg\varphi$" is *stronger* than "$TS \not\models \varphi$". The two are guaranteed to coincide only for CTL and only for transition systems with one initial state.

(b) Give a transition system with exactly one initial state such that $TS \not\models \Box a$ and $TS \not\models \neg\Box a$.

Solution.



$s_0^\omega \models \Box a$, so $TS \not\models \neg\Box a$; $s_0 s_1^\omega \not\models \Box a$.

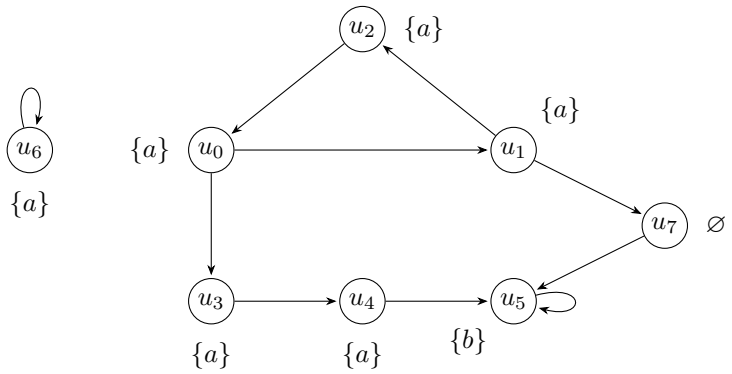
(c) True or false?

	LTL, one init.	CTL, one init.	CTL, several
$TS \models \varphi \wedge \psi \iff TS \models \varphi$ and $TS \models \psi$	T	T	T
$TS \models \varphi \vee \psi \iff TS \models \varphi$ or $TS \models \psi$	F	T	F

Solution. $\wedge$ commutes with every universal quantifier. $\vee$ does not: in the system of (b), $TS \models \Box a \vee \neg\Box a$ but neither disjunct holds. For CTL with one initial state, the claim reduces to the state semantics of $\vee$.

Problem 4 Computing fixpoints

This is a borderline lame exercise to just tell you what you should have learned from the CTL model checking lecture. Consider the following transition system:



(a) Compute $Sat(\exists(a \,\mathsf{U}\, b))$. (Reminder: You are computing the least fixpoint by iterating through $T_0 = Sat(b)$, $T_{i+1} = T_i \cup \{s \in Sat(a) \mid Post(s) \cap T_i \neq \varnothing\}$.)

Solution. $T_0 = \{u_5\}$, $T_1 = \{u_4, u_5\}$, $T_2 = \{u_3, u_4, u_5\}$, $T_3 = \{u_0, u_3, u_4, u_5\}$, $T_4 = \{u_0, u_2, u_3, u_4, u_5\}$, $T_5 = \{u_0, u_1, u_2, u_3, u_4, u_5\} = T_6$. Note $u_7 \notin$ (neither a nor b) and $u_6 \notin$ (never reaches b).

(b) Compute $Sat(\exists\Box a)$. (Reminder: you are computing the greatest-fixpoint by iterating through $T_0 = Sat(a)$, $T_{i+1} = \{s \in T_i \mid Post(s) \cap T_i \neq \varnothing\}$.)

Solution. $T_0 = \{u_0, u_1, u_2, u_3, u_4, u_6\}$, $T_1 = \{u_0, u_1, u_2, u_3, u_6\}$ (u_4 drops), $T_2 = \{u_0, u_1, u_2, u_6\}$ (u_3 drops) $= T_3$.

(c) Now repeat (a), but start from $T_0 = S$ (all states) and replace $\cup$ by recomputation: $T_{i+1} = Sat(b) \cup \{s \in Sat(a) \mid Post(s) \cap T_i \neq \varnothing\}$. What set do you get, and which CTL formula does it compute?

Solution. All of S except u_7: $\{u_0, \ldots, u_6\}$. This is the *greatest* solution of $F \equiv b \vee (a \wedge \exists \bigcirc F)$, i.e. $\exists(a \,\mathsf{W}\, b) = \exists(a \,\mathsf{U}\, b) \vee \exists\Box a$. The difference with (a) is exactly u_6.

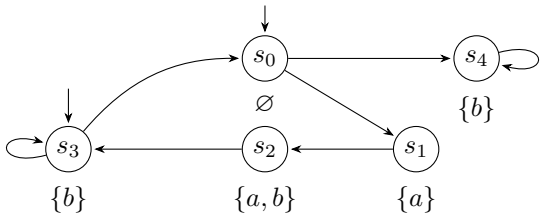
(d) Repeat (b) starting from $T_0 = \varnothing$. What do you get, and why is this useless?

Solution. $\varnothing$ immediately — the least solution of $F \equiv a \wedge \exists \bigcirc F$ is always *false*. $\exists\Box$ needs the greatest fixpoint; starting from below can never "discover" an infinite path.

Problem 5 CTL Model Checking

For the following transition system and the given formulas, determine which states satisfy each given formula:

$\Phi_1 = \forall(a \,\mathsf{U}\, b) \vee \exists \bigcirc (\forall\Box b)$
$\Phi_2 = \forall\Box\forall(a \,\mathsf{U}\, b)$
$\Phi_3 = (a \wedge b) \rightarrow \exists\Box\exists \bigcirc \forall(b \,\mathsf{W}\, a)$
$\Phi_4 = \forall\Box\exists\Diamond\Phi_3$



Solution. Bottom-up analysis exactly as in the syntax-tree picture from the slides:

- $\forall(a \,\mathsf{U}\, b)$: $\{s_1, s_2, s_3, s_4\}$ (s_0 has neither a nor b). $\forall\Box b$: $\{s_4\}$ (s_3 can return to s_0). $\exists \bigcirc \forall\Box b$: $\{s_0, s_4\}$.
  $Sat(\Phi_1) = S$.
- $Sat(\Phi_2) = \{s_4\}$: from s_1, s_2, s_3 one reaches s_0, where $\forall(a \,\mathsf{U}\, b)$ fails.
- $\forall(b \,\mathsf{W}\, a) \equiv \neg\exists((b \wedge \neg a) \,\mathsf{U}\, (\neg a \wedge \neg b))$: $\{s_1, s_2, s_4\}$ (s_0 fails at once; s_3 fails via $s_3 s_0$). $\exists\bigcirc$ of it: $\{s_0, s_1, s_4\}$. $\exists\Box$ of that: $\{s_0, s_4\}$ (witness $s_0 s_4^\omega$). Only s_2 satisfies $a \wedge b$, and $s_2 \notin \{s_0, s_4\}$, so $Sat(\Phi_3) = \{s_0, s_1, s_3, s_4\}$.
- $Sat(\Phi_4) = S$: Every state except s_2 already satisfies Φ_3, and $s_2 \rightarrow s_3$.

$TS \models \Phi_1, \Phi_4$; $TS \not\models \Phi_2$; $TS \models \Phi_3$ (both initial states s_0, s_3 satisfy it).

Problem 6 LTL Expansion Laws: Understanding Least/Greatest Fixpoint Solutions

For each equation over LTL path sets, give the *least* and *greatest* solutions as standard LTL formulas (or state that the solution is not LTL-expressible).

equation	least	greatest
$X \equiv a \wedge \bigcirc X$	false	$\Box a$
$X \equiv a \vee \bigcirc X$	$\Diamond a$	true
$X \equiv \bigcirc X$	false	true
$X \equiv b \vee (a \wedge \bigcirc X)$	$a \,\mathsf{U}\, b$	$a \,\mathsf{W}\, b$
$X \equiv \Diamond(a \wedge \bigcirc X)$	false	$\Box\Diamond a$
$X \equiv \Box(a \wedge \bigcirc X)$	false	$\Box a$
$X \equiv a \wedge \bigcirc \bigcirc X$	false	none in LTL

Solution. Note that "false" is the empty set, and "true" is the set of all paths.

Cheap trick: Plug in both false and true and see if the equation is satisfied. If they are then they must be (respectively) the least and greatest fixpoints. This leaves you with the remaining fixpoint to figure out. This trick would get you results for all rows but row 4 above.

Row 5: any solution Y satisfies $Y \Rightarrow \Diamond(a \wedge \bigcirc Y)$, so from a Y-path we see a and are back in Y; repeating gives infinitely many a's, so $Y \Rightarrow \Box\Diamond a$, and $\Box\Diamond a$ is itself a solution. Row 7: the greatest solution is "a holds at every even position", which is *not* LTL-expressible! Fixpoints are strictly more expressive than LTL (and CTL) — this is the motivation for the μ-calculus.

(a) Reverse direction: write $\Diamond\Box a$ as the least or greatest solution of an equation that uses the formula's arguments.

Solution. $\Diamond\Box a$: least solution of $X \equiv \Box a \vee \bigcirc X$.

(b) Row 5 hides a second fixpoint inside $\Diamond$. Unfolding $\Diamond$ by its own expansion law produces the following two equations that use only a, $\wedge$, $\vee$, $\bigcirc$:

$$X \equiv Y \qquad\qquad Y \equiv (a \wedge \bigcirc X) \vee \bigcirc Y$$

Compute X as:

1. the least solution for both X and Y;
2. the greatest solution for both;
3. the greatest for X, and for each candidate X the least Y. Note this is formally written as $\nu X.\, \mu Y.\, (a \wedge \bigcirc X) \vee \bigcirc Y$ (read greatest X such that least Y such that ...).
4. Now swap the two from the previous part: $\mu Y.\, \nu X.\, (a \wedge \bigcirc X) \vee \bigcirc Y$.
5. Now redo this analysis for row 6, $X \equiv \Box(a \wedge \bigcirc X)$. Unfold $\Box$ by its expansion law $\Box\psi \equiv \psi \wedge \bigcirc\Box\psi$. What kind of fixpoint does the hidden $\Box$ need?

Solution.

1. false: $(\mathit{false}, \mathit{false})$ is a solution, and it is the smallest one.
2. true: $(a \wedge \bigcirc \mathit{true}) \vee \bigcirc \mathit{true} = \mathit{true}$, so $(\mathit{true}, \mathit{true})$ is a solution.
3. $\Box\Diamond a$. For a fixed X, the least Y is $\Diamond(a \wedge \bigcirc X)$; the outer greatest fixpoint then gives row 5.
4. $\Diamond\Box a$. For a fixed Y, the greatest X is "a forever, or a until a step into Y". The outer least fixpoint then forces the play to stop calling on Y after finitely many steps, so eventually a holds forever.
5. $X \equiv Y$, $Y \equiv (a \wedge \bigcirc X) \wedge \bigcirc Y$. The hidden $\Box$ is a *greatest* fixpoint, the same kind as the outer X (we want the greatest solution of row 6).
   We can safely substitute a variable and get rid of the other one: $X \equiv a \wedge \bigcirc X \wedge \bigcirc X \equiv a \wedge \bigcirc X$. The greatest solution is $\Box a$, matching the table.
   If you are interested in knowing precisely why substitution is safe, and generally learning more than the scope of the course, you can look up Bekić's lemma (for the precise reason for the safety of substitution), and read more about *modal logic* to understand fixpoints and their applications.

Key point: When two variables need *different* kinds of fixpoint, the *order of nesting* matters. Cases 1 and 2 also show that $\Box\Diamond a$ is not the least or greatest solution of any *single-kind* reading of these equations.

Problem 7 Conditions for Fixpoint Solutions

(a) Show that no set of infinite paths X satisfies $X \equiv \neg \bigcirc X$. *Hint:* think about a path of the form p^ω, where p is an atomic proposition.

Solution. The equation says: for every path π, $\pi \in X \iff \pi^1 \notin X$, where π^1 is π with its first position dropped. Let $\pi = p^\omega$ for some atomic proposition p; then $\pi^1 = \pi$. Now let X be *any* set of paths. Either $\pi \in X$ or $\pi \notin X$:

- if $\pi \in X$, the equation requires $\pi^1 \notin X$, i.e. $\pi \notin X$ — contradiction;
- if $\pi \notin X$, the equation requires $\pi^1 \in X$, i.e. $\pi \in X$ — contradiction.

So X is not a solution. Since X was arbitrary, the equation has no solution.

(b) What property do the expansion law equations have that guarantees them a (least) fixpoint solution, which this one lacks?

Solution. Monotonicity: $X \subseteq Y$ implies $f(X) \subseteq f(Y)$. By Knaster–Tarski, a monotone function on a complete lattice has least and greatest fixpoints (Tarski, *Pacific J. Math.* 5, 1955). $f(X) = \neg \bigcirc X$ is not monotone: $\mathit{false} \subseteq \mathit{true}$, but $f(\mathit{false}) = \neg \bigcirc \mathit{false} = \mathit{true} \not\subseteq \mathit{false} = \neg \bigcirc \mathit{true} = f(\mathit{true})$. We will see more on this front in the next course topic when we study dataflow analyses.

Problem 8 Two Unknowns

Consider the following three simple constraints about three unknown LTL formulas F, G, and H:

$$F \equiv a \vee G \qquad\qquad G \equiv b \wedge \bigcirc F$$

Find (standard non-recursive) LTL formulas to stand for F and G above such that the constraints are satisfied and the formulas represent the *smallest* set of paths satisfying the constraints.

(a) $F \equiv b \,\mathsf{U}\, a$ (b) $G \equiv b \wedge \bigcirc(b \,\mathsf{U}\, a)$

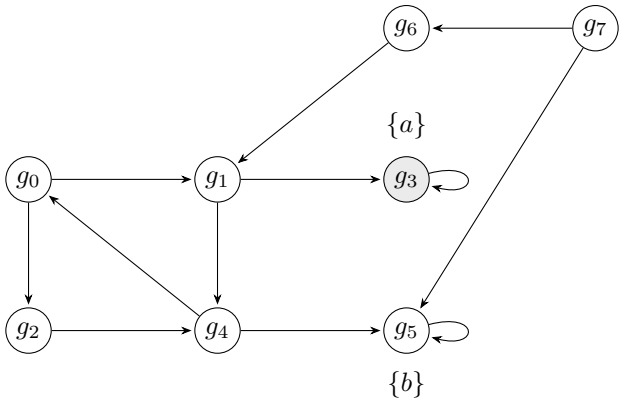
What if we are interested in the formulas representing the *largest* set of paths satisfying the constraints?

(a) $F \equiv b \,\mathsf{W}\, a$ (b) $G \equiv b \wedge \bigcirc(b \,\mathsf{W}\, a)$

Solution. Substitute G into F: $F \equiv a \vee (b \wedge \bigcirc F)$, which is exactly the expansion law of $b \,\mathsf{U}\, a$. Least solution: $F = b \,\mathsf{U}\, a$; greatest: $F = b \,\mathsf{W}\, a$. Then G is determined by its own equation.

Problem 9 CTL and Games

Two players, **Eve** ($\exists$) and **Adam** ($\forall$), push a single token along the edges of the graph below. They *alternate*: whoever is to move picks a successor of the current state and moves the token there. **Eve wants the token to reach an a-state** (here, g_3) and Adam wants to prevent this forever. These games are called *reachability games* for this reason.



Unlabelled states are labelled $\varnothing$. Ignore b until part (e).

(a) Warm up to understand the game: Put the token on g_7 with Eve to move. Who wins, and how? Same question for g_0.

Solution. **From g_7, Eve wins.** She moves to g_6; Adam has no choice but g_1; Eve moves to g_3. **From g_0, Adam wins.** Adam's strategy: *at g_1 go to g_4; at g_0 go to g_2; at g_4 do anything.* Check it. If Eve plays $g_0 \to g_1$, Adam plays g_4. If Eve plays $g_0 \to g_2$, Adam is forced to g_4. Either way the token is at g_4 with Eve to move. If she goes to g_5 she is stuck there forever. If she goes to g_0, Adam answers g_2, Eve is forced to g_4, and now *Adam* is at g_4 and can end it at g_5. The token never reaches g_1 with Eve to move, which is the only way into g_3.

(b) Let X be the set of states from which Eve wins *if it is her turn*, and Y the set from which Eve wins *if it is Adam's turn*. Explain why (X, Y) is a solution of

$$X \equiv a \vee \exists \bigcirc Y \qquad\qquad Y \equiv a \vee \forall \bigcirc X.$$

What is the difference between a *least* solution and a *greatest* solution here, and which do we want?

Solution. Eve, to move at s, wins iff the token is already at a, or she can pick *some* successor from which she wins with Adam to move: $a \vee \exists \bigcirc Y$. Adam, to move at s, cannot stop Eve iff the token is at a, or *every* successor he may pick is a win for Eve with her to move: $a \vee \forall \bigcirc X$. So the turn structure is exactly $\exists\bigcirc$ (Eve chooses) alternating with $\forall\bigcirc$ (Adam chooses).
Least: Eve must reach a in *finitely many* moves. A solution that is too big would also credit Eve with plays that loop forever without seeing a. That is the same reason $\exists(\Phi \,\mathsf{U}\, \Psi)$ is a least fixpoint.

(c) Compute the least solution by simultaneous iteration from $(X_0, Y_0) = (\varnothing, \varnothing)$.

Solution.

i	$X_i = a \vee \exists \bigcirc Y_{i-1}$	$Y_i = a \vee \forall \bigcirc X_{i-1}$	what changed
1	$\{g_3\}$	$\{g_3\}$	the target itself
2	$\{g_1, g_3\}$	$\{g_3\}$	Eve at g_1 moves to g_3
3	$\{g_1, g_3\}$	$\{g_3, g_6\}$	Adam at g_6 is forced to g_1
4	$\{g_1, g_3, g_7\}$	$\{g_3, g_6\}$	Eve at g_7 moves to g_6
5	same as round 4		stable

So Eve wins from $\{g_1, g_3, g_7\}$ if it is her turn to move, and from $\{g_3, g_6\}$ if it is Adam's turn to move. This agrees with (a): $g_7 \in X$, $g_0 \notin X$.

(d) What does round i mean in game terms? How does this help devise a winning strategy for Eve from the table? What is the greatest solution?

Solution. **Meaning of the rounds.** X_i (Y_i) is the set of states from which Eve can force a within $i-1$ moves if it is Eve's (Adam's) turn. Examples: $g_1 \in X_2$ (1 move), $g_6 \in Y_3$ (2 moves), $g_7 \in X_4$ (3 moves). This is the game version of Problem 10(c), where the iterates of $\exists$U are bounded untils.
Strategy. Rank a *state–turn pair* by its first entry round:

$$r_E(s) = \min\{i \mid s \in X_i\}, \qquad r_A(s) = \min\{i \mid s \in Y_i\},$$

with rank ∞ if the state never enters that column. At a non-target state with finite $r_E(s)$, Eve chooses a successor t with $r_A(t) < r_E(s)$; one exists by $\exists\bigcirc$. At a non-target state with finite $r_A(s)$, every successor t satisfies $r_E(t) < r_A(s)$, by $\forall\bigcirc$. The rank decreases on each move until the target is reached. This is the *attractor* construction for reachability games (Grädel, Thomas & Wilke (eds.), *Automata, Logics, and Infinite Games*, LNCS 2500, 2002, Ch. 2).
Greatest solution. The greatest solution is simply (S, S) (where S is the set of all states). With $Y = S$, we get $\exists \bigcirc Y = S$, so we have $X = S$, and similarly $Y = S$. In game terms, the greatest fixpoint counts *infinite* play as a win for Eve. Since no state is terminal, play never gets stuck, so Eve would "win" everywhere.

(e) *Safety.* Change Eve's goal to: **keep the token away from b (i.e. g_5) forever**; Adam wants to reach b. These are called *safety games* with the idea that the states being avoided are unsafe ones. Write the two equations (in the style of (b)), decide between the least/greatest fixpoint appropriate for this case, and compute it.

Solution.

$$X \equiv \neg b \wedge \exists \bigcirc Y$$
$$Y \equiv \neg b \wedge \forall \bigcirc X$$

Greatest fixpoint: Eve's goal has now changed to play *forever*, why avoiding b. This has a similar flavour to $\exists\Box$. We iterate starting from (S, S), and alternately remove states from columns until we reach the greatest fixpoint.

i	X_i	Y_i
1	S'	S'
2	S'	$\{g_0, g_1, g_2, g_3, g_6\}$ (g_4, g_7: Adam can move to g_5)
3	$\{g_0, g_1, g_3, g_4, g_6, g_7\}$	same (g_2: Eve forced to $g_4 \notin Y_2$)
4	same	$\{g_1, g_2, g_3, g_6\}$
5	$\{g_0, g_1, g_3, g_6, g_7\}$	same
6	same	$\{g_3, g_6\}$
7	$\{g_1, g_3, g_7\}$	$\{g_3, g_6\}$ stable

Problem 10 Bounded Until

Define *bounded until*: $\pi \models \varphi \,\mathsf{U}^{\leq k}\, \psi$ iff ψ holds at some position $j \leq k$ of π and φ holds at all positions before j.

(a) Write an expansion law for $\varphi \,\mathsf{U}^{\leq 0}\, \psi$ and for $\varphi \,\mathsf{U}^{\leq k+1}\, \psi$.

Solution. $\varphi \,\mathsf{U}^{\leq 0}\, \psi \equiv \psi$; $\quad \varphi \,\mathsf{U}^{\leq k+1}\, \psi \equiv \psi \vee (\varphi \wedge \bigcirc(\varphi \,\mathsf{U}^{\leq k}\, \psi))$.

(b) For ordinary U we had to say "least". Why is there no least/greatest choice to make here?

Solution. The recursion is on k, which strictly decreases; it bottoms out at $k = 0$. A well-founded recursion has exactly one solution. The cycle $X = \ldots X \ldots$ that allowed several solutions is gone.

(c) Prove that the iterates of the least-fixpoint algorithm for $\exists(\Phi \mathsf{U} \Psi)$ satisfy $T_k = Sat(\exists(\Phi \mathsf{U}^{\leq k} \Psi))$.

Solution. Induction on k. $T_0 = Sat(\Psi)$ matches (a). Step: $T_{k+1} = T_k \cup (Sat(\Phi) \cap \exists \bigcirc T_k)$; by IH and (a) this is $Sat(\exists(\Phi \,\mathsf{U}^{\leq k}\, \Psi)) \cup Sat(\Phi \wedge \exists \bigcirc \exists(\Phi \,\mathsf{U}^{\leq k}\, \Psi)) = Sat(\exists(\Phi \,\mathsf{U}^{\leq k+1}\, \Psi))$.

(d) Consequence: if state s first appears in T_k, what does k tell you? Check it on u_1 in Problem 4.

Solution. k is the length of the *shortest* witness. The algorithm is a backward breadth-first search. u_1 enters at $k = 5$; its shortest witness is $u_1u_2u_0u_3u_4u_5$ (5 steps; going via u_7 is not allowed since $u_7 \not\models a$).

(e) Tell the analogous story for the $\exists\Box\Phi$ iteration: what does T_k mean, and why does stabilization justify "forever"?

Solution. $T_k = \{s \mid$ some path from s satisfies Φ at positions $0, \ldots, k\}$. Once $T_{k+1} = T_k$, every state of T_k has a successor in T_k, so one can extend a Φ-path step by step forever. Finiteness of S guarantees stabilization within $|S|$ rounds.

Problem 11 CTL Model Checking Time Complexity

(a) For every $k \geq 1$, design a transition system on which the $\exists\Box a$ iteration of Problem 4(b) removes states in exactly k rounds before stabilizing.

Solution. A chain $v_1 \to v_2 \to \cdots \to v_k \to w$, all v_i labelled $\{a\}$, w labelled $\varnothing$ with a self-loop. Round i removes exactly v_{k-i+1}.

(b) Give an upper bound on the number of rounds of either iteration in terms of $|S|$. Does the size of Φ matter?

Solution. At most $|S|$ rounds: each non-final round adds (resp. removes) at least one state. The formula size only multiplies the cost, since each subformula is computed once, bottom-up.

(c) The naive iteration costs $O(|S| \cdot (|S| + |\to|))$ per operator. How can $\exists(\Phi \,\mathsf{U}\, \Psi)$ be done in time linear in $|S| + |\to|$?

Solution. Backward search: start from $Sat(\Psi)$ and follow edges backwards, adding a predecessor if it satisfies Φ; each edge is examined once. For $\exists\Box\Phi$, restrict to $Sat(\Phi)$ and keep states that can reach an SCC containing a cycle (or decrement successor counters). Overall CTL model checking is $O(|\Phi| \cdot (|S| + |\to|))$ (Clarke, Emerson & Sistla, *ACM TOPLAS* 8(2), 1986).