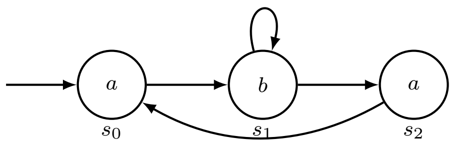
6.0 ECTS/4.5h VU Programm- und Systemverifikation (184.741)
29 September 2023

Kennzahl (study id)	Matrikelnummer (student id)	Familienname (family name)	Vorname (first name)	Platz (seat)

1.) Consider the following Kripke Structure:



For each formula, give the states of the Kripke structure for which the formula holds. In other words, for each of the states from the set $\{s_0, s_1, s_2\}$, consider the computation trees starting at that state, and for each tree, check whether the given formula holds on it or not.

(a) $\mathbf{EG}\, a$ *Solution:* $\emptyset$

(b) $\mathbf{EG}\,\mathbf{F}\, a$ *Solution:* $\{s_0, s_1, s_2\}$

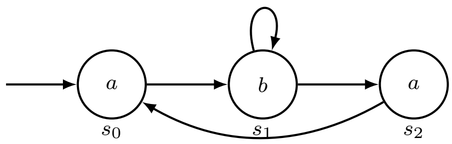
(c) $\mathbf{A}(a \wedge \mathbf{X}\, b)$ *Solution:* $\{s_0\}$

(d) $\mathbf{A}(a\, \mathbf{U}\, b)$ *Solution:* $\{s_0, s_1, s_2\}$

(e) $\mathbf{E}(b\, \mathbf{U}\, a)$ *Solution:* $\{s_0, s_1, s_2\}$

(f) $(\mathbf{AX}\, a) \vee (\mathbf{AX}\, b)$ *Solution:* $\{s_0, s_2\}$

2.) Consider the following Kripke structure with initial state s_0:



Use the tableaux algorithm from the lecture to compute the sets of states in which the formula $\mathbf{EG}\,(\mathbf{EX}\, a)$ (and its subformulas) hold.

- For every subformula, compute the states for which it holds!
- For fixpoints, list every step of the computation!

Solution: $\{s_1\}$. We use the tableaux algorithm:

- States that satisfy a: $\{s_0, s_2\}$.
- States that satisfy $\mathbf{EX}\, a$: these are the states with some successor satisfying a, that is, $\{s_1, s_2\}$.
- States that satisfy $\mathbf{EG}\,\mathbf{EX}\, a$: these are the states where some path completely contained within states satisfying $\mathbf{EX}\, a$ start. We compute a fixpoint, starting with $\{s_1, s_2\}$. In each step, we remove elements whithout a successor in the set.

  s_1 has itself as a successor, but s_2 does not have a successor within $\{s_1, s_2\}$. Hence, we remove s_2 from the set of states and we are left with $\{s_1\}$.

  Now, s_1 is the only element in the set and it has itself as a successor, so we have reached a fixpoint at $\{s_1\}$.

3.) Consider the following formula in propositional logic; is it satisfiable?

1	2	3	4	5	6	$\sum$
/18	/10	/10	/09	/06	/07	/60

- If yes, **provide all satisfying assignments** and **explain how you arrived at that number**.
- If not, **provide the CDCL steps leading to that conclusion**. In particular, you must provide the propagated literals and reason clauses leading to each conflict, and the clauses learned from such conflicts.

$$\begin{aligned}(\neg x_1 \vee \neg x_2) \ \wedge \ (x_1 \vee x_2) \ \wedge \ (\neg x_2 \vee \neg x_3) \ \wedge \ (x_2 \vee x_3) \wedge \\ (\neg x_3 \vee \neg x_4) \ \wedge \ (x_3 \vee x_4) \ \wedge \ (\neg x_4 \vee \neg x_5) \ \wedge \ (x_4 \vee x_5) \wedge \\ (\neg x_5 \vee \neg x_6) \ \wedge \ (x_5 \vee x_6) \ \wedge \ (\neg x_6 \vee \neg x_7) \ \wedge \ (x_6 \vee x_7) \wedge \\ (\neg x_1 \vee \neg x_6 \vee x_7) \ \wedge \ (\neg x_1 \vee \neg x_7 \vee x_6) \ \wedge \ (\neg x_6 \vee \neg x_7 \vee x_1) \ \wedge \ (x_1 \vee x_6 \vee x_7)\end{aligned}$$

Solution: A satisfying assignment μ must satisfy either x_1 or $\neg x_1$.

- If μ satisfies x_1, then by unit propagation we conclude that:
  - μ satisfies $\neg x_2$, because otherwise μ cannot satisfy the clause $\neg x_1 \vee \neg x_2$.
  - μ satisfies x_3, because otherwise μ cannot satisfy the clause $x_2 \vee x_3$.
  - μ satisfies $\neg x_4$, because otherwise μ cannot satisfy the clause $\neg x_3 \vee \neg x_4$.
  - μ satisfies x_5, because otherwise μ cannot satisfy the clause $x_4 \vee x_5$.
  - μ satisfies $\neg x_6$, because otherwise μ cannot satisfy the clause $\neg x_5 \vee \neg x_6$.
  - μ satisfies x_7, because otherwise μ cannot satisfy the clause $x_6 \vee x_7$.

  However, then μ falsifies the clause $\neg x_1 \vee \neg x_7 \vee x_6$. Therefore, there is no satisfying assignment that also satisfies x_1.
- If μ satisfies $\neg x_1$, then by unit propagation we conclude that:
  - μ satisfies x_2, because otherwise μ cannot satisfy the clause $x_1 \vee x_2$.
  - μ satisfies $\neg x_3$, because otherwise μ cannot satisfy the clause $\neg x_2 \vee \neg x_3$.
  - μ satisfies x_4, because otherwise μ cannot satisfy the clause $x_3 \vee x_4$.
  - μ satisfies $\neg x_5$, because otherwise μ cannot satisfy the clause $\neg x_4 \vee \neg x_5$.
  - μ satisfies x_6, because otherwise μ cannot satisfy the clause $x_5 \vee x_6$.
  - μ satisfies $\neg x_7$, because otherwise μ cannot satisfy the clause $\neg x_6 \vee \neg x_7$.

  We then conclude that, if there is a satisfying assignment that also satisfies $\neg x_1$, then it must be unique assignment satisfying $\neg x_1, x_2, \neg x_3, x_4, \neg x_5, x_6, \neg x_7$. This assignment does indeed satisfy each clause in the formula. Hence, there is exactly one satisfying assignment that also satisfies $\neg x_1$.

In total, there is exactly one satisfying assignment.

4.) Consider the following formulas in Equality Logic with Uninterpreted Functions (EUF); are they satisfiable?

- If yes, provide a satisfying interpretation.
- If not,
  (a) encode the formula as an equisatisfiable formula in equality logic without uninterpreted functions, and
  (b) give the reasoning based on equivalence classes that leads to this conclusion.

(a) $(g = h) \wedge (a = b) \wedge (a = c) \wedge (e \neq i) \wedge (d = e) \wedge (f = e) \wedge (h = i) \wedge (z(c) \neq z(i)) \wedge (a = i)$

Solution: This formula is unsatisfiable. Let us first encode it without uninterpreted functions. We define z_x as $z(x)$:

$$(g = h) \wedge (a = b) \wedge (a = c) \wedge (e \neq i) \wedge (d = e) \wedge (f = e) \wedge (h = i) \wedge (z_c \neq z_i) \wedge (a = i)$$

We then obtain the following equivalence classes:

$$\{a, b, c, i, g, h\} \qquad \{d, e, f\} \qquad \{z_c\} \qquad \{z_i\}$$

We must also include the functional constraint $c = i \to z_c = z_i$. Since c and i are in the same equivalence class, we can then add the atom $z_c = z_i$, so the final equivalence classes are:

$$\{a, b, c, i, g, h\} \qquad \{d, e, f\} \qquad \{z_c, z_i\}$$

We now check that for each disequality $x \neq y$ variables x, y are in different equivalence classes. This holds for $e \neq i$, but it does not hold for $z_c \neq z_i$, so the formula is unsatisfiable.

(b) $(g = h) \wedge (a = b) \wedge (a = c) \wedge (e \neq i) \wedge (d = e) \wedge (f = e) \wedge (h = i) \wedge (z(b) \neq z(f))$

Solution: This formula is satisfiable. For example, the interpretation I with domain $1, 2, 3$ given by

$$I(a) = I(b) = I(c) = 1 \quad I(d) = I(e) = I(f) = 2 \quad I(g) = I(h) = I(i) = 3 \quad I(z) = x \mapsto x$$

satisfies the formula.

(c) $(a = b) \wedge (d = e) \wedge (c = b) \wedge (e = f) \wedge (z(a) = z(d)) \wedge (z(c) \neq z(f))$

Solution: This formula is unsatisfiable. Let us first encode it without uninterpreted functions. We define z_x as $z(x)$:

$$(a = b) \wedge (d = e) \wedge (c = b) \wedge (e = f) \wedge (z_a = z_d) \wedge (z_c \neq z_f)$$

We then obtain the following equivalence classes:

$$\{a, b, c\} \qquad \{d, e, f\} \qquad \{z_a, z_d\} \qquad \{z_c\} \qquad \{z_f\}$$

We must also include the following functional constraints:

$$\begin{array}{lll} (a = c) \to (z_a = z_c) & (a = d) \to (z_a = z_d) & (a = f) \to (z_a = z_f) \\ (c = d) \to (z_c = z_d) & (c = f) \to (z_c = z_f) & (d = f) \to (z_d = z_f) \end{array}$$

Since a and c are in the same equivalence class, the atom $z_a = z_c$ can be added. Similarly, the atom $z_d = z_f$ can be added because d and f are in the same equivalence class. The final equivalence classes are:

$$\{a, b, c\} \qquad \{d, e, f\} \qquad \{z_a, z_c, z_d, z_f\}$$

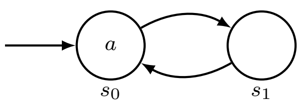
We now check that for each disequality $x \neq y$ variables x, y are in different equivalence classes. This does not hold for $z_c \neq z_f$, so the formula is unsatisfiable.

5.) The unquantified equality logic formula $(a = b) \wedge (c = d) \wedge (f(a) \neq f(c))$ logically implies the formula $d \neq b$.

Solution: True. The question is equivalent to whether $(a = b) \wedge (c = d) \wedge (f(a) \neq f(c)) \wedge (d = b)$ is unsatisfiable, which it is.

6.) The LTL formula $a \wedge \mathbf{G}\,(a \to \mathbf{X}\,\mathbf{X}\,a)$ is logically equivalent to the LTL formula $\mathbf{G}\,a$.

Solution: False. Consider the Kripke structure



This Krikpe structure falsifies $\mathbf{G}\,a$ but satisfies $a \wedge \mathbf{G}\,(a \to \mathbf{X}\,\mathbf{X}\,a)$

7.) The CTL formula $\mathbf{AG}\,\mathbf{AG}\,a$ is logically equivalent to the LTL formula $\mathbf{EG}\,\mathbf{AG}\,a$.

Solution: True. Let us first check that $\varphi = \mathbf{AG}\,\mathbf{AG}\,a$ implies $\psi = \mathbf{EG}\,\mathbf{AG}\,a$. Given a Kripke structure K that satisfies φ, let s be an arbitrary initial state, and a path π based on s. Then, the path π satisfies $\mathbf{G}\,\mathbf{AG}\,a$, and the existence of such a π shows that the state s also satisfies ψ. Since s was an arbitrary initial state, then K satisfies ψ.

Now let us check that ψ implies φ. Given any Kripke structure K satisfying ψ, we consider an arbitrary initial state s. Because K satisfies ψ, we know that there is some path σ based on s that satisfies $\mathbf{G}\,\mathbf{AG}\,a$. In particular, the state s satisfies $\mathbf{AG}\,a$. Now, let π be an arbitrary path based on s, and let $i \geq 0$ be arbitrary. We show that π_i satisfies $\mathbf{AG}\,a$. To do that, let θ be an arbitrary path based on π_i. Then, we can construct a path τ with $\tau_j = \pi_j$ for $j < i$ and $\tau_j = \theta_{j-i}$ for $j \geq i$. The path τ is based on s, so it satisfies $\mathbf{G}\,a$. Hence, the path θ also satisfies $\mathbf{G}\,a$.

Since θ was arbitrary, we have shown that the state π_i satisfies $\mathbf{AG}\,a$. Since i was arbitrary too, we have shown that π satisfies $\mathbf{G}\,\mathbf{AG}\,a$. Since π was arbitrary, we have shown that s satisfies φ. And since s was an arbitrary initial state of K, we have shown that K satisfies φ.

8.) There are formulas that can be represented as a BDD but not as a CNF formula.

Solution: False. All BDDs can be represented as a propositional formula, and all propositional formulas can be represented as a CNF formula.

9.) For any formula in unquantified equality logic, if there is an interpretation (that satisfies the formula) with an infinite domain, there is also an interpretation with a finite domain.

Solution: True, because we can convert a formula in unquantified equality logic to a formula in unquantified equality logic without function symbols.

10.) The equivalence logic formula $(a = b) \wedge (e = f) \wedge (c \neq b) \wedge (c = d) \wedge (z(a) = z(f)) \wedge (z(b) = z(c))$ is satisfiable.

Solution: True. The interpretation I with domain $\{1, 2\}$ and

$$I(a) = I(b) = I(e) = I(f) = 1 \qquad I(c) = I(d) = 2 \qquad I(z) = x \mapsto 1$$

satisfies this formula.