בדיקת תכונות בטיחות רגולריות

הרצאה בקורס מבוא לאימות תוכנה בשיטות פורמליות

הפקולטה למדעי המחשב והמידע | אוניברסיטת בן-גוריון

גרא וייס

5/20/2026

מטרות ההרצאה

נחזור לתכונות בטיחות רגולריות

- נגדיר תכונת בטיחות רגולרית דרך רישות רעות.
- נראה למה מספיק להסתכל על רישות רעות מינימליות.
- נבדיל בין תכונות בטיחות רגולריות ולא רגולריות.

נבנה אלגוריתם בדיקה

- נתרגם בדיקת תכונת בטיחות רגולרית לבדיקת שְׁמוּרָה.
- נבנה מכפלה של מערכת מעברים עם אוטומט סופי.
- נשתמש בהפרת הַשְׁמוּרָה כדי להפיק דוגמה נגדית סופית.

תזכורת: רישות רעות

תכונת בטיחות P_{safe} היא תכונה שבה כל הפרה נחשפת אחרי מספר סופי של צעדים.

$$BadPref(P_{\text{safe}}) = \{\rho \in (2^{AP})^* \mid \forall \sigma \in (2^{AP})^\omega \ (\rho\sigma \notin P_{\text{safe}})\}$$

רישא רעה

מרגע שראינו אותה, שום המשך אינסופי כבר לא יכול לתקן את ההפרה.

דוגמה נגדית סופית

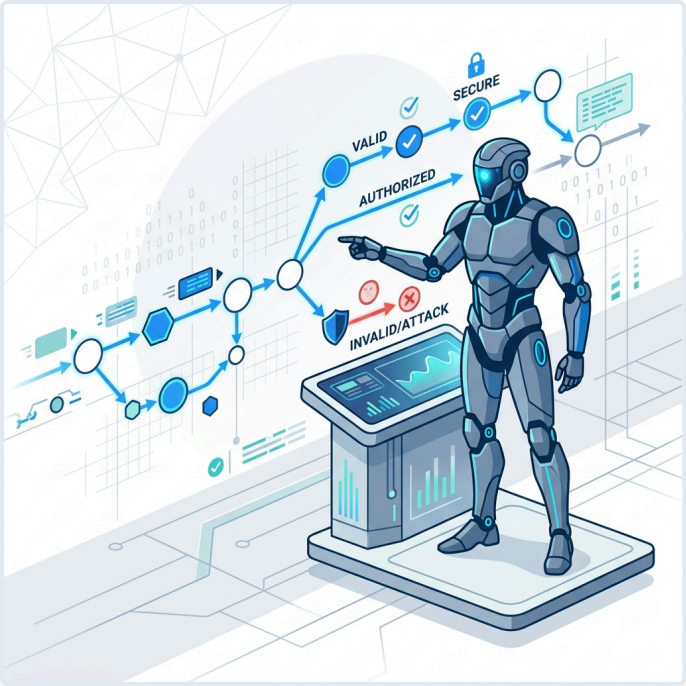
לכן בדיקת בטיחות מחזירה עדות סופית: מסלול שמגיע לרישא רעה.

תכונת בטיחות רגולרית

תכונת בטיחות P_{safe} היא **רגולרית** אם קבוצת הרישות הרעות שלה היא שפה רגולרית:

קבוצת הרישות הרעות $\mathit{BadPref}(P_{\mathrm{safe}})\subseteq (2^{AP})^*$ היא שפה רגולרית.

קיים אוטומט סופי $\mathcal{A}$ מעל האלפבית 2^{AP} כך ש־ $\mathcal{L}$



רישות רעות מינימליות

רישא רעה ρ היא **מינימלית** אם אף רישא ממש שלה אינה רעה.

$$MinBadPref(P_{\text{safe}})$$

למה זה שימושי?

אוטומט עבור רישות מינימליות עוצר בדיוק בנקודת ההפרה הראשונה, ולכן הדוגמה הנגדית שהוא מייצר בדרך כלל קצרה וברורה יותר.

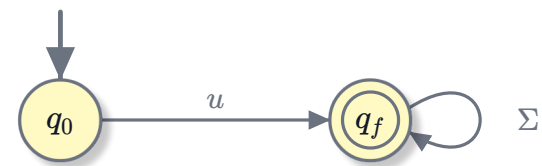
משפט

P_{safe} רגולרית אם ורק אם $\mathit{MinBadPref}(P_{\mathrm{safe}})$ רגולרית.

למה המינימליות שומרת רגולריות?

ממינימליות לכל הרישות הרעות

אם יש אוטומט שמקבל את הרישות הרעות המינימליות, מוסיפים לכל מצב מקבל לולאות על כל אות באלפבית. מרגע שהגענו להפרה, כל המשך נשאר רישא רעה.



מכל הרישות הרעות למינימליות

אם יש אוטומט שמקבל את כל הרישות הרעות, מסירים את כל המעברים היוצאים ממצבים מקבלים. כך מתקבלות רק מילים שבהן הקבלה מתרחשת לראשונה בסוף המילה.

האינטואיציה: רישא רעה היא "נקודת אל־חזור"; רישא רעה מינימלית היא נקודת אל־חזור הראשונה.

דוגמאות

שְׁמוּרָה

כל שְׁמוּרָה היא תכונת בטיחות רגולרית.

$\mathit{MinBadPref} = (2^{AP})^* \neg\Phi$

ראינו אלגוריתמים לבדיקת תכונות שְׁמוּרָה

תכונת בטיחות רגולרית שאינה שְׁמוּרָה

אור אדום נדלק רק אם בצעד הקודם דלק אור צהוב. צריך לזכור צעד אחד אחורה.

אין לנו **עדיין** אלגוריתם לתכונות שאינן שְׁמוּרָה

בטיחות לא רגולרית

בכל רישא סופית, מספר המטבעות שהוכנסו אינו קטן ממספר המשקאות שסופקו. כאן צריך זיכרון לא חסום.

לא נתאר אלגוריתם לתכונות שאינן רגולריות

דוגמה: מניעה הדדית

בתכונת מניעה הדדית דורשים ששני תהליכים לא יהיו באזור הקריטי בו־זמנית.

$$P_{\text{mutex}} = \{\sigma \mid \forall i \geq 0\ (\sigma[i] \not\models (crit_1 \wedge crit_2))\}$$

רישא רעה מינימלית

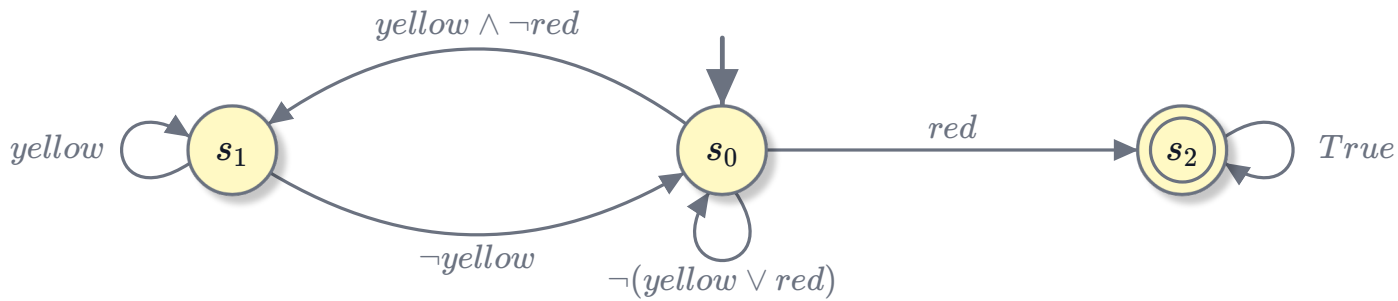
מסלול שמסתיים בפעם הראשונה במצב שבו $crit_1\land crit_2$ נכון.

אוטומט הבדיקה

מספיק אוטומט קטן: כל עוד לא ראינו הפרה נשארים במצב תקין; כשנראה $crit_1\land crit_2$ עוברים למצב מקבל.

דוגמה: רמזור

נדרוש שאור אדום לא יידלק אם לא דלק אור צהוב בדיוק בצעד הקודם.

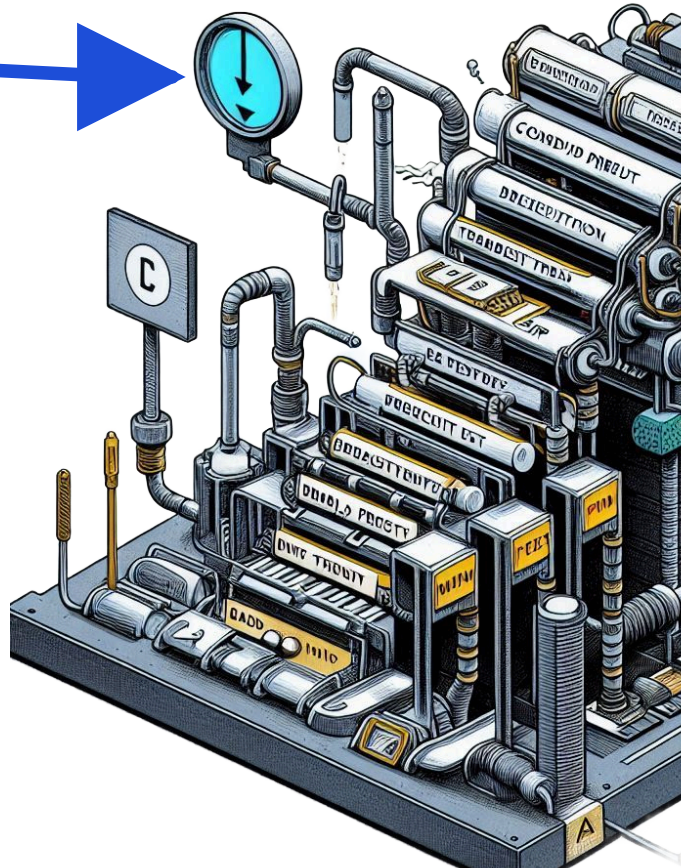


האוטומט זוכר אם בצעד הקודם דלק צהוב; אם אדום מופיע בלי זיכרון כזה, עוברים למצב מקבל.

מוסיפים רכיב בדיקה למערכת

נוסיף למערכת רכיב
שיזהה שנצפתה
רישא רעה

ונבדוק את תכונת
הַשְׁמוּרָה "אף פעם לא
נצפית רישא רעה"



חימום: מונה פשוט

נבדוק את התכונה: בכל שלושה צעדים רצופים, לפחות אחד מקיים ψ.

$$P = \{\sigma \mid \forall i\ (\sigma[i] \models \psi \ \vee\ \sigma[i+1] \models \psi \ \vee\ \sigma[i+2] \models \psi)\}$$

המערכת המקורית

בכל צעד עוברת ממצב s למצב t ומייצרת תווית $L(t)$.

+

מנגנון התראה

שומר מונה $c\in\{0,1,2,3\}$ של מספר הצעדים הרצופים שבהם לא התקיים ψ.

מונה 0

ראינו ψ

מונה 1

צעד אחד בלי ψ

מונה 2

שני צעדים בלי ψ

התראה

שלושה צעדים בלי ψ

הרכבת המונה עם המערכת

נבנה מערכת מעברים חדשה כמכפלה סינכרונית של המערכת המקורית עם מונה דטרמיניסטי, כך שהמונה זוכר כמה צעדים רצופים עברו בלי ψ.

$$TS_{\#} = \langle S \times \{0,1,2,3\}, Act, \rightarrow_{\#}, I_{\#}, AP \cup \{bad\}, L_{\#} \rangle$$

מצבים התחלתיים

מתחילים באותו מצב מערכת, והמונה נקבע לפי התווית הראשונה:

$I_{\#}=\{\langle s,0\rangle\mid s\in I,\ L(s)\models\psi\}\cup\{\langle s,1\rangle\mid s\in I,\ L(s)\not\models\psi\}$

כלל גזירה: עדכון המונה והמעבר במכפלה

המעברים במערכת $TS_{\#}$ מוגדרים על ידי הכלל:

$$\frac{s \xrightarrow{\alpha} t, \quad c' = \begin{cases} 0 & L(t) \models \psi \\ \min(c+1,3) & L(t) \not\models \psi \end{cases}}{\langle s,c\rangle \xrightarrow{\alpha}_{\#} \langle t,c'\rangle}$$

תוויות

התווית המקורית נשמרת, ונוסיף התראה אם ראינו שלושה צעדים רצופים בלי ψ:

$L_{\#}(\langle s,c\rangle)=L(s)\cup\begin{cases}\{bad\} & c=3\\ \emptyset & c\neq 3\end{cases}$

בדיקת שמורה על המערכת החדשה

על המערכת שקיבלנו נוכל לבדוק את תכונת השמורה:

$P_{inv}=\{\sigma\mid \forall i\ge 0\ (\sigma[i]\models\neg bad)\}$

$TS\models P\iff TS_{\#}\models P_{inv}$

רעיון האלגוריתם

נריץ את המערכת ואת אוטומט הרישות הרעות במקביל.

מערכת המעברים מייצרת עקבה:

$L(s_0)L(s_1)L(s_2)\cdots$

+

האוטומט קורא את התוויות ומזהה מתי נוצרה רישא רעה.

אם מגיעים למצב מקבל של האוטומט, מצאנו רישא רעה ולכן מצאנו הפרה של תכונת הבטיחות.

המכפלה עם אוטומט הרישות הרעות

נשתמש באוטומט סופי:

$$\mathcal{A} = \langle Q, 2^{AP}, \delta, Q_0, F \rangle$$

נבנה מערכת מעברים חדשה שבה כל מצב שומר שני רכיבים:

$$TS \times \mathcal{A} = \langle S \times Q, Act, \rightarrow_{\times}, I_{\times}, Q, L_{\times} \rangle$$

הרכיב הראשון הוא מצב המערכת; הרכיב השני הוא מצב האוטומט אחרי קריאת התוויות שראינו עד כה.

הגדרת המכפלה

המצבים ההתחלתיים:

$$I_\times = \{\langle s_0, q\rangle \mid s_0 \in I \wedge \exists q_0 \in Q_0\ (q \in \delta(q_0, L(s_0)))\}$$

המעברים:

$$\frac{s \xrightarrow{\alpha} t \ \wedge\ p \in \delta(q, L(t))}{\langle s, q\rangle \xrightarrow{\alpha}_\times \langle t, p\rangle}$$

התווית במכפלה משקפת את מצב האוטומט:

$$L_\times(\langle s, q\rangle) = \{q\}$$

מה המכפלה מייצגת?

מסלול סופי במכפלה:

$$\langle s_0, q_1 \rangle \langle s_1, q_2 \rangle \cdots \langle s_n, q_{n+1} \rangle$$

קיים אם ורק אם $s_0s_1\cdots s_n$ הוא מסלול סופי התחלתי של TS, וקיימת ריצה של האוטומט:

$$q_0 \xrightarrow{L(s_0)} q_1 \xrightarrow{L(s_1)} q_2 \cdots \xrightarrow{L(s_n)} q_{n+1}$$

לכן מצב מקבל במכפלה פירושו: העקבה הסופית שנוצרה עד כה שייכת לשפת הרישות הרעות.

הַשְׁמוּרָה שמחליפה את הבטיחות

במכפלה כבר אין צורך לדבר על כל השפה הרגולרית. מספיק לבדוק שאף פעם לא מגיעים למצב אוטומט מקבל.

$$P_{\text{inv}} = \{\sigma \in (2^Q)^\omega \mid \forall i \geq 0\ (\sigma[i] \cap F = \emptyset)\}$$

אם הַשְׁמוּרָה מתקיימת, אין רישא רעה באף עקבה של המערכת.

אם הַשְׁמוּרָה מופרת, המסלול שמגיע ל־F הוא דוגמה נגדית סופית.

אינטואיציית הבניה

מערכת המעברים
מייצרת עקבות

האוטומט עוקב אחר
העקבות שנוצרות

$TS\times\mathcal{A}$

קיים מעבר מערכת
המעברים המקורית

קיים מעבר באוטומט
הקורא את התווית של t

$$\frac{s\xrightarrow{\alpha}{\color{blue

מערכת המעברים החדשה מתקדמת בשני הרכיבים,
זה שמייצג את מערכת המעברים המקורית וזה שעוקב אחר האוטומט

דוגמה: מערכת בת 3 מצבים

הגדרת המערכת והאוטומט

מערכת TS:

$S = \{0, 1, 2\}$

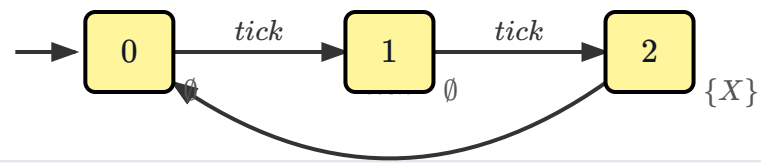
$L(0) = L(1) = \emptyset,\ L(2) = \{X\}$

אוטומט $\mathcal{A}$:

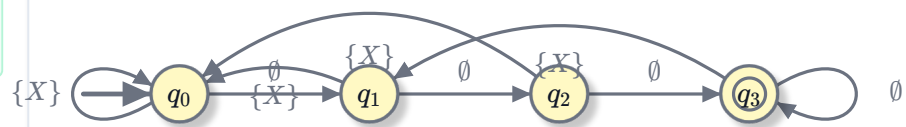
$Q = \{q_0, q_1, q_2, q_3\}$

$F = \{q_3\}$

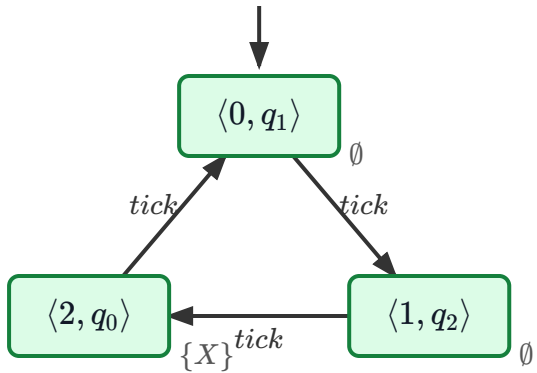
מערכת המעברים TS



אוטומט הרישות הרעות $\mathcal{A}$



$TS \times \mathcal{A}$



התוצאה: התכונה מתקיימת!

אף מצב נגיש במערכת המכפלה אינו מכיל את המצב המקבל q_3, לכן $TS \models P_{\text{safe}}$.

דוגמה: מערכת בת 4 מצבים

הגדרת המערכת והאוטומט

מערכת TS:

$S = \{0, 1, 2, 3\}$

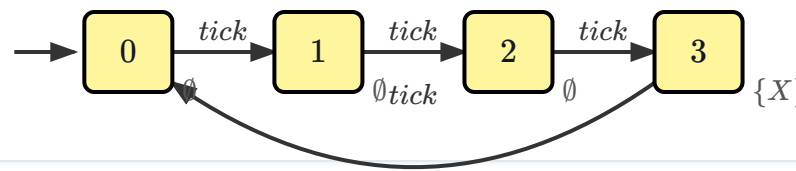
$L(0) = L(1) = L(2) = \emptyset,\ L(3) = \{X\}$

אוטומט $\mathcal{A}$:

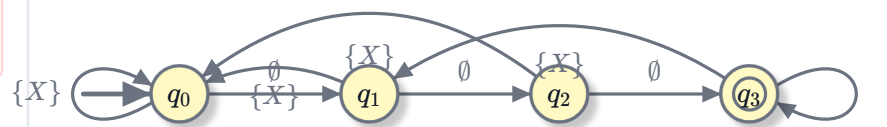
$Q = \{q_0, q_1, q_2, q_3\}$

$F = \{q_3\}$

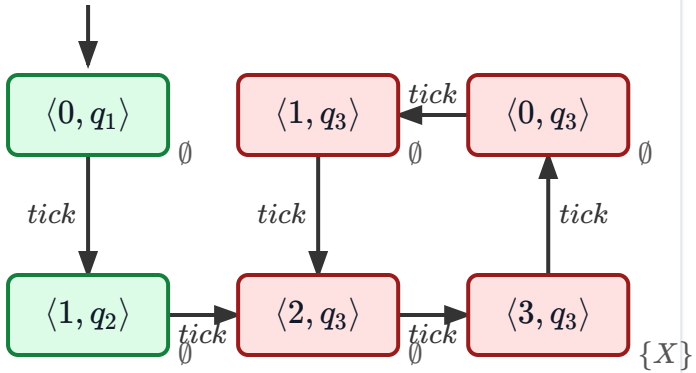
מערכת המעברים TS



אוטומט הרישות הרעות $\mathcal{A}$



$TS \times \mathcal{A}$



התוצאה: התכונה מופרת!

מערכת המכפלה מגיעה למצבים אדומים (המכילים את המצב המקבל q_3) ונכנסת ללולאה בתוכם. לכן $TS \not\models P_{\text{safe}}$.

המשפט המרכזי

נניח ש־TS סופית וללא מצבים סופניים, וש־$\mathcal{A}$ מקבלת את $\mathit{BadPref}(P_{\mathrm{safe}})$. אז התנאים הבאים שקולים:

1. $TS\models P_{\mathrm{safe}}$.
2. $\mathit{Traces}_{fin}(TS)\cap\mathcal{L}(\mathcal{A})=\emptyset$.
3. $TS\times\mathcal{A}\models P_{\mathrm{inv}}$.

זו הרדוקציה: בדיקת תכונת בטיחות רגולרית מצטמצמת לבדיקת שְׁמוּרָה על מערכת מכפלה.

דוגמה נגדית

אם בדיקת הַשְׁמוּרָה נכשלת, נקבל מסלול התחלתי במכפלה:

$$\langle s_0, q_1\rangle\langle s_1, q_2\rangle \cdots \langle s_n, q_{n+1}\rangle$$

אם $q_{n+1}\in F$, אז $L(s_0)L(s_1)\cdots L(s_n)\in\mathit{BadPref}(P_{\mathrm{safe}})$.

ההיטל על רכיב המערכת, $s_0s_1\cdots s_n$, הוא מסלול אמיתי של TS שמסביר את ההפרה.

מצבים סופניים ומעברים חסרים

אם באוטומט יש מעברים חסרים, המכפלה עלולה ליצור מצבים סופניים שאינם קיימים במערכת המקורית.

הבעיה

מסלול במערכת יכול להמשיך, אבל רכיב האוטומט "נתקע" כי אין מעבר מתאים.

הפתרון

משלימים את האוטומט באמצעות מצב מלכודת לא מקבל, ומוסיפים אליו את כל המעברים החסרים.

אחרי השלמה, האוטומט יודע להמשיך לקרוא כל מילה מעל 2^{AP}.

האלגוריתם

קלט: מערכת מעברים סופית TS ותכונת בטיחות רגולרית P_{safe}.

1. בונים אוטומט סופי $\mathcal{A}$ עבור $\mathit{BadPref}(P_{\mathrm{safe}})$ או עבור $\mathit{MinBadPref}(P_{\mathrm{safe}})$.
2. משלימים את האוטומט אם צריך.
3. בונים את מערכת המכפלה $TS\times\mathcal{A}$.
4. בודקים את הַשְׁמוּרָה: לא מגיעים למצב שבו רכיב האוטומט שייך ל־F.
5. אם הַשְׁמוּרָה מופרת, מחזירים את המסלול הסופי כדוגמה נגדית.

שיפור מעשי

כאשר אפשר, עדיף לבנות אוטומט עבור הרישות הרעות המינימליות.

פחות חיפוש מיותר

האוטומט מקבל בדיוק ברגע שבו ההפרה מתגלה לראשונה.

דוגמאות נגדיות קצרות

המסלול המוחזר מסתיים בנקודת ההפרה הראשונה, ולא אחרי המשכים לא רלוונטיים.

מבחינה תיאורטית זה שקול; מבחינה אלגוריתמית זה בדרך כלל נוח יותר.

סיבוכיות

גודל מערכת המכפלה הוא מכפלת הגדלים של המערכת ושל האוטומט.

$$O(|TS| \cdot |\mathcal{A}|)$$

הזיכרון הדרוש הוא ליניארי בגודל המכפלה.

הזמן הדרוש לבדיקת הַשְׁמוּרָה הוא ליניארי בגודל המכפלה.

מה לקחת מכאן?

תכונת בטיחות רגולרית

הפרות מזוהות על ידי אוטומט סופי שקורא רישות סופיות של עקבות.

בדיקה על ידי מכפלה

מכפילים את המערכת באוטומט הרישות הרעות, ובודקים שְׁמוּרָה פשוטה.

דוגמה נגדית

הגעה למצב מקבל באוטומט נותנת רישא רעה ומסלול סופי שמסביר אותה.

העלות

המחיר הוא גודל המכפלה: $O(|TS|\cdot|\mathcal{A}|)$.