הקורס מתחלק לשני חלקים:
1. הוכחת נכונות של תוכניות בכלים של לוגיקה.
2. מערכות עם מספר סופי של מצבים. כלים אוטומטיים לאימות המערכת.

נתעסק בשני סוגים של מערכות:
1. מערכות טרנספורמטיביות: קלט ← תוכנית ← פלט.
2. מערכות תגובתיות (ראקטיביות) – למשל מערכת הפעלה או מעגל חשמלי.

מצב של מערכת: השמה מסויימת למשתני המערכת.
מרחב המצבים: (state-space): מספר המצבים האפשריים במערכת.
מרחב המצבים הישיגיים במערכת: reachable state space.

	הוכחת נכונות	אימות אוטומטי
מספר המצבים	אינסופי או גדול מאוד	סופי וקטן יחסית
סוג התוכנית	טרנספורמטיבית	ריאקטיבית
מפרט	לוגיקה מסדר ראשון (למשל תחשיב היחסים)	לוגיקה טמפורלית (עיתית)
חישוביות	בלתי כריעה undecidable	כריעה
אוטומציה	עזרה מהמשתמש	אוטומציה מלאה
כלים	Theorem proves	RuleBase

דוגמה למפרט בלוגיקה מסדר ראשון (עבור תוכניות מיון): $\forall 0 \le i, j \le n \;\; a[i] \le a[j]$

שיטות להוכחת נכונות:

המטרה: להוכיח שתוכנית מקיימת את המפרט שלה, בעזרת כלים מקובלים בלוגיקה ובמתמטיקה.

האמצעים:
1. שפת מפרטים לתיאור הפונקציונאליות המבוקשת מהמערכת.
2. שפת תכנות עם סמנטיקה פורמאלית.
3. כללי הוכחה.

דוגמה: עבור תוכנית P ומפרט φ נדרוש כי $P \models \varphi$ (P מספק את φ).

נאותות (תקפות) ושלמות:
נאותות: האלגוריתם מחזיר תשובה נכונה.
שלמות: האלגוריתם מחזיר תשובה כן/לא תוך זמן סופי.

אבני יסוד להגדרות:
א. משתנים: $\overline{X}(x_1, x_2, ..., x_n)$
ב. מצב σ של התוכנית – פונקציה ממשתני התוכנית לתחום הערכים שלהם. הערך של x במצב σ הוא: $\sigma(x) = 5$.
ג. מפרט: זוג טענות $< q_1(\overline{x}), q_2(\overline{x}) >$ כאשר $q_1(\overline{x}), q_2(\overline{x})$ הן טענות מתחשיב היחסים מעל משתני התוכנית $\overline{x}$. המשמעות, תנאי קדם $q_1(\overline{x})$ גורר את תנאי $q_2(\overline{x})$ עם סיום התוכנית.
דוגמה: $< x > 0 \wedge b = T, y > x \wedge b = F >$ כלומר, אם בתחילת התוכנית הערך של x גדול מאפס והערך של המשתנה b הוא T, אז בסוף התוכנית (אם יש לה סוף) מתקיים שהערך של המשתנה y גדול מהערך של המשתנה x והערך של המשתנה b הוא F.

ד. חישוב של תוכנית ממצב σ מסומן ב $\pi(P,\sigma)$ הינו סדרה סופית של מצבים $\sigma_0,...,\sigma_k$ או סדרה אינסופית של מצבים $\sigma_0,\sigma_1,...$, כך שלכל i המעבר מ σ_i ל σ_{i+1} הוא בהתאם לתוכנית.

ה. $Val(\pi)$ מסמן את המצב הסופי של החישוב, אם קיים. אם החישוב הוא אינסופי אזי $Val(\pi)=\perp$.
$\perp$ הוא ערך לא מוגדר, שונה מכל הערכים (Bottom).

ו. $\sigma \models q(\bar{x})$ אם הטענה $q(\bar{x})$ נכונה כאשר במקום המשתנים החופשיים ב $q(\bar{x})$ מציבים את הערכים המתאימים להם ב σ.

<u>דוגמה:</u> $q(y)=\forall x(x>y \vee x<y \vee x|_2)$
כלומר, לכל x יתקיים ש $x>y$ או ש $x<y$ או ש x זוגי.

$\sigma_1(y)=1$ $\quad$ $\sigma_1 \not\models q(y)$

(כי עבור $x=1$ לא מתקיים $x>y$ וגם לא מתקיים $x<y$ וגם לא מתקיים ש x זוגי)

$\sigma_2(y)=4$ $\quad$ $\sigma_2 \models q(y)$ (כי לכל x שלם או שהוא גדול מ 4 או שהוא קטן מ 4 או שהוא זוגי)

עבור כל $q(\bar{x})$ $\quad$ $\perp \not\models q(\bar{x})$

<u>נכונות חלקית:</u>
תוכנית P היא נכונה חלקית (partially correct) ביחס למפרט $<q_1(\bar{x}),q_2(\bar{x})>$ אם ורק אם לכל חישוב π של התוכנית P שמתחיל ממצב התחלתי σ_0 שמספק את $q_1(\bar{x})$, (מסמנים $\sigma_0 \models q_1(\bar{x})$), אם החישוב עוצר $\left(val(\pi(P,\sigma_0)) \neq \perp\right)$ אזי $q_2(\bar{x})$ מתקיים בסוף החישוב - $\left(val(\pi(p,\sigma_0)) \models q_2(\bar{x})\right)$.

<u>במילים אחרות:</u> $\sigma_0 \models q_1(\bar{x}) \wedge Val(\pi(P,\sigma_0)) \neq \perp \rightarrow Val(\pi(P,\sigma_0)) \models q_2(\bar{x})$

<u>סימון:</u>
$\{q_1\}P\{q_2\}$ וניתן לכתוב $\models \{q_1\}P\{q_2\}$ $\quad$ $\models_0 \{q_1\}P\{q_2\}$

<u>דוגמאות:</u> רק תוכנית שלעולם לא עוצרת מספקת את המפרט $\{True\}P\{False\}$. כל תוכנית מספקת את המפרט $\{False\}P\{True\}$.

<u>דוגמה:</u>
מפרט: "בסוף התוכנית ערכו של x כפול מערכו בהתחלה".
הביטוי $<true, x=2x>$ אינו נכון.
ביטוי נכון ניתן לכתוב באמצעות משתנה חדש y: $<y=x, x=2y>$.
$\forall x \forall y \ \{y=x\}P\{x=2y\}$

<u>נכונות מלאה:</u>
תוכנית P נכונה באופן מלא (Total correctness) אם ורק אם כל חישוב π שמתחיל ממצב σ_0 שמספק את $q_1(\bar{x})$ עוצר, ומצבו הסופי מספק את $q_2(\bar{x})$.
<u>סימון:</u> $<q_1>P<q_2>$.
כל נכונות מלאה היא גם נכונות חלקית, אבל להפך לא מתקיים.
$\sigma_0 \models q_1(x) \rightarrow Val(\pi(p,\sigma_0)) \models q_2(x)$ אם π איננו סופי אז $Val(\pi)=\perp$ ו- $\perp \not\models q_2(x)$