**Operational-semantics**

**Class Notes by Itay Rosenblatt and Eylon Yogev**

ניתוח מתקדם של שפות תכנות – שיעור מס' 2

הגדרת שפות תכנות מורכבות משני חלקים: סינטקס וסמנטיקה.

סינטקס:

* תחביר השפה, כיצד היא נראית – דקדוק, ביטויים רגולריים, טוקנים.

סמנטיקה:

* משמעות השפה.
* ישנם מספר כלים המבטאים את משמעות השפה: אינטרפרטר, קומפיילר וכו'.
* ניתן, בכדי להבין את משמעות (הסמנטיקה) של השפה, להריץ את הקומפיילר ולפי אופן פעולתו, לנסות ולהבין את משמעות השפה.
* דרך זו הינה מסורבלת ממס' סיבות:
	+ קשה ולא נוח לעקוב אחר ביצועי הקומפיילר.
	+ לא ניתן באמצעות שיטה זו להוכיח תכונות אודות השפה.

בהמשך ההרצאה נראה דרכים טובות יותר להגדיר את הסמנטיקה של השפה, בכדי שנוכל להוכיח תכונות אודותיה.

למה צריך סמנטיקה פורמאלית?

* נהוג להניח כי אם משהו קשה להגדרה אז הוא גם קשה למימוש וכן גם קשה להשתמש בו.
(קשה להגדיר = קשה לממש = קשה להשתמש).

כשיש פיצ'רים מסובכים (הורשה מרובה ב C++ למשל) קשה מאוד להשתמש בהם.

* נחפש להגדיר דברים בצורה פשוטה ולכן יהיה ניתן לממש ולהשתמש בהם בצורה פשוטה.
* יתרונות בתיאור פורמאלי (ופשוט יחסית) של הסמנטיקה של השפה:
	+ הבנה ולמידה של השפה למי שרוצה להשתמש בה.
	+ הוכחת תכונות של תוכנית מסוימת.
	+ הוכחת/הפרכת שקילות של תוכניות.
	+ יצירה אוטומטית של אינטרפרטר שמקורו מההגדרה המופשטת של השפה. משתמשים בכך בעיקר לשם יצור אב טיפוס של שפות.
	+ יצירה אוטומטית של קומפיילר לפי הסמנטיקה הפורמאלית של השפה. רעיון טוב אך קשה למימוש.

ישנם שלוש גישות עיקריות להגדרת סמנטיקה של שפה:

1. **סמנטיקה אופרטיבית** היא הצורה הטבעית ביותר, קרובה לאינטרפרטר מופשט. היא מתארת כיצד התוכנה פועלת על כל מצב במערכת, לרוב ע"י תיאור של מתמיר מצבים.
נראה שתי דוגמאות לסמנטיקה אופרטיבית – טבעית ומבנית.
	1. יתרונות: קלה להגדרה, קלה להבנה.
	2. חיסרון – קשה להוכיח תכונות על השפה בצורה זו.
2. **סמנטיקה ציונית (Denotational)** היא הגדרה אשר אין בה קשר הדוק למימוש.
בשיטה זו, משמעות התוכנית היא דבר מופשט,היא פונקציה או יחס בין הקלט לפלט.
בשיטה זו נרצה להגדיר ביטויים מתמטיים שיבטאו את פעולת התוכנית על הקלט.
	1. יתרונות: ניתן באמצעות הביטויים המתמטיים לבטא את אופן פעולת התוכנית על כל הקלטים וערכי המשתנים האפשריים.
	2. חסרונות:
		1. קשה למצוא הגדרות מתמטיות שיקיפו את כלל ההתנהגות.
		2. ישנן בעיות אינהרנטיות בסמנטיקה זו – כמו איך להגדיר מקבילות, או תכונות נוספות שאין להם תכונות ציוניות (או קשות לפירמול).
3. **סמנטיקה אקסיומטית** מגדירה את משמעות של תוכנית על ידי מה שניתן להוכיח עליה. שיטה זו מבוססת על כלים בלוגיקה. הסמנטיקה תנסה להוכיח תכונות על התוכנית, על פי ההשפעה של התוכנית על טענות(assertions) במהלך הריצה שלה. ובעזרת הטענות נראה מה ניתן להוכיח אודות התוכנית.

כעת נרחיב על כל אחת מן השיטות.

סמנטיקה אופרטיבית

הסמנטיקה מתארת את מהלך ריצת התוכנית.
בעזרת סמנטיקה זו, מתארים את התוכנית כמתמיר מצבים.

**הערה:**ישנה תוכנה של מייקרוסופט בשם "טרמינטור" שמנסה להוכיח האם תוכנית מסתיימת. היא עושה זאת ע"י חיפוש של משתנים שגדלים או קטנים באופן עקבי, ומנסה להוכיח שהריצה מסתיימת. התוכנה אינה מנסה לפתור את בעיית העצירה, אלא מנסה להוכיח תכונה אודות רוב התוכנות הנכתבות ע"י בני אדם (תוכנות סבירות).
(<http://research.microsoft.com/en-us/um/cambridge/projects/terminator/>)

אבחנה שמסייעת בביצוע המשימה : ברוב התוכניות, אם הן מסתיימות הן, אזי הן מסתיימות בעקבות תכונות פשוטות יחסית שניתנות לזיהוי בקלות.

סמנטיקה ציונית

בהינתן תוכנית, אזי הפלט של הסמנטיקה הציונית תהיה פונקציה הפועלת על הקלטים של התוכנית.

לדוגמא, בהינתן פונקצית עצרת (המקבלת את הקלט n), הגדרת הסמנטיקה של התוכנית תהיה הפונקציה הבאה:

אופן הגדרה זה נותן לנו כלים מתמטים שבעזרתם ניתן יהיה להוכיח תכונות מסוימות על התוכנית.
חשוב לציין, כי יתכן והסמנטיקה תספק לנו הגדרות שניתן בעזרתן להוכיח תכונות אך ורק באופן ידני, ולא באופן ממוחשב (כדוגמת שקילות תוכניות).

כאמור, ישנה בעיה לתאר דברים מקביליים. ובכלל, דברים שיש להם קשר הדוק למימוש קשים לתיאור בצורה זו.

סמנטיקה אקסיומטית

בהינתן טענה (assertion), נרצה כללים לבניית שאר הטענות בהמשך התוכנית, וע"י כך להגיע לאקסיומה בסוף התוכנית, שהיא בעצם התכונה שהוכחנו על התוכנית.

לדוגמא, נצטרך למצוא עבור כל לולאה מהי השמורה שלה (באם היא קיימת) – כלומר, מה תמיד מתקיים בעת הכניסה ללולאה.

האתגר העיקרי בתהליך זה הוא למצוא ולנסח את השמורות של הלולאות.

יתכן וניתן למצוא מספר שמורות ללולאה, או לנסח את השמורה במספר דרכים. באמצעות ניסוח מספר שמורות ניתן יהיה להוכיח מספר תכונות שונות אודות התוכנית.

לדוגמא:

**הערה:**

כל מה שלא ניתן להוכיח, משאיר לכותב הקומפיילר מרווח לבצע אופטימזציות ושינויים. כל זאת כל עוד הוא לא פוגע בשמורות שכן ניתנות להוכחה.
כותב הקומפיילר צריך לדאוג כי טענות שהיו נכונות בשלב ההגדרה, יהיו נכונות גם ברמת המימוש.

*אנו נתמקד בעיקר בסמנטיקה הראשונה – סמנטיקה אופרטיבית.*

ניתוח סטאטי (Static Analysis) –
ניתוח סטאטי הוא גזירה של התכונות שנשמרות במהלך התוכנית, בכל ריצה אפשרית.

בכדי להוכיח טענות על התוכנית עבור כלל הקלטים, נשתמש באבסטרקציה.
נגדיר פונקציות תרגום מערכים קונקרטיים לערכים אבסטרקטיים, ונגדיר את הסמנטיקה על ערכים אלה.

ייצוג מופשט של הנתונים

קבוצת המצבים האפשריים

קבוצת המצבים האפשריים

ייצוג מופשט של הנתונים

אבסטרקציה

המחשה

*סמנטיקה אופרטיבית*

*סמנטיקה אבסטרקטית*

S הצהרה

S הצהרה

אחד האתגרים המשמעותיים הוא לדעת לבחור באופן אוטומאטי את קבוצות הערכים המופשטות, עבור תוכנית נתונה.
בחירה נכונה של קבוצות מופשטות תאפשר לנו להוכיח באופן נוח ונכון תכונות אודות התוכנית.

לדוגמא: פונקציה שמכפילה שני מספרים ניתן לתאר באופן מופשט, ע"י הקבוצות הבאות:

בכדי להדגים את הגדרת סמנטיקה באופן פורמאלי, נגדיר שפה פשוטה יחסית: שפת While.

הסינטקס של השפה הוא:

הערות:

1. ניתן להשתמש בסוגריים בשפה בכדי להגדיר קדימות בין פעולות.
2. אנו נניח כי התוצאות של פעולות מתמטיות ובוליאניות מתנהגות באופן הרגיל.
3. פעולת SKIP מתנהגת כאילו לא ביצעו אף פעולה.

דוגמאות לפקודות בשפה WHILE:

**סימונים כללים:**

1. קבוצות סינטקטיות:
	1. Var - קבוצת המשתנים.
	2. Stm – קבוצת ההצהרות (statements).
	3. Aexp – קבוצת הביטויים האריתמטיים ( arithmetic expressions) .
	4. Bexp – קבוצת הביטויים הבוליאניים( Boolean expressions) .
2. קבוצות סמנטיות:
	1. N - המספרים הטבעיים {0,1,2…..}
	2. T – ערכי האמת {tt,ff}.
	3. מיפוי סופי של המשתנים לערכם – Var 🡪 N
	4. חיפוש ערך של משתנה במצב s – s x
	5. עדכון ערך משתנה במצב s - s [x🡪5]

**לדוגמא:**

1. [x🡪1, y🡪7, z🡪16] y = 7
2. [x🡪1, y🡪7, z🡪16] x = 1
3. [x🡪1, y🡪7, z🡪16] z = 16
4. [x🡪1, y🡪7, z🡪16] t =
5. [x🡪1, y🡪7, z🡪16] [x🡪y] x = 7

**כעת נגדיר את הסמנטיקה של הביטויים:**

הערה:
הן לביטויים האריתמטיים והן לביטויים הבוליאניים אין השלכות נוספות (Side effect).

סמנטיקת הביטויים האריתמטיים -

* A[[Aexp]] : State 🡪 N. הסמנטיקה של ביטוי מתמטי הוא מיפוי מהמצב לערך מספרי כלשהו.
* נגדיר בצורה אינדוקטיבית את המשמעות הסמנטית של הביטויים:
	+ - פירוש של ערך מספרי.
	+ - ערכו של המשתנה X בטבלא של S.
* הסמנטיקה הינה צירופית – משמעות של מבנה מסוים מוגדר באמצעות המשמעות של תתי הביטויים שלו.
* בעקבות עובדות אלה, ניתן להוכיח תכונות בעזרת אינדוקציה מבנית – אינדוקציה על עומק עץ הגזירה. לדוגמא: הוכחה כי חיבור הוא קומוטטיבי, הוכחה על שימור ערך של משתנה מסוים במהלך הריצה וכו'.

סמנטיקת הביטויים הבוליאניים-

* B[[Bexp]] : State 🡪 T
* נגדיר בצורה אינדוקטיבית את המשמעות הסמנטית של הביטויים:
	+ על פי הסמנטיקה מוגדר כי פעולת ה "=" מוגדרת בין שני ביטויים אריתמטיים.
	+ על פי הסמנטיקה מוגדר כי פעולת ה"∧" מוגדרת בין שני ביטויים בוליאניים.

באותו אופן ניתן להגדיר את המשמעות של שאר הביטויים המתמטיים והבוליאניים.

**סמנטיקה אופרטיבית טבעית**

נראה שלוש צורות להגדיר סמנטיקה של משפטים, כאשר הרעיון הכללי הוא סמנטיקה אבסטרקטית המתעלמת מההתנהגויות הפנימיות של המשפט.
הסמנטיקה הטבעית מגדירה את ההשפעות/התכונות הכלליות של התוכנית. הסמנטיקה מתעלמת מחישובים שאינם מסתיימים.
זוהי סמנטיקה חישובית, כלומר בהינתן תוכנית ומצב ניתן לחשב את הערך שיחזיר האינטרפרטר.

זוהי סמנטיקה אלגנטית ופשוטה.

סימונים ומשמעותם:

* - ההצהרה S מופעלת על המצב הסופי s.
* – ההצהרה S מופעלת על הקלט i. ההצהרה מסתיימת ואחד מהמצבים האפשריים הוא o.
* משמעות התוכנית P על המצב s (קלט) הוא כל המצבים o כך שקיים i עבורו .
* המשמעות של הצהרות מורכבות מוגדר באמצעות תתי ההצהרות המרכיבים אותן.

כעת נגדיר את הסמנטיקה הטבעית עבור השפה WHILE:

תחילה נגדיר את הסמנטיקה עבור האקסיומות (ns = natural semantics, ass=assignment, comp= compound):

כעת נגדיר את כללי ההיסק –

עבור משפט IF-

 במידה והתנאי נכון, אזי שיערוך משפט IF , שקול לשערוך משפט ה--THEN

במידה והתנאי אינו נכון,שיערוך המשפט שקול לשערוך משפט ה-ELSE אם קיים, אחרת שקול ל-SKIP-

עבור לולאת WHILE-

 במידה והתנאי נכון, נשתמש בהגדרת ה-WHILE עצמו בכדי להגדיר את המשמעות.

 במידה והתנאי אינו נכון, שיערוך לולאת ה-WHILE שקולה ל-SKIP

דוגמאות:

נניח כי הינו המצב בו .

אזי:

כאשר נרצה להוכיח כי נקבל עץ גזירה, כאשר:

* הוא השורש
* העלים הם האקסיומות (הרלוונטיות) של השפה
* הצמתים הפנימיים הם כללי ההיסק, כאשר ילדיו (הישירים) של כל צומת הם התנאים המקדימים לכלל ההיסק.

דוגמא 1:



עץ גזירה המוכיח כי:

דוגמא 2:
נראה שהביטוי  *שקול לביטוי .
ניקח לדוגמא מקרה בו במצב , x:=3 ונראה עץ גזירה מתאים המראה כי:*

*נבנה עץ גזירה שהשורש שלו הוא בביטוי הנ"ל (הריבועים עם המסגרת האדומה הם כללי ההיסק והאקסיומות שהופעלו).*

**הערות:**

* ניתן באופן אוטומטי לבנות אינטרפרטר ע"י כך שהוא מריץ את התוכנית- מתחיל מהשורש ומפעיל את החוקים עד שהוא מגיע לאקסיומות.
* בשפת WHILE לביטוי עץ הגזירה ומצב s’ יחידים (עבור תוכנית S ומצב s).

**שקילות סמנטית:**

S1 ו S2 ייקראו שקולים סמנטית אם לכל s ולכל s’ מתקיים

לדוגמא:
הביטוי שקול סמנטית לביטוי

הוכחה:

 is divided into followed by

when – in other words, this is a derivation tree for S; while b do S.

When , we obtain skip.

On the other side, according to the while semantics:

* is the definition for The "while" semantics provided
* is the definition for the while semantics provided

 So, is reconstructed from

**דטרמיניזם:**

הסמנטיקה של שפת While היא דטרמיניסטית (זו תכונה שאינה בהכרח נכונה לכל שפה), באופן פורמאלי:

הוכחה [לקוח מ <http://www.clei.cl/cleiej/papers/v12i2p2.pdf>]



מכיוון שהסמנטיקה היא דטרמיניסטית אזי המשמעות הסמנטית של כל משפט היא פונקציה
 , וכן בסמנטיקה זו קיימת משמעות רק לחישובים שמסתיימים.
 דוגמאות:

נגדיר כעת סמנטיקה שתקפה גם לחישובים שאינם מסתיימים.

**סמנטיקה מבנית**

סמנטיקה זו דומה לסמנטיקה הקודמת אך הפעולות בה הן קטנות יותר, כלומר היא יותר LOW LEVEL.
הרעיון הכללי הוא שבמקום להגדיר מצב סופי מגדירים מצב התחלתי ומהו המשפט הבא שצריך לבצע.

לכל מצב S נכתוב את הכלל והמשמעות היא שבשלב הראשון של הביצוע של S על i נגיע למצב .

כאשר יכול להיות:

1. מצב סופי (ואז אין הבדל בין הסמנטיקה הטבעית)
2. שלב ביניים כלומר אילו עוד משפטים S’ צריך לבצע על s’.
3. יכול להיות תקוע(stuck). כלומר הגענו למצב בו אין יותר כללים לביצוע ואנו תקועים בחישוב. מצב זה לא היה מתואר בסמנטיקה הטבעית מכיוון שהוא לא מוביל למצב סופי.

חישוב שבסמנטיקה הטבעית התבטא בצעד אחד, יבוטא כאן כשרשרת הצעדים שתוביל לאותו מצב.

**הגדרת הסמנטיקה:**

נגדיר תחילה אקסיומות – הסמנטיקה של האקסיומות מוגדרות באופן זהה לאלה בסמנטיקה הטבעית.

כעת, נגדיר את כללי ההיסק (sos = structural operational semantic) -

נפריד בהגדרת החוקים באם הביטוי (S) הוא אטומי או שיש לו שלבי ביניים.

 **סדרות גזירה:**

בדומה לעץ גזירה בסמנטיקה הטבעית, הוכחה בסמנטיקה המבנית מבוטאת ע"י סדרת גזירה.
סדרת גזירה של הצהרה S שמתחילה במצב s יכולה להיות:

1. סדרה סופית כך ש:
	1. הוא מצב תקוע או מצב סופי.
2. סדרות אינסופיות - עבור כל i.

הערה:
סדרה אינסופית זו לא הייתה מופיעה בסמנטיקה הטבעית.

סימונים –

* - גזירה תוך i צעדים.
* – גזירה תוך מספר סופי כלשהו של צעדים (\* מתייחס למספר סופי).

כל אחד מן הצעדים ניתן לתאר באמצעות עץ גזירה.

**דוגמא:**

נשתמש תחילה בכלל ההרכבה הראשון כאשר:

לכן נקבל:

וכעת נפעיל את משפט הגזירה השני (מכיוון שהפעולה הבאה כעת היא אטומית):

מכיוון שניתן לעשות את המעבר הנ"ל בצעד אחד, ולכן לפי כלל ההיסק נקבל את השקילות:

לבסוף נבצע את אקסיומת ההשמה, ונקבל:

נתבונן בדוגמא נוספת- :

Assume: s = {x = 3, y=4, p=1}

סיום תוכנית:

* התוכנית מסתיימת אם קיים מסלול לסיום, ובאופן פורמאלי:
S מסתיימת על s אם ורק אם, קיים רצף גזירות סופי המתחיל ב -
* נאמר כי התוכנית מסתיימת בצורה טובה אם מצב הסיום אינו STUCK.
* נאמר כי S נכנסת ללולאה על s אם קיים רצף גזירות אינסופי המתחיל ב -

תכונות הסמנטיקה:

1. נאמר כי S1, S2 שקולים סמנטית אם:
	1. עבור כל s ועבור כל שהוא או סופי או STUCK, מתקיים:
	2. יש רצף גזירה אינסופי המתחיל מ - אם ורק אם קיים
	רצף אינסופי המתחיל מ - .
2. דטרמיניזם:
3. את ההפעלה ניתן לחלק לשני חלקים:
4. הרכביות (sequential composition) – אם , אזי קיים מצב s' ומספרים k1,k2 כך ש:

כאשר נרצה להוכיח תכונות בסמנטיקה זו, נוכיח אותן באמצעות אינדוקציה על אורך סדרת הגזירה.

בדומה לסמנטיקה טבעית, גם כאן משמעות ההצהרה S הינה פונקצית מיפוי ממצב אחד למשנהו:

ניתן להוכיח כי בשפת WHILE אין הבדל בין שתי הסמנטיקות.
באופן כללי, שלבי ההוכחה הם כדלקמן:

1. בהינתן עץ גזירה עבור נראה כי קיימת סדרת גזירה באורך k כך ש
2. בהינתן סדרת גזירה באורך k כך ש נראה כי קיים עץ גזירה המבטא

בכך נוכיח הכלה בשני הצדדים, ועל כן הסמנטיקות שקולות.
*(הוכחה מלאה ניתן למצוא ב-* [*http://www.daimi.au.dk/~bra8130/Wiley\_book/wiley.html*](http://www.daimi.au.dk/~bra8130/Wiley_book/wiley.html)

*עמודים 40-44).*