Turingol

שפה וסביבת פיתוח ללימוד מכונות טיורינג

ד"ר דוד פייטלסון

דצמבר 2023

# מבוא

מסמך זה מתאר שפה עילית וסביבת פיתוח ללימוד מכונות טיורינג.

# הצורך בשפה עילית

מכונת טיורינג היא מכונת מצבים שפועלת על סרט. עד עכשיו כשרצינו לתאר מכונות מצבים או מכונות מחסנית השתמשנו בדיאגרמות שבהן תיארנו ישירות את המצבים ואת המעברים. אבל כשמנסים להשתמש בטכניקה זו כדי לתאר מכונות טיורינג נתקלים בבעיה, אפילו עבור מכונות פשוטות יחסית. כמות המצבים והמעברים הדרושה כדי לתאר את המכונה פשוט גדולה מדי. התוצאה היא דיאגרמה כל כך גדולה ומסובכת עד שקשה לנו מאוד להבין אותה. המצב דומה לניסיון לכתוב תוכנית מחשב בשפת מכונה. והפתרון גם הוא דומה. במקום לכתוב ישירות בשפת המכונה, נשתמש בכלי שמאפשר לנו לכתוב בשפה גבוהה יותר ויתרגם את התוכנית שלנו למכונת מצבים.

# התקנת הסביבה

פרטי ההתקנה משתנים מעט בהתאם למערכת ההפעלה.

## Windows 10

הורידו את הקובץ הבא, הריצו אותו ועקבו אחרי הוראות ההתקנה.

[Turingol-Win32.zip](https://afekacollege.sharepoint.com/sites/msteams_cfab3a/Class%20Materials/Turingol-Win32.zip)

## Mac OS

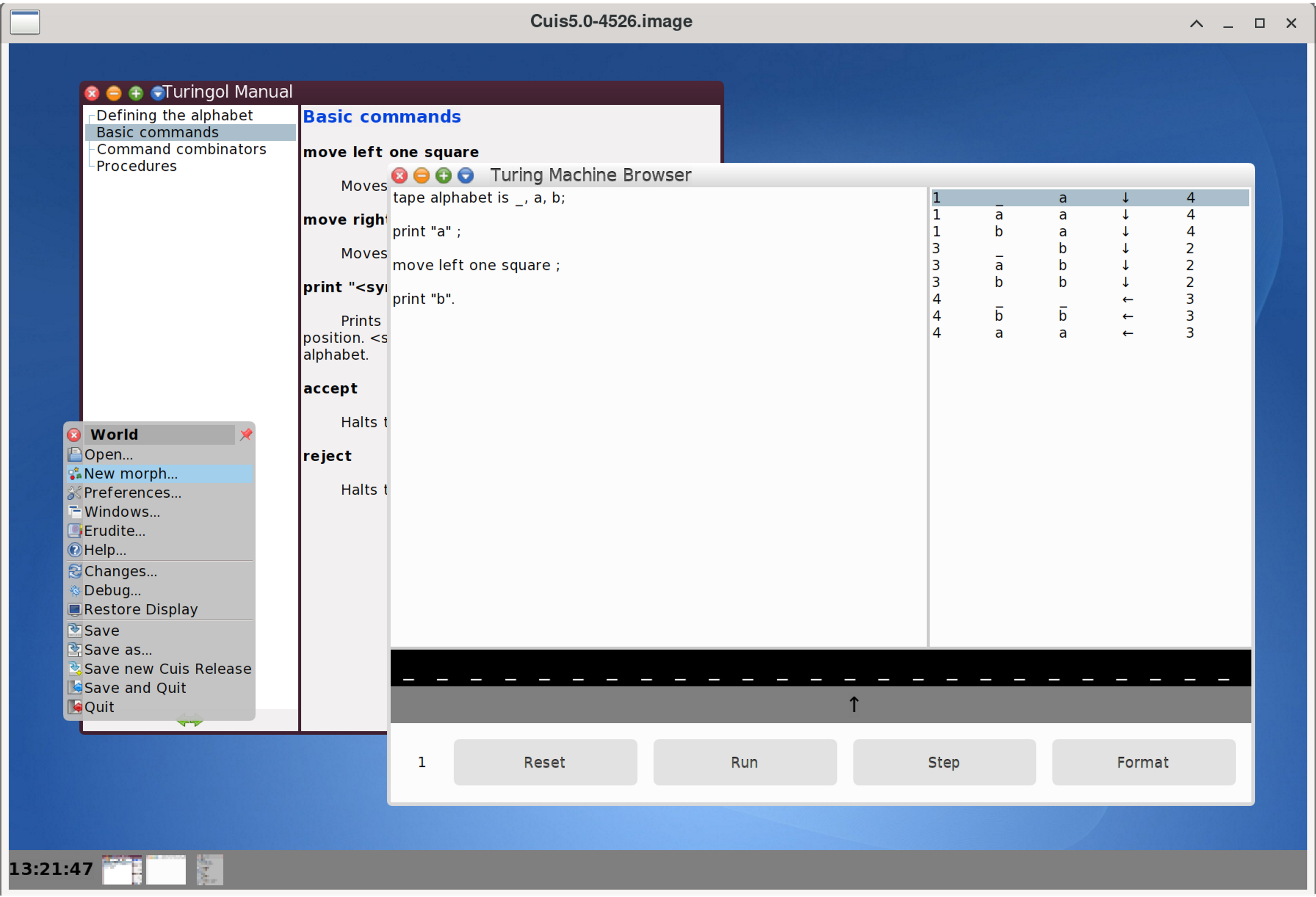
## Linux

# היכרות עם סביבת העבודה

בסביבת העבודה תמצאו כמה רכיבים:

1. כלי לכתיבה והרצה של מכונות טיורינג. זהו הכלי העיקרי שאיתו תעבדו,
2. מדריך קצר של פקודות השפה,
3. סביבת ניהול שבעזרתה אפשר לשמור את מצב המערכת ולנהל את הכלים השונים.

ניתן לראות מראה כללי של סביבת הפיתוח באיור 1.



איור 1 מראה כללי של סביבת הפיתוח.

# כתיבת תוכניות

את התוכנית נכתוב בחלק השמאלי של חלון העבודה. לאחר שסיימנו לכתוב את התוכנית נבקש מהמערכת לבנות ממנה מכונת מצבים. לשם כך יש לוודא שהעכבר נמצא על אזור הכתיבה וללחוץ על הכפתור הימני בעכבר. בתפריט יש לבחור Accept. אם אין שגיאות תחביר המערכת תבנה את מכונת המצבים ותציג את טבלת המעברים בחלון הימני. אם יש שגיאות תחביר המערכת תסמן באדום את החלק של התוכנית שאותו היא לא מבינה ולא תבנה מכונה.

# טיפול בשגיאות

## שגיאות תחביר

אם בתוכנית שכתבנו יש שגיאה תחבירית המערכת תסמן באדום את הטקסט שאותו לא הצליחה להבין.

## שגיאות בשמות משתנים או פרוצדורות

אם טעינו בשם פרוצדורה או משתנה, המערכת תסמן את הפניה השגויה באדום על גבי התוכנית שלנו. טיפ: כדי לסלק בקלות את הודעת השגיאה אפשר ללחוץ על Ctrl-Z (פעולת Undo).

## טעויות נפוצות

1. לשכוח לסיים את התוכנית עם נקודה,
2. לשכוח לסיים פקודה עם נקודה פסיק,
3. לפנות לפרוצדורה או למשתנה בשם לא נכון.

# צפיה במכונת המצבים

 אפשר לראות דיאגרמה של מכונת המצבים בעזרת התפריט שזמין מחלון טבלת המעברים. את התפריט נקבל על ידי לחיצה על כפתור ימין כאשר העכבר ממוקם על טבלת המעברים. אם נלחץ על show נראה את מכונת המצבים כדיאגרמה. המצב ההתחלתי יסומן בחץ אדום ומצבי העצירה יסומנו בעיגולים כפולים.

Cuis5.O-4526.image 
x 
0 000 Turing Machine Browser 
tape alphabet is 
print "a" , 
O e O e State Machine Diagram 
move left 
print "b". 
1 
1 
write b 
4 
h 
1 
write a 

איור 2 הצגה של מכונת המצבים שהקומפיילר מייצר.

# פקודות בסיסיות

## הגדרת אלפבית הסרט

כל תוכנית חייבת להתחיל בפקודה שמגדירה את הסמלים שיופיעו על סרט המכונה. לדוגמה, הפקודה הבאה מגדירה אלפבית סרט של שלושה סימנים:

tape alphabet is \_ , a , b ;

שימו לב:

1. האלפבית תמיד חייב להכיל את הסמל \_ שמציין את התוו הריק (blank).
2. יש לסיים את הפקודה בנקודה פסיק.

## כתיבת סמל בודד

הפקודה שרושמת סמל על הסרט נראית כך:

print "a"

שימו לב: יש להקיף את הסמל בגרשיים.

## הזזת הראש

יש שתי פקודות בסיסיות להזזת הראש:

move right one square

move left one square

## פקודות עצירה

כשבונים מכונה לקבלת שפה יש להשתמש בשתי הפקודות הבאות כדי לעצור את המכונה במצב מקבל או לא מקבל:

accept

reject

# שילוב פקודות

## ביצוע סדרת פעולות

כדי לבצע מספר פעולות בזו אחר זו נחבר אותן בעזרת נקודה פסיק. לדוגמה, התוכנית הבאה רושמת ab על הסרט ועוצרת על ה-b:

print "a";

move right one square;

print "b".

שים לב, יש לסיים את התוכנית בנקודה.

## תנאים

אפשר לשנות את בקרת התוכנית בהתאם לתוכן הסרט. לדוגמה, התוכנית הבאה הופכת "a" ל-"b":

if tape symbol is "a" then print "b".

## תוויות וקפיצות

כדי לבצע פעולות מורכבות יותר כמו לולאות, אפשר לתת תווית לנקודה בתוכנית ולהעביר את הבקרה לנקודה זו באופן דינמי בעזרת פקודת go to. לדוגמה, התוכנית מחליפה בין "a" ל-"b" בכל מחרוזת הקלט:

loop:

move right one square;

if tape symbol is "a" then { print "b"; go to loop } ;

if tape symbol is "b" then print "a" ;

go to loop.

שימו לב לסוגריים המסולסלים בתנאי הראשון. הם מגדירים סדרת פקודות שתתבצע כולה אם התנאי מתקיים ולא תתבצע כלל אם התנאי לא מתקיים.

# פרוצדורות

## הגדרת פרוצדורה

המבנה הכללי של פרוצדורה נראה כך:

to <name> ... { <statement> ... }

ראשית תופיע מילת המפתח to לאחריה שם הפרוצדורה ולבסוף גוף הפרוצדורה בתוך סוגריים מסולסלים. שם הפרוצדורה יכול להכיל פרמטרים, הפרמטרים מסומנים על ידי התחילית $. לדוגמה

to move left until $x or $y {

loop :

move left one square ;

if tape symbol is $x then go to exit;

if tape symbol is $y then go to exit;

go to loop;

exit : ;

}

מגדיר פרוצדורה שנקראת move left until $x or $y כאשר $x ו-$y הם שני פרמטרים לפרוצדורה.

## קריאה לפרוצדורה

כדי לקרוא לפרוצדורה נכתוב את שם הפרוצדורה כפקודה ונספק לה ערכים לפרמטרים. לדוגמה, כדי לקרוא לפרוצדורה שהגדרנו קודם לכן, נכתוב כך:

move left until "a" or "\_"

הפרוצדורה תזיז את ראש המכונה שמאלה עד שהראש יגיע לתא שמכיל את הסמל “a” או שתגיע לתא ריק (מכיל את הסמל “\_”).

# הרצת המכונה

כדי להזין קלט למכונה יש למקם את העכבר מעל לסרט ולחוץ על הכפתור הימני. בתפריט שיפתח אפשר לבחור לרשום סמל יחיד מעל המיקום הנוכחי של הראש, או להכניס סדרה של סמלים בבת אחת.

אפשר לשנות את מיקום הראש על ידי בחירתו והזזתו בעזרת העכבר תוך כדי לחיצה על כפתור שמאלי.

בצד שמאל של פאנל הכפתורים התחתון אפשר לראות תמיד את המצב שבו נמצאת כרגע המכונה. כאשר המצב הוא מצב עצירה צבע הרקע יהיה אדום, אחרת צבע הרקע יהיה ירוק.

לחיצה על הכפתור Reset תעביר את המכונה למצב ההתחלתי שלה. לחיצה על הכפתור Step תגרום למכונה לבצע צעד חישוב יחיד. לחיצה על הכפתור Run תגרום למכונה לבצע 100 צעדי חישוב לכל היותר. המכונה תעצור תמיד אם הגיעה למצב עצירה. הגבלת כמות צעדי החישוב נועדה למנוע מהסימולטור לרוץ ללא הגבלה במקרה והמכונה לא תעצור. לחיצה על כפתור Format תנקה את הסרט ותחזיר אותו למצב שכל התאים בו מכילים את הסמל הריק (blank).

# תרגילים

1. כתבו תוכנית עבור כל אחת מהפקודות הבסיסיות בשפה וציירו את מכונת המצבים המתקבלת.
2. כתבו תוכנית שמבצעת שתי פעולות בסיסיות זו אחר זו, וציירו את המכונה המתקבלת. האם אפשר לבנות מכונה יעילה יותר שתבצע אותה פעולה? אם כן, הציעו כיצד אפשר לשפר את תהליך בניית המכונה.
3. כתבו פרוצדורות עבור הפעולות השימושיות הבאות:
   1. להתקדם ימינה/שמאלה עד שמגיעים לתוו כלשהו
   2. להתקדם ימינה/שמאלה כל עוד רואים תוו כלשהו
4. עבור מחרוזות מעל האלפבית הבינארי נגדיר את להיות המספר שהמחרוזת x מייצגת. לדוגמה, , כתבו תוכנית שמחשבת את הפונקציה   
   כלומר, בהינתן מחרוזת שמייצגת מספר בינרי, הפונקציה מחשבת מחרוזת שמייצגת את המספר הבא אחריו.
5. כתבו תוכנית שמזהה את השפה