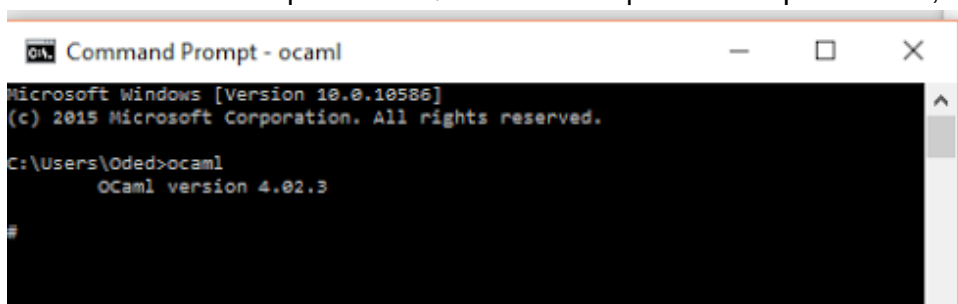


הוראות להתקנת OCaml

סביבת הפיתוח ל OCaml נתמכת בצורה טובה יותר במערכת ההפעלה לינוקס. הוראות התקנה לסביבת לינוקס נמצאות ב <https://github.com/realworldocaml/book/wiki/Installation-Instructions>. בלינוקס ניתן להתקין גם את הקומפיילר ל OCaml, גם את ספריית Core, וגם את utop. בסביבת Windows ניתן להתקין בקלות את הקומפיילר בלבד, מה שמספיק בהחלט לפתרון התרגיל בקורס.

להתקנה בסביבת Windows יש להוריד את קובץ ההתקנה מ: <http://protz.github.io/ocaml-installer>. במהלך ההתקנה אין צורך להתקין את Cygwin (אין בעיה גם להתקין). לאחר שההתקנה הסתיימה, יש להפעיל את ה Command Prompt. ניתן לעשות זאת ע"י לחיצה על מקש "התחל" וכתיבת "cmd". בחלון שנפתח, אפשר להריץ את ה toplevel של OCaml ע"י הרצת הפקודה :ocaml



```
Command Prompt - ocaml
Microsoft Windows [Version 10.0.10586]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\Oded>ocaml
OCaml version 4.02.3
#
```

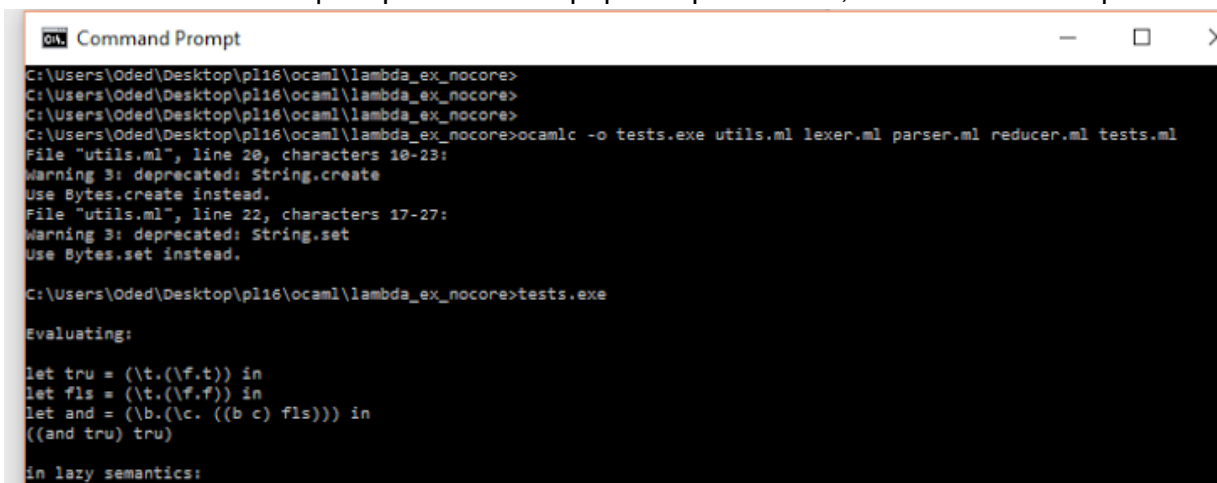
אם הצלחתם להריץ את זה, ההתקנה הסתיימה בהצלחה. שימו לב שלצורך ההתקנה הבסיסית אין צורך להריץ שום דבר שקשור ל opam או ל Cygwin.

ה toplevel משמש לעבודה אינטראקטיבית ב ocaml, וניתן לקרוא על הפקודות שזמינות בו כאן: <http://caml.inria.fr/pub/docs/manual-ocaml/toplevel.html> בתרגולים השתמשנו ב utop, שהוא תחליף יותר משוכלל ל toplevel הבסיסי, אבל אין שום בעיה לעבוד עם ה toplevel הבסיסי.

כדי לקמפל קוד OCaml לקובץ exe, מפעילים את הפקודה ocamlc. לדוגמה, כדי לקמפל את קוד התרגיל מריצים:

```
ocamlc -o tests.exe utils.ml lexer.ml parser.ml reducer.ml tests.ml
```

ולאחר שהקמפול הסתיים בהצלחה, אפשר להריץ את הקובץ tests.exe מתוך ה Command Prompt:



```
Command Prompt
C:\Users\Oded\Desktop\pl16\ocaml\lambda_ex_nocore>
C:\Users\Oded\Desktop\pl16\ocaml\lambda_ex_nocore>
C:\Users\Oded\Desktop\pl16\ocaml\lambda_ex_nocore>
C:\Users\Oded\Desktop\pl16\ocaml\lambda_ex_nocore>ocamlc -o tests.exe utils.ml lexer.ml parser.ml reducer.ml tests.ml
File "utils.ml", line 20, characters 10-23:
Warning 3: deprecated: String.create
Use Bytes.create instead.
File "utils.ml", line 22, characters 17-27:
Warning 3: deprecated: String.set
Use Bytes.set instead.
C:\Users\Oded\Desktop\pl16\ocaml\lambda_ex_nocore>tests.exe
Evaluating:
let tru = (\t.(f.t)) in
let fls = (\t.(f.f)) in
let and = (\b.(c. ((b c) fls))) in
((and tru) tru)
in lazy semantics:
```