

תרגיל 9 – מידול וניתוח מערכות סייבר פסיקליות

ברק לביא 315330985

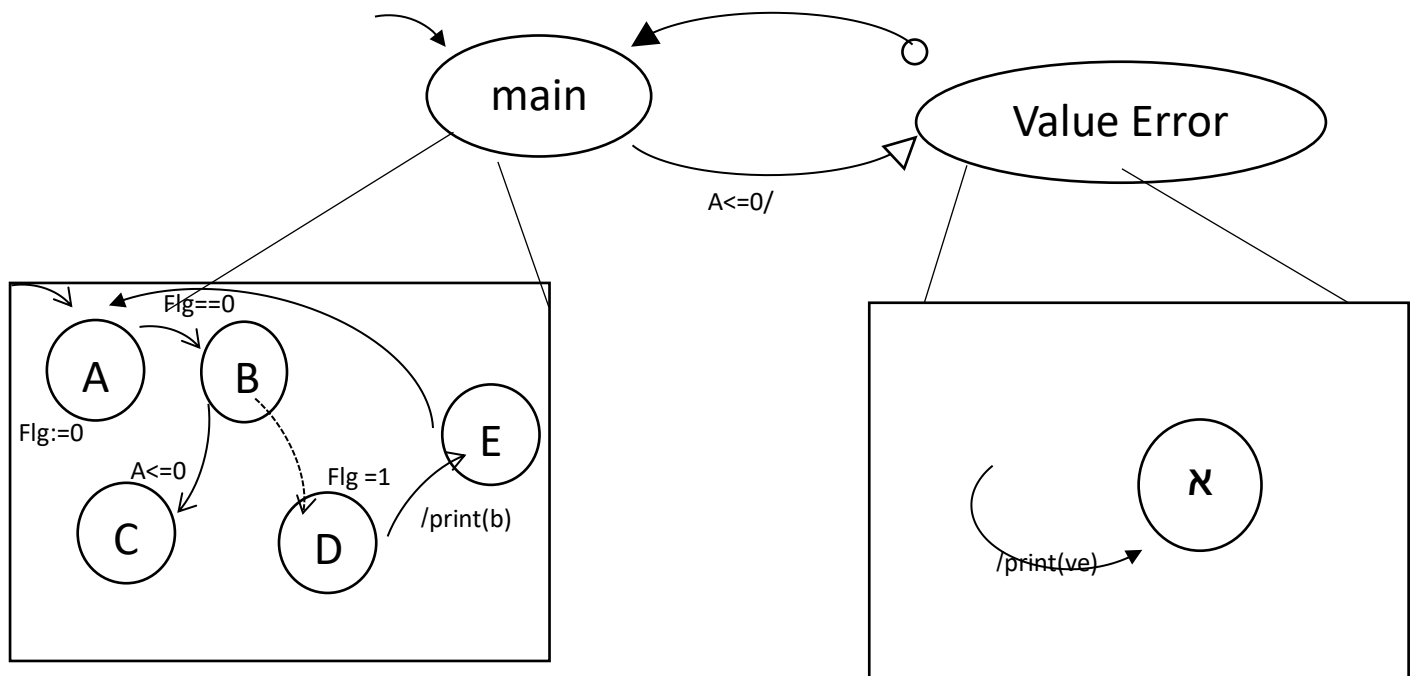
נוי כשר 314963810

שאלה 1: מודל היררכי

Variable: flg: 0, b: int

Inputs: a: int,

Outputs: b:int , ve: exception

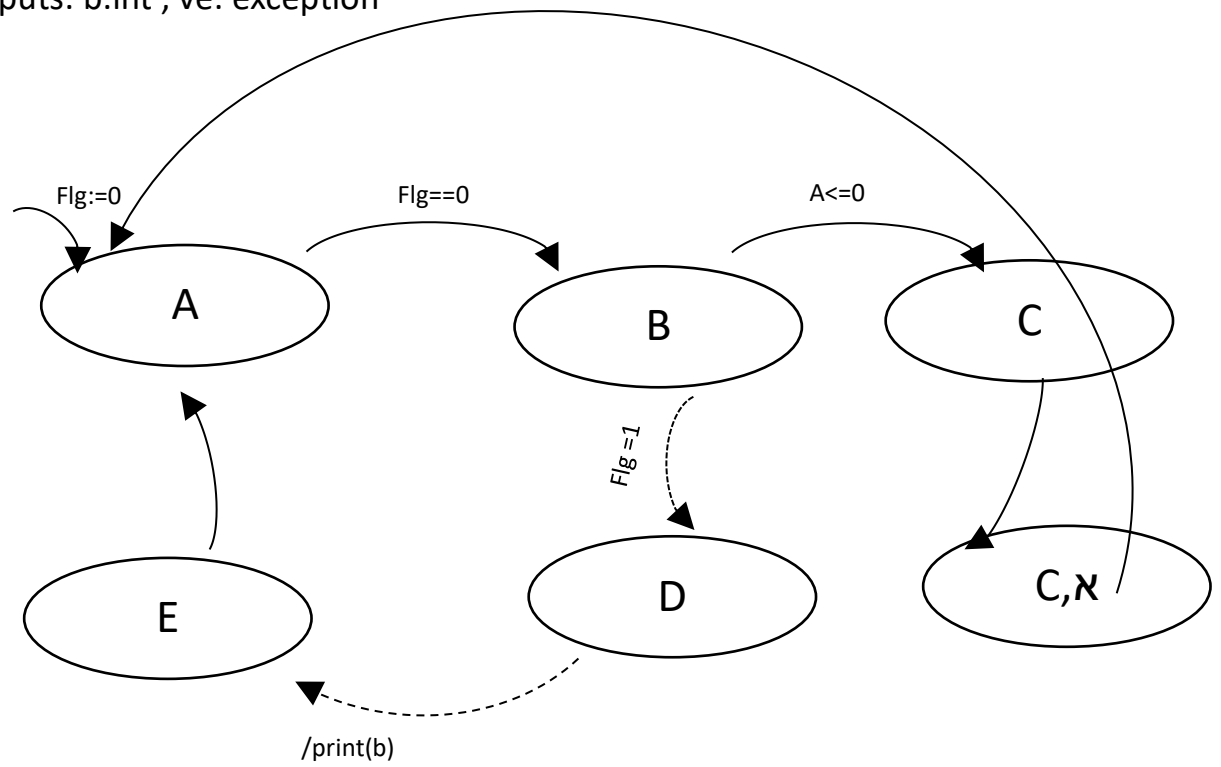


מודל משוטח:

Variable: flg: 0, b: int

Inputs: a: int,

Outputs: b: int , ve: exception



שאלה 2: מודל הבית החכם

בתהליך הראשי הוגדרו מספר משתני עזר גלובאליים:

- Counter אשר סוכם את מספר האנשים הנמצאים בבית החכם.
- מערך outputs אשר מציג את הפלטים לפי הסדר של [טעות, אורות, מוזיקה, רובוט].
- End – משתנה בוליאני אשר הופך ל-True כאשר עולה שגיאה והתהליכים המקביליים צריכים להסתיים.

התהליך שקורה במקביל לתהליך הראשי הינו תהליך דגימת החיישנים (1).

כל עוד לא עלתה שגיאה, נקרא לפונקציה main_fun (2) אשר מתפעלת את הבית החכם.

אם עלתה שגיאה, התהליכים מסתיימים ומתאחדים בפקודת Join.

הודעה שגיאה מסוג שגיאת זמן ריצה תופיע על המסך.

```
if __name__ == '__main__':
    counter = Value('i', 0)
    outputs = Array('i', [0, 0, 0, 0]) # Err, L, M, R - all of them "off"
    end = Value('i', 0)

    l = Lock()
    q = Queue()
    p_queue = Process(target=inputs, args=(q, l))
    p_queue.start()

    time.sleep(2)
    while end.value == 0:
        main_fun(q, outputs, counter, l, end)

    p_queue.terminate()
    p_queue.join()
    if outputs[0] == 1: #exception
        raise RuntimeError("Error!")
```

(1) תהליך דגימת החיישנים:

דגימה רנדומלית עבור שלושת החיישנים – Door, In, Out.

לפני כל דגימה, על ה-Process לקחת את המפתח ולאחריה הוא מחזיר אותו.

בין דגימה לדגימה לקחנו שנייה של דיליי על מנת לא להעמיס על המערכת.

כל דגימה נכנסת לאובייקט מסוג תור.

```
def inputs(q, l): # getting inputs from sensors - door, in, out
    while True:
        l.acquire()
        q.put([random.choice([True, False]), random.choice([True, False]), random.choice([True, False])]) # Door, In, Out
        time.sleep(1)
        l.release()
```

(2) פונקציית Main_fun:

נשלף את האיבר הראשון בתור של הדגימות שנלקחו, ונבחן את הדגימה הנבחרת על ידי מערכת תנאים "אם-אז" על מנת להבין איך לטפל. מערכת תנאים זו לקוחה מהשקף במצגת מההרצאה.

```
def main_fun(communicator:Queue, outputs, counter, l, end):
    time.sleep(1)
    l.acquire()
    data = communicator.get()
    print('inputs', data)

    if data == [True, True, False]:...
    elif data == [True, True, True] and counter.value >= 1:...
    elif data == [True, False, True] and counter.value - 1 > 0:...
    elif data == [True, False, True] and counter.value - 1 == 0:...
    elif data == [True, False, False] or data == [True, False, True] and counter.value - 1 < 0 or

    print('outputs', outputs[:])
    print('Amount of people in house', counter.value)
    print()
    l.release()
```

ויזואליזציה:

בסיום הטיפול בכל דגימה רנדומלית, נציג את הפלטים כגרף קטגוריאלי (כלומר האם פלט מסוים קיבל 1 או 0, וזאת נעשה עבור כל פלט).

```
# visualization
fig = plt.figure()
names = ['Err', 'L', 'M', 'R']
ax = plt.bar(names, outputs)
plt.show()
time.sleep(3)
```