



# Reactive Instance Variables

## RIVar

RIVka Altshuler

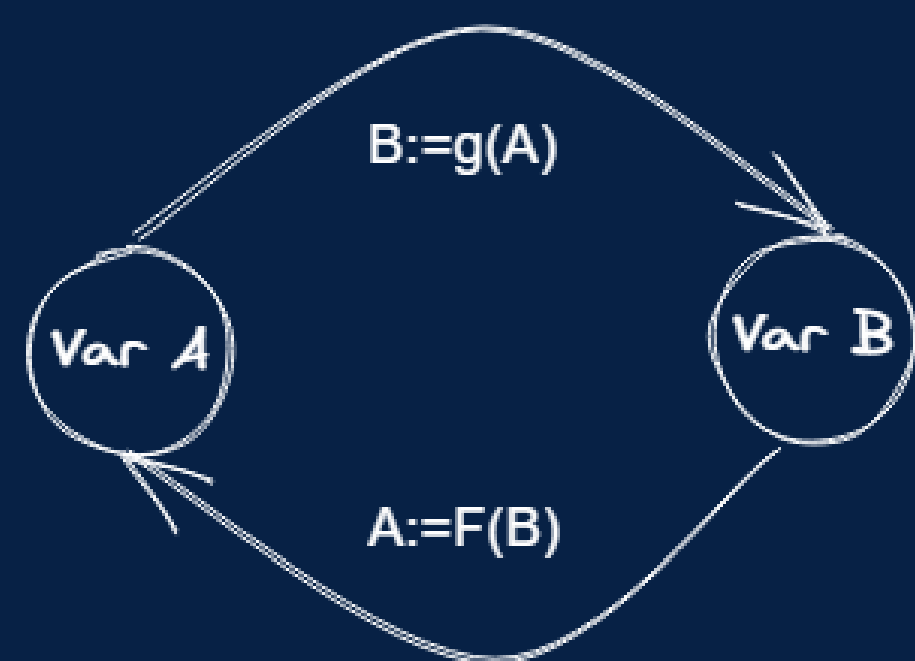
Advisor: David H. Lorenz

Dept. of Math and Computer Science  
Open University of Israel

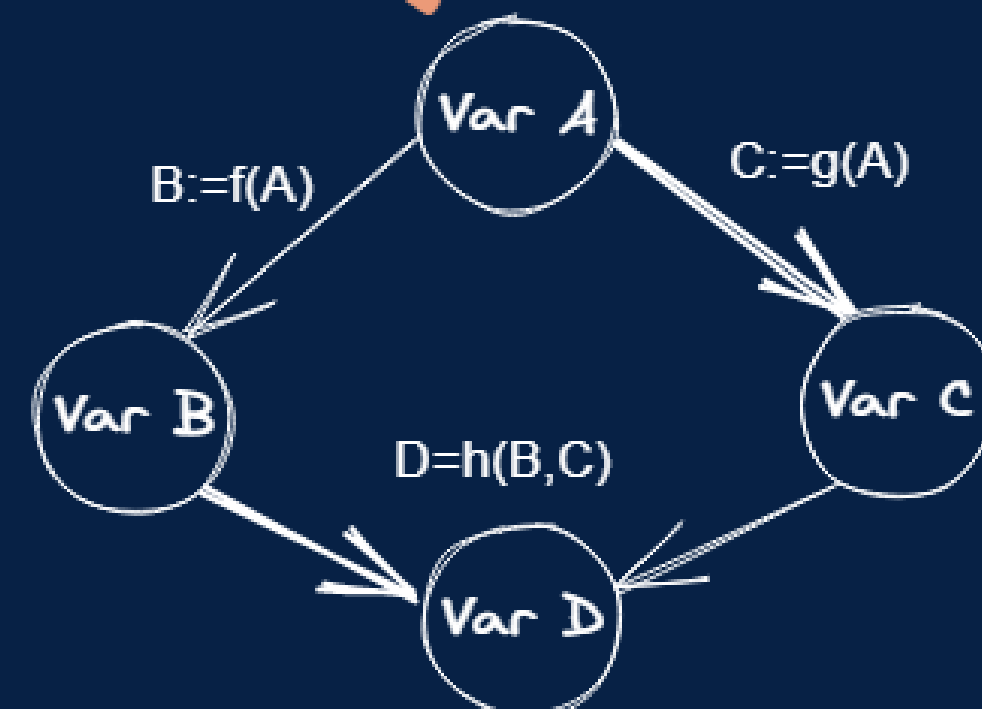
Goal Use Excel-like Formula in your Object's Class

State-of-the-Art Reactive variables by their nature suffer from:

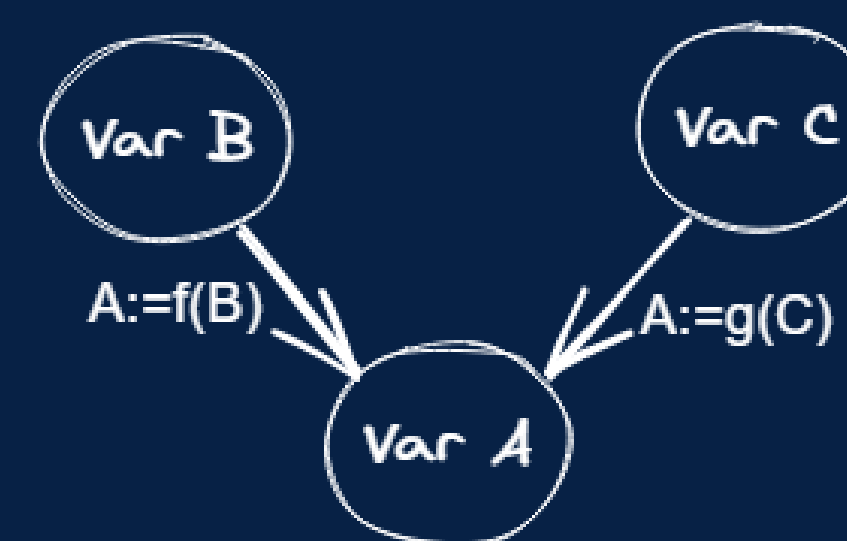
(1) cycles



(2) glitches

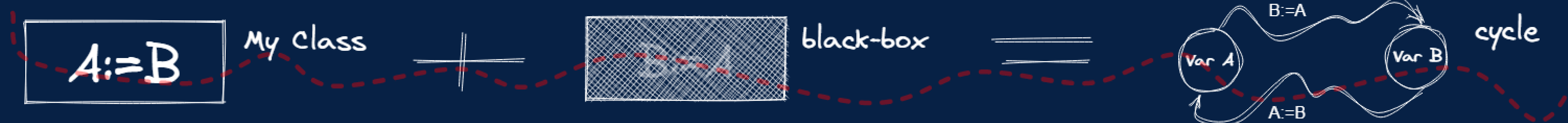


(3) reassignments



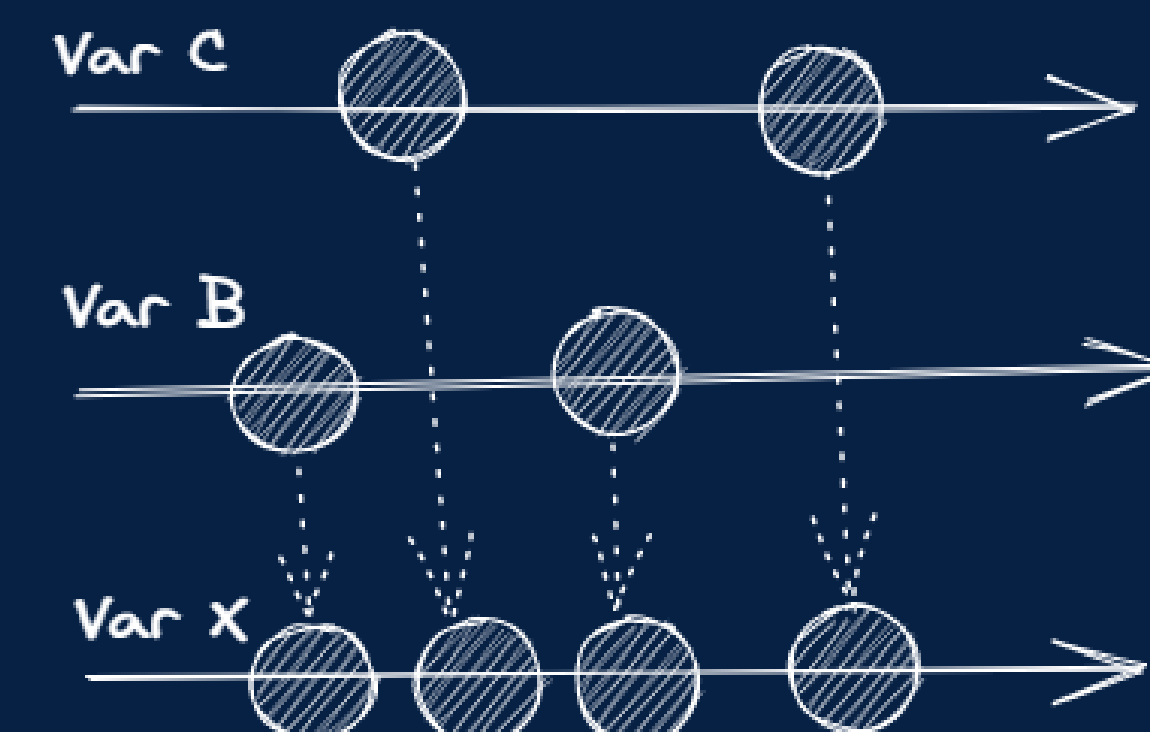
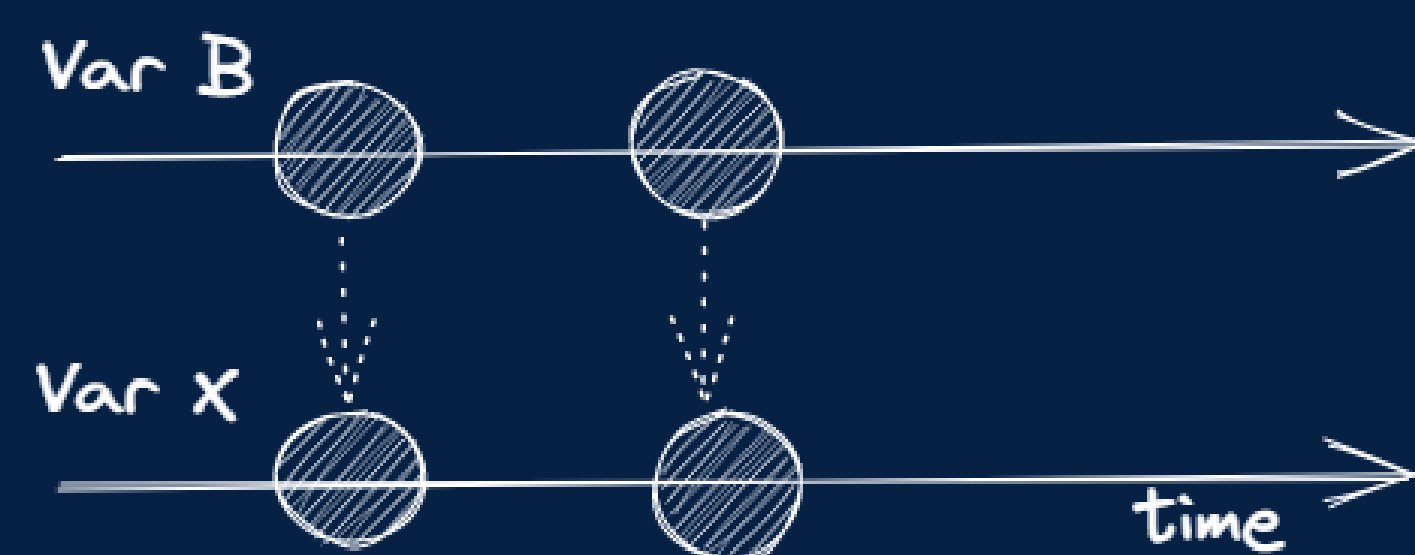
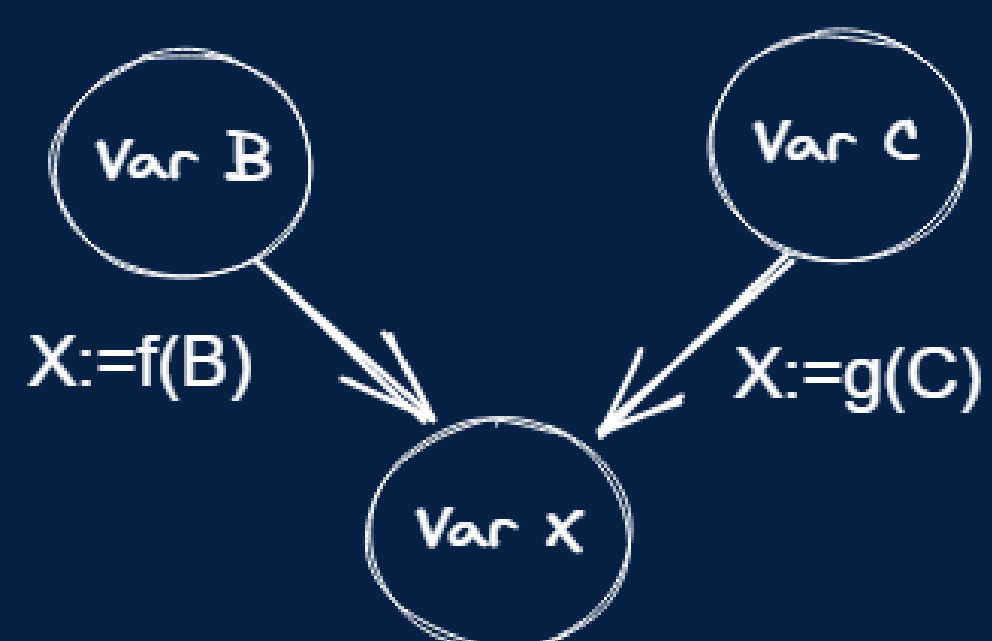
## Limitation of Current Approaches

Current approaches include: avoidance, iterations, and central management. However, when combining reactive variables with instance variables in OOP -- RIVars are subject to unsupported constructions, unpredictable results, and no-isolation.



## Our Approach

(1) support multiple assignments; (2) streams: values over time; (3) merge streams



על

רגל אחת

(Reactive Instance Variable) מושג מופע ריאקטיבי (Reactive Variable) הוא שילוב של משתנה ריאקטיבי (FRP), מן הפרדיגמה של תכנות ריאקטיבי-פונקציונלי (FRP), יחד עם משתנה מופע (Instance Variable) מן הפרדיגמה של תכנות מונחה עצמים (OOP). בדומה למשתנה ריאקטיבי, ניתן להגדיר לו חישוב אוטומטי ע"י קישור לביטוי. בדומה למשתנה מופע, הוא יכול להיות חלק מממשק (Interface).

(1) ממשק ספק/לקוח

בין מחלקת אב לבן

```
class Pump : Bag
{
    public RIVar<decimal> Rate = new ...
    public RIVar<decimal> Dose = new ...
    public RIVar<decimal> Duration = new ...

    public Pump()
    {
        base.Amount.Set(Duration.Mul(Dose));
        base.Volume.Set(Duration.Mul(Rate));
    }
}
```

```
class Pump
{
    public RIVar<decimal> Rate = new ...
    public RIVar<decimal> Dose = new ...
    public RIVar<decimal> Duration = new ...

    public Pump(IBag bag)
    {
        bag.Amount.Set(Duration.Mul(Dose));
        bag.Volume.Set(Duration.Mul(Rate));
    }
}
```

## Motivation

דוגמא

OOP ממשק המשתמש שלהלן מורכב מחלק עליון וחלק תחתון, שהם שני micro-frontends המתקשרים ביניהם באמצעות אירועים. נדרש לבצע חישובים שונים בתגובה לשינויים בשדות. כל חלק נדרש לבצע את החישובים שלו מבלי להכיר מה הם החישובים של החלק האחר. הואיל ולרופא נושא באחריות למינון שהוא רושם, אסור למערכת לשנות (על ידי חישוב) ערך שקבע הרופא באופן בלתי צפוי.

Drug	VolumeOffFluid	Concentration
<del>100</del> 99	300	0.33
Dose	Duration	Rate
10	10	30

אולם, על אף שהחישובים עיקריים, חישוב עלול לגרום לשינויים. הסיבה היא אפקט אפשרי של איבוד מידע, למשל כאשר משתמשים בייצוג עשרוני.

\*ערכים מחושבים מופעים בגופן *Italics*.

## FRP

בדוגמא שלהלן, בכל הוספה של מקור נתונים למינון (Amount), מעדכנים את מקור הנתונים של משתנה Alert (כמוצג בשלבים א'-ג'). אם שוכחים לעדכן (בעיה רלוונטית במערכות גדולות), נוצר חוסר עקביות בין הערכים של Amount ו-Alert. בנוסף (שלב ד'), כאשר מגדירים חישוב מן המשתנה Amount, מגדירים משתנה Amount נוסף, ולא מעדכנים את מקור הנתונים של Alert. כך נוצרים מספר משתנים שמציינים את Amount וחוסר עקביות בשימוש בהם. שלב א': Alert ישתנה אוטומטית בהתאם למינון (Amount).

```
Amount=FromInput()
Alert=IsAbnormal(Amount)
```

שלב ב': המינון יכול להיקבע על ידי מכפלה של Concentration ו-Volume.

```
AmountByInput=FromInput()
AmountByConcentrationAndVolume=Concentration*Volume
Alert=Or(IsAbnormal(AmountByInput),IsAbnormal(AmountByConcentrationAndVolume))
```

שלב ג': המינון יכול להקבע על ידי מכפלה של Duration ו-Dose.

```
AmountByConcentrationAndVolume=Concentration*Volume
AmountByDoseAndDuration=Dose*Duration
AmountByInput=FromInput()
Alert=Or(IsAbnormal(AmountByInput),IsAbnormal(AmountByConcentrationAndVolume),IsAbnormal(AmountByDoseAndDuration))
```

שלב ד': במידה והוזן Amount ו-Duration, יש לחשב את Dose.

```
AmountByInputOrByConcentrationAndVolume=Merge(
    AmountByConcentrationAndVolume,AmountByInput)
DoseByDoseAndDuration=AmountByInputOrByConcentrationAndVolume / Duration
```

## Patent

מימון

- "Reactive Calculated Signals Method".  
US Provisional Application No. 63/061204. Aug 5, 2020.

NuGet

GitHub



brandrivka@gmail.com  
lorenz@openu.ac.il