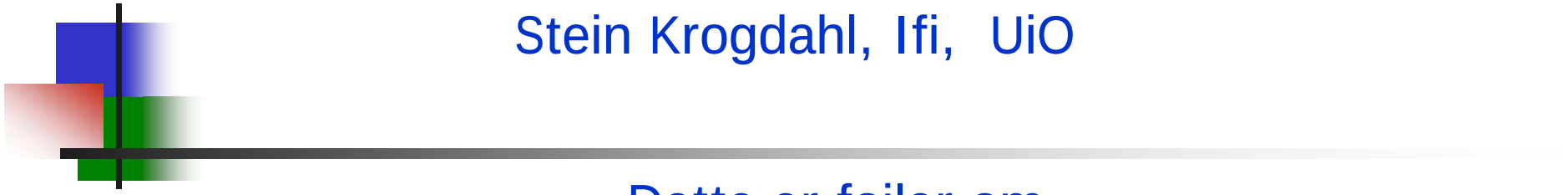


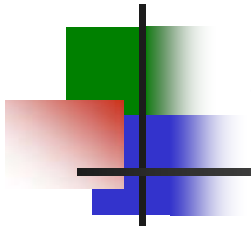
INF5110 – 7. mai 2014

Stein Krogdahl, Ifi, UiO

Dette er foiler om

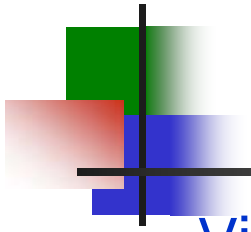
global data-analyse

NB: Disse foilene er også pensum (men stoffet finnes bare her på disse foilene.)



Global dataflyt-analyse, eksempler:

- Gitt en TA-instruksjon der variablen x brukes:
  - Finn alle de tilordninger (definisjoner) der denne verdien på x kan ha vært satt
- Gitt en tilordning der x blir satt:
  - Finn alle de steder der denne verdien av x kan bli brukt
- Disse og liknende ting:
  - Kan ”lett” bergenes ved en kompletterings-algoritme omtrent som når vi finner First og Follow.
- Vi skal se på en slik algoritme som finner:
  - Hvilke variable som *faktisk* er i live etter hver basal blokk



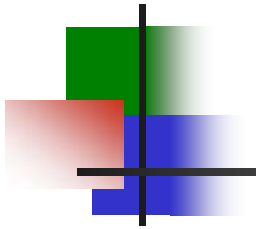
Vi skal se på global dataflyt-analyse for å finne: Hvilke variable er *egentlig* i live etter en basal blokk?

- Vi har tidligere antatt at *alle* programvariable er i live etter hver basal blokk, og at *ingen* temporær-variable er det.

- Den analysen vi ser på her vil gi svar på hvilke variable som *faktisk* er i live etter hver blokk.

  - Altså, gitt en blokk: For hvilke variable det er mulig at deres verdi på slutten av blokka vil bli brukt videre i utførelesen.

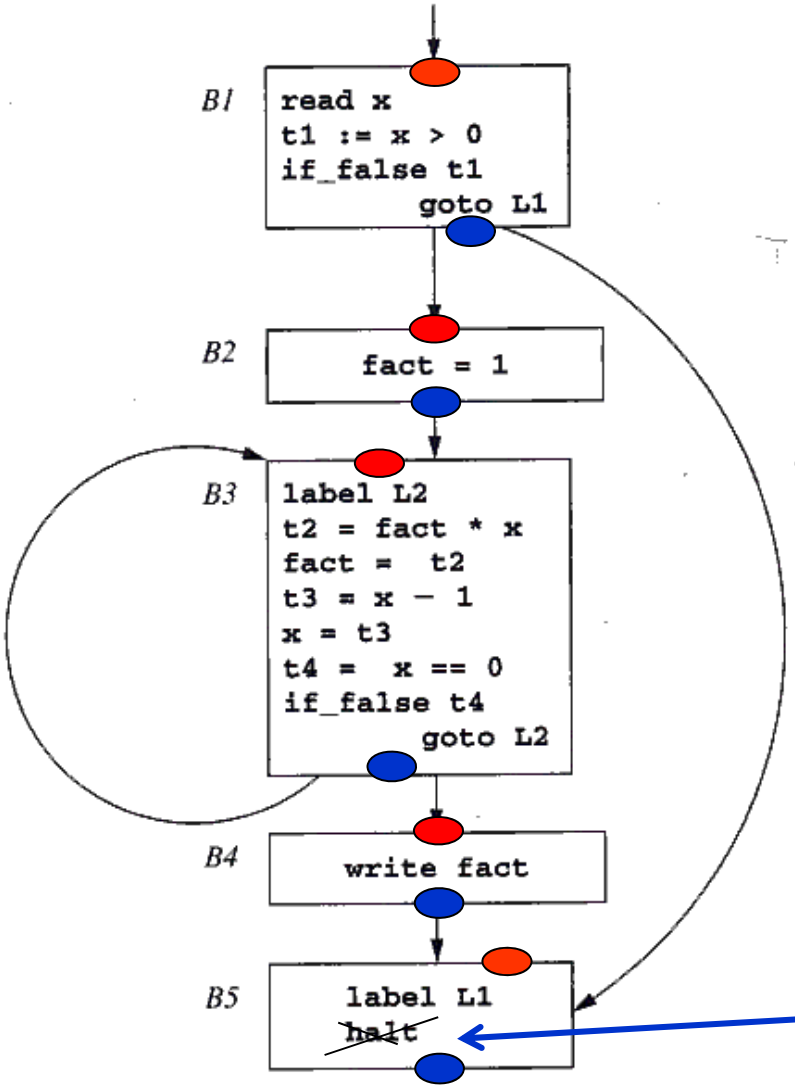
  - Vi kan godt her også tillate at temporære variable kan bære verdier fra en basal blokk til en annen (og altså kan være i live etter en blokk).



Data for algoritmen som finner:
Hvilke variable er i live etter en basal blokk

- Hver basal blokk har, ut fra strukturen på flytgrafen, en mengde av (direkte) *etterfølgere* og *forgjengere* (mengder av basale blokker), og disse vil altså være konstant gjennom algoritmen!
  - Begge disse mengdene kan inkludere blokken selv
- Under algoritmen har hver basal blokk to mengder av variable knyttet til seg (og disse vil stadig få flere elementer gjennom algoritmen)
  - InLive: Dette skal bli mengen av variable som er i live helt i starten av den aktuelle blokka. Initielt er denne mengden tom for alle blokker.
  - OutLive: Dette skal bli mengden av variable som er i live helt på slutten av den aktuelle blokka. Det er *den* vi egentlig er interessert i (for å kunne initialisere algoritmen som finner NextUse etc. med så *få variable* som mulig). Også disse skal fra starten være tomme.

Eksempel på de forskjellige mengder m.m.



Forgjenger-mengdene er:

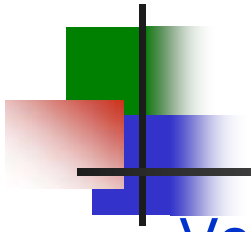
- For B1: ingen (starten av alg.)
- For B2: B1
- For B3: B2 og B3
- For B4: B3
- For B5: B1 og B1

: InLive

: OutLive

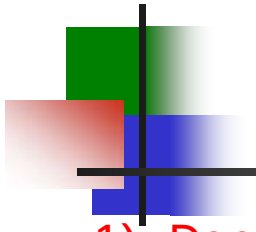
Denne instruksjonen tenker vi oss byttet ut med:

return (fact)



Mer global dataflyt-analyse

- Ved “global analyse” ser vi på *hele* flytgrafen for en *metode*
- Vi ser på hver basal blokk, og koker den informasjonen vi trenger om *hver variabel* i *hver blokk* ned til ett av følgende tre tilfeller:
  1) Blir den verdien som er i variablen ved inngang av blokken brukt før den eventuelt blir gitt ny verdi ved tilordning?
  2) Får variablen ny verdi ved tilordning før den eventuelt blir brukt?
  3) Blir variablen hverken brukt eller får ny verdi?
- Denne informasjonen kan lett samles inn for hver blokk og hver variabel.
- Algoritmen vil generelt arbeide seg “bakover”, både gjennom de basale blokkene, og langs kantene.



Hva kan vi i de tre tilfellene slutte om:
Innholdet av InLive når vi kjenner OutLive

1) Den verdien som er i variablen x ved inngang av blokken blir brukt før x eventuelt blir gitt ny verdi ved tilordning. Eksempler:

x = x + y	a = x + y
y = u + v	y = u + v

Her skal InLive inneholde x uansett om den var med i OutLive eller ikke

2) Variablen x får ny verdi ved tilordning før den eventuelt blir brukt. Eksempler:

x = u + v	*x = u + v*
a = x + y	*a = w + y*

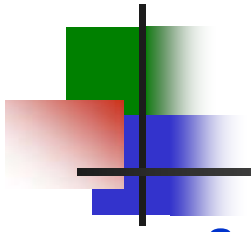
Her skal InLive *ikke* inneholde x uansett om den var med i OutLive eller ikke

3) Variablen x blir hverken brukt eller får ny verdi i blokka. Eksempel:

a = w + y

w = u + v

Her skal InLive inneholde x hvis og bare hvis den var med i OutLive



Kompletterings-algoritmen:

- Selve algoritmen går på at vi utfører følgende kompletteringssteg så lenge ett av de tre *gir noe nytt* for en eller annen variabel og en eller annen blokk:
  - Tilbakeføring gjennom en blokk:

    Velg en variabel og en blokk, og se om det er tilfelle 1), 2) eller 3). Ut fra det og innholdet i OutLive, avgjør så om x skal med i InLive.
  - Tilbakeføring fra InLive til forgjengerenes OutLive:

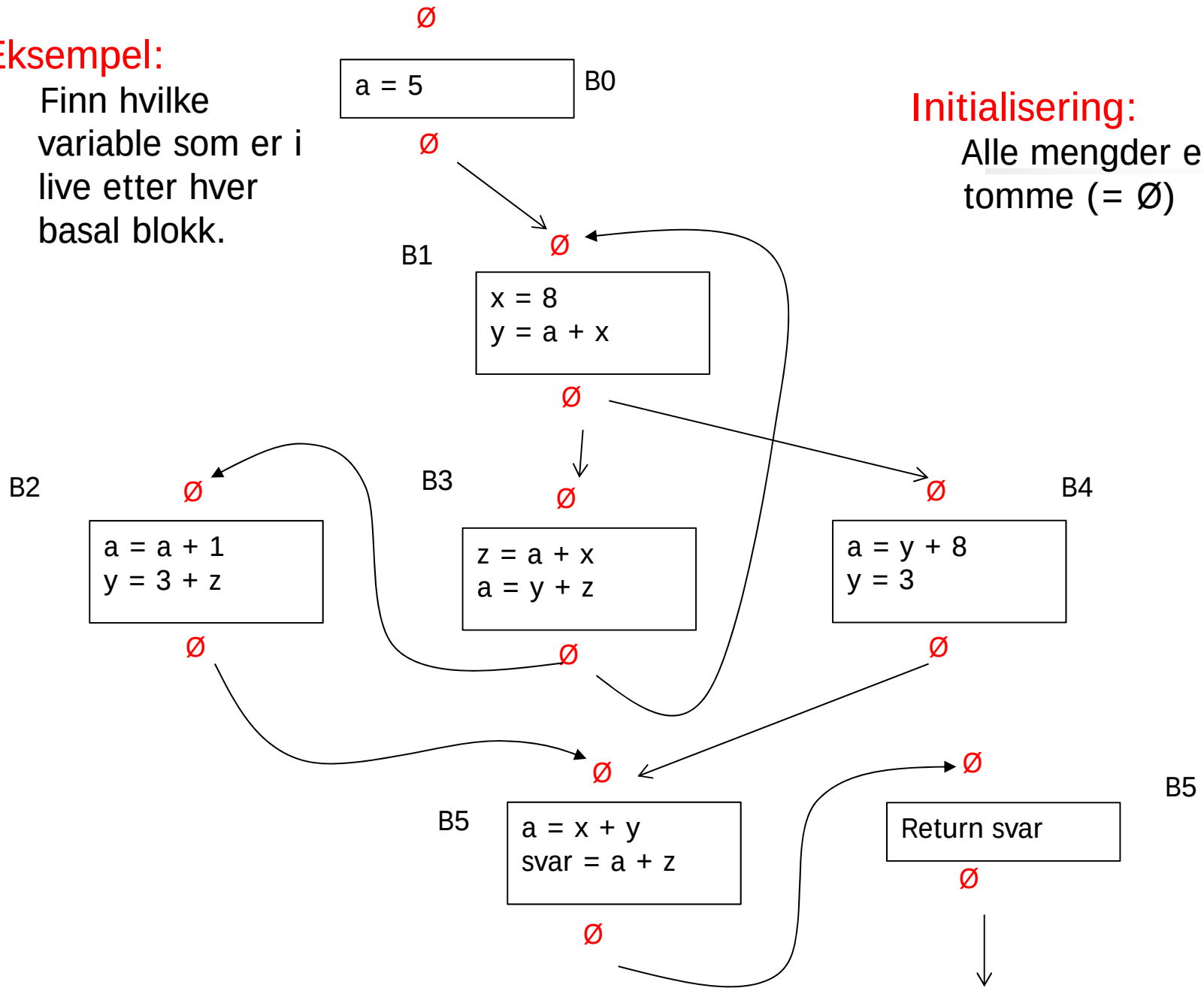
    Se på en InLive-mengde, og sørg for alle dens variable er inkludert i OutLine-mengdene til alle forgjenger-blokker
  - Virker fordi alt som her legges inn i en av mengdene InLive eller OutLive ved disse skrittene *må* være med i de respektive mengdene, og fordi vi til slutt må få med alle som skal være med (siden alt er endelig).
  - Det som til slutt står i OutLive-mengdene er nå de riktige (og disse kan så brukes som initialisering til algoritmen som finner next-use etc.)

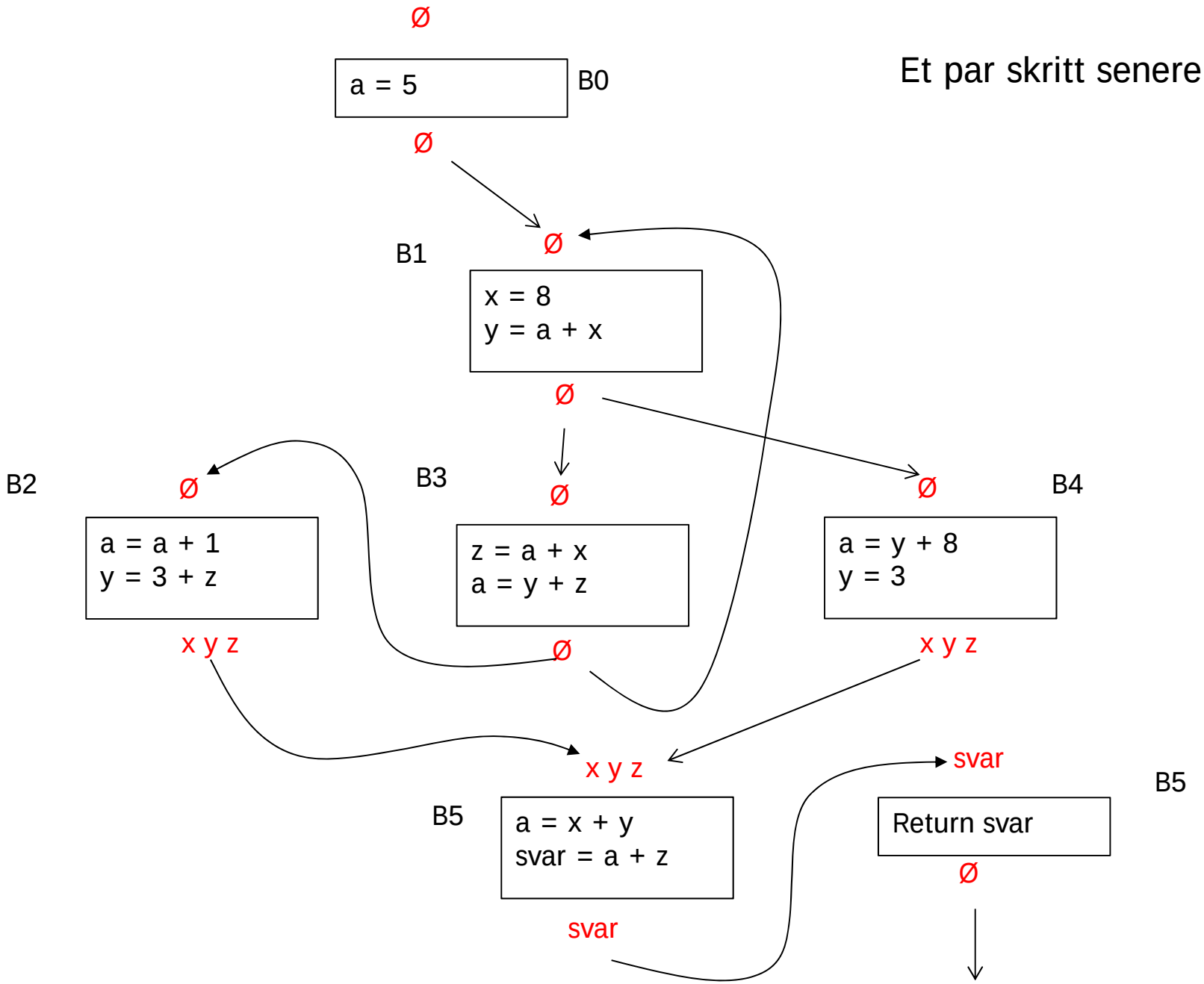
Eksempel:

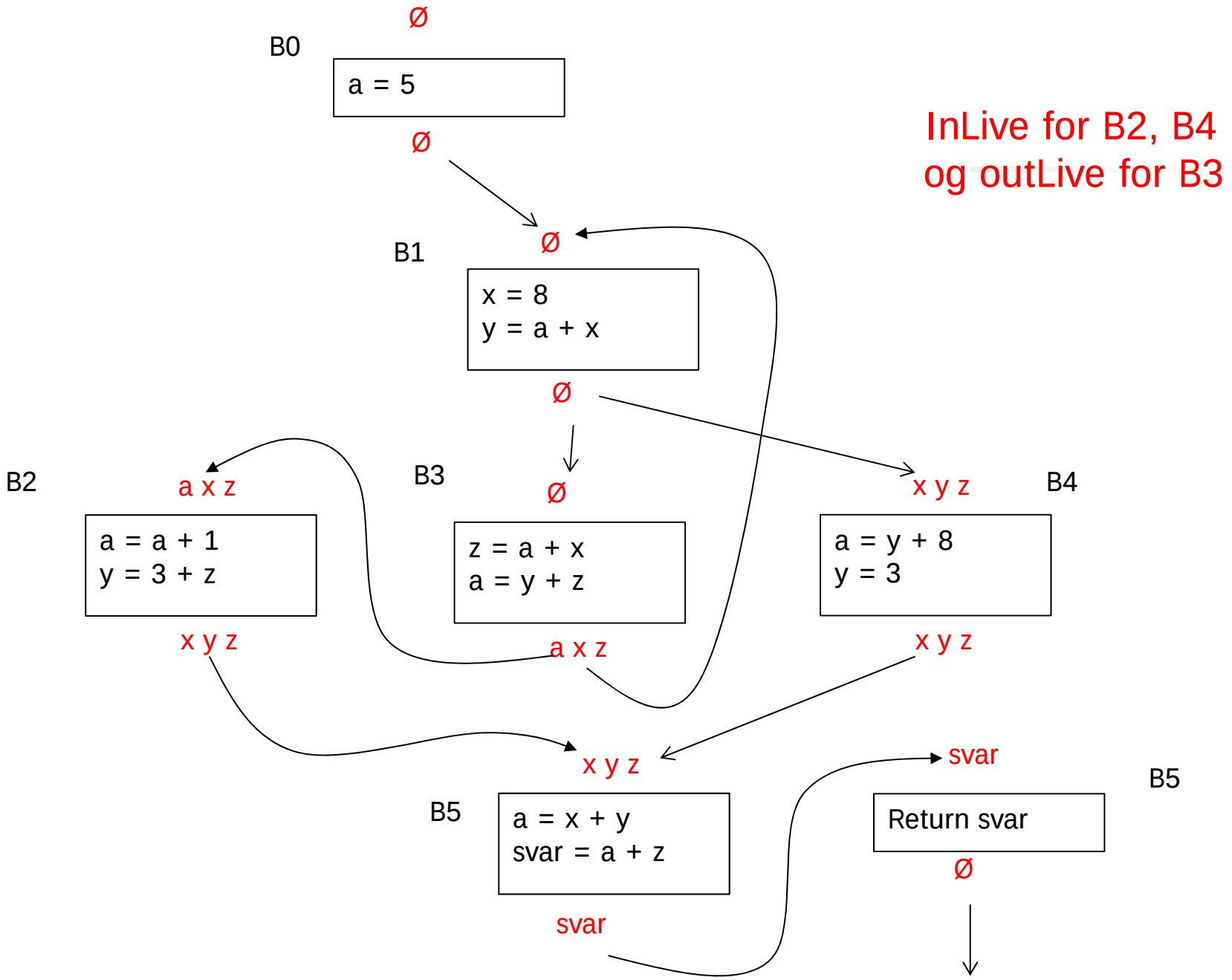
Finn hvilke variable som er i live etter hver basal blokk.

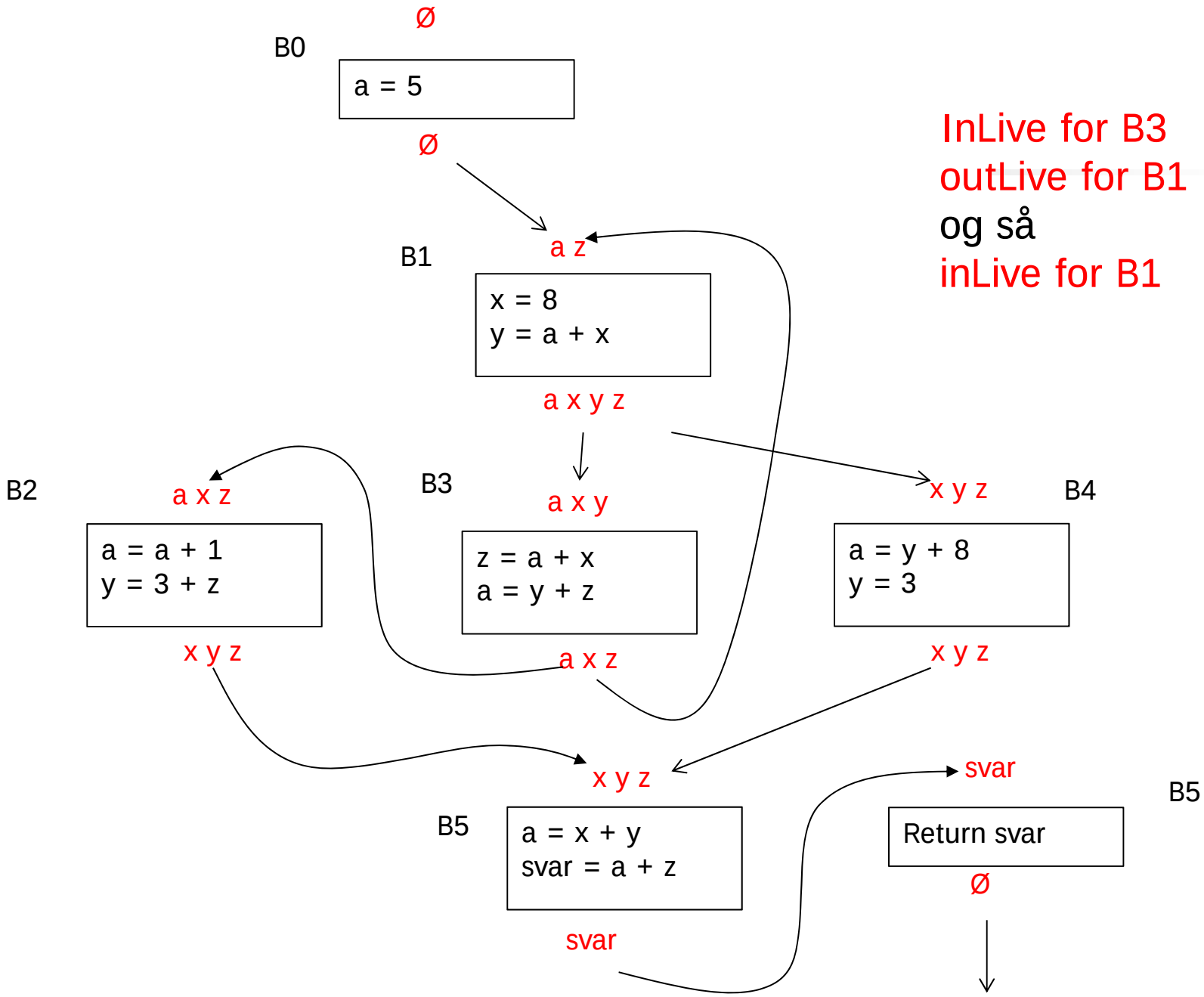
Initialisering:

Alle mengder er tomme (= Ø)

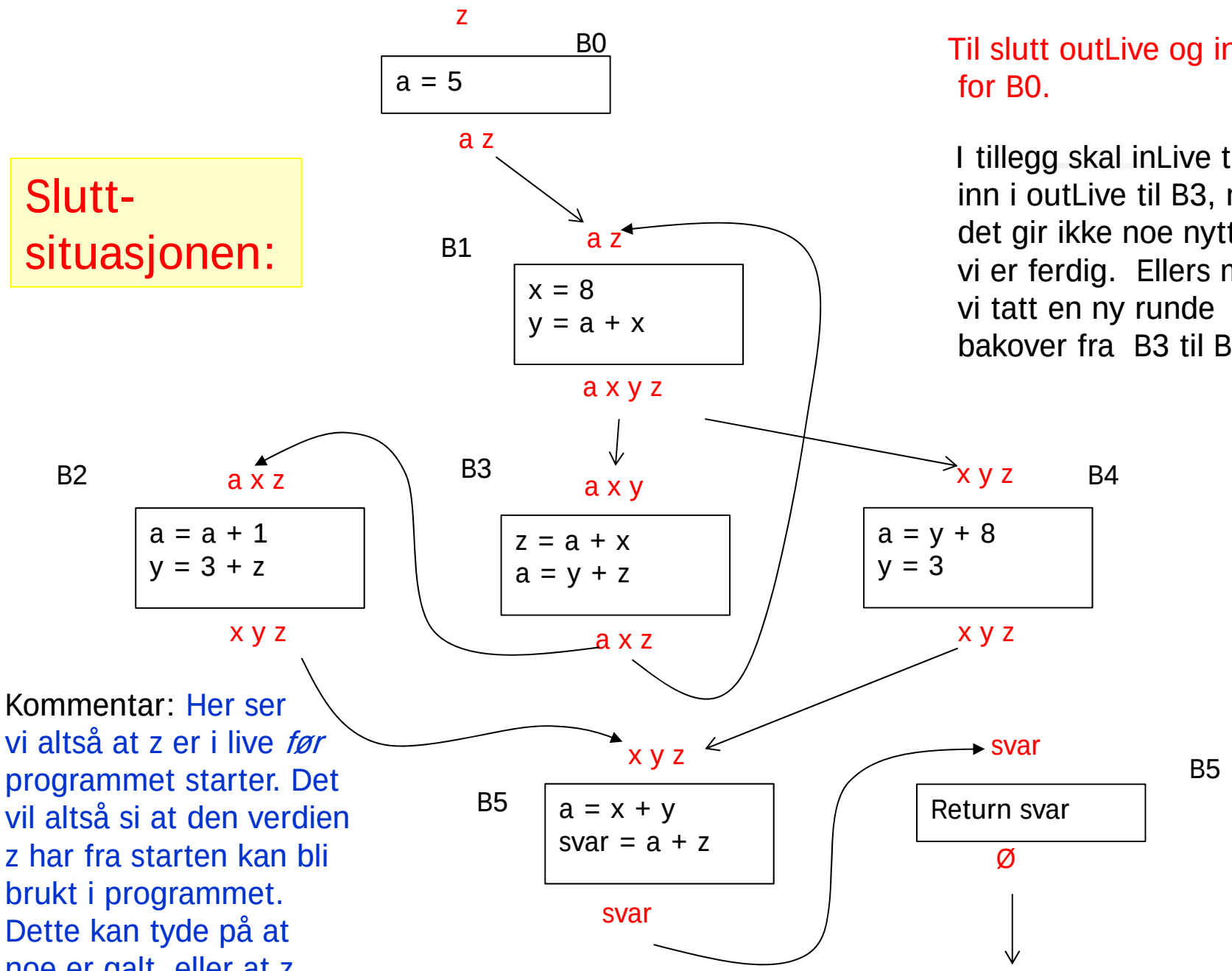



Et par skritt senere
Ø
a = 5
B0
Ø
B1
Ø
x = 8
y = a + x
Ø
B2
Ø
a = a + 1
y = 3 + z
x y z
B3
Ø
z = a + x
a = y + z
Ø
B4
Ø
a = y + 8
y = 3
x y z
x y z
B5
a = x + y
svar = a + z
svar
svar
B5
Return svar
Ø


Ø
B0
a = 5
Ø
InLive for B2, B4
og outLive for B3
Ø
B1
x = 8
y = a + x
Ø
B2
a x z
a = a + 1
y = 3 + z
x y z
B3
Ø
z = a + x
a = y + z
a x z
x y z
B4
a = y + 8
y = 3
x y z
x y z
B5
a = x + y
svar = a + z
svar
svar
B5
Return svar
Ø


Ø
B0
a = 5
Ø
a z
B1
x = 8
y = a + x
a x y z
B2
a x z
a = a + 1
y = 3 + z
x y z
B3
a x y
z = a + x
a = y + z
a x z
x y z
B4
a = y + 8
y = 3
x y z
x y z
B5
a = x + y
svar = a + z
svar
svar
B5
Return svar
Ø
InLive for B3
outLive for B1
og så
inLive for B1

Slutt-situasjonen:



Til slutt outLive og inLive for B0.

I tillegg skal inLive til B1 inn i outLive til B3, men det gir ikke noe nytt, så vi er ferdig. Ellers måtte vi tatt en ny runde bakover fra B3 til B2 osv.

Kommentar: Her ser vi altså at z er i live *før* programmet starter. Det vil altså si at den verdien z har fra starten kan bli brukt i programmet. Dette kan tyde på at noe er galt, eller at z f.eks. er en parameter til hele metoden

En oppgave:

En liten vri på situasjonen gir dette til en mer interessant oppgave.