Sensorveiledning mappeoppgave 2 i SOK-2030

I oppgaveteksten blir studentene bedt om å redegjøre for de økonomiske modeller og løsningskonsepter som de velger å bruke i sin besvarelse. I bedømmelsen av oppgavene vil det vektlegges hvordan studentene har brukt de ulike økonomiske modellene til å svare på spørsmålene i oppgaveteksten, samt en generell forståelse i bruk av faglige begreper, modeller og løsningskonsepter. I kurset har vi brukt Python som dataverktøy for å løse modeller, og i besvarelsene skal studentene legge ved utskrift av Python-kodene som er brukt for å løse oppgavene. For oppgave 1 er det utarbeidet koder for løsning av Stackelberg modell i notebook for kurset (som da er tilgjengelig for studentene). For oppgave 2 må studentene utarbeide egne koder, men kan ta utgangspunkt i liknende oppgaveløsninger som ligger i notebook.

I oppgave 1 skal studentene regne ut optimal markedstilpasning i et marked med Olivita som lederbedrift og Dr Choice AS som følgerbedrift. Her skal de bruk Stackelberg model for å komme frem til markedsløsningen. Modellen løses i et to-trinns spill, hvor lederbedriften gjør sitt valg av kvantum på trinn 1, og deretter gjør følgerbedriften sitt valg på trinn 2. Modellen løses ved baklengs induksjon, hvor vi starter med trinn 2 og finner reaksjonsfunksjonen (RF2) til følgerbedriften. RF2 settes så inn i profittfunksjonen til lederbedriften, som deriveres og gir et optimalt nivå på 28 200 solgte flasker av Olivita. Optimalt salg av Easy Choice Omega 3 er på 14 100 flasker. Markedspris for begge produktene er på kr 285. Som lederbedrift i dette markedet vil Olivita oppnå en profitt på 3 627 000, mens følgerbedriften Dr Choice AS vil tjene 313 800 på sitt salg. Se vedlagt løsning i Python.

I dynamiske spill med homogene produkter og kvantumskonkurranse, som Stackelberg modell, er det slik at lederbedriften kommer best ut og oppnår høyest profitt.

I oppgave 2 a) skal studentene vurdere om en fusjon mellom to av tre bryggerier vil være lønnsom. For å kunne si noe om lønnsomheten ved en mulig fusjon, må en se på markedstilpasningen og profitten til bedriftene før fusjonen, og sammenligne dette med profitten som selskapene vil oppnå etter fusjonen. Ved Cournot konkurranse og symmetriske bedrifter vil aldri en fusjon mellom 2 av 3 selskaper være lønnsomt. Ved asymmetri mellom selskapene kan kostnadsbesparelser gjør en fusjon lønnsom for det fusjonerte selskapet.

Før fusjon vil produksjonsnivået for Bryggeri 13 og Graff Brygghus være på 10 125 flasker per år, mens Mack Mikrobryggeri, som er noe mer effektiv i sin produksjon, vil produsere 10 875 flasker per år. Prisen i markedet vil være på kr 50.50. Total profitt for Bryggeri 13 og Graff Brygghus er på kr 110 062, og for Mack Mikrobryggeri er profitten på 173 062.

Ved fusjon mellom Mack Mikrobryggeri og Bryggeri 13 vil det nye selskapet ha en mer effektiv produksjon, med marginalkostnader på kr 7, og en kostnadsbesparelse på de faste kostnadene på kr 100 000.

Etter fusjonen vil det nye selskapet produsere 14 250 flasker, og Graff Brygghus vil produsere 13 500 flasker per år. Markedsprisen vil nå være på kr 64. Profitt for det fusjonerte selskapet vil være på 312 250, og for Graff Brygghus vil profitten bli på kr 429 000.

For å se på lønnsomhet av fusjon sammenligner vi profitten før og etter fusjonen. Profitt før fusjon vil være på 283 124 for Mack Mikrobryggeri og Bryggeri 13, og etter fusjon vil de tjene 312 250. Graff Brygghus er de som tjener mest på en slik fusjon, med en økning i profitt på kr 318 938. For utregning se vedlagt løsning i Python.

I Oppgave 2 b) skal studentene bruke en modell for vertikale relasjoner. I første del av oppgaven, vil det være duopol i oppstrømsleddet og monopol i nedstrømsmarkedet. Siden vi har to produksjonsbedrifter i oppstrømmarkedet vil vi anta kvantum som strategisk variable og likevekten kan løses ved Counot modell. I andre del av oppgaven, vil bedriftene ha monopol i begge ledd i verdikjeden.

Modellen løses i et to-trinns spill, hvor oppstrømsbedriftene velger pris til nedstrømsaktøren i trinn 1, og i trinn 2 velger nedstømsaktøren optimal pris og kvantum til sluttbruker. For å løse modellen starter vi i trinn 2, og finner optimal tilpasning for RGH der marginalinntekten (*175 - 4Q*) er lik innkjøpskostnaden per flaske mikroøl *r*. Dette gir *Q =(175 – r)/4*, som også er den direkte etterspørselen til bryggeriene Mack mikrobrygg 13 og Graff Brygghus. Bryggeriene setter pris *r = 63*. Dette vil gi et salg for RGH på 28 000 flasker per år, til en pris på kr 119 per flaske mikroøl til sluttbruker. Profitt for RGH er på kr 968 000, og for bryggeriene er profitten på 284 000.

Ved en fusjon mellom Mack Mikrobrygg 13 og Graff Brygghus vil vi få monopol i begge markedene. Dette vil gi økte priser, lavere kvantum og høyere profitt for bedriftene. Solgt kvantum i sluttbrukermarkedet er på 21 000 flasker til en pris på kr 133 per flaske mikroøl. Pris fra det fusjonerte bryggeriet til RGH vil være på kr 91. Profitt for RGH reduseres til 382 000, og for bryggeriet er profitten på 764 000.

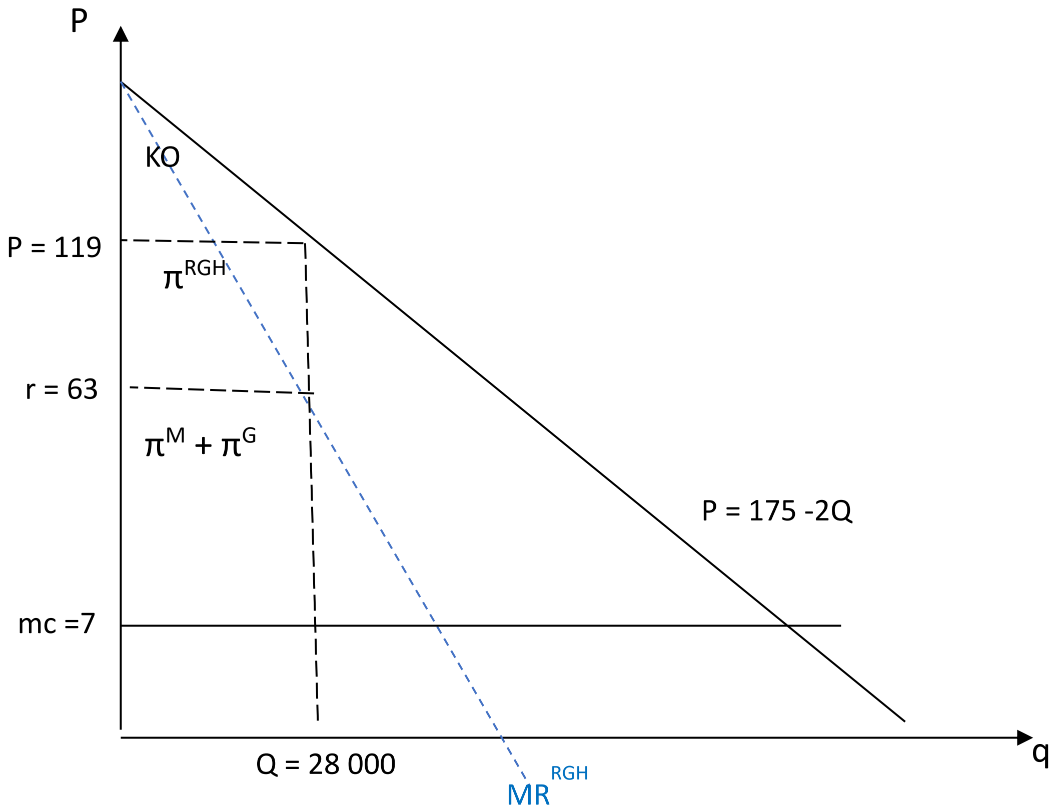
For bryggeriene vil denne fusjonen være lønnsom. Samlet profitt før fusjon er på 568 000, og etter fusjon tjener selskapet 196 000 mer.

I oppgave 2 c) skal studentene redegjøre for de samfunnsøkonomiske konsekvenser av en fusjon mellom Mack Mikrobrygg 13 og Graff Brygghus. Her bør de komme inn på dobbel marginalisering ved monopol i begge leddene i verdikjeden, hvor oppstømsaktøren oppnår dobbel så stor profitt som nedstømsaktøren (når vi ser bort i fra faste kostnader). De bør gi en forklaring om at konkurranse gjør at en vil ta ut en lavere margin i oppstømsleddet.

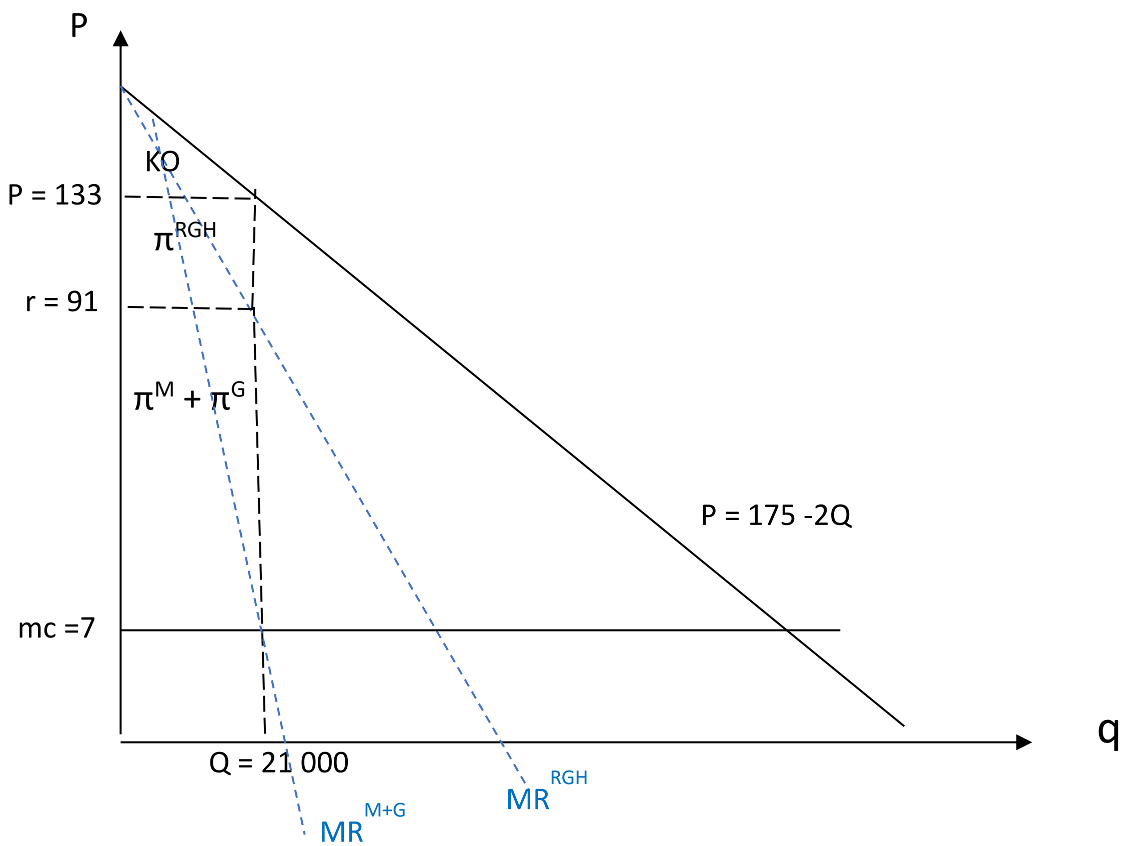
Markedskonsekvensene av fusjon i oppstømsleddet vil være redusert kvantum fra 28 000 til 21 000 flasker og en prisøkning til sluttbruker fra kr 119 til kr 133. Profitten til RGH vil reduseres som følge av fusjon; fra 968 000 til 382 000. For de fusjonerte oppstrømsbedriftene vil profitten øke fra 2* 284 000 til 764 000.

En kan også se på konsumentoverskuddet i sluttbrukermarkedet; som før fusjon vil være på 784 000 og på 441 000 etter fusjon. Altså en betydelig reduksjon i konsumentoverskudd som følge av monopol i begge ledd i verdikjeden.

Figuren nedenfor illustrer optimal tilpasning og dobbel marginalisering. Vedlegg fra Python viser koder for utregning av oppgaven.



Figur 1. Tilpasning i restaurantmarkedet før en fusjon i oppstrømsleddet



Figur 2. Tilpasning i restaurantmarkedet etter fusjon i oppstrømsleddet

Python-kode for oppgave 1)

Ved Stackelberg modell er kvantum bedriftens handlingsvariable og bedriftene gjør sine valg sekvensielt. Lederbedriften velger først hvor mye den vil produser og deretter velger følgerbedriften sitt volum. Prisen er gitt ved etterspørselsfunksjonen.

Vi antar at vi har to bedrifter som konkurrer med kvantum som strategisk variabel. Invers ettersprøsel er gitt ved: P(Q) = 990-1/60(q1 + q2). Vi antar at bedriften har identiske marginalkostnader lik 50 per produsert flaske og faste kostnader på 3 millioner.

```
from sympy import *
from matplotlib import pyplot as plt
import numpy as np
```

```
q1, q2,c, a, b, pi,i=symbols('q1 q2 c a b pi i')

def P_demand(Q,a,b):
    return 990-1/60*Q

def profit(q1,q2,c,a,b):
    return (P_demand(q1+q2,a,b)-50)*q1
```

Modellen løses i to trinn. På trinn 1 velger lederbedriften sitt kvantum og på trinn 2 velger følgerbedriften sitt optimale kvantum. Modellen løses ved baklengs induksjon, dvs av vi starter på trinn 2

Vi dereiverer profittfunksjon til bedrift 2 mhp q2: π2 = (P-c)*q2 = (100-1/60(q1+q2)-50)*q2

```
d_profit2_Q=diff(profit(q2,q1,c,a,b),q2)
d_profit2_Q
```

$$-0.0166666666666667q_1 - 0.0333333333333333q_2 + 940$$

Setter den derivert lik 0 og finner reaksjonsfunksjon til bedrift 2

```
Q2_sol1=solve(d_profit2_Q,q2)[0]
Q2_sol1
```

$$28200.0 - 0.5q_1$$

På trinn 1 sette vi reaksjonsfunksjonene til bedrift 2 inn i bedrift 1 sin profittfunksjon, og deriverer dette utrykket mhp q1. Maks π1 =(990-1/60(q1+q2)-50)*q1 gitt q2=28200-0.5q1

```
d_profit1_Q=diff(profit(q1,Q2_sol1,c,a,b),q1)
d_profit1_Q
```

$470.0 - 0.0166666666666667q_1$

For å finne optimalt kvantum til lederbedriften setter vi uttrykket over lik 0

```
Q1_sol=solve(d_profit1_Q,q1)[0]
Q1_sol
```

28200.0

Vi setter så optimalt valg av q1 inn i reaksjonsfunksjonen til bedrift 2

```
Q2_sol2=Q2_sol1.subs({q1:Q1_sol})
Q2_sol2
```

14100.0

```
def P_demand(q1,q2):
    return 990-1/60*(q1+q2)
```

```
# Markedspris:
P_demand(q1,q2).subs({q1:Q1_sol,q2:Q2_sol2})
```

285.0

```
# Profitt for bedriftene er lik (p -c)q:
def profit(q1):
    return (P_demand(q1,q2).subs({q1:Q1_sol,q2:Q2_sol2})-50)*Q1_sol-3000000

round(profit(q1),2)
```

3627000.0

```
# Profitt for bedriftene er lik (p -c)q:
def profit(q2):
    return (P_demand(q1,q2).subs({q1:Q1_sol,q2:Q2_sol2})-50)*Q2_sol2-3000000

round(profit(q2),2)
```

313500.0

Python-kode for oppgave 2a)

Vi antar at vi har tre bedrifter som konkurrer med kvantum som strategisk variabel. Invers etterspørsel er gitt ved: P(Q) = 150-Q og marginalkostnaden til to av bedriftene er på 30. Den tredje bedriften har potensielt høyere marginalkostnader; lik 30b, hvor b>1.

```
from sympy import *
from matplotlib import pyplot as plt
import numpy as np
```

```
q1,q2,q3,c1,c2,c3, a, b=symbols('q1 q2 q3 c1 c2 c3 a b')
```

```
def P_demand(Q,a,b):
    return a-b*Q

def profit(q1,q2,q3,c,a,b):
    return (P_demand(q1+q2+q3,a,b)-c)*q1
```

```
d_profit1_Q=diff(profit(q1,q2,q3,c1,a,b),q1)
d_profit2_Q=diff(profit(q2,q1,q3,c2,a,b),q2)
d_profit3_Q=diff(profit(q3,q2,q1,c3,a,b),q3)
display(d_profit1_Q)
display(d_profit2_Q)
display(d_profit3_Q)
```

$a - bq_1 - b(q_1 + q_2 + q_3) - c_1$

$a - bq_2 - b(q_1 + q_2 + q_3) - c_2$

$a - bq_3 - b(q_1 + q_2 + q_3) - c_3$

```
sol=solve([d_profit1_Q,d_profit2_Q,d_profit3_Q],[q1,q2,q3])

display(sol[q1])
display(sol[q2])
display(sol[q3])
```

$$\frac{a - 3c_1 + c_2 + c_3}{4b}$$

$$\frac{a + c_1 - 3c_2 + c_3}{4b}$$

$$\frac{a + c_1 + c_2 - 3c_3}{4b}$$

```python
cournot=lambdify(
    (a,b,c1,c2,c3),
    (sol[q1],sol[q2],sol[q3])
)
```

```python
cournot(175,4,10,10,7)
```

```
(10.125, 10.125, 10.875)
```

```python
a_value=175
b_value=4
c1_value=10
c2_value=10
c3_value=7
```

```python
q1sol, q2sol, q3sol=cournot(a_value,b_value,c1_value,c2_value,c3_value)
print (P_demand(q1sol+q2sol+q3sol,a_value,b_value))
```

```
50.5
```

```python
print(f"""Løsningen er at
bedrift 1 produserer {q1sol} med profitt {profit(q1sol,q2sol,q3sol,c1_value,a_value,b_value)},
bedrift 2 produserer {q2sol} med profitt {profit(q2sol,q1sol,q3sol,c2_value,a_value,b_value)} og
bedrift 3 produserer {q3sol} med profitt {profit(q3sol,q1sol,q2sol,c3_value,a_value,b_value)}
prisen blir {P_demand(q1sol+q2sol+q3sol,a_value,b_value)}""")
```

```
Løsningen er at
bedrift 1 produserer 10.125 med profitt 410.0625,
bedrift 2 produserer 10.125 med profitt 410.0625 og
bedrift 3 produserer 10.875 med profitt 473.0625
prisen blir 50.5
```

Horisontal fusjon

Anta at 2 og 3 fusjonerer, og at all produksjon flyttes til bedrift 2. Ny tilpasning blir da:

```
def P_demand1(Q,a,b):
    return a-b*Q

def profitF(q1,q2,c,a,b):
    return (P_demand(q1+q2,a,b)-c)*q1
```

```
d_profitF1_Q=diff(profitF(q1,q2,c1,a,b),q1)
d_profitF2_Q=diff(profitF(q2,q1,c2,a,b),q2)

display(d_profitF1_Q)
display(d_profitF2_Q)
```

$$a - bq_1 - b(q_1 + q_2) - c_1$$

$$a - bq_2 - b(q_1 + q_2) - c_2$$

```
sol=solve([d_profitF1_Q,d_profitF2_Q],[q1,q2])

display(sol[q1])
display(sol[q2])
```

$$\frac{a - 2c_1 + c_2}{3b}$$

$$\frac{a + c_1 - 2c_2}{3b}$$

```
cournot=lambdify(
    (a,b,c1,c2),
    (sol[q1],sol[q2])
)
```

```
cournot(175,4,10,7)
```

```
(13.5, 14.25)
```

```python
a_value=175
b_value=4
c1_value=10
c2_value=7
```

```python
q1sol, q2sol=cournot(a_value,b_value,c1_value,c2_value)
print (P_demand(q1sol+q2sol,a_value,b_value))
```

```
64.0
```

```python
print(f"""Løsningen er at
bedriftene produserer {q1sol} enheter som gir profitt lik {profitF(q1sol,q2sol,c1_value,a_value,b_value)} og
prisen i markedet blir {P_demand1(q1sol+q2sol,a_value,b_value)}""")
```

```
Løsningen er at
bedriftene produserer 13.5 enheter som gir profitt lik 729.0 og
prisen i markedet blir 64.0
```

```python
print(f"""Løsningen er at
bedrift 1 produserer {q1sol} med profitt {profitF(q1sol,q2sol,c1_value,a_value,b_value)} og
bedrift 2 produserer {q2sol} med profittF {profitF(q2sol,q1sol,c2_value,a_value,b_value)}
prisen blir {P_demand1(q1sol+q2sol,a_value,b_value)}""")
```

```
Løsningen er at
bedrift 1 produserer 13.5 med profitt 729.0 og
bedrift 2 produserer 14.25 med profittF 812.25
prisen blir 64.0
```

Python-kode for oppgave 2b) – Vertikal separasjon

Del 5 Relasjoner mellom bedrifter

Vertikal separasjon

Vi antar et marked bestående av to oppstrømsbedrifter, M1 og M2, og en nedstrømsbedrifter, R. Invers etterspørsel er gitt ved:

P(Q) = 175 - 2Q

Oppstrømsbedriftene produserer innsatsfaktor som selges til nedstrømsbedriften til pris lik w. Oppstrømsbedriftene har marginalkostnader lik 7, og nedstrømsbedriften ingen produksjonkostnader, kun nnkjøpskostander på w.

To-trinns spill:

```
Trinn 1: Oppstrømsbedriftene velger optimal pris til nedstrømsbedriftene; w
Trinn 2: Nedstrømsbedriften velger optimalt kvantum og pris i sluttbrukermarkedet; P
```

Vi bruker baklengs indukjon og starter med Trinn 2.

```
import sympy as sp
from sympy import *
import numpy as np
from matplotlib import pyplot as plt
```

Trinn 2: Nedstrømsbedriften velger optimalt produksjonsnivå som monopolist

```
qR1,qM1, qM2,c,k,w, A, B, QR, QM, P =symbols('q^R_1 q^M_1 q^M_2 c k w A B Q^R Q^M P ',
                                             positive=True)
```

```
def demand_R(QR):
           return (175-2*QR)
```

```
def marginalrevenue_R(QR):
    return (175-4*QR)
```

Optimal tilpasning der MR = MC

```
# Optimal tilpasning for RGH der MR = MC
equ_R1 = sp.Eq(marginalrevenue_R(QR),w)
equ_R1
```

$175 - 4Q^R = w$

Trinn 1: Oppstrømsbedriftene velger optimal pris til nedstrømsbedriftene

På trinn 1 vil etterspørselen være lik optimalt kvantum fra trinn 2: w = 175 - 4Q

```
# Etterspørsel til oppstrømsbedriftene
QR_eq = equ_R1
QR_eq
```

$175 - 4Q^R = w$

Oppstømsbedriftene velger optimalt kvantum og pris i Cournot konkurranse

```
def demand_1M(qM1):
           return (175-4*(qM1+qM2))
```

```
def demand_2M(qM2):
           return (175-4*(qM1+qM2))
```

```
def marginalrevenue_1M(qM1):
     return (175-8*qM1-4*qM2)
```

```
def marginalrevenue_2M(qM2):
     return (175-4*qM1-8*qM2)
```

```
# Optimal tilpasning for M1 der MR = MC

equ_M1 = sp.Eq(marginalrevenue_1M(qM1),7)
equ_M1
```

$-8q_1^M - 4q_2^M + 175 = 7$

```python
# Optimal tilpasning for M2 der MR = MC

equ_M2 = sp.Eq(marginalrevenue_2M(qM2),7)
equ_M2
```

$$-4q_1^M - 8q_2^M + 175 = 7$$

```python
#reaksjonsfunksjon til bedrift 1
qM1_equ = sp.solve(equ_M1, qM1)[0]
qM1_equ
```

$$21 - \frac{q_2^M}{2}$$

```python
#reaksjonsfunksjon til bedrift 2
qM2_equ = sp.solve(equ_M2, qM2)[0]
qM2_equ
```

$$21 - \frac{q_1^M}{2}$$

```python
#  setter uttrykk for qM2 inn i qM1 og finner optimalt kvantum til M1:

qM1_unresv_eq = qM1_equ.subs(qM2, qM2_equ)
qM1_opt = sp.solve(sp.Eq(qM1, qM1_unresv_eq), qM1)[0]
qM1_opt
```

$$14$$

```python
# Otimalt kvantum M1
qM1_unresv = sp.Eq(qM1, qM1_opt)
qM1_unresv
```

$$q_1^M = 14$$

```
# setter uttrykk for qM1 inn i qM2 og finner optimalt kvantum til M2:

qM2_unresv_eq = qM2_equ.subs(qM1, qM1_equ)
qM2_opt = sp.solve(sp.Eq(qM2, qM2_unresv_eq), qM2)[0]
qM2_opt
```

14

```
# Otimalt kvantum M1
qM2_unresv = sp.Eq(qM2, qM2_opt)
qM2_unresv
```

$q_2^M = 14$

```
# Totalt kvantum fra oppstrømsbedriftene
QM_eq = qM1_opt + qM2_opt
QM_eq
```

28

```
# Finner pris (w) ved å sette totalt kvantum inn i etterspørselen

w_opt=demand_1M(qM1).subs({qM1:qM1_opt,qM2:qM2_opt})
w_opt
```

63

```
# Optimal pris til nedstrømsbedriftene

w_equ = sp.Eq(w, demand_1M(qM1).subs({qM1:qM1_opt,qM2:qM2_opt}))
w_equ
```

$w = 63$

Optimalt kvantum og pris i nedstrømsmarkedet

```
# Setter inn for w i optimalt kvantum for nedstømsbedriften fra Trinn 2

QR_opt2=equ_R1.subs({w:w_opt})
sp.simplify(QR_opt2)
```

$Q^R = 28$

```
# Optimal pris til sluttbruker

P_equ = sp.Eq(P, demand_R(QR).subs({QR:QM_eq}))
P_equ
```

$P = 119$

```
# Profitt for nedstrømsbedriften er lik (p - w)Q:
def profitt(QR):
    return (demand_R(QR).subs({QR:QM_eq})-w.subs({w:w_opt}))*QM_eq

sp.simplify(profitt(qR1))
```

1568

```
# Profitt for oppstrømsbedriftene er lik (w-c)q:
def profitt(qM1):
    return (demand_1M(qM1).subs({qM1:qM1_opt,qM2:qM2_opt})-7)*qM2_opt

sp.simplify(profitt(qM1))
```

784

Python-kode for oppgave 2b) – Vertikal integrasjon

Vertikale fusjoner og dobbel-marginalisering

Vi antar at vi har to bedrifter, en oppstrømsbedrift Mack 13/Graff (r) og en nedstrømsbedrift RGH (d). Invers etterspørsel er gitt ved: P(Q) = 175 - 2Q. Oppstrømsbedriften produserer innsatsfaktor som selges til nedstrømsbedriften til pris lik r. Oppstrømsbedriften har marginalkostnader lik 7, og nedstrømsbedriften har kun innkjøpskostander på r.

To-trinns spill:

```
Trinn 1: Oppstrømsbedriften velger optimal pris r
Trinn 2: Nedstrømsbedriften velger kvantum og pris i sluttbrukermarkedet
```

Vi bruker baklengs indukjon og starter med Trinn 2.

```python
import sympy as sp
from sympy import *
import numpy as np
from matplotlib import pyplot as plt
```

```python
A, B,Qd,Qr,QI,P,c, r=symbols('A B Qd Qr QI P c r')
```

```python
def marginalrevenue(Qd):
    return (175-4*Qd)

def marginalcost(r):
    return (r)
```

```python
# trinn 2: optimal tilpasning for RGH
Q=sp.symbols('Qd', real=True, positive=True)
equ=sp.Eq(marginalrevenue(Qd),marginalcost(r))
equ
```

$$175 - 4Qd = r$$

Optimalt kvantum og pris for nedstrømsbedriften

```
Qd_equ=sp.solve(equ,Qd)[0]
Qd_equ
```

$$\frac{175}{4} - \frac{r}{4}$$

```
def demand(Qd):
    return (175-2*Qd)
```

```
P_eq=demand(Qd_equ)
P_eq
```

$$\frac{r}{2} + \frac{175}{2}$$

Trinn 1. Optimal valg av r

Oppstrømsbedriften selger samme kvantum som nedstrømsbedriften(Qd) og står ovenfor etterspørsel r = 175 - 4Q.

```
def demand(Qr):
    return (175-4*Qr)

def marginalrevenue(Qr):
    return (175-8*Qr)

def marginalcost(c):
    return (7)
```

```
Qr=sp.symbols('Qr', real=True, positive=True)
equ=sp.Eq(marginalrevenue(Qr),marginalcost(c))
equ
```

$$175 - 8Qr = 7$$

```python
Qr_equ=sp.solve(equ,Qr)[0]
Qr_equ
```

21

```python
# optimal pris fra oppstrømsbedriften
r_eq=demand(Qr_equ)
r_eq
```

91

```python
# Finner pris ved å sette totalt kvantum og pris fra oppstrømsbedriften inn i etterspørselen til nedstrømsbedriften
P_eq.subs({Qd:Qd_equ,r:r_eq})
```

133

```python
# Optimalt kvantum i sluttbrukermarkedet Qd:
Qd_equ.subs({Qd:Qd_equ,r:r_eq})
```

21

```python
# profitt for oppstrømsbedriften
def profitt(Qr):
    return (r_eq-marginalcost(c))*Qr_equ
print (profitt(Qr_equ))
```

1764

```python
# profitt for nedstrømsbedriften
def profitt(Qd):
    return (P_eq.subs({Qd:Qd_equ,r:r_eq})-r_eq)*Qd_equ.subs({Qd:Qd_equ,r:r_eq})
print (profitt(Qr_equ))
```

882