Report Project 1

Carl-Anton Grandelius, cagra@kth.se & Suhaib Abdi Muhummed, muhummed@kth.se

Course code: IX1500_HT2024
Date: 2024-09-10

1 Summary (Sammanfattning med Resultat)

I detta projekt har vi i Task 1 analyserat olika rörelser som en partikel kan utföra i ett xy-plan. Partikeln kan antingen röra sig uppåt (U), där både x- och y-koordinaterna ökar med 1, eller nedåt (L), där x-koordinaten ökar med 1 medan y-koordinaten minskar med 1.

I Task 2 fokuserade vi på ett röstningsscenario där två kandidater, A och B, tävlar om ett pris. Vi undersökte hur rösterna kan räknas så att A alltid leder över B. A får 9 röster och B får 2 röster. Genom att tillämpa Ballot Theorem kunde vi beräkna antalet sekvenser där A alltid leder under hela rösträkningen och sannolikheten för detta utfallet.

1.1 Task 1

1. **Deluppgift a**: Antalet möjliga vägar från punkten $(0, 3)$ till $(7, 2)$. Här har vi totalt 7 rörelser, där 3 är U och 4 är L. Antalet möjliga vägar beräknas med binomialkoefficienten $\binom{7}{3} = 35$. Följande tabell visar alla möjliga sekvenser av U- och L-rörelser:

LLLLUUU	LLLULUU	LLLUULU	LLLUULU
LLULLUU	LLULULU	LLULUUL	LLUULLU
LLUULUL	LLUUULL	LULLLUU	LULLULU
LULLUUL	LULULLU	LULULUL	LULUULL
LUULLLU	LUULLUL	LUULULL	LUUULLL
ULLLLUU	ULLLULU	ULLLUUL	ULLULLU
ULLULUL	ULLUULL	ULULLLU	ULULLUL
ULULULL	ULUULLL	UULLLLU	UULLLUL
UULLULL	UULULLL	UUULLLL	

2. **Deluppgift b**: Antalet vägar som vidrör eller korsar x-axeln vid $(3, 0)$ eller $(5, 0)$. Det finns totalt 7 sådana vägar, uppdelat på följande sätt:

   För punkten $(3, 0)$:

   För punkten $(5, 0)$:

LLLUUUL	LLLUULU
LLLULUU	LLLLUUU

ULLLLUU
LULLLUU
LLULLUU

Antalet vägar som vidrör x-axeln subtraherades från det totala antalet möjliga vägar för att räkna ut de giltiga vägarna som inte vidrör x-axeln.

3. **Deluppgift c**: Antalet vägar från (0, 3) till (7, 2) som aldrig vidrör x-axeln. För att beräkna detta subtraherades b) från det totala antalet vägar, vilket gav 28 vägar som aldrig vidrör x-axeln. Dessa vägar är:

LLULULU	LLULUUL	LLUULLU	LLUULUL
LLUUULL	LULLULU	LULLUUL	LULULLU
LULULUL	LULUULL	LUULLLU	LUULLUL
LUULULL	LUUULLL	ULLLULU	ULLLUUL
ULLULLU	ULLULUL	ULLUULL	ULULLLU
ULULLUL	ULULULL	ULUULLL	UULLLLU
UULLLUL	UULLULL	UULULLL	UUULLLL

4. **Deluppgift D** Antalet vägar från (7, 6) till (20, 5) som aldrig vidrör eller korsar x-axeln: Det totala antalet möjliga vägar beräknades till 1716 via $\binom{13}{7}$. Genom att identifiera de vägar som korsar x-axeln vid specifika punkter räknades 13 fall, vilket resulterade i 1703 vägar som aldrig vidrör eller korsar x-axeln.

   Antalet förflyttningar måste vara 13 eftersom förändringen i X-led är 13:

$$(x_0, y_0) \text{ till } (x_1, y_1)$$

$$(7, 6) \text{ till } (20, 5)$$

$$x_1 - x_0 = 13$$

   Förändringen i y-axeln är samtidigt:

$$y_1 - y_0 = -1$$

   Därför måste vi ha en mer L (som sänker partikeln i y) än U (som höjer partikeln i y):

$$7L, 6U$$

   Det totala antalet vägar från start- till slutpunkten ges av:

$$\binom{13}{7} = \frac{13!}{7!(13-7)!}$$

$$\binom{13}{7} = \frac{13!}{7!6!}$$

$$\binom{13}{7} = \frac{13 \times 12 \times 11 \times 10 \times 9 \times 8}{6 \times 5 \times 4 \times 3 \times 2}$$

$$\binom{13}{7} = \frac{13 \times 11 \times 9 \times 8}{6}$$

$$\binom{13}{7} = \frac{10296}{6} = 1716$$

X-axeln kan bara nås vid två punkter med de givna L- och U-rörelserna, förutsatt att vi i slutändan vill nå vårt slutmål. Detta syns tydligt om man markerar punkterna i ett koordinatsystem. Punkter på X-axeln som kan nås är:

$$(13, 0) \text{ (fall 1)}$$

$$(15, 0) \text{ (fall 2)}$$

I fall 1 måste vi inleda med 6 L-rörelser:

$$(7, 6) + 6 \times (1, -1) = (13, 0)$$

Därefter måste de resterande 6 U-rörelserna och 1 L-rörelse utföras.

$$\binom{7}{1} = \frac{7!}{1!(7-1)!}$$

$$\binom{7}{1} = \frac{7!}{6!} = 7$$

Dessa kombinationer är:

LLLLLLLUUUUUU
LLLLLLULUUUUU
LLLLLLUULUUUU
LLLLLLUUULUUU
LLLLLLUUUULUU
LLLLLLUUUUULU
LLLLLLUUUUUUL

I fall 2 måste vi avsluta med 5 U-rörelser:

$$(15, 0) + 5 \times (1, 1) = (20, 5)$$

Dessförinnan måste 7 L-rörelser och 1 U-rörelse utföras.

$$\binom{8}{1} = \frac{8!}{1!(8-1)!}$$

$$\binom{8}{1} = \frac{8!}{7!} = 8$$

Dessa kombinationer är

ULLLLLLLUUUUU
LULLLLLLUUUUU
LLULLLLLUUUUU
LLLULLLLUUUUU
LLLLULLLUUUUU
LLLLLULLUUUUU
LLLLLLULUUUUU
LLLLLLLUUUUUU

Båda fallen där linjen vidrör X-axeln ger:

$$8 + 7 = 15 \text{ fall, där fall } 1 \cap \text{fall } 2 = 3$$

Det slutgiltiga antalet fall där X-axeln vidrörs är därmed:

$$15 - 2 = 13$$

Samtliga fall där X-axeln vidrörs:

LLLLLLLUUUUUU	LLLLLLULUUUUU
LLLLLLUULUUUU	LLLLLLUUULUUU
LLLLLLUUUULUU	LLLLLLUUUUULU
LLLLLLUUUUUUL	ULLLLLLLUUUUU
LULLLLLLUUUUU	LLULLLLLUUUUU
LLLULLLLUUUUU	LLLLULLLUUUUU
LLLLULLLUUUUU	

1.2 Task 2

För att lösa denna uppgift använder vi en liknande metodik som för tidigare Task, där vi representerar rösterna som steg i ett rutnät:

- Varje röst på A motsvarar ett steg uppåt U (ökar både x- och y-koordinaten med 1).
- Varje röst på B motsvarar ett steg nedåt L (ökar x-koordinaten men minskar y-koordinaten).

Slutpunkten är vid $(11, 7)$, eftersom A har fått 9 röster och B har fått 2, vilket innebär en skillnad i höjd på $9 - 2 = 7$.

1.2.1 Totalt antal möjliga vägar utan restriktioner

Det totala antalet möjliga sätt att arrangera rösterna (stegen U och L) utan restriktioner är binomialkoefficienten:

$$\binom{11}{9} = \binom{11}{2} = \frac{11!}{9!2!} = \frac{11 \times 10}{2 \times 1} = 55$$

1.2.2 Antal vägar där A alltid leder över B – Användning av Ballot Theorem

Enligt Ballot Theorem är antalet sätt att arrangera rösterna så att A=9 alltid har fler röster än B=2 är:

$$\frac{9-2}{9+2} \times \binom{11}{9} = \frac{7}{11} \times 55 = 35$$

2 Formulas and Equations

2.1 Task 1

(a) **Binomialkoefficient**:

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

Binomialkoefficienten används för att beräkna hur många sätt man kan välja k objekt från n objekt. I detta fall användes den för att räkna antalet möjliga kombinationer av uppåtsteg (U) och nedåtsteg (L) i vår modell.

(b) **Förflyttningar**: Antalet förflyttningar som krävs för att nå en viss punkt bestäms genom enkel subtraktion av x- och y-koordinaterna. Detta ger en tydlig bild av hur många steg uppåt och nedåt som behövs för att nå en given position i vårt rutnät.

(c) **Korrigering för vägar som korsar x-axeln**: För att räkna ut de vägar som aldrig vidrör eller korsar x-axeln subtraherades antalet vägar där x-axeln korsades från det totala antalet möjliga vägar. Detta gav oss en slutlig beräkning av giltiga vägar som uppfyller våra krav.

Se deluppgift d) ovan för en mer detaljerad genomgång av denna metod, som tillämpades med ”papper och penna”.

2.2 Task 2

(a) **Ballot Theorem**:

$$P(\text{A leder alltid}) = \frac{a-b}{a+b} \times \binom{a+b}{a}$$

Ballot Theorem används för att beräkna sannolikheten att kandidat A alltid leder över kandidat B under hela rösträkningen. Formeln ger oss antalet giltiga sekvenser där A alltid har fler röster än B vid varje steg i processen.

3 Discussion

I denna inlämning har vi fått lära oss hur man kan använda binomialkoefficienter, kombinatorik och sätta ut punkter i koordinatsystem för att beräkna antal vägar och rörelser i ett xy-plan. Vi tog reda på antalet rörelsevägar som en partikel kan ta mellan olika punkter, samt hur vi kan sätta ett villkor som att x-axeln måste, eller inte får vidröras.

Vid en utvärdering av våra lösningar kan vi säga att vi fick rätt svar genom att titta på kombinationer, försöka förstå problemet genom att själva titta på koordinatsystem och till slut använda binomialkoefficienten. En alternativ metod hade kunnat vara att använda någonting som kallas Catalans tal, för att räkna om resultatet t.ex. när x-axeln ska undvikas.

I **Task 2** tillämpade vi ytterligare kombinatoriska metoder och sannolikhetsteori för att lösa problemet med att arrangera röster där en kandidat, A, alltid leder över en annan kandidat, B. Genom att använda Ballot Theorem kunde vi beräkna sannolikheten att A leder över B under hela rösträkningen. Det visade sig att sannolikheten var 63,6%, vilket beror på förhållandet mellan antalet röster för A och B. Även i detta fall kunde Catalans tal ha använts som ett alternativ för att lösa problem där sekvenser undviker att vidröra eller korsa x-axeln.

4 Code

Här nedan följer den fullständiga koden (1,5 sidor). Den finns också tillgänglig på Github: https://github.com/suhaib921/IX150$_HT24_50140-$

4.1 Task 1

```
from itertools import permutations

# a) Antal vägar från (0, 3) till (7, 2)

vagar = set(permutations(['U'] * 3 + ['L'] * 4)) # 3 U och 4 L rörelser
print("a) Antalet vägar från (0, 3) till (7, 2): ", len(vagar))
for vag in vagar:
    print(''.join(vag))

# b) Antal vägar från (0, 3) till (7, 2) som vidrör x-axeln
def vagar_som_vidror_x_axeln():
    vagar = set(permutations(['U'] * 3 + ['L'] * 4)) # 3 U och 4 L rörelser
    vagar_som_vidror = set()

    for vag in vagar:
        x, y = 0, 3
        for steg in vag:
            if steg == 'L':
                y -= 1
            else:
                y += 1
            if y == 0:  # Ej framme, men vi vidrör x-axeln
                vagar_som_vidror.add(vag)
                break
    return vagar_som_vidror

vagar = set(permutations(['U'] * 3 + ['L'] * 4)) # 3 U och 4 L rörelser
print("b) Antal vägar som vidrör x-axeln: ", len(vagar))
for vag in vagar:
    print(''.join(vag))


# c) Antal vägar från (0, 3) till (7, 2) som EJ vidrör x-axeln
vagar = set(permutations(['U'] * 3 + ['L'] * 4))
vagar = vagar - vagar_som_vidror_x_axeln()
print("Antal vägar från (0, 3) till (7, 2) som EJ vidrör x-axeln: ", len(vagar))
for vag in vagar:
    print(''.join(vag))

# d) Antal vägar från (7, 6) till (20, 5) som EJ vidrör x-axeln
def vagar_ej_vidror_x_axeln_7_6_till_20_5():
    vagar = set(permutations(['U'] * 6 + ['L'] * 7))

    icke_vagar_som_vidror = set()
```

```
    vagar_som_vidror = set()
    for vag in vagar:
        x, y = 7, 6  # Startpunkt (7, 6)
        korsat_x_axeln = False
        for steg in vag:
            if steg == 'L':
                y -= 1
            else:
                y += 1
            if y == 0:  # Om vägen når x-axeln
                korsat_x_axeln = True
                break
        if not korsat_x_axeln:
            icke_vagar_som_vidror.add(vag)
        else:
            vagar_som_vidror.add(vag)

    print("Antal vägar från (7, 6) till (20, 5) som EJ vidrör x-axeln: ",
    len(icke_vagar_som_vidror),
    ". De som vidrör x-axeln: ", len(vagar_som_vidror))
    i = 0
    for vag in icke_vagar_som_vidror:
        if (i < 20):
            print(''.join(vag))
            i = i + 1

vagar_ej_vidror_x_axeln_7_6_till_20_5()
```

4.2 Task 2

```
from math import comb

# Funktion för att beräkna giltiga sekvenser där A alltid leder över B
def ballot_problem(a, b):
    # Formeln för ballot-problemet:
    giltiga_sekvenser = (a - b) * comb(a + b, b) // (a + b)
    return giltiga_sekvenser

# Totalt antal sätt att arrangera rösterna utan restriktioner
def totalt_antal_satt(a, b):
    return comb(a + b, b)
```

```python
# Antal giltiga sekvenser och totala sekvenser
a_roster = 9
b_roster = 2

giltiga_sekvenser = ballot_problem(a_roster, b_roster)
totala_sekvenser = totalt_antal_satt(a_roster, b_roster)

# Sannolikheten att A alltid leder över B
sannolikhet = giltiga_sekvenser / totala_sekvenser

print(f"Antal giltiga sekvenser där A alltid leder: {giltiga_sekvenser}")
print(f"Totalt antal sekvenser utan restriktioner: {totala_sekvenser}")
print(f"Sannolikhet att A alltid leder: {sannolikhet:.4f}")
```