

OEFENINGEN

BIJ HET VAK

“FUNDAMENTEN
VOOR DE
INFORMATICA”
(PROF. BART DEMOEN)

1^{ste} BACHELOR INFORMATICA
ACADEMIEJAAR 2005-2006

OPGAVEN

Jon Sneyers
jon@cs.kuleuven.be
Dept. Computerwetenschappen
K.U.Leuven

Inhoudsopgave

1	Inleiding op complexiteit	4
1.1	Talen en eindige automaten	4
1.1.2	Reguliere expressies en talen	4
1.1.3	Eindige automaten en reguliere talen	5
1.1.4	Niet deterministische eindige automaten	6
1.2	Turing machines	7
1.2.1	Wat is een Turing machine	7
1.2.2	Turing machines en functies	8
1.3	Analyse van algoritmen	8
1.3.1	Tijdscomplexiteit van algoritmen	8
1.3.2	Het bepalen van de complexiteit in enkele concrete voorbeelden	9
1.4	De klassen P en NP	10
1.4.1	Turing machines, talen en de klasse P	10
1.4.2	Niet deterministische Turing machines en de klasse NP	11
1.4.3	De klasse NP -Compleet	12
2	Grafentheorie	13
2.1	Inleiding	13
2.2	Grafen	13
2.2.1	Allerhande paden	13
2.2.2	Voorstelling van grafen	14
2.2.3	Isomorfisme van grafen	15
2.2.4	Gewogen grafen	17
2.2.5	Vlakke grafen	17
2.2.6	Het kleuren van grafen	18
2.3	Bomen	19
2.3.1	Inleiding	19
2.3.2	Eigenschappen van bomen	19
2.3.3	Een meer compacte voorstelling voor bomen	20
2.3.4	Opspannende bomen	20
2.3.5	Minimale opspannende bomen	20
2.3.6	Doorlopen van bomen	21
2.3.7	Spelbomen	21
2.4	Netwerkmodellen en Petri-netten	21
2.4.2	Maximale stroming	21

2.4.3	Matching	22
2.4.4	Petrinetten	23
2.5	Overzichtsoefeningen	24
3	Vastepuntstheorie	27
3.2	Orderrelaties	27
3.2.1	Basisbegrippen	27
3.2.2	Monotone en continue afbeeldingen	28
3.3	De stellingen van Tarski en Kleene	28
4	Overzichtsoefeningen	30

Hoofdstuk 1

Inleiding op complexiteit

1.1 Talen en eindige automaten

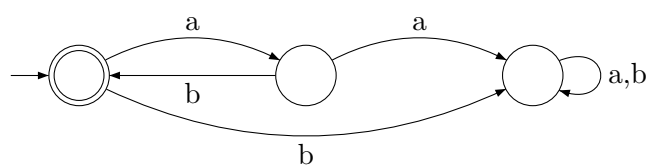
1.1.2 Reguliere expressies en talen

1. Zijn de volgende verzamelingen talen? Zo ja, op welk alfabet Σ ? Is er dan een algoritme om na te gaan of een string uit Σ^* tot de taal behoort of niet? (getallen worden voorgesteld in decimale notatie)
 - (a) de natuurlijke getallen tussen 100 en 200 (inclusief)
 - (b) de natuurlijke getallen groter dan of gelijk aan 100
 - (c) priemgetallen kleiner dan 20
 - (d) alle priemgetallen
 - (e) de lege verzameling
 - (f) alle eindige rijen van natuurlijke getallen
 - (g) alle oneindige rijen van cijfers
 - (h) de rationale getallen, in komma-notatie
 - (i) de rationale getallen, in breuk-notatie
 - (j) de reële getallen tussen 0 en 1
 - (k) alle woorden in het Nederlands
 - (l) alle zinnen in het Nederlands
 - (m) (syntactisch correcte) Java-programma's
 - (n) (syntactisch correcte) eindigende Java-programma's
 - (o) een willekeurige deelverzameling van $\{a, b, c\}^*$
 - (p) een willekeurige eindige deelverzameling van $\{a, b, c\}^*$
2. Zijn de talen uit de vorige oefening reguliere talen? Zo ja, geef een reguliere expressie die de taal bepaalt (enkel voor de zes eerste reguliere talen). Gebruik C als afkorting voor $(0|1|2|3|4|5|6|7|8|9)$, de reguliere expressie die één cijfer voorstelt.
3. Zoek een reguliere expressie voor de volgende talen op $\Sigma = \{a, b, c\}$:

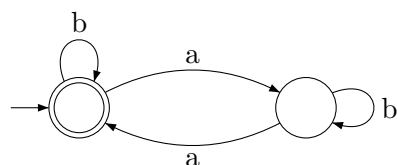
- (a) alle strings waarin bac (minstens) twee keer voorkomt
 - (b) alle strings die op het einde drie keer hetzelfde symbool hebben
 - (c) alle strings waarin a niet twee keer na elkaar voorkomt
 - (d) alle strings waarin a juist 1 keer voorkomt en b minstens 2 keer
4. Een uitgebreide reguliere expressie heeft twee extra operaties: A^+ (één of meer keer A) en $A^?$ (nul of één keer A). Kunnen er meer talen voorgesteld worden met uitgebreide reguliere expressies dan met gewone reguliere expressies? Zo ja, geef een uitgebreide reguliere expressie die een niet-reguliere taal bepaalt. Zo nee, waarom niet?
5. Waar of niet waar? Argumenteer.
- (a) Eindige talen zijn regulier.
 - (b) Reguliere talen zijn eindig.
 - (c) De unie van twee reguliere talen is een reguliere taal.
 - (d) Elke deelverzameling van een taal is een taal.
 - (e) Elke deelverzameling V van een reguliere taal R ($V \subseteq R$) is een reguliere taal.
 - (f) Elke verzameling V die een reguliere taal R omvat ($R \subseteq V$) is een reguliere taal.
 - (g) De verzameling van reguliere expressies over een gegeven alfabet Σ is zelf een reguliere taal.

1.1.3 Eindige automaten en reguliere talen

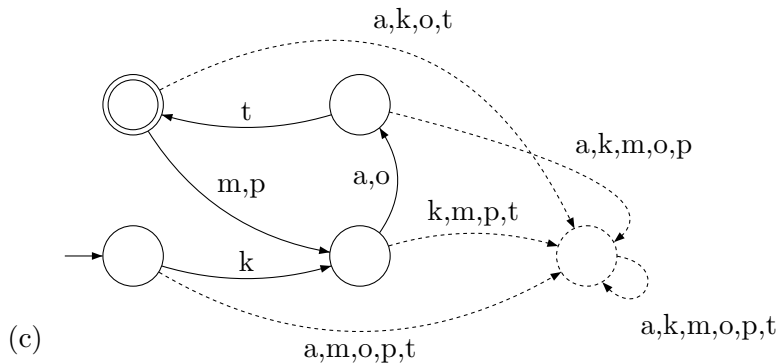
1. Welke taal wordt herkend door deze automaten? Geef een reguliere expressie die deze taal bepaalt.



(a)



(b)



2. Construeer een eindige automaat die de taal herkent die bepaald wordt door de volgende reguliere expressies:

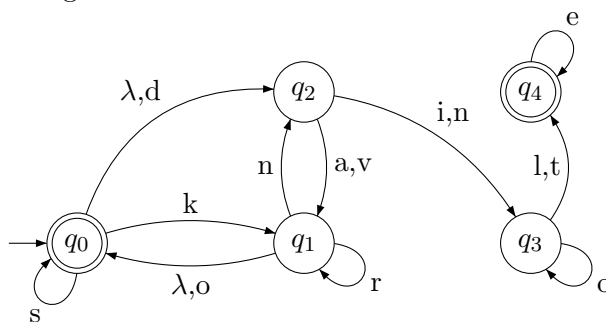
- (a) $(bla)^*$
- (b) $(d|w)(at|ie|aar)$
- (c) $(b|\lambda)a(na)^*s$

3. Bewijs de volgende stellingen, waarbij L_1 en L_2 reguliere talen zijn over een alfabet Σ .

- (a) De concatenatie L_1L_2 is een reguliere taal.
- (b) $L_1 \cup L_2$ is een reguliere taal.
- (c) Het complement van L_1 , dit is $L_1^C = \Sigma^* \setminus L_1$, is een reguliere taal.
- (d) $L_1 \cap L_2$ is een reguliere taal.

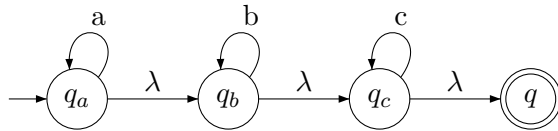
1.1.4 Niet deterministische eindige automaten

1. Welke van de volgende woorden worden aanvaard door deze niet deterministische eindige automaat?



dit, dat, dar, nar, koor, noot, viool, riool, dans, dan, krokodil, ski, skoda, kokosnoot, ananas, ole, knit, knot, knol, dante, sandaal, sars, silo, ark, dank, danone, kar, knars, knal, kras, krant

2. Maak van de niet deterministische eindige automaat van het voorbeeld bovenaan bladzijde 18 in de cursus een deterministische eindige automaat.



3. Waar of niet waar? Argumenteer.

- Voor elke niet deterministische eindige automaat is er ook een deterministische eindige automaat die dezelfde taal herkent.
- Sommige niet deterministische eindige automaten herkennen niet-reguliere talen.
- Als er voor een bepaalde string $x \in \Sigma^*$ een pad is van de begintoestand naar een aanvaardbare eindtoestand zodat de concatenatie van de labels van de gevolgde pijlen de string x oplevert, dan wordt x aanvaard.
- Als er voor een bepaalde string $x \in \Sigma^*$ een pad is van de begintoestand naar een niet-aanvaardbare eindtoestand zodat de concatenatie van de labels van de gevolgde pijlen de string x oplevert, dan wordt x niet aanvaard.
- Er bestaan niet deterministische eindige automaten zonder λ -pijlen en meerdere pijlen vanuit dezelfde toestand met hetzelfde label, die toch geen deterministische eindige automaten zijn.

4. Construeer een deterministische eindige automaat die dezelfde taal herkent als de **(Extra)** niet-deterministische eindige automaat uit oefening 1.

1.2 Turing machines

1.2.1 Wat is een Turing machine

- Construeer een Turing machine met $\Sigma = \{\#, 0, 1\}$ die
 - overtollige nullen aan de linkerkant wegdoet (dus met invoer 00010110111 moet er uiteindelijk 10110111 op de band staan);
 - bij geen enkele invoer eindigt (op twee verschillende manieren);
 - strings met een even aantal nullen en oneven aantal enen aanvaardt en andere strings verwerpt;
 - bij invoer 10001 of 1000001 in een oneindige lus gaat, maar bij invoer 1001 of **(Extra)** 100001 eindigt.
- Geef een voorbeeld van een Turing machine waarvan de verzameling van invoerstrings die aanvaard worden *geen* reguliere taal is.
- Een *read-only* Turing machine is een Turing machine die de symbolen op de magneetband niet overschrijft, m.a.w. $P(\dots, \sigma) = (\dots, \sigma, \dots)$, waarbij P het programma van de Turing machine is. Bewijs of geef een tegenvoorbeeld:

- (a) Elke eindige automaat kan als een read-only Turing machine geschreven worden. Met andere woorden: Voor elke (deterministische) eindige automaat A op een alfabet Σ bestaat er een read-only Turing machine M met invoersymbolen $T = \Sigma$, die bij invoer i stopt in een aanvaardbare eindtoestand als i aanvaard wordt door A , en stopt in een niet aanvaardbare eindtoestand als i verworpen wordt door A .
- (b) Elke steeds eindigende read-only Turing machine kan als een eindige automaat **(Extra)** geschreven worden. Met andere woorden: Voor elke read-only Turing machine M met invoersymbolen T die eindigt voor elke invoer $i \in T^*$ bestaat er een deterministische eindige automaat A op alfabet $\Sigma = T$, die de string i aanvaardt als M bij invoer i stopt in een aanvaardbare eindtoestand, en die i verworpt als M bij invoer i stopt in een niet aanvaardbare eindtoestand.
4. Geef commentaar op de volgende stelling: “Eindige automaten en read-only Turing machines zijn essentieel hetzelfde”.

1.2.2 Turing machines en functies

1. Construeer in detail een TM M die de functie $f : (\mathbb{N} \times \mathbb{N}) \rightarrow \mathbb{N} : (m, n) \mapsto m \oplus n$ berekent, met $m \oplus n$ gelijk aan $m - n$ als $m \geq n$ en 0 als $m < n$. De invoer is gegeven in unaire notatie en de twee getallen worden gescheiden door het symbool B . Uitvoer in unaire notatie.
2. Waar of niet waar? Argumenteer.
 - (a) Er zijn oneindig veel Turing machines.
 - (b) Het aantal Turing machines met n toestanden en m symbolen is eindig.
3. Construeer (in detail) een TM M die de functie $f : \mathbb{N} \rightarrow \{0, 1\} : n \mapsto n \bmod 2$ (rest **(Extra)** bij deling door 2) berekent. (invoer in unaire notatie, uitvoer is 0 of 1)
4. Construeer (in detail) een TM M die de functie $f : \mathbb{N} \rightarrow \mathbb{N} : n \mapsto n \operatorname{div} 2$ (Euclidisch **(Extra)** quotiënt bij deling van n door 2) berekent. Alles in unaire notatie.
5. Gebruik oefening 3 en 4 om een TM te beschrijven (enkel in woorden, niet in detail) **(Extra)** die de functie CODEER, voor het omzetten van de unaire naar de binaire notatie, berekent.
6. Beschrijf een TM die de functie DECODEER (van binair naar unair) berekent. **(Extra)**

1.3 Analyse van algoritmen

1.3.1 Tijdscomplexiteit van algoritmen

1. Vergelijk volgende functies f en g op gebied van asymptotisch gedrag. De vier mogelijkheden zijn: 1) f is $O(g)$ (maar niet g is $O(f)$), 2) g is $O(f)$ (maar niet f is $O(g)$), 3) f is $\theta(g)$, 4) f is niet $O(g)$ en g is niet $O(f)$. Bewijs.
 - (a) $f(x) = x^2$ en $g(x) = x^2 + 1$

- (b) $f(x) = a$ en $g(x) = b$ en $a > b > 0$
- (c) $f(x) = x^a$ en $g(x) = x^b$ en $a > b > 0$
- (d) $f(x) = |\sin(x\frac{\pi}{2})|$ en $g(x) = |\cos(x\frac{\pi}{2})|$
- (e) $f(x) = \sin(x\frac{\pi}{2}) + 2$ en $g(x) = \cos(x\frac{\pi}{2}) + 2$
- (f) $f(x) = \log_2(x)$ en $g(x) = x$
- (g) $f(x) = x^2$ en $g(x) = x^2 - x$
- (h) $f(x) = x!$ en $g(x) = 2^x$
- (i) $f(x) = \frac{x!}{(x-5)!}$ en $g(x) = x^5$

2. Beschouw twee functies $f, g : \mathbb{N} \rightarrow \mathbb{R}^+$. Toon aan dat

$$\lim_{n \rightarrow +\infty} \frac{f(n)}{g(n)} = \begin{cases} 0 & \Rightarrow f \text{ is } O(g) \text{ maar niet } g \text{ is } O(f) \\ A \in \mathbb{R}_0^+ & \Rightarrow f \text{ is } O(g) \text{ en } g \text{ is } O(f) \\ +\infty & \Rightarrow g \text{ is } O(f) \text{ en } f \text{ is niet } O(g) \end{cases}$$

3. f, g en h zijn functies van \mathbb{N} naar \mathbb{R}^+ . Waar of niet waar? Motiveer: bewijs als waar, geef tegenvoorbeeld als niet waar.

- (a) f is $O(g) \Rightarrow \forall n \in \mathbb{N}: f(n) \leq g(n)$.
- (b) f is $O(n^2)$ en g is ook $O(n^2) \Rightarrow fg$ is $O(n^4)$.
- (c) f is $O(g) \Rightarrow f$ is $O(g+h)$.
- (d) $g : \mathbb{N} \rightarrow \mathbb{R}^+ : n \mapsto f(n+1)$ is asymptotisch equivalent met f .
- (e) Als f en g asymptotisch equivalent zijn en h is $O(g)$, dan zijn f en $g+h$ asymptotisch equivalent.
- (f) f is niet $O(g) \Rightarrow g$ is $O(f)$.
- (g) Als f asymptotisch equivalent is met n^2 , dan is f een veeltermfunctie.

4. Bewijs dat de relatie “is asymptotisch equivalent met” wel degelijk een equivalentie- (Extra) relatie is, m.a.w. bewijs dat ze reflexief, symmetrisch en transitief is.

5. Toon aan dat in de rij functies (Extra)

$$\log_2 n \quad n \quad n \log_2 n \quad n^2 \quad \dots \quad n^k \quad \dots \quad 2^n \quad n!$$

voor elke functie f die links van een functie g staat geldt dat f is $O(g)$ en g is niet $O(f)$.

1.3.2 Het bepalen van de complexiteit in enkele concrete voorbeelden

1. Bereken de complexiteit van de volgende algoritmes:

- (a)

```
for (int i=0; i<n; i++)
  for (int j=0; j<n; j++)
    for (int k=0; k<n; k++)
      [een elementaire opdracht]
```

```
(b) for (int i=0; i<n; i++)
      for (int j=0; j<i*i; j++)
        for (int k=0; k<i; k++)
          [een elementaire opdracht]
```

```
(c) for (int i=0; i<n; i++)
      for (int j=0; j<i*i; j++)
        for (int k=0; k<j; k++)
          [een elementaire opdracht]
```

(d) De volgende recursieve functie:

(Extra)

```
int fibonacci(int n) {
  if (n < 3) return 1
  else return (fibonacci(n-1) + fibonacci(n-2));
}
```

2. Beschrijf Turing machines die de volgende functies berekenen. Wat is hun complexiteit? Kan dat nog beter?

- (a) $f : \mathbb{N} \rightarrow \mathbb{N} : x \mapsto x + 1$, in unaire notatie
- (b) $f : \mathbb{N} \rightarrow \mathbb{N} : x \mapsto x + 1$, in binaire notatie
- (c) $f : \mathbb{N} \rightarrow \mathbb{N} : x \mapsto 2x$, in unaire notatie
- (d) $f : \mathbb{N} \rightarrow \mathbb{N} : x \mapsto 2x$, in binaire notatie

3. Wat is de complexiteit van...

- (a) een woord opzoeken in een woordenboek (probleemgrootte: aantal bladzijden van het woordenboek)
- (b) gras maaien (probleemgrootte: oppervlakte gazon)
- (c) elkaar gelukkig nieuwjaar wensen (probleemgrootte: aantal mensen)
- (d) een ballon opblazen (probleemgrootte: diameter van de ballon)
- (e) in de witte gids de naam zoeken die bij een telefoonnummer hoort (probleemgrootte: aantal bladzijden van de witte gids)

1.4 De klassen P en NP

1.4.1 Turing machines, talen en de klasse P

1. Bewijs.

- (a) Elke recursieve taal is recursief opsombaar.
- (b) Elke reguliere taal is recursief.

2. Bewijs: Als L een taal is over alfabet Σ :

L is een recursieve taal

\Downarrow

de functie $f : \Sigma^* \rightarrow \{0, 1\} : x \mapsto \begin{cases} 1 & \text{als } x \in L \\ 0 & \text{als } x \notin L \end{cases}$ is Turing berekenbaar.

3. Zij M een TM die de functie $f : \mathbb{N} \rightarrow \mathbb{N}$ berekent. Laat zien dat:
 - (a) we mogen veronderstellen dat M slechts één aanvaardbare eindtoestand heeft. M.a.w. verander M in een TM M' die f berekent, die slechts één aanvaardbare eindtoestand heeft, en die dezelfde asymptotische tijdscomplexiteit heeft als M .
 - (b) we mogen veronderstellen dat indien M stopt, de leeskop boven het meest linkse symbool van de uitvoer staat. M.a.w. verander M in een TM M' die f berekent, die bij beëindiging de leeskop boven het meest linkse symbool van de uitvoer heeft staan, en die dezelfde asymptotische tijdscomplexiteit heeft als M .
4. Zij M een TM die de functie $f : \mathbb{N} \rightarrow \mathbb{N}$ berekent, en N een TM die de functie $g : \mathbb{N} \rightarrow \mathbb{N}$ berekent. Veronderstel dat ze alletwee werken met unaire invoer en uitvoer. Construeer een TM die de functie $g \circ f$ berekent.
5. Illustreer de volgende stellingen door de Turing machines die de recursieve talen beslissen aan te passen of aan elkaar te koppelen (op hoog niveau, niet in detail).
 - (a) Het complement van een recursieve taal is recursief.
 - (b) De unie van twee recursieve talen is recursief.
 - (c) De doorsnede van twee recursieve talen is recursief.
 - (d) Het verschil ($L_1 \setminus L_2$) tussen twee recursieve talen is recursief.
6. Waar of niet waar? Bewijs of geef een tegenvoorbeeld. **(Extra)**
 - (a) Alle recursieve talen zitten in **P**.
 - (b) Elke recursief opsombare taal is recursief.

1.4.2 Niet deterministische Turing machines en de klasse NP

1. Construeer een niet-deterministische TM die gegeven een bedrag en een aantal munten bepaalt welke munten nodig zijn om dat bedrag samen te stellen. De invoer is als volgt:

... # # # 1 1 1 1 = 1 + 1 1 + 1 1 1 1 1 + 1 # # # ...

Dus het bedrag dat moet gevormd worden is 4 cent, met 4 munten : 2 van 1 cent, eentje van 2 cent en eentje van 5 cent. De uitvoer ziet er dan als volgt uit:

... # # # 1 1 1 1 = 1 + 1 1 + X X X X X + 1 # # # ...

2. Stel dat je een DTM M_d hebt die invoer van de vorm

$$\underbrace{1\ 1\ 1\ \dots\ 1\ 1\ 1}_{n+1\ \text{keer}\ 1}\ D\ \underbrace{1\ 1\ 1\ \dots\ 1\ 1\ 1}_{m+1\ \text{keer}\ 1}$$

in polynomiale tijd aanvaardt als $1 < n < m$ en n een deler is van m , en alle andere invoer verwerpt.

Construeer nu een NDTM die de taal $L = \{n \in \mathbb{N} \mid n \text{ is niet priem}\}$ in polynomiale tijd aanvaardt (invoer in unaire notatie).

3. Waar of niet waar?
- Een NDTM die ($\forall n \in \mathbb{N}$) bij elke invoer van lengte n voor elk uitvoeringspad in 2^n stappen eindigt in een aanvaardbare eindtoestand is van polynomiale tijd.
 - Een NDTM die ($\forall n \in \mathbb{N}$) bij elke invoer van lengte n voor elk uitvoeringspad in 2^n stappen eindigt in een aanvaardbare eindtoestand beslist een taal uit \mathbf{P} .
 - Een NDTM die ($\forall n \in \mathbb{N}$) bij elke invoer van lengte n voor elk uitvoeringspad in 2^n stappen eindigt in een niet aanvaardbare eindtoestand is van polynomiale tijd.
 - Een NDTM die ($\forall n \in \mathbb{N}$) bij elke invoer van lengte n één uitvoeringspad heeft dat na 10^{1000} stappen eindigt in een aanvaardbare eindtoestand, en oneindig veel andere uitvoeringspaden heeft die na $n!$ stappen eindigen in een aanvaardbare eindtoestand, is van polynomiale tijd.
 - Er zijn talen die in polynomiale tijd door een NDTM beslist worden maar waarvoor geen DTM bestaat die de taal in polynomiale tijd beslist.
 - NDTM's zijn expressiever dan DTM's in de zin dat er talen zijn die wel door een NDTM beslist worden maar waarvoor geen DTM bestaat die die taal beslist.

1.4.3 De klasse NP -Compleet

1. Beschouw de volgende talen:

$$L_1 = \{(n, p) \mid n \in \mathbb{N}, p = 1 \text{ als } n \text{ priem is en } p = 0 \text{ als } n \text{ niet priem is}\}$$

$$L_2 = \{(n, p) \mid n \in \mathbb{N}, p = 0 \text{ als } n \text{ priem is en } p = 1 \text{ als } n \text{ niet priem is}\}$$

Geef een polynomiale transformatie om aan te tonen dat $L_1 \propto L_2$.

2. Een recursieve taal L wordt beslist in polynomiale *ruimte* door een Turing Machine (**Extra**) M , indien M de taal L beslist en de *ruimtecomplexiteit* $O(n^k)$ is voor een $k \in \mathbb{N}$. De (slechtste-geval) ruimtecomplexiteit van een Turing machine is voor een gegeven invoeromvang n het maximum aantal niet-blanco posities dat tijdens de uitvoering op de magneetband staat. **PSPACE** is de klasse van talen die in polynomiale *ruimte* door een TM beslist worden. Stel dat $\mathbf{P} \neq \mathbf{NP}$, bewijs dan dat $\mathbf{P} \subsetneq \mathbf{PSPACE}$.

Hoofdstuk 2

Grafentheorie

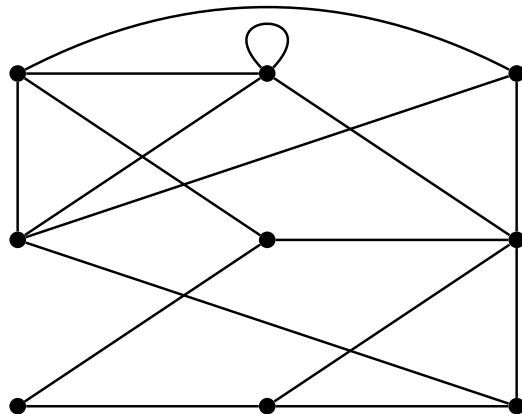
2.1 Inleiding

1. Het “missionarissen en kannibalen”-probleem lijkt op “de wolf, de geit en de kool”. Aan de oever van een rivier zijn drie missionarissen, drie kannibalen en een bootje. Niemand kan de rivier oversteken zonder de boot, en er kunnen maximaal twee mensen in de boot. Iedereen kan roeien, maar er moet natuurlijk wel iemand in de boot zitten om de boot naar de overkant te roeien. Als er op één van de oevers (strikt) meer kannibalen zijn dan missionarissen, dan worden de missionarissen opgegeten. Kunnen ze alle zes heelhuids de overkant bereiken?

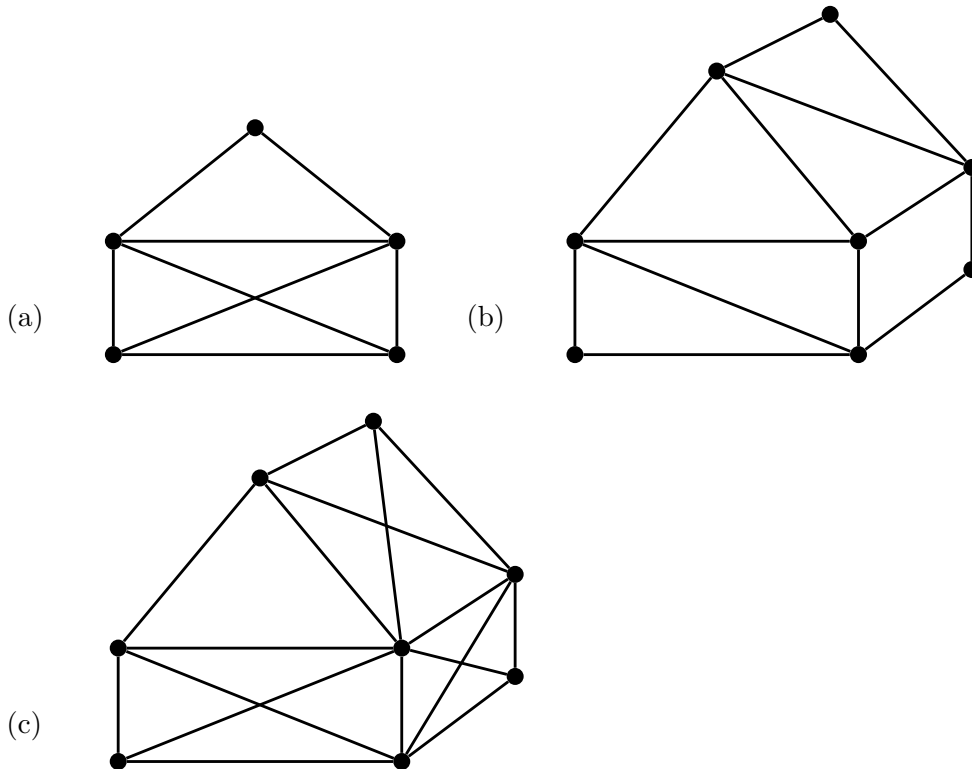
2.2 Grafen

2.2.1 Allerhande paden

1. Geef de graad van elke knoop in de graaf $(\{a, b, c, d, e\}, \{(a, b), (b, e), (d, d), (a, d), (e, a)\})$. Hoeveel kringen zijn er in deze graaf?
2. Op een departement zijn er 25 personen. Elke persoon heeft ruzie met exact 5 andere personen. Kan dat?
3. Zoek indien mogelijk een Hamiltoniaanse kring voor de volgende graaf:



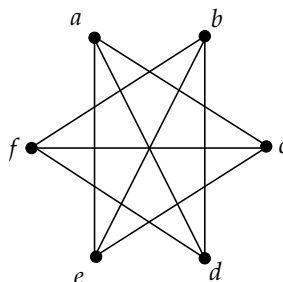
4. Zoek indien mogelijk een Euleriaanse kring/pad voor de volgende grafen. Gebruik de methode uit het bewijs van stelling 2.1.15.



5. Veralgemeen stelling 2.1.15 en 2.1.16 naar gerichte grafen. Bewijs.

2.2.2 Voorstelling van grafen

1. Hoeveel bits zijn overbodig in buurmatrices van enkelvoudige niet-gerichte grafen?
2. Geef de buurmatrix A en incidentiematrix I voor de volgende graaf:

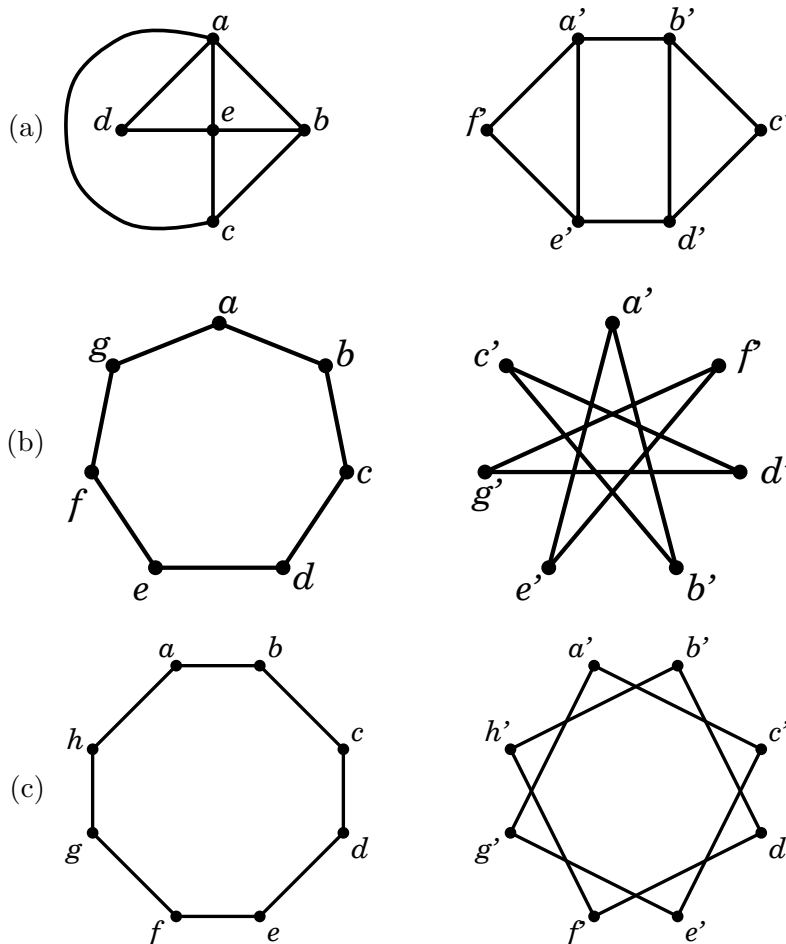


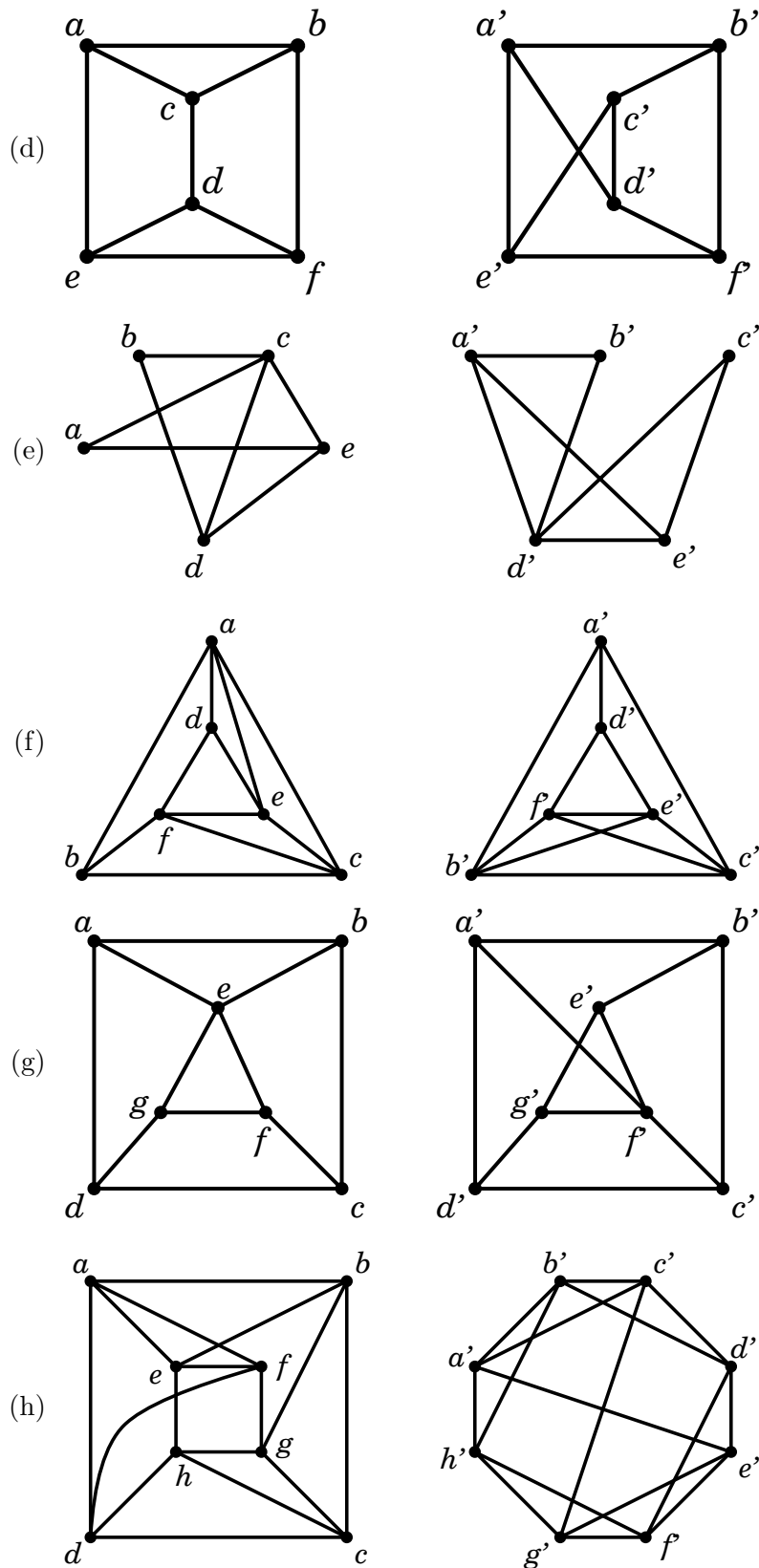
3. Veralgemeen de definitie van *buurmatrix* zodat stelling 2.2.1 ook geldig is voor niet-enkelvoudige, gerichte grafen.

4. Bewijs: als I de incidentiematrix is van een niet-gerichte, enkelvoudige graaf, dan is $II^T = A'$ gelijk aan de buurmatrix A voor die graaf, behalve op de diagonaalelementen $A'[i, i]$, die gelijk zijn aan het aantal bogen in de knoop i .
5. Hoe ziet de buurmatrix van K_n eruit? Hoeveel 1'tjes staan er in de buurmatrix van K_n ? Wat is de relatie met het aantal bogen in K_n ?

2.2.3 Isomorfisme van grafen

1. Geef een voorbeeld van twee grafen die evenveel bogen en evenveel knopen hebben en die niet isomorf zijn.
2. Zijn de volgende paren van grafen isomorf? Indien ja, geef het isomorfisme, indien nee, geef een onder isomorfisme invariante eigenschap waaraan de ene graaf voldoet en de andere niet.

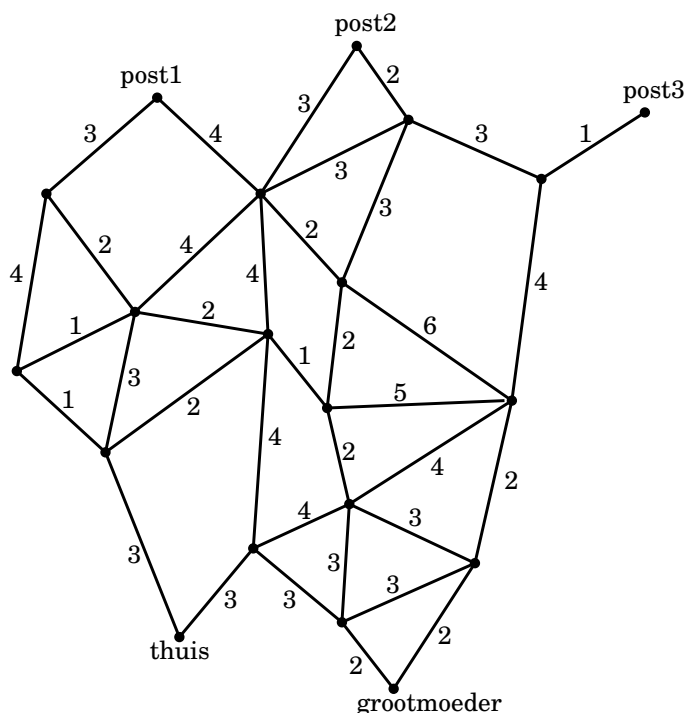




2.2.4 Gewogen grafen

1. Hoe kan je het algoritme van Dijkstra aanpassen zodat het het kortste pad berekent vanuit 1 knoop naar alle andere knopen?
2. Gegeven het volgende stratenplan, wat is de kortste weg om bij mij thuis te vertrekken, langs 1 van de postkantoren te gaan, en dan bij mijn grootmoeder aan te komen?

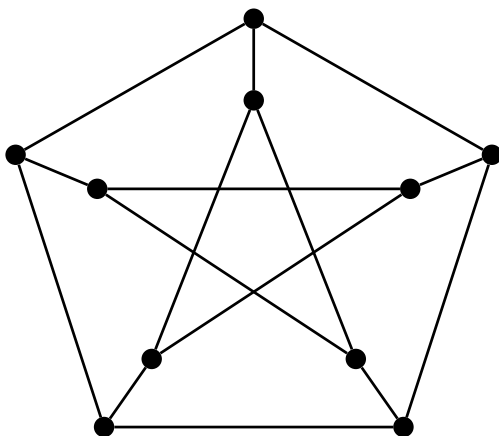
Hint: pas de graaf aan zodat je het kortste-pad algoritme van Dijkstra kan gebruiken om de kortste weg van thuis naar grootmoeder te vinden (waarom kan dat niet rechtstreeks?).



2.2.5 Vlakke grafen

1. Toon aan dat als G een vlakke graaf is met e bogen, v knopen, f zijvlakken en c componenten, dat dan $v - e + f - c = 1$. Voor $c = 1$ reduceert dit zich tot Eulers formule voor vlakke grafen.
2. Teken een vlakke, enkelvoudige graaf met 3 knopen met graad 4, 4 knopen met graad 5 en 1 knoop met graad 6. Indien dit niet mogelijk is, argumenteer waarom.
3. Wat is het maximaal aantal bogen in een enkelvoudige, vlakke graaf met n knopen?
4. Toon aan dat "... is homeomorf met ..." een equivalentierelatie is. (Extra)
5. Bewijs dat Petersen's graaf (zie hieronder) niet vlak is.

Hint: gebruik de stelling van Kuratowski.

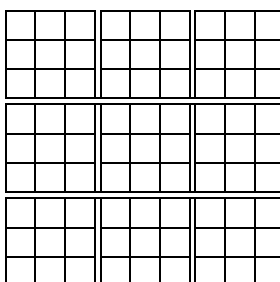


2.2.6 Het kleuren van grafen

1. Het examen FVI vindt plaats in auditorium 200L 00.07, en daar zijn n rijen van banken (meer dan 2), en op elke rij zijn er m plaatsen (meer dan 2). Er zijn nm studenten, dus elke zitplaats is bezet. Omdat iedereen zo dicht bij elkaar zit, zijn er verschillende vragenreeksen opgesteld, om afkijken te ontmoedigen.

Als je veronderstelt dat je kan afkijken bij je linker- en rechterbuur, en ook bij degene die recht voor jou zit, hoeveel vragenreeksen zijn er dan nodig om ervoor te zorgen dat niemand kan afkijken? En als je niet kan afkijken bij degene die recht voor jou zit, maar wel bij degenen die schuin voor jou zitten? Wat als je bij alle vijf kan afkijken (links, rechts, links-voor, voor, rechts-voor) ?

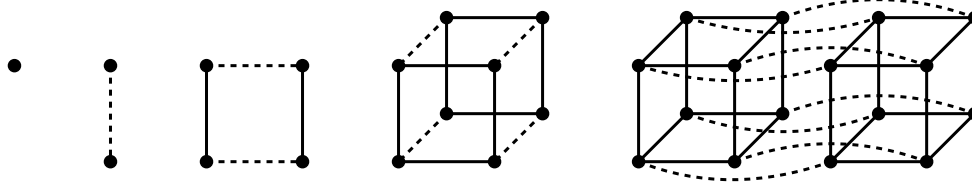
2. Een *sudoku*-puzzel bestaat uit een vierkant bord van 9 vakjes op 9 vakjes, dat gevormd wordt door 9 kleinere vierkantjes van 3x3 vakjes:



De bedoeling is om in alle vakjes een cijfer tussen 1 en 9 in te vullen zodat elk klein vierkantje, elke rij, en elke kolom alle cijfers tussen 1 en 9 bevat.

Formuleer sudoku als een probleem van het kleuren van een graaf. (Hoeveel bogen heeft de graaf die je construeerde?)

3. Een n -kubus bestaat uit twee $(n-1)$ -kubussen met de overeenkomstige "hoekpunten" van de twee $(n-1)$ -kubussen met elkaar verbonden. Een 1-kubus bestaat uit een enkel punt. Een aantal n -kubussen worden getoond in de figuur.

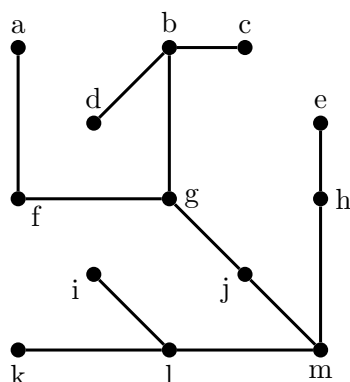


- (a) Voor welke n heeft een n -kubus een Hamiltoniaanse kring?
 - (b) Wat is het minimaal aantal kleuren waarmee een n -kubus kan gekleurd worden?
 - (c) Voor welke n is een n -kubus vlak?
4. Zijn de duale grafen van twee isomorfe grafen isomorf?

2.3 Bomen

2.3.1 Inleiding

1. Kies in onderstaande boom een wortel zodat de hoogte van de boom minimaal is.



Teken een boom waarvoor de “beste wortel” niet uniek is.

2. Twee bomen B_1 en B_2 met hoogtes h_1 en h_2 kunnen samengevoegd worden tot een nieuwe boom door een boog toe te voegen van de wortel van de ene boom naar de wortel van de andere. Je wilt de hoogte van de nieuwe boom zo klein mogelijk wil houden. Welke van de twee oorspronkelijke wortels moet je dan als nieuwe wortel aanduiden, en wat is de hoogte van de nieuwe boom?

2.3.2 Eigenschappen van bomen

1. Een volledige, binaire boom met alle bladeren op hoogte n heeft evenveel bladeren als het aantal deelverzamelingen van een verzameling met n elementen. Verklaar.
Hint: een binaire boom kan je bekijken als een beslissingsboom met telkens twee keuzes.
2. Hoeveel kleuren heb je nodig om een boom te kleuren?

3. Bewijs dat voor een samenhangende graaf G geldt:

(Extra)

G is een volledige, binaire boom $\Leftrightarrow \exists i \in \mathbb{N} : G$ heeft i knopen met graad 3, 1 knoop met graad 2, $i + 2$ knopen met graad 1, en geen knopen met andere graad.

2.3.3 Een meer compacte voorstelling voor bomen

1. Geef een voorbeeld van een boom met hoogte meer dan 10 die in de compactere graafvoorstelling een “hoogte” (als je bogen die een kring vormen wegdoet tot je een boom krijgt) minder dan 5 heeft.

2.3.4 Opspannende bomen

1. Geef een opspannende boom voor...
- (a) K_n (breedte-eerst constructie)
 - (b) K_n (diepte-eerst constructie)
 - (c) $K_{n,m}$ (breedte-eerst constructie)
 - (d) $K_{n,m}$ (diepte-eerst constructie)

2.3.5 Minimale opspannende bomen

1. Bij de Belgische spoorwegen besluit men om een nieuw soort treinen te laten rijden waarvoor een aanpassing aan de spoorlijn nodig is. Die aanpassing is erg duur, dus ze willen in een eerste fase zo weinig mogelijk kilometers spoorlijn aanpassen, terwijl toch vanuit elke grote stad elke andere grote stad kan bereikt worden met de nieuwe trein. Welke spoorlijnen kunnen ze het beste aanpassen? Zijn er nog andere mogelijkheden? Zo ja, welke oplossing zou jij verkiezen als goede verbindingen van provinciehoofdsteden naar Brussel belangrijk zijn?

In de volgende tabel staan de lengtes van de spoorlijnen tussen de verschillende steden die verbonden moeten worden (als er geen getal staat is er geen rechtstreekse verbinding):

	Charleroi	Gent	Hasselt	Kortrijk	Leuven	Luik	Mechelen	Mons	Namen
Aarlen						128			80
Antwerpen		49	66		51		17		
Brugge		22		39					
Brussel	38	32		66	25		25	46	57
Charleroi								30	30
Gent				20					
Hasselt					39	58			
Kortrijk								64	
Leuven						38	17		64
Luik									38

2. Gegeven een gewogen, samenhangende graaf G en 1 van zijn knopen s . Bestaat er dan steeds een opspannende boom T van die graaf zodat er voor elke knoop d van

G een kortste pad p bestaat van s naar d zodat p volledig in T ligt. Bestaat er ook een minimaal opspannende boom met die eigenschap? Toon aan.

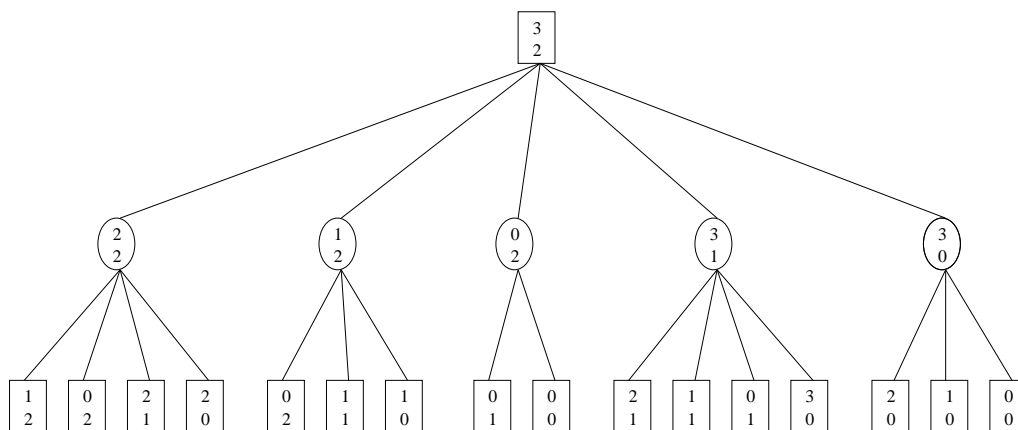
- De algoritmes van Prim en Kruskal zijn gulzige algoritmes. Bedenk een ander gulzig (**Extra**) algoritme en bewijs of het optimaal is (vb wissel).

2.3.6 Doorlopen van bomen

- Vergelijk diepte-eerste en breedte-eerst doorlopen van een binaire boom met hoogte d voor het zoeken van een element op hoogte $k \leq d$ op gebied van tijdscomplexiteit en geheugengebruik.

2.3.7 Spelbomen

- Beschouw de eerste 3 niveaus van de spelboom voor $\text{nim}(3,2)$:

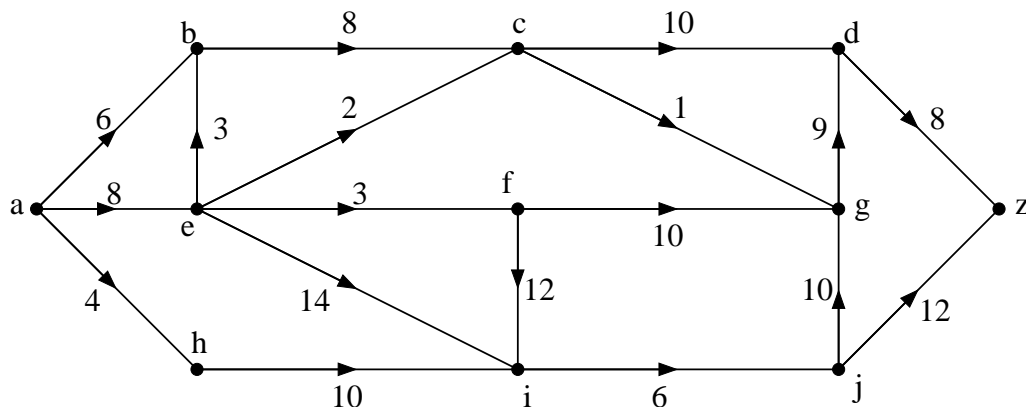


Zoek een gepaste evaluatiefunctie die aan elke knoop op diepte 2 een getal toekent. Label dan de knopen van de boom m.b.v. die evaluatiefunctie en pas het α - β algoritme toe.

2.4 Netwerkmodellen en Petri-netten

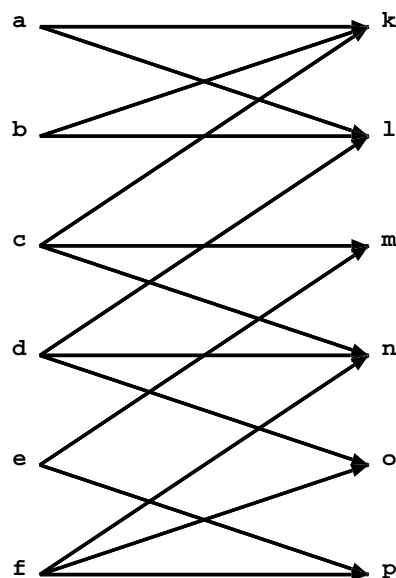
2.4.2 Maximale stroming

- Vind alle minimale snedes op Figuur 2.65 in de cursus (p. 125).
- Voer het algoritme voor het vinden van de maximale stroming (Algoritme 4.2.3) toe op de volgende graaf, en zoek alle minimale snedes.



2.4.3 Matching

1. Toon aan dat er voor de volgende graaf geen maximale matching is die boog (c, k) bevat:

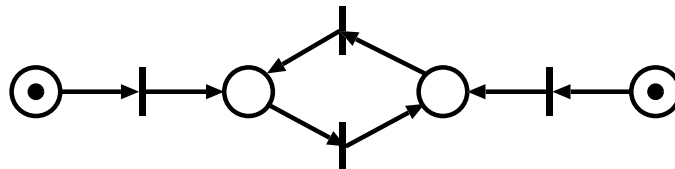


2. Is elke matching een deelgraaf van een maximale matching?
3. (a) Stel een lessenrooster op dat aan de volgende voorwaarden voldoet:
 - Er mogen ten hoogste drie lessen op één dag vallen: 1 's ochtends, 1 op de middag en 1 in de late namiddag;
 - Per week zijn er 3 lessen van FVI en SOCS, 2 lessen van Statistiek en Logica en 1 les van Filosofie en OP;
 - Voor elk vak met meer dan 1 les per week: niet alle lessen van een vak mogen op dezelfde dag vallen;
 - De lessen van SOCS moeten in het begin van de week vallen (ma/di/wo) en die van OP mogen niet in het begin van de week vallen;

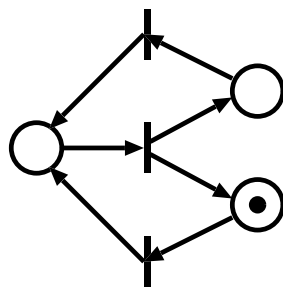
- De lessen van Fundamenten mogen niet op donderdag doorgaan en die van Statistiek mogen niet op woensdag of vrijdag vallen;
 - Logica en Filosofie mogen niet op maandag of vrijdag vallen;
 - Woensdag en vrijdag zijn er in de late namiddag geen lessen, en maandagochtend ook niet.
- (b) Het barteam van Wina stelt voor om op donderdagochtend geen lessen meer te organiseren. Is dat mogelijk?
4. Toon aan dat er een volledige matching bestaat als in een matching-probleem alle (Extra) knopen in V juist 2 uitgaande bogen hebben en alle knopen in W juist 2 inkomende.

2.4.4 Petrinetten

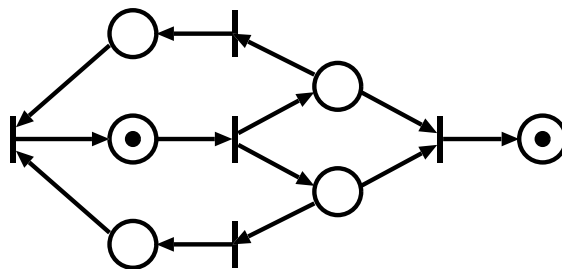
1. Geef voor de volgende Petrinetten aan of de markering levend is of dood, of de markering begrensd is of onbegrensd, en of de markering veilig is of niet.



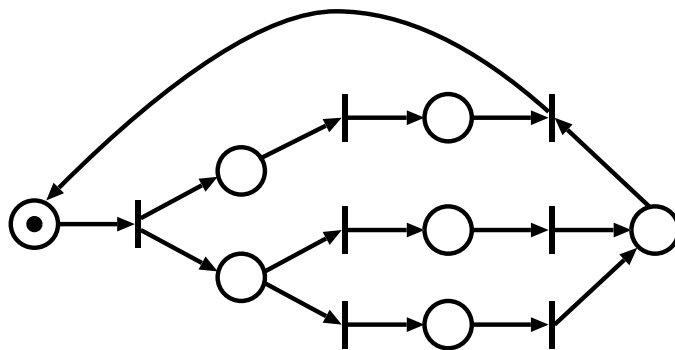
(a)



(b)



(c)



(d)

2. Construeer een Petrinet voor volgend stukje programmacode:

```
int a = 1;
int s = 0;
while (true) {
    s += a;
    a++;
}
```

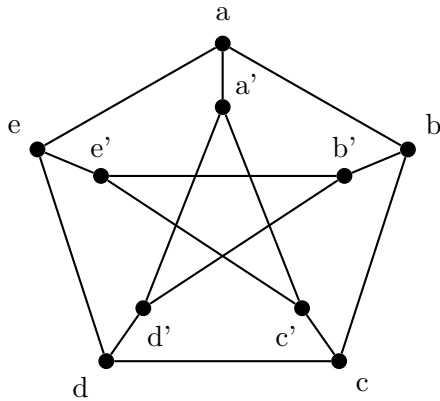
3. Construeer een Petrinet voor volgend stukje programmacode:

```
int a = N;
int s = 0;
while (a < 10) {
    a++;
    s = 2*s + 10;
}
s *= a;
```

4. Construeer een Petrinet dat de verkeerslichten op een kruispunt modelleert. Er zijn **(Extra)** twee verkeerslichten; ze moeten afwisselend groen worden en ze mogen niet allebei tegelijkertijd groen zijn: ze moeten eerst allebei rood zijn voor er terug een licht groen wordt.

2.5 Overzichtsoefeningen

1. Toon aan dat voor elke enkelvoudige graaf geldt dat $e \leq \frac{v-1}{2} \cdot v$.
2. Een schaakbord is 8×8 vakjes groot. Een paard mag alleen zetten doen waarbij het 2 vakjes in de ene richting (horizontaal/verticaal) gaat en dan 1 vakje in de andere richting.
 - (a) Kan een paard het schaakbord overlopen zodat deze elke mogelijke zet juist éénmaal doet en terug op zijn beginpositie komt?
 - (b) Kan een paard elke mogelijke zet juist éénmaal doen en op een andere positie dan zijn beginpositie terechtkomen?
3. Het complement van een enkelvoudige graaf G is een enkelvoudige graaf \overline{G} met dezelfde knopen als G . De graaf \overline{G} heeft een boog van knoop a naar knoop b (met $a \neq b$) als en slechts als G geen boog heeft van a naar b .
Toon aan dat elke enkelvoudige graaf met 11 knopen of meer ofwel zelf niet vlak is ofwel een complement heeft dat niet vlak is.
4. Welke eigenschappen blijven bewaard onder een rijreductie?
5. Toon aan dat de volgende graaf geen Hamiltoniaanse kring heeft.



9. Een wiel met n spaken W_n is een graaf met $(n + 1)$ knopen m, v_1, \dots, v_n , en bogen $(v_1, v_2), (v_2, v_3), \dots, (v_{n-1}, v_n), (v_n, v_1)$ en (m, v_i) voor elke i . Geef de minimale k waarvoor W_n een k -kleuring heeft als een functie van n . Bewijs dat je functie correct is.
10. Toon aan dat elke duale graaf van een vlakke graaf die een Euleriaanse kring heeft, kan gekleurd worden met 2 kleuren.

Hoofdstuk 3

Vastepuntstheorie

3.2 Orderelaties

3.2.1 Basisbegrippen

1. Toon aan dat de relatie “is een veelvoud van” een partiële orderelatie is op \mathbb{N} . Teken een Hasse-diagramma voor deze relatie op $\{1, 4, 7, 8, 14, 21, 56\} \subset \mathbb{N}$. Orden de getallen uit $\{0, 20, 40, 30, 10\}$ van klein naar groot volgens deze relatie. Als dit niet gaat, kan je dan een element weglaten zodat dit wel gaat?
2. Veronderstel dat R een orderelatie is op een verzameling A . Toon aan dat R^{-1} ook een orderelatie is. ($R^{-1} = \{(x, y) \mid (y, x) \in R\}$)
3. Bewijs: “... is een voorouder van (of gelijk aan) ...” (notatie $\dots \preceq \dots$) is een partiële orderelatie op de verzameling K van knopen van een boom B (met wortel w).

Geef een bovengrens voor elke deelverzameling $X \subseteq K$. Hebben alle deelbomen een ondergrens, infimum, supremum? Voor welke bomen B is B, \preceq een complete tralie?

4. Waar of niet waar? Argumenteer.
 - (a) Het gesloten interval $[a, b] \subset \mathbb{Z}$ met orde \leq is een complete tralie.
 - (b) Het gesloten interval $[a, b] \subset \mathbb{Q}$ met orde \leq is een complete tralie.
 - (c) Het gesloten interval $[a, b] \subset \mathbb{R}$ met orde \leq is een complete tralie.
 - (d) Het open interval $]a, b[\subset \mathbb{N}$ met orde \leq is een complete tralie.
 - (e) Het open interval $]a, b[\subset \mathbb{R}$ met orde \leq is een complete tralie.
5. Beschouw een willekeurige verzameling A en neem $X \subseteq \mathcal{P}(A)$. Toon aan dat in de (Extra) geordende verzameling $\mathcal{P}(A), \subseteq$ geldt dat

$$\sup(X) = \bigcup_{C \in X} C \quad \text{en} \quad \inf(X) = \bigcap_{C \in X} C$$

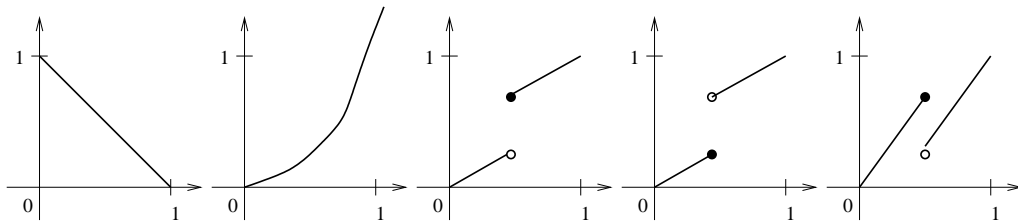
6. Neem een willekeurige oneindige verzameling A en neem (Extra)

$$X = \{B \subseteq A \mid B \text{ is eindig}\}$$

Toon aan dat X in $\mathcal{P}(A)$ slechts één bovengrens heeft, namelijk A .

3.2.2 Monotone en continue afbeeldingen

- Is de afbeelding $T : \mathbb{N} \rightarrow \mathbb{N} : x \mapsto x + 1$ monotoon op
 - \mathbb{N}, \leq ?
 - $\mathbb{N}, |$?
 - \mathbb{N}, \geq ?
- Is de afbeelding $T : \mathbb{N} \rightarrow \mathbb{N} : x \mapsto \begin{cases} 2x & \text{als } x \text{ even is} \\ x & \text{als } x \text{ oneven is} \end{cases}$ monotoon op
 - \mathbb{N}, \leq ?
 - $\mathbb{N}, |$?
- Geef een voorbeeld van een complete tralie L, \leq en een deelverzameling $X \subseteq L$ waarbij X **niet** gericht is.
- Zijn er continue afbeeldingen die niet monotoon zijn? En monotone afbeeldingen die niet continu zijn?
- Hieronder vind je een aantal afdrucken van grafieken op de complete tralie $[0, 1], \leq$. **(Extra)** Welke grafieken stellen een continue afbeelding op $[0, 1]$ voor? (Continu in de zin van deze cursus!). Kun je een algemene kenschets geven van de continue afbeeldingen op deze complete tralie (geen bewijzen nodig)?



3.3 De stellingen van Tarski en Kleene

- Gegeven een (niet noodzakelijk samenhangende) graaf $G = (V, E)$. Beschouw de afbeelding T die een deelverzameling X van V afbeeldt op $X \cup \{v \in V \mid (a, v) \in E \wedge a \in X\}$, m.a.w. T voegt aan een verzameling knopen al haar burens toe. Bestaan er vaste punten van T ? Wat zijn de vaste punten van T ?
- Beschouw de monotone afbeelding

$$T : [0, 1] \rightarrow [0, 1] : \begin{cases} T(x) = \frac{1}{4} + \frac{1}{2}x & \text{als } 0 \leq x < \frac{1}{2} \\ T(x) = \frac{3}{4} + \frac{1}{4}x & \text{als } \frac{1}{2} \leq x \leq 1 \end{cases}$$

Toon aan dat $T \uparrow \omega$ geen vast punt is. Is T continu?

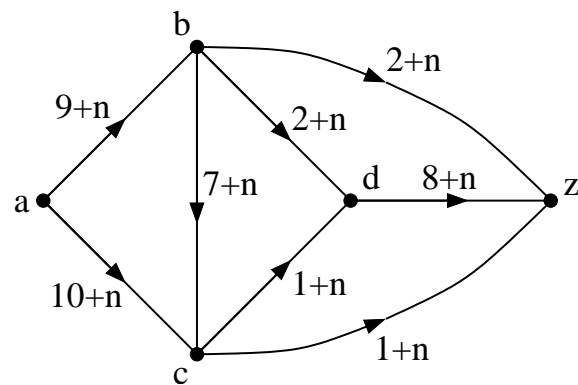
- Werk de details uit van het voorbeeld dat geschetst wordt op pagina 154.
- Geef een niet-continue functie $T : [0, 1] \rightarrow [0, 1]$ op de complete tralie $[0, 1], \leq$ waar- **(Extra)** voor $T \uparrow \omega$ toch een vast punt is.

5. Beschouw een graaf $G = (V, E)$ en de relatie R op $V \times V$ waarbij $R = \{(x, y) \mid (x, y) \in E \text{ of } (y, x) \in E\}$. M.a.w. xRy als de knoop x een buur is van y . Wat is de transitieve sluiting van deze relatie? Als we T definiëren zoals in de cursus, is er dan een $n \in \mathbb{N}$ waarvoor de transitieve sluiting van R samenvalt met $T \uparrow n$? Bewijs. Waarmee is de graaf $(V, T \uparrow n)$ isomorf als G samenhangend is? **(Extra)**

Hoofdstuk 4

Overzichtsoefeningen

1. De *Busy Beaver* functie $S(n)$ geeft het maximaal aantal stappen dat een Turing machine met alfabet $\Sigma = \{\#, 1\}$ en $n+1$ toestanden kan doen op de lege invoerband, zonder in een oneindige lus te gaan. De toestanden zijn q_0, q_1, \dots, q_{n-1} en een aanvaardbare eindtoestand h , en “blijf staan” is geen toegelaten beweging van de schrijfkop (alle Turing machines kunnen zonder “blijf staan” geschreven worden).
Geef $S(1)$ en (een ondergrens voor) $S(2)$ en bewijs (in grote lijnen) dat de functie S niet Turing berekenbaar is.
2. Teken de mogelijke onderlinge liggingen van de klassen **P**, **NP** en **NPC**.
3. Beschouw de talen $HC = \{G \mid G \text{ is een graaf met een Hamiltoniaanse kring}\}$ en $TSP = \{(G, k) \mid G \text{ heeft kring met gewicht } \leq k \text{ die alle knopen aandoet}\}$.
Bewijs dat $HC \propto TSP$.
4. Wat zou er gebeuren met stelling 4.3.4, uit welke equivalentieklassen voor \sim zou **P** bestaan en hoe verandert **NPC** als je in de definitie van *polynomiale transformatie* de voorwaarde weglaat dat f in polynomiale tijd berekend kan worden? Wat als je de voorwaarde vervangt door “ f kan in lineaire tijd berekend worden”?
5. Waar of niet waar? Bewijs of geef een tegenvoorbeeld.
 - (a) Alle bomen zijn tweeledige grafen.
 - (b) Op sommige grafen die n -kleurbaar zijn kunnen rijreducties toegepast worden die de graaf herleiden tot een graaf die niet n -kleurbaar is.
 - (c) Op geen enkele graaf die niet n -kleurbaar is kan een rijreductie worden toegepast die de graaf herleidt tot een graaf die wel n -kleurbaar is.
 - (d) Alle vlakke grafen met een 2-kleuring zijn bomen.
6. Bespreek voor het volgende netwerk de maximale stroming en de minimale snede als functie van $n \in \mathbb{N}$.



7. Bereken de asymptotische tijdscomplexiteit van algoritme 4.2.3 (p. 120), voor een transportnetwerk met n knopen en maximale stroming F . Kan je een voorbeeld geven waarvoor deze worst-case complexiteit bereikt wordt?