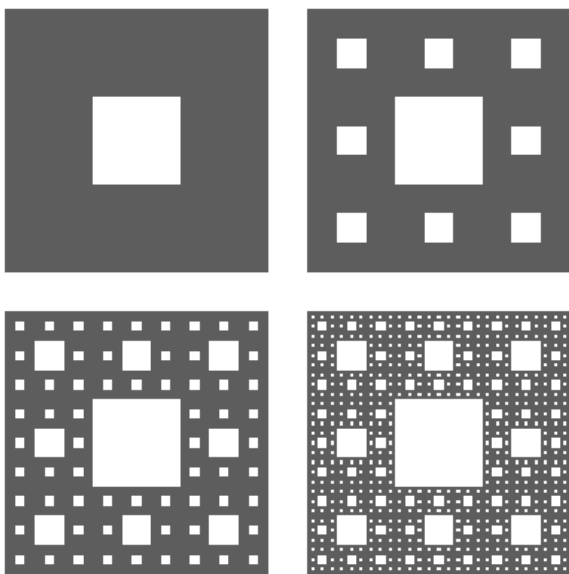


Dit tentamen duurt twee uur en telt 32 vragen. Bij elke vraag is steeds precies één antwoord het juiste. Wel kunnen andere antwoorden “bijna juist” of “deels juist” zijn. Mochten er meerdere goede antwoorden zijn, dan geldt het beste antwoord. Omdat er verschillende versies van de opgaven bestaan, correspondeert de volgorde van de opgaven niet altijd met de volgorde van de stof zoals die behandeld is in de colleges.

Het gebruik van elektronische apparatuur, inclusief rekenmachines, is verboden. Het doorseinen van antwoorden, in welke vorm dan ook, wordt gewaardeerd met een nul.

Veel succes!

1. De fractale dimensie van Sierpinski's tapijt

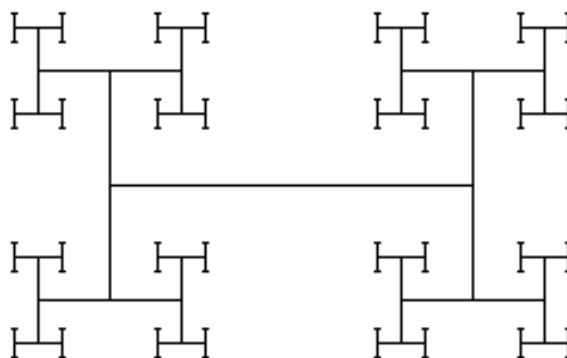


is gelijk aan

- (a) $\log 3 / \log 4$
 (b) $\log 4 / \log 3$
 (c) $\log 3 / \log 8$
 ✓ $\log 8 / \log 3$
2. Welke van de hieronder gegeven termen zijn alternatieve aanduidingen voor de term iterated function system?

- (i) Iterated non-linear fractal.
 (ii) Multiple reduction copy machine.
 (iii) Set of linear contractions.
 (iv) Recursive function system.
- (a) (2i) en (2ii).
 ✓ (2ii) en (2iii).
 (c) (2iii) en (2iv).
 (d) Het goede antwoord staat er niet bij.

3. De H-boom, waarvan hieronder het begin getoond wordt, fungeert in VLSI-ontwerp als uitgangspunt voor circuit layouts. Alle onderdelen van een chip ontvangen de klok-puls dan met een ongeveer vergelijkbare vertraging. Er zijn meer toepassingen van H-bomen, zoals antenne-ontwerp en efficiënte vlakvulling.



Laat F = “forward”, $+$ = “links 90° ”, $-$ = “rechts 90° ”, $[\cdot]$ = “vergroot 0.65”. De H-boom kan worden gegenereerd met de volgende Lindenmayer formule:

$$\begin{cases} \text{Axioma: } [F]-F \\ \text{Productieregel:} \end{cases}$$

- (a) $F \rightarrow +F - F$
 ✓ $F \rightarrow [+F][-F]$
 (c) $F \rightarrow +F| - F$
 (d) $F \rightarrow [+F][[-F]]$

4. Flake plaatst Lindenmayer-systemen in de Chomsky hiërarchie van formele talen. Volgens hem kunnen L-systemen worden gezien als een bepaald type grammatica waarin non-terminals parallel worden herschreven. (Meerdere turtles tegelijkertijd worden aangestuurd.) Naar welk grammaticaal type refereert Flake?

- (a) De reguliere grammatica.
 ✓ De context-vrije grammatica.
 (c) De context-gevoelige grammatica.
 (d) De semi-beslisbare grammatica.

5. Beschouw de volgende *incorrecte* definitie van een contractie:

Een contractie is een afbeelding f van een metrische ruimte X naar zichzelf, zodanig dat voor elk tweetal elementen $x, y \in X$ geldt dat er een constante $c < 1$ te vinden is, zó dat $d(fx, fy) \leq c \cdot d(x, y)$.

Deze definitie is incorrect om de volgende reden:

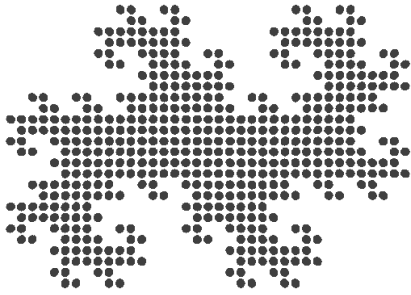
- (a) Een contractie is een *lineaire* afbeelding.
 ✓ De constante c mag niet afhangen van x en y .
 (c) Het “ \leq ”-teken in $d(fx, fy) \leq c \cdot d(x, y)$ moet worden vervangen door een “ $<$ ”-teken.
 (d) Om een andere reden.

★ De volgende definitie is correct.

Een contractie is een afbeelding f van een metrische ruimte X naar zichzelf, zodanig dat er een constante $c < 1$ te vinden is, zodanig dat voor elk tweetal elementen $x, y \in X$ geldt dat $d(fx, fy) \leq c \cdot d(x, y)$.

Merk op dat de phrase “zodanig dat er een constante $c < 1$ te vinden is” naar voren is geschoven.

6. Geef het IFS dat hoort bij de volgende afbeelding.



(a) Het systeem

$$\begin{bmatrix} x \\ y \end{bmatrix} \mapsto \begin{bmatrix} 1/2 & -1/2 \\ -1/2 & 1/2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} -d \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \end{bmatrix} \mapsto \begin{bmatrix} 1/2 & -1/2 \\ -1/2 & 1/2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} d \\ 0 \end{bmatrix}.$$

✓ Het systeem

$$\begin{bmatrix} x \\ y \end{bmatrix} \mapsto \begin{bmatrix} 1/2 & -1/2 \\ 1/2 & 1/2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} -d \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \end{bmatrix} \mapsto \begin{bmatrix} 1/2 & -1/2 \\ 1/2 & 1/2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} d \\ 0 \end{bmatrix}.$$

(c) Het systeem

$$\begin{bmatrix} x \\ y \end{bmatrix} \mapsto \begin{bmatrix} 1/2 & -1/2 \\ -1/2 & 1/2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} -d \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \end{bmatrix} \mapsto \begin{bmatrix} -1/2 & 1/2 \\ 1/2 & -1/2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} d \\ 0 \end{bmatrix}.$$

(d) Het systeem

$$\begin{bmatrix} x \\ y \end{bmatrix} \mapsto \begin{bmatrix} -1/2 & 1/2 \\ 1/2 & -1/2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} -d \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \end{bmatrix} \mapsto \begin{bmatrix} 1/2 & -1/2 \\ -1/2 & 1/2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} d \\ 0 \end{bmatrix}.$$

7. De Mandelbrot fractal kan worden verkregen door de volgende vraag te beantwoorden:

- ✓ Voor welke waarden van c convergeert de iteratie van $z^2 + c$ in het complexe vlak?
- (b) Voor welke waarden van c convergeert de iteratie van $2^z + c$ in het complexe vlak?
- (c) Voor welke waarden van z convergeert de iteratie van $z^2 + c$ in het complexe vlak?
- (d) Voor welke waarden van z convergeert de iteratie van $2^z + c$ in het complexe vlak?

★ Antwoord (c) definieert de Julia fractal.

8. De kleuren in de Mandelbrot fractal kunnen worden gegenereerd door de volgende methode.

- (a) Meet de afstand tot $(2, 0)$ na, zeg, 50 iteraties.
- (b) Meet de afstand tot de eenheidscirkel na, zeg, 50 iteraties.
- (c) Tel het aantal iteraties voordat je buiten de eenheids-cirkel komt.
- ✓ Tel het aantal iteraties voordat je verder dan twee van de oorsprong af komt.

9. Het aantal verschillende omgevingen van een cel in een 2-dimensionale cellulaire automaat met radius R en aantal toestanden K , waarin de toestand van de cel zelf ook meetelt, is gelijk aan:

- (a) $(K^2)^{2R+1}$
- ✓ $\sqrt{K^{(2R+1)^2}}$, i.e., $K^{\wedge(2R+1)^2}$
- (c) $(2R+1)^{2K}$
- (d) $(2R+1)^{K^2}$, i.e., $2R+1^{\wedge K^2}$

10. Het aantal verschillende regels van een 1-dimensionale cellulaire automaat met radius R en aantal toestanden K is gelijk aan:

- ✓ $\sqrt{K^{K^{(2R+1)}}}$, i.e., $K^{\wedge(K^{\wedge(2R+1)})}$
- (b) $(2R+1)^{K^{(2R+1)}}$, i.e., $(2R+1)^{\wedge(K^{\wedge(2R+1)})}$
- (c) $K^{K^{2R}}$, i.e., $K^{\wedge(K^{\wedge(2R)})}$
- (d) $(2R)^{K^{2R}}$, i.e., $(2R)^{\wedge(K^{\wedge(2R)})}$

11. Wat zegt de Pareto verdeling (ook wel: Zip's regel) over de verdeling van de omvang van lawine's in cellulaire automaten die gebouwd zijn om zandhopen te simuleren?

- (a) Een lawine met omvang s vindt plaats met kans $P(X = s) = e^{-\lambda} \lambda^s / s!$.
- (b) Een lawine met omvang s vindt plaats met kans $P(X \leq s) = e^{-\lambda} \lambda^s / s!$.
- ✓ De frequentie f van lawines met een bepaalde omvang is omgekeerd evenredig aan de omvang s volgens $f(s) = as^{-b}$.
- (d) De frequentie f van lawines met een bepaalde omvang is omgekeerd evenredig aan de omvang s volgens $f(s) = a \log(s) s^{-b}$.

★ Syllabus, Chapter 2, p. 35.

12. Noem twee mogelijke effecten van de combinatie van evolutie en leren.

- (a) Bratley effect en Lamarckiaans leren.
- (b) Bratley effect en latent leren.
- ✓ Baldwin effect en Lamarckiaans leren.
- (d) Baldwin effect en latent leren.

★ Syllabus, Chapter 2, pp. 27-28.

13. Strategic Bugs is een ALife model dat in 1992 voorgesteld werd door Bedau en Packard. Welk doel hadden Bedau en Packard met Strategic Bugs?

- (a) Het bestuderen van strategisch gedrag: in welke mate manifesteert strategisch gedrag zich in de populatie en waarom?
- ✓ Het bestuderen van evolutionaire activiteit: in welke mate worden nieuwe genetische vernieuwingen tot stand gebracht in de populatie?
- (c) Het bestuderen van emergent gedrag: hoe kan (nuttig) groepsgegedrag ontstaan uit individuele handelingen en strategieën?
- (d) Het genetisch programmeren van bugs zodat uiteindelijk de best genetisch geëvolueerde bug een specifiek voedselspoor volledig kan aflopen.

★ Syllabus.

14. Flake bespreekt Kolmogorov-complexiteit als een maat voor de complexiteit van fractals en andere constructies. Zelf noemt hij deze maat overigens “algoritmische complexiteit” en verwijst in een voetnoot naar de officiële term “Kolmogorov-complexiteit”. Welke fractal bezit de hoogste Kolmogorov-complexiteit?

- ✓ Een IFS met 1000 wezenlijk verschillende contracties.
 (b) Sierpinski’s tapijt (volledig uitgeïtereerd).
 (c) De Mandelbrot set (volledig uitgeïtereerd).
 (d) De Hilbert curve (volledig uitgeïtereerd).

★ Het gaat om de maximale lengte van het kortste programmaatje dat nodig is om een bepaalde fractal te produceren. Sierpinski’s tapijt, de Mandelbrot set en de Hilbert curve zijn allemaal met relatief redelijk korte programmaatjes te genereren. Echter, het is onmogelijk de informatie die is vervat in 1000 wezenlijk verschillende contracties in R^2 te comprimeren tot een even kort programmaatje van vergelijkbare lengte. Immers, alle $1000 \times (4 + 2) = 6000$ matrix entries moeten een plaats krijgen in dat programmaatje.

15. Direct daarna bespreekt Flake een alternatief voor algoritmische complexiteit. Welk alternatief is dat en welke benaming gebruikt Flake daarvoor?

- (a) Computationale complexiteit.
 (b) Georganiseerde complexiteit.
 (c) Entropie.
 ✓ Stochastische onregelmatigheid.

16. We beschouwen de volgende vier beweringen over local hill-climbing (LHC).

- (i) LHC wordt meestal vanaf verschillende beginpunten gestart.
 (ii) Een nadeel van LHC is dat deze vaak in een lokaal maximum terecht komt.
 (iii) LHC vindt altijd de beste oplossing.
 (iv) LHC is geschikt voor zoeken op zowel discrete als continue ruimten. (Denk aan roosterpunten Z^2 voor discreet en het platte vlak R^2 voor continu.)

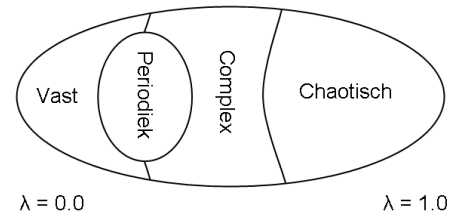
Welke beweringen zijn waar?

- ✓ Alle behalve (16iii).
 (b) Alle behalve (16iv).
 (c) Alle behalve (16iii) en (16iv).
 (d) Het goede antwoord staat er niet bij.

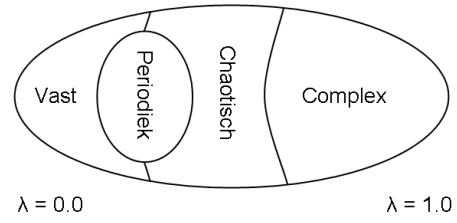
★ Omdat in de oorspronkelijke opgave per ongeluk nog “lokaal minimum” stond, en antwoord (d) dan ook goed zou zijn, is deze opgave afgekeurd.

17. Langton’s λ -parameter deelt de regelruimte van 1-dimensionale cellulaire automaten op in verschillende gebieden. Een schematische representatie van die gebieden is de volgende:

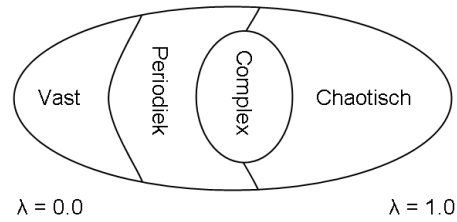
- (a) Figuur 1:



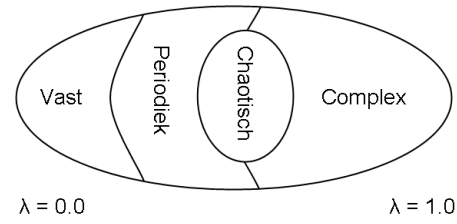
(b) Figuur 2:



✓ Figuur 3:



(d) Figuur 4:



18. In de context van genetische algoritmen wordt onder de term “bitmask” meestal een patroon van bits verstaan die de volgende functie heeft.

- ✓ Een bitmask geeft aan op welke plekken chromosomen bij kruising hun materiaal mogen uitwisselen.
 (b) Een bitmask verifieert of een schema nog in een populatie aanwezig is.
 (c) Een bitmask verifieert met welk percentage een schema nog in een populatie aanwezig is.
 (d) Een bitmask is een hulpconstructie om zg. kritieke plekken in chromosoomschema’s te definiëren.

★ Syllabus, Chapter 3.

19. In het werkcollege werd aangetoond dat, als A en B aftelbare verzamelingen zijn, dat dan ook het Cartesisch product $A \times B$ een aftelbare verzameling is.

Welke van de volgende beweringen is **NIET** waar?

- (a) Als de verzamelingen A_1, \dots, A_n stuk voor stuk aftelbaar zijn, dan is de verzameling

$$\prod_{i=1}^n A_i = A_1 \times \dots \times A_n$$

dat ook.

- (b) Als de aftelbaar oneindige rij van verzamelingen A_1, \dots, A_i, \dots aftelbaar zijn, dan is de verzameling

$$\bigcup_{i=1}^{\infty} A_i = A_1 \cup \dots \cup A_i, \dots$$

dat ook.

- (c) De verzameling van alle eindige bitstrings (van verder willekeurige lengte) is aftelbaar.

- ✓ De verzameling van alle (naar rechts) oneindige bitstrings is aftelbaar. (Een naar rechts oneindige bitstring is bijvoorbeeld "11101101101111110...". Dergelijke strings hebben wel een begin maar geen eind.)

★ (19a) Een Cartesisch product van twee aftelbare verzamelingen is aftelbaar, dus $A_1 \times A_2$ is aftelbaar. Dit herhalend kunnen we concluderen dat $A_1 \times A_2 \times A_3 = (A_1 \times A_2) \times A_3$ aftelbaar is. Zo doorgaand kunnen we concluderen dat $A_1 \times \dots \times A_n$ aftelbaar is.

(19b) Als A_1, \dots, A_i, \dots een aftelbaar oneindige rij van aftelbaar oneindige verzamelingen is, dan kunnen we deze onder elkaar zetten:

$$\begin{array}{ccccccc} A_1 & = & a_{11} & \dots & a_{i1} & \dots & \\ A_2 & = & a_{12} & \dots & a_{i2} & \dots & \\ A_3 & = & a_{13} & \dots & a_{i3} & \dots & \\ \vdots & & \vdots & \ddots & \vdots & \vdots & \\ A_j & = & a_{1j} & \dots & a_{ij} & \dots & \\ \vdots & & \vdots & & \vdots & \vdots & \end{array}$$

Deze verzameling $\{a_{ij}\}_{i,j}$ is af te tellen met behulp van bijvoorbeeld de zig-zag methode, of door een spiraal te tekenen door alle a_{ij} met als middelpunt a_{11} .

(19c) Alle eindige bitstrings zijn te nummeren door eerst alle bitstrings ter lengte 1 op te noemen. Dat zijn er eindig veel. Dan alle bitstrings ter lengte 2, etc.

Dan blijft over de verzameling van alle (naar rechts) oneindige bitstrings. Deze zou niet aftelbaar moeten zijn. (De overaftelbaarheid van deze verzameling kan inderdaad worden aangetoond met behulp van Cantor's diagonaalargument. Maar dit laatste werd niet behandeld.)

20. Eindige CA: een CA met eindig veel toestanden. Aftelbaar oneindige CA: een CA met aftelbaar oneindig veel toestanden.

- (i) Een **eindige** CA op **eindig** veel cellen neemt oneindig vaak dezelfde globale configuratie aan.
- (ii) Een **eindige** CA op **aftelbaar oneindig** veel cellen neemt oneindig vaak dezelfde globale configuratie aan.
- (iii) Een **aftelbaar oneindige** CA op **eindig** veel cellen neemt oneindig vaak dezelfde globale configuratie aan.
- (iv) Een **aftelbaar oneindige** CA op **aftelbaar oneindig** veel cellen neemt oneindig vaak dezelfde globale configuratie aan.

Welke beweringen zijn waar?

- ✓ Alleen (20i).
- (b) Alleen (20i) en (20ii).
- (c) Alleen (20i) en (20iii).
- (d) Het goede antwoord staat er niet bij.

★ (20i) Waar: dit is uitgelegd op het college.

(20ii) Onwaar: bekijk de meest eenvoudige glider in een verder leeg grid. Elke keer is de globale configuratie weer anders.

Wel is het zo dat er voor eindige CAs met aftelbaar oneindig veel cellen tenminste één globale configuratie bestaat zodanig dat elke cel oneindig vaak wordt "bezoekt". Bv. de lege configuratie bij een glider in een verder leeg grid wordt oneindig vaak bezocht omdat gliders nooit terugkeren.

(20iii) Onwaar: intuïtief is dit in te zien door je te realiseren dat je één (verder willekeurig gekozen) cel kunt laten aftellen. Die ene cel zorg er dan voor dat de globale configuratie steeds veranderd.

(20iv) Onwaar: immers, dit is een sterkere versie dan (20iii).

21. Welke van de volgende beweringen over proportionele selectie (fitness proportionate selection) zijn waar?

- (i) FPS is invariant onder optellen.
- (ii) FPS is invariant onder vermenigvuldigen.
- (iii) Er is een gevaar van voortijdige convergentie als fitte individuen dankzij FPS te vroeg en te vaak geselecteerd worden.
- (iv) Bij FPS bestaat het gevaar dat fitnesswaarden te dicht bij elkaar gaan liggen, waardoor de selectiedruk wegvalt.

Welke beweringen zijn waar?

- (a) Bewering (21i) en (21ii).
- (b) Bewering (21ii) en (21iii).
- (c) Bewering (21iii) en (21iv).
- ✓ Een andere combinatie van beweringen is waar.

★ De enige onware bewering is (21ii).

22. Welke van de volgende beweringen over toernooi-selectie zijn waar?

- (i) De toernooi-grootte kan in principe vrij gekozen worden, tussen nul en de populatiegrootte.
- (ii) Toernooi-selectie is lastiger te implementeren dan FPS.
- (iii) Toernooi-selectie is inefficiënter (rekenintensiever) dan FPS.
- (iv) De gebruikelijke toernooi-grootte ligt tussen de twee en drie.
- (a) Bewering (22i) en (22ii).
- (b) Bewering (22ii) en (22iii).
- (c) Bewering (22i) en (22iv).
- ✓ Een andere combinatie van beweringen is waar.

★ Alleen (22iv) is waar.

23. De s-expressie E is gelijk aan

$$(/ (\text{SQRT} (+ 4 (- X 5))) (- Y (- W))).$$

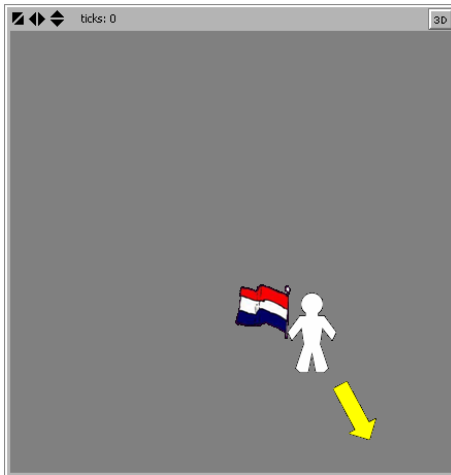
Hoeveel kruisingen van E met zichzelf zijn er mogelijk zonder E weer terug te krijgen?

- (a) 30
- (b) 90
- ✓ 110
- (d) Het goede antwoord staat er niet bij.

★ De s-expressie

$(/ (\text{SQRT} (+ 4 (- X 5))) (- Y (- W)))$ bezit 11 sub-expressies. (Vijf lijsten, waarvan één top-expressie, en vijf bladeren.) bij twee ouders is elke sub-expressie is een mogelijk punt van kruising. Deze punten moeten wel verschillend zijn. Bij Ouder 1 zijn er dus 11 keuzes, en bij Ouder 2 zijn er dan nog maar $11 - 1 = 10$ keuzes. Dat geeft 110 mogelijke kruisingen.

24. In een (zogenaamd) nieuwe versie van Netlogo is het mogelijk de randen met elkaar te identificeren volgens de Möbiusband. We vinken de volgende vakjes aan in de Settings dialoog: “world does not wrap horizontally” en “world wraps vertically according to the Möbius strip”.



In het figuur zie je een ventje naar (voor ons) rechtsonder lopen met een vlag (voor ons) links in zijn hand. Op een gegeven moment duikt dit ventje boven weer op.

- Hij loopt dan naar linksonder met het vlaggetje rechts in zijn hand.
- (b) Hij loopt dan naar rechtsonder met het vlaggetje rechts in zijn hand.
- (c) Hij loopt dan naar linksonder met het vlaggetje links in zijn hand.
- (d) Hij loopt dan naar rechtsonder met het vlaggetje links in zijn hand.

★ Eigenlijk bevindt het mannetje zich ook nog aan de andere kant van het oppervlak en kijken we hem op zijn rug.

25. Hier staan een aantal voorwaarden voor recombinitie op genotypen.
- (i) Allelen moeten in dezelfde volgorde blijven staan.
 - (ii) Allelen mogen niet herhaald worden.
 - (iii) Elk genotype dient te beginnen met hetzelfde allel.

Aan welke voorwaarden dient een standaard GA-implementatie van het handelsreizigerprobleem te voldoen?

- (a) Aan (25i) en (25ii).
- (b) Aan (25ii) en (25iii).
- (c) Aan (25i) en (25iii).
- Het goede antwoord staat er niet bij.

★ (25i): onjuist. De uitdaging is juist om de allelen zo te ordenen dat de kortste route wordt gelopen.

(25ii): juist. Steden moeten precies één keer bezocht worden.

(25iii): onjuist. Dat een toer moet beginnen en eindigen in dezelfde stad legt daarmee de begin- en eindstad nog niet vast.

26. We evalueren de volgende twee s-expressies:

- (a) $(\text{eval } (\text{eval } '(+ 1 (* 2 3))))$
- (b) $(\text{eval } '(\text{eval } '(+ 1 (* 2 3))))$

Krijgen we dezelfde uitkomst?

- Ja.
- (b) Nee, (26a) geeft een run-time error.
- (c) Nee, (26b) geeft een run-time error.
- (d) Nee, beide expressies evalueren naar verschillende waarden.

27. Geef het aantal mogelijke kruisingen van $(\text{if } (\text{food-ahead}) (\text{right}) (\text{left}))$ en $(\text{if } (\text{not food-ahead}) (\text{right}) (\text{move}))$.

- (a) 4
- (b) 9
- 10
- (d) 16

28. In de ontwikkeling van een algoritme voor GP is er de keuze tussen twee sets van logische operatoren:

- (i) $\{\neg, \wedge\}$
- (ii) $\{\neg, \wedge, \rightarrow, \vee, \equiv\}$

Welke van de twee ga je gebruiken, en waarom? Neem in overweging dat de logische operatoren \neg en \wedge volstaan om elke propositie-logische bewering uit te drukken.

- (a) Verz. (28i) is beter geschikt dan verzameling (28ii), omdat Verz. (28i) minder operatoren bevat.
- (b) Verz. (28i) is beter geschikt dan verzameling (28ii), omdat Verz. (28i) logisch compleet is.
- Verz. (28ii) is beter geschikt dan verzameling (28i), omdat Verz. (28ii) een grotere variëteit aan operatoren bevat.
- (d) Beide verzamelingen zijn even geschikt.

29. Naast reproductie, mutatie en recombinitie spreekt (de groep van) John Koza nog over een vierde genetische operator, namelijk een wijziging in de architectuur (an *architecture-altering operation*).

Hiermee wordt het volgende bedoeld.

- (a) Structurele wijzigingen in de programmatuur waarmee GP-elementen worden gemanipuleerd, bv. een besluit om LISP lijsten anders te representeren (bv. als dubbel-gelinkte lijst met sentinel).
- Structurele wijzigingen in de basiselementen van GP bouwstenen, bv. een wijziging van het aantal functie-argumenten van een specifieke functie.
- (c) Essentiële wijzigen in de experimentele opzet van GP sessies, met name waar het de instelling van experiment-specifieke parameters betreft.
- (d) Essentiële wijzigen in de programmatuur waarmee GP-elementen worden gemanipuleerd, bv. de omschakeling naar een andere GP omgeving (Eng.: “environment” of “testbed”).

30. John Holland publiceerde in 1975 de zg. *schema-stelling*. Deze stelling wordt algemeen beschouwd als de eerste theoretische bouwsteen voor genetisch programmeren. Een versie van de schema-stelling is de volgende.

$$p(H_{t+1}) \geq p(H_t) \frac{|H_t|}{|X_t|} \frac{f(H_t)}{f(X_t)} (1 - \epsilon_c)(1 - \epsilon_m)$$

Er zijn nogal wat kanttekeningen te plaatsen bij de schema-stelling. Welke van de onderstaande kanttekeningen is niet behandeld in dit vak¹ en is bovendien verkeerd geciteerd?

- (a) De fitness functie is niet uniek en kan daarom bovenstaande ongelijkheid beïnvloeden (Grefenstette en Baker, 1989).
- (b) Voor lange chromosomen is een populatie geen representatieve sample van een schema (Mühlenbein, 1991).
- ✓ Schemata zijn arbitrair. Bij chromosomen van lengte γ kunnen $\gamma!$ schemata verzonnen worden (Resnick, 1991).
- (d) Het rechterlid is weinig informatief als ondergrens (Vose, 1993).

★ Deze vraag is afgekeurd omdat we het er achteraf over eens werden dat deze opgave niet controleert op inhoudelijke kennis, maar op formele niet-vakinhoudelijke kennis.

31. Beschouw de volgende grammatica voor het genereren van rekenkundige expressies:

```

<expr> ::= ( <expr> <op> <expr> )
          | <atom>
<op>    ::= + | - | * | /
<atom>  ::= <var> | <num>
<var>   ::= h | v | m
<num>   ::= 0 | 0.1 | 0.5 | 1 | 5 | 10
          | 50 | 100 | 500

```

Als chromosomen nemen we bitstrings met lengte 10. Hoe groot moet de lengte van een codon minimaal zijn om alle grammaticaal geldige expressies te kunnen genereren?

- (a) Drie.
- ✓ Vier.
- (c) Acht.
- (d) Het goede antwoord staat er niet bij.

★ Drie is te weinig. In dat geval kan het getal 500 niet worden geproduceerd.

32. We gebruiken de gegevens in Vraag 31 en nemen aan dat codons een vastgestelde lengte 3 bezitten. Geef de minimale waarde van *maxwrap* om de bitstring “0000000001” een geldige expressie te laten produceren.

- (a) Nul.
- (b) Twee.
- ✓ Vier.
- ✓ Het goede antwoord staat er niet bij.

★ De theorie en JGE doen beiden iets anders. Bij de eerste drie codons is duidelijk wat te doen: ze zijn allemaal “000”. De vierde codon kan gekozen worden als “1” (één bit in plaats van drie) of “100” (drie bits, de laatste bit van de string, daarna wrappen zodat we twee nullen van het begin van de string consumeren). Theoretisch is de laatste interpretatie correct. In dat geval krijgen we dus de waarde “100” (binair) = 4 (decimaal) = $\langle \text{expr} \rangle \langle \text{op} \rangle \langle \text{expr} \rangle$, waardoor we de boom blijven vergroten en we na vier wraps geen correcte expressie hebben.

JGE en Netlog JGE doen het anders. Deze lezen de laatst overgebleven bit als 1 (decimaal), en hebben dientengevolge vier wraps nodig om $((h + h) + h) + h$ te produceren.

Einde.

¹ Met “niet behandeld” wordt bedoeld: werd niet besproken in hoorcolleges, en komt niet voor in handouts, syllabi, slides, of werkcollege-opgaven.