

Toets Kunstmatige Intelligentie (INFOB2KI)

30 januari 2015
11:00 - 13:00

MC	
I	
II	
T	

Vooraf

- Mobiele telefoons en dergelijke dienen uitgeschakeld te zijn.
- Het eerste deel van het tentamen bestaat uit 10 multiple-choice vragen. **Alleen als je in dit deel minstens 5 antwoorden goed hebt, zal het tweede deel van het tentamen nagekeken worden!**^a
- Het tweede deel van het tentamen bestaat uit 6 open vragen voor maximaal 70 punten.
- Controleer of je alle vragen hebt!

Tijdens

- Lees de vragen zorgvuldig. Licht je antwoorden duidelijk toe (in eerste deel optioneel)!
- Gedurende het tentamen mag geen materiaal van de cursus geraadpleegd worden (gesloten-boek tentamen), op een 'spiekbriefje' (A4tje **eigen** aantekeningen) na.
- Je mag gebruik maken van een *NIET*-programmeerbare rekenmachine.

Bij inleveren

- Controleer voor je weggaat of je je **naam** en alle antwoorden hebt ingevuld. (Zet bij niet ingevulde antwoorden een streepje.)
- Lever alle bladen in, en toon je "spiekbriefje" en collegekaart.

Binnenkort

- Graag de vakevaluatie invullen. Bedankt!

Succes!

^aNormering: 3 pntn per vraag, dus max 30 pntn; als #correct < 5, dan cijfer = (# pntn)/3 (= # correct).

Deel I (versie 1): kies steeds één antwoord, dat het beste past.

1. Welke uitspraak met betrekking tot neurale netwerken is **NIET** waar?
 - A. Neurale netwerken imiteren (tot op zekere hoogte) het menselijk brein.
 - B. Neurale netwerken leren van voorbeelden.
 - C. De trainingstijd van een neuraal netwerk is *onafhankelijk* van de grootte van het netwerk.
 - D. Artificiële neuronen werken anders dan biologische neuronen.

2. Welke uitspraak beschrijft een eigenschap van **truncation selection** (de selectiemethode voor evolutionaire algoritmen waarin de top τ % uit een populatie wordt geselecteerd en elk individu $100/\tau$ keer wordt opgenomen in de volgende generatie)?
 - A. Selectie van individuen gebeurt op basis van absolute fitness in plaats van relatieve fitness.
 - B. Het selectieproces convergeert snel voor lage waarden van τ .
 - C. Selectie is makkelijk parallel te doen.
 - D. Deze selectiemethode wordt in de literatuur als min of meer de standaard gezien.

3. In de kansrekening geldt voor 2 **niet onafhankelijke** proposities a en b dat:

- A. $P(a \wedge b) = P(a) - P(b) + P(a \vee b)$
- B. $P(a \wedge b) = P(a) \cdot P(b)$
- C. $P(a \wedge b) = P(a | b) \cdot P(b) + P(a | \neg b) \cdot P(\neg b)$
- D. $P(a \wedge b) = P(b | a) \cdot P(a)$

4. Beschouw een **fuzzy set** A en zijn support, core en alpha-cut. Welke relatie geldt er voor de drie operatoren $\text{supp}(A)$, $\text{core}(A)$, en A_α met $\alpha \in \langle 0, 1 \rangle$?

- A. $\text{supp}(A) \subseteq A_\alpha \subseteq \text{core}(A)$
- B. $\text{core}(A) \subseteq A_\alpha \subseteq \text{supp}(A)$
- C. $A_\alpha \subseteq \text{supp}(A) \subseteq \text{core}(A)$
- D. $\text{core}(A) \subseteq \text{supp}(A) \subseteq A_\alpha$

5. De '**Sussman Anomaly**' illustreert een zwakte van welke type planners?

- A. lineaire planners
- B. klassieke planners
- C. deterministische planners
- D. 'interleaved' planners

6. Welke uitspraak geldt voor een (**finite**) **state machine** (FSM)?

- A. In elke toestand zijn typisch alle acties beschikbaar.
- B. Een toestand representeert *gedrag* in plaats van een (*statische*) *wereld*.
- C. Een state machine is alleen 'finite' als er een 'terminal state' aanwezig is.
- D. Beloningen worden toegekend tijdens transities.

7. Wat is een nadeel van het gebruik van **separation cones** (kegels) voor het voorkomen van botsende AI-karakters ('collision avoidance')?

- A. Een karakter kan vast komen te zitten in een hoek ('corner trap').
- B. Elkaar passerende karakters vermijden elkaar onvoldoende.
- C. Het gebruik van de kegels kan niet altijd een botsing voorkomen.
- D. De 'cone-check' (zitten karakters in elkaars kegel?) is computationeel zwaar.

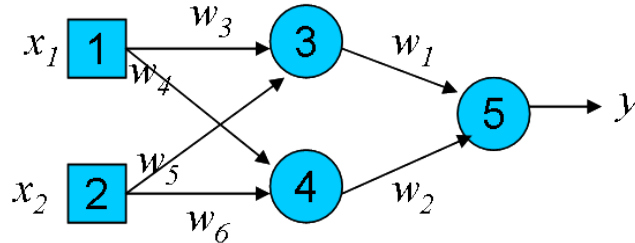
8. Welke uitspraak met betrekking tot **emergent cooperation** in groepen is waar?
- A. Bij 'emergent cooperation' wordt individueel gedrag bepaald door het groepsgedrag.
 - B. Bij 'emergent cooperation' bepaalt een *leider* het gedrag van de individuen.
 - C. Bij 'emergent cooperation' is het groepsgedrag moeilijk te voorspellen uit de regels van de individuen.
 - D. 'Emergent cooperation' is een eenvoudige manier om zeer intelligent-ogend groepsgedrag te bewerkstelligen.

9. Een **POMDP** is een generalisatie van een MDP waarbij
- A. Beloningen niet direct observeerbaar zijn.
 - B. Acties niet direct observeerbaar zijn.
 - C. Toestanden niet direct observeerbaar zijn.
 - D. Observaties slechts in sommige 'timesteps' beschikbaar zijn.

10. Beschouw een **statische wereld waarin alles bekend, compleet voorspelbaar en observeerbaar is**. Welke van onderstaande (type) planners lijkt het meest geschikt om planningproblemen in deze wereld op te lossen?
- A. FSM: finite state machine
 - B. MDP: Markov decision process
 - C. SharedPlan
 - D. HTN: Hierarchical task network

Deel II

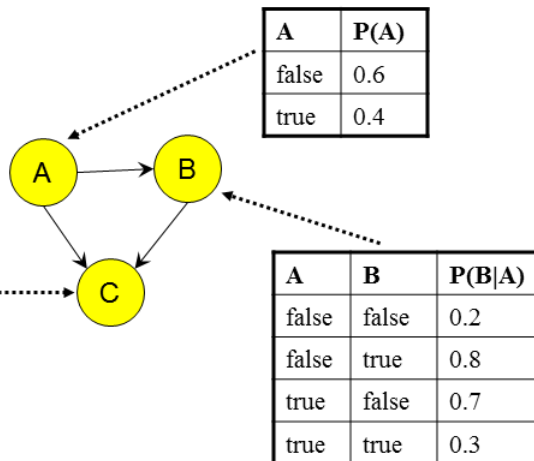
Opgave 1 (10 ptn.) Beschouw het volgende neurale netwerk, waarbij x_1 en x_2 binaire inputs zijn, $w_1 = w_4 = w_6 = 1$, $w_3 = w_5 = -1$, en $w_2 = 2$. Knoop 3 gebruikt de *sign function* met drempelwaarde $\theta_3 = -\frac{1}{2}$ als activatiefunctie; knopen 4 en 5 gebruiken daarvoor de *step function* met de volgende drempelwaarden: $\theta_4 = 1\frac{1}{2}$ voor knoop 4, en $\theta_5 = 0$ voor knoop 5.



Het neurale netwerk implementeert een logische operator $OP(x_1, x_2)$. Welke operator is OP ?

Opgave 2 (10 ptn.)

A	B	C	P(C A,B)
false	false	false	0.4
false	false	true	0.6
false	true	false	0.9
false	true	true	0.1
true	false	false	0.8
true	false	true	0.2
true	true	false	0.5
true	true	true	0.5



A	P(A)
false	0.6
true	0.4

A	B	P(B A)
false	false	0.2
false	true	0.8
true	false	0.7
true	true	0.3

Bereken uit bovenstaand Bayesiaans netwerk de volgende kans: $P(A = true \mid C = false)$; geef tenminste 2 cijfers achter de komma.

$$P(A = true \mid C = false) =$$

Opgave 3 (10 ptn.)

Stel je ontwerpt een koffieserveerrobot 'Douwe' voor de Uithof, op basis van een FSM. Om Douwe te testen beperk je het koffie rondbrengen tot 6 gebouwen. Bovendien gebruik je uitsluitend losse bits om (draadloos) een plan aan Douwe door te geven: dat wil zeggen, Douwe krijgt een '0' of een '1' door en gaat dan koffie serveren op de bestemming, zoals hieronder gespecificeerd:

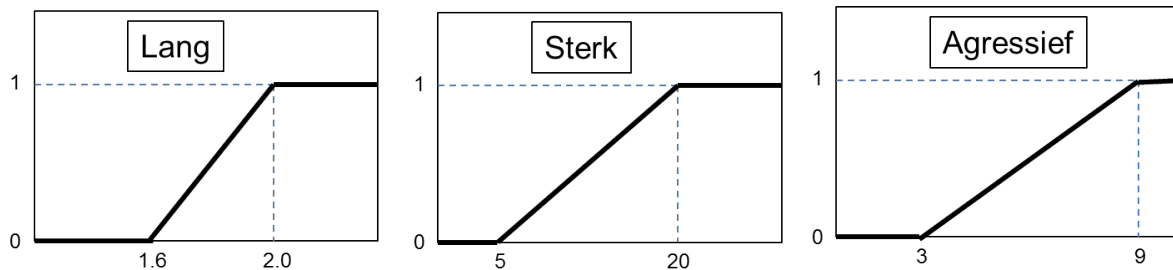
BBG:	als '0', blijf in BBG	als '1', ga naar MIN
MIN:	als '0', ga naar UNNIK	als '1', ga naar KRUYT
UNNIK:	als '0', blijf in UNNIK	als '1', ga naar RUPP
KRUYT:	als '0', ga naar WENT	als '1', ga naar BBG
RUPP:	als '0', ga naar KRUYT	als '1', ga naar MIN
WENT:	als '0', ga naar UNNIK	als '1', blijf in WENT

1) Teken het toestandsdiagram voor de FSM. 2) Als Douwe start in het BBG en voortaan alleen een reeks '01'-en (dwz 01010101...) krijgt, waar zal hij dan **nooit** koffie serveren?

Opgave 4 (15 ptn.) Beschouw een game waarin de NPCs (non-player characters) gebruikmaken van een regelgebaseerd systeem om te bepalen of ze de speler wel of niet aan gaan vallen. Het systeem bestaat uit twee if-then regels met een fuzzy preconditionie en een actie als gevolg. Of de actie daadwerkelijk uitgevoerd wordt hangt allereerst af van de membershipwaarde van de (volledige) preconditionie. Het NPC beschikt over de volgende regels:

- r1: IF Zelf = Sterk AND Zelf = NOT(Lang) THEN Aanvallen
- r2: IF Zelf = Lang OR Zelf = Agressief THEN Aanvallen

waarbij de fuzzy sets als volgt zijn gedefinieerd:



Er zijn nu twee implementatie opties om op basis van deze regels een besluit te nemen:

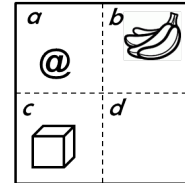
- I Val aan als voor tenminste 1 van de regels de membershipwaarde van de preconditionie > 0.6 is.
- II Val aan als het gemiddelde van de membershipwaarden van beide preconditionies > 0.5 is.

Beschouw nu een NPC met een lengte van 1.9, een kracht van 15 en agressie van 5. Geef voor zowel implementatie I als II aan of de NPC zal aanvallen, op welke regels en membershipwaarden je deze conclusie baseert, en welke definities je hierbij voor de operatoren AND, NOT en OR hebt gebruikt.

Opgave 5 (15 ptn.) Beschouw een aap in een kamer waar bananen aan het plafond hangen. De aap kan alleen bij de bananen als er een doos onder staat en hij daar op klimt. Initieel staan de aap en de doos op de grond, de aap in lokatie a en de doos op lokatie c ; de bananen hangen bij lokatie b . De aap kan zelf verplaatsen, de doos duwen, op de doos klimmen en bananen grijpen. Hoe komt de aap aan de bananen? Dit planningsprobleem kunnen we modelleren in STRIPS met de volgende begin- en doeltoestanden en acties:

Begintoestand: on(aap, vloer), on(doos, vloer), at(aap, a),
 at(bananen, b), at(doos, c), status(bananen, hangend)
Doeltoestand: status(bananen, gepakt)

Acties: go(X , Y); push(D , X , Y); climb_on(D); grab(B)



- 1) Geef een volledige specificatie van de vier acties, gebruikmakend van de STRIPS syntax.
- 2) Kan STRIPS een plan maken waarin de aap een doeltoestand bereikt waarin *tenminste* 'status(bananen, gepakt)' geldt? Zoja, geef het plan en de resulterende doeltoestand. Zonee, leg uit waarom STRIPS dat niet kan.

Opgave 6 (10 ptn.)

Onderstaande tabel laat een populatie van strings zien. Iedere string staat voor een binair getal n met *fitness* $f = \frac{1}{2} \cdot n^2$.

Ouders worden op basis van *tournament selection* met $\sigma = 2$ uit de initiële populatie geselecteerd. Er heeft al 1 ronde van tournament selection plaatsgevonden op een (andere) random volgorde van de gegeven strings; de resulterende ouders zijn al gegeven.

Gebruik *tournament selection* tussen opeenvolgende string nummers (linkerkolom) om de overige ouders te selecteren. Genereer vervolgens een nieuwe populatie, die de oude vervangt. Gebruik *cross-over* voor recombinatie; *mutatie* gebeurt alleen op het middelste bit (met kans 1). Voor de eerste twee paren is het crossover punt aangegeven (met '!'); het laatste paar heeft een crossover punt tussen het tweede en derde bit van **links**.

Vul onderstaande tabel in, bereken de totale fitness in de nieuwe populatie en concludeer op basis daarvan of de nieuwe generatie beter is dan de oude.

String nr.	Initial	f	Parents	Offspring	f
1	00010	2	0!1100		
2	10101		1!0101		
3	00110		100!11		
4	10011				
5	01100				
6	10001				