Modeluitwerking Tentamen Computationele Intelligentie
Universiteit Leiden – Informatica
Vrijdag 11 Januari 2013

Opgave 1. Python

Gegeven is de volgende (slechte) Python code:

```
def t(x):
  def p(y):
    return x*y
  return p

def o(n):
  for i in range(0,n):
    if i % 2 == 1:
      yield i

def l(v,p):
  l2 = 0
  for i in v:
    l2 += p(i)
  return l2

def f(n):
  return l(list(o(n)),t(2))
```

a. Geef de uitvoer van de functie-aanroep `l([1,2,3],lambda x:x+2)`; beschrijf in voldoende detail hoe je aan deze uitvoer komt.

```
[1,2,3] -> [3,4,5] -> 12
```

b. Geef de uitvoer van de functie-aanroep `f(5)`; beschrijf in voldoende detail hoe je aan deze uitvoer komt.

```
list(o(5)) = [1,3]
l(list(o(5)),t(2)) = 2 + 6 = 8
```

c. Herschrijf functie `l` zodanig dat deze gebruik maakt van *list comprehension.*

```
def l(v,p):
  return sum([p(i) for i in v])
```

d. Herschrijf functie `f` zodanig dat deze dezelfde uitvoer geeft, slechts één for lus bevat en geen functieaanroep anders dan `range(0,n)`.

```
def f(n):
  l2 = 0
  for i in range(0,n):
    if i % 2 == 1:
      l2 += i * 2
  return l2
```

Opgave 2. Logica

Bewijs door middel van natuurlijke deductie:

a. $\neg(p \vee q) \models \neg p$

1. $\neg(p \vee q)$
2. $|p$ (assumption)
3. $|p \vee q$ ($\vee$ introduction, 2)
4. $|\bot$ ($\neg$ elimination, 1, 3)
5. $\neg p$ ($\neg$ introduction, 2-4)

b. $(q \vee p) \wedge \neg p \models q$

1. $(q \vee p) \wedge \neg p$
2. $(q \vee p)$ ($\wedge$ elimination 1,2)
3. $\neg p$ ($\wedge$ elimination 1,1)
4. $|q$ (assumption)
5. $|q$ (copy 4)
6. $|p$ (assumption)
7. $|\bot$ ($\neg$ elimination 3,6)
8. $|q$ ($\bot$ elimination, 7)
9. q ($\vee$ elimination 2, 4-5, 6-8)

Zet de volgende formule om in CNF; schrijf ze daarbij zo simpel mogelijk:

c. $((p \to q) \to r) \vee (q \to p)$

$$\neg((\neg p) \vee q) \vee r \vee \neg q \vee p = (p \wedge \neg q) \vee r \vee \neg q \vee p = (p \vee r \vee \neg q) \wedge (\neg q \vee r \vee p) = p \vee r \vee \neg q$$

Geef een interpretatie over universum $\mathcal{U} = \{1, 2\}$ die de volgende formule over predicaatsymbolen $\mathcal{P} = \{P/2, Q/1\}$ en functiesymbool $\mathcal{F} = \{a/0\}$ waar maakt:

d. $\exists x(P(x, a) \wedge Q(x))$

$a_I = 1$
$P_I = \{(1, 1)\}$
$Q_I = \{(1)\}$

Opgave 3. Solvers

a. Een probleem dat je met een SAT solver op kunt lossen is het Hamilton-pad probleem. Je kunt dit bijvoorbeeld doen door booleaanse variabelen x_{ij} te gebruiken die aangeven of in stap i knoop j bezocht wordt. Beschrijf hoe de CNF formule eruit ziet waarmee je het Hamilton-pad probleem oplost, gebruikmakend van deze variabelen.

We nemen aan dat n het aantal knopen is. Deze formule bestaat uit verschillende clauses:

- clauses die afdwingen dat voor alle i er minimaal één x_{ij} waar is (voor alle i met $i \in \{1, \ldots, n\}$: $x_{i1} \vee \cdots \vee x_{in}$) (voor elke stap is er tenminste één knoop die bezocht wordt)
- clauses die afdwingen dat voor alle j er minimaal één x_{ij} waar is (voor elke knoop is er tenminste één stap waarin deze bezocht wordt)
- clauses die afdwingen dat voor alle i er geen twee x_{ij} en $x_{ij'}$ met $j \neq j'$ waar zijn (voor alle $i, j \neq j' : \neg x_{ij} \vee \neg x_{ij'}$)
- clauses die afdwingen dat voor alle j er geen twee x_{ij} en $x_{i'j}$ met $i \neq i'$ waar zijn

- clauses die afdwingen dat knopen alleen opeenvolgend in een kring kunnen voorkomen als deze met elkaar verbonden zijn (voor alle knopen i en j waar geen boog tussen bestaat: $\neg x_{ki} \vee \neg x_{(k+1)j}$, waarbij $k \in \{1, \ldots, n\}$)

In principe is het goed als deze vijf elementen genoemd worden, ook als de formules niet volledig gegeven zijn.

Merk op dat in woorden beschreven mag worden voor welke variabelen een clause gebouwd wordt, maar dat de clause zelf wel formeel gegeven moet worden.

b. Los het volgende integer lineair programma op.

$$\begin{array}{rl} \text{maximaliseer} & 2x - 3y \\ \text{waar} & x + y \leq 10 \\ & x \leq 5 \\ & 0 \leq x, y \in \mathbb{N} \end{array}$$

Het optimum is $x = 5, y = 0$ en heeft waarde 10. Omdat dit probleem slechts twee variabelen heeft kan dit ook in een 2D tekening aannemelijk gemaakt worden.

c. Geef twee verschillen tussen integer lineair programmeren en constraint programming.

- integer lineair programmeren ondersteunt alleen lineaire contraints; CP kent globale constraints (zoals alldifferent constraints)
- ILP maakt gebruik van interior point of simplex algoritme; deze worden in CP (meestal) niet gebruikt
- in ILP kan ook gebruik gemaakt worden van reële variabelen; in CP is dit meestal niet het geval.

Opgave 4. Fuzzy Logic

Gegeven zijn de volgende twee Fuzzy sets over domein $X = \{1, 2, 3, 4\}$: $A = \sum_{x \in X} \mu_A(x)/x$ en $B = \sum_{x \in X} \mu_B(x)/x$ met $\mu_A(x) = (4 - x)/3$ en $\mu_B(x) = (x - 1)/3$.

a. Teken deze fuzzy sets en geef de *support* van deze fuzzy sets.

De support van A is $\{1, 2, 3\}$ en van B is dat $\{2, 3, 4\}$

b. Gegeven is de volgende *crisp* set: $C = \{a, b\}$ over het universum $\{a, b, c\}$. Met behulp van deze verzameling maken we het volgende Mamdani systeem:

als x is A en x is B, dan y is C

Bepaal de membership functie van de fuzzy relatie tussen x en y zoals berekend door dit systeem.

$x \downarrow$	a	b	c	$\leftarrow y$
1	0	0	0	
2	$\frac{1}{3}$	$\frac{1}{3}$	0	
3	$\frac{1}{3}$	$\frac{1}{3}$	0	
4	0	0	0	

Opgave 5. Evolutionaire Algoritmen

a. Geef een voorbeeld van een genotype voor de onderstaande problemen:

- een evolutionary strategy voor het vinden van het minimum van een functie $f(x, y, z)$;
- genetisch programmeren voor het vinden van een functie van x naar y.

Geef een korte motivatie.

- een vector van reële getallen; evolutionaire algoritmen werken in het simpelste geval op een vector reële getallen; bijvoorbeeld $(0.0, 1.0, 2.0)$ waar opeenvolgend x, y en z gegeven zijn
- een boom waarin de interne knopen operatoren zijn en de bladeren x, y of constanten; genetisch programmeren werkt doorgaans op dit soort bomen. Bijvoorbeeld $+(+(x, y), y)$ (gebruikmakend van prefix notatie).

b. Geef de Gray-code voor de binaire string 0101110.

0111001

c. In evolutionary strategies wordt gebruik gemaakt van *strategy parameters*. Beschrijf wat deze zijn en waarvoor ze gebruikt worden.

Strategy parameters zijn (meestal) mutatieparameters die in elk individu in een populatie worden opgenomen; deze parameters bepalen in welke mate de getallen in een individu gemuteerd worden. Als bijvoorbeeld elk getal in een individu gemuteerd volgens een normaalverdeling, kan de variantie van deze normaalverdeling als parameter in het individu worden opgenomen. Het idee is dat evolutie vervolgens ook gebruikt kan worden om de juiste strategieparameters te bepalen.

Opgave 6. Swarm Intelligence

a. Pas twee iteraties van het standaard algoritme voor Ant Colony Optimisation toe om het kortste pad te vinden van knoop A naar knoop D in de volgende grafe, waarbij voor elke boog zowel de afstand (getal 1) als de pheromonen (getal 2) gegeven zijn. Neem aan dat $\alpha = Q = 1$ (parameters in de kansberekening); $\rho = 0.2$ (verdamping); er zijn 2 mieren. Geef duidelijk tussenstappen aan; geef met name aan met welke kansen mieren bogen kiezen en welke bogen ze uiteindelijk nemen.

$$\begin{tikzcd} A \arrow[r, "{(1,2)}"] \arrow[d, "{(2,1)}"'] & B \arrow[d, "{(1,0.5)}"] \\ C \arrow[r, "{(2,3)}"'] & D \end{tikzcd}$$

We nemen aan dat beide mieren het pad $A \to B \to D$ bewandelen en zowel de eerste als de tweede keer ditzelfde pad kiezen.

In de eerste ronde zijn dit de kansen waarmee beide mieren de bogen op dit pad kiezen:

- $A \to B$ met kans $2/3$, $A \to C$ met kans $1/3$;
- $B \to D$ met kans 1

De lengte van dit pad is 2. De gewichten worden als volgt aangepast:

- $A \to B : 0.8 \cdot 2 + 1/2 \cdot 2 = 2.6$
- $B \to D : 0.8 \cdot 0.5 + 1/2 \cdot 2 = 1.4$
- $A \to C : 0.8 \cdot 1 = 0.8$
- $C \to D : 0.8 \cdot 3 = 2.4$

In de tweede ronden zijn dit de kansen waarmee beide mieren de bogen op het pad kiezen:

- $A \to B$ met kans $2.6/(2.6 + 0.8) \approx 0.76$, $A \to C$ met kans ≈ 0.24;

- $B \to D$ met kans 1

De gewichten worden als volgt aangepast:

- $A \to B : 0.8 \cdot 2.6 + 1/2 \cdot 2 \approx 3.1$
- $B \to D : 0.8 \cdot 1.4 + 1/2 \cdot 2 \approx 2.1$
- $A \to C : 0.8 \cdot 0.8 = 0.64$
- $C \to D : 0.8 \cdot 2.4 = 1.92$

b. Beschrijf het verschil tussen *synchrone* en *asynchrone* particle swarm optimization.

Particle swarm optimisation algoritmes lossen optimalisatieproblemen op. Particles stellen hierbij mogelijke oplossingen voor. Nabijgelegen deeltjes beïnvloeden elkaar. By *synchrone* PSO worden de posities en snelheden van alle deeltjes "tegelijk" aangepast op basis van de posities en snelheden van de andere deeltjes in de vorige tijdstap. By *asynchrone* PSO worden de snelheden en posities individueel aangepast en vinden de updates niet voor alle deeltjes tegelijk plaats.