

Beoordeling: 20 punten per vraag

Opgave 1: Basisbegrippen

- a. Wat is het verschil tussen de **taak** en het **domein** van een te bouwen kennissysteem? Geef twee voorbeelden van het hergebruik van onderdelen van een kennissysteem, waarin gebruik gemaakt wordt van dit onderscheid.

De taak is de activiteit van de gebruiker, waarbij het systeem behulpzaam moet zijn, of de activiteit die het systeem zelf moet uitvoeren. Het domein is het toepassingsgebied, met aparte achtergrondkennis.

Vb 1. diagnose van kapotte auto's – design van auto's (zelfde domein, andere taak)

Vb2. diagnose van kapotte auto's – diagnose van zieke mensen (zelfde taak; ander domein) (slide 1.9).

- b. Volgens Newell (1982) is kennis datgene wat we toeschrijven aan een agent, zodat we zijn gedrag kunnen begrijpen met behulp van het 'principe van rationaliteit'. Wat is het **principe van rationaliteit**?

"If an agent has knowledge that one of its actions will lead to one of its goals, then the agent will select that action (slide 1.13)."

Het gedrag van een agent wordt verklaard door wat ie (blijkbaar) wil (goals), en welke acties die kan uitvoeren. Als er een actie is die tot een goal leidt, moet de agent die ook uitvoeren, anders is ie irrationeel.

Opgave 2: Kennisacquisitie

- a. Wat is **participerend ontwerpen** (participatory design)?

Dat is een vorm van ontwerpen, waarbij mensen in verschillende rollen (stakeholders) vanaf het begin bij het ontwerp zijn betrokken (participeren). (Stefik, p 301)

- b. Noem minstens vier soorten **belanghebbenden** (stakeholders) bij het ontwerpproces.

Wat is hun rol in het ontwerpproces? Wat is hun primaire belang?

<i>opdrachtgever:</i>	<i>- geeft opdracht; betaalt;</i>
	<i>- een winstgevend systeem ; een goedkoop systeem</i>
<i>eindgebruiker:</i>	<i>- gebruikt systeem dagelijks ; stuurt ontwerp proces bij</i>
	<i>- een bruikbaar systeem (= nuttig & gebruiksvriendelijk</i>
<i>kennisingenieur</i>	<i>- bouwt systeem; verzameld kennis</i>
	<i>- tevreden klant; weinig onderhoud; technische uitdaging</i>
<i>domein expert</i>	<i>- brengt kennis in</i>
	<i>- wordt betaald voor kennis</i>

Opgave 3: Productieregels

- a. Wat is **conflict resolutie**? Geef een korte beschrijving van vijf conflict resolutie strategieën.

Een productie systeem werkt door het steeds herhalen van drie stappen. Conflict resolutie wordt gebruikt bij het selecteren van een regel in de tweede stap.

1. verzamel alle regels die toepasbaar zijn t.o.v. 'working memory' (WM); deze verzameling heet de conflict set
2. selecteer een regel uit de conflict set dmv een conflict resolutie strategie
3. voer de regel uit; hierdoor verandert WM.

1. selecteer de **eerste** regel (boven-beneden; links-rechts)
2. selecteer de **meest specifieke** regel (meeste eisen in de Condities)
3. selecteer de regel met hoogste **prioriteitswaarde**
4. selecteer de regel, die past op de **meest recent** toegevoegde feiten aan WM
5. selecteer de regel, met de hoogste **zekerheidsfactor**.

Andere mogen ook (zie slide 4.36)

- b. Een van de voordelen van productieregels, is dat ze kennis inzichtelijk (transparent) maken, ook voor de eindgebruiker. Leg uit, hoe je met behulp van productieregels en forward en backward chaining op zogenaamde "**How?**" en "**Why?**" vragen antwoord kunt geven.

Een 'how' vraag, vraagt naar de afleiding van een bepaald antwoord van het systeem.

Een zogenaamde 'backward trace' vormt zo'n antwoord. Van de conclusie wordt uitgelegd welke regels gebruikt zijn om die af te leiden, en welke regels gebruikt zijn om de tussenresultaten af te leiden, net zo lang tot je bij gegeven feiten uitkomt.

U: How?

S: **by** rule 10 Fruit = Orange, if Fruitclass = tree &
Color=orange & seedclass=multiple,
and by rule 3 Fruitclass = tree, if Shape = round and Diam < 4,
and by rule 5 Seedclass = multiple, if Seedcount > 1,
and Color=orange, Shape = round, Diam = 1.5, seedcount=10 are **given**.

Een 'why' vraag, vraagt naar de reden waarom een bepaalde vraag gesteld wordt aan de gebruiker. Het antwoord is een 'forward trace' dat loopt van het mogelijke antwoord op de vraag die gesteld is aan de gebruiker, via de gebruikte regels, naar het einddoel.

U: What kind of fruit is this?

(goal: fruit = X ?)

S: Is the shape of the fruit round?

(question: Shape=round ?)

U: Why?

S: **Because by** rule 3 if its shape is round and its diameter is smaller than 4, then it grows on a tree,

and by rule 10 if it grows on a tree and its color is orange and it produces many seeds, then it is an Orange. (linguistic output by NL generator)

So, is the shape of the fruit round?

Opgave 4: Tijd en Ruimte

De volgende tabel geeft een overzicht van Allen's (1984) relaties tussen tijdsintervallen.

Relation	Condition	Symbol
X before Y	$X^+ < Y^-$	$<$
X equals Y	$X^- = Y^-, X^+ = Y^+$	$=$
X meets Y	$X^+ = Y^-$	m
X overlaps Y	$X^- < Y^-, X^+ > Y^-, X^+ < Y^+$	o
X during Y	$(X^- < Y^-, X^+ \leq Y^+) \text{ OR } (X^- \geq Y^-, X^+ < Y^+)$	d
X starts Y	$X^- = Y^-, X^+ < Y^+$	s
X finishes Y	$X^+ = Y^+, X^- < Y^-$	f

Tabel 1: Allen's zeven interval relaties

Allen's tijdsrekening maakt gebruik van een compositietabel, die het mogelijk maakt twee relaties R en S te combineren tot een nieuwe relatie, op grond van de beschikbare informatie. Een deel van deze compositietabel staat hieronder.

$R \setminus S$	$<$	$>$	d	m
$<$	$<$	any	$<, o, m, d, s$	$<$
$>$	any	$>$	$>, mi, o, d, f$	$>, oi, mi, d, f$
d	$<$	$>$	d, fi, si	$<, m$
m	$<$	$>, oi, mi, d, s$	$m, <$	$<$

a. Vul de ontbrekende delen van de tabel aan op de plaats van de '...'.

Let op! Er zat een foutje in de definitie van during. Het moet zijn $(X^- > Y^-, X^+ \leq Y^+)$ OR $(X^- \geq Y^-, X^+ < Y^+)$. Ik heb er wel rekening mee gehouden.

Een tip: maak tekeningetjes. dan kun je het vrij eenvoudig aflezen.

$< >$	any
$< m$	$<$
$> d$	$>, mi, o, d, f$ (mi is inverse van m)
$> m$	$>, oi, d, f$ (oi is inverse van o)
$d >$	$>$
$d d$	d, fi, si (fi, si is inverse van f, s resp.)
$m <$	$<$
$m >$	$>, oi, mi, d$
$m d$	$m, <$
$m m$	$<$

- b. Wat is het verschil tussen de begrippen **linear time** en **branching time**? Welk tijdsbegrip zou jij kiezen voor het bouwen van een ‘surveillance robot’?

Onder linear time verstaan we het tijdsbegrip dat uitgaat van één mogelijk verleden, en één mogelijke toekomst. Bij branching time worden er meer toekomsten mogelijk geacht (zgn mogelijke werelden). Een ‘surveillance robot’ (zie slides Frans Voorbraak) moet plannen wat ie gaat doen. In die plannen moet ie rekening kunnen houden met verschillende routes, of verschillende uitkomsten van zijn acties (onzekerheid). We kiezen dus voor branching time.

Opgave 5: Onzekerheid en Vaagheid

- a. Een doctor weet dat de ziekte ‘meningitis’ (M) in 50% van de gevallen een stijve nek (S) veroorzaakt.

$$P(S|M) = 1/2$$

De kans dat een patiënt überhaupt meningitis krijgt is ongeveer 1 op 50.000.

$$P(M) = 1/50.000$$

De kans dat een patiënt een stijve nek krijgt is ongeveer 1 op 20.

$$P(S) = 1/20.$$

Bereken de **voorwaardelijke kans** op meningitis, gegeven dat een patiënt een stijve nek heeft.

$$\text{Gevraagd: } P(M|S)?$$

*Regel van Bayes: $P(M|S) = P(S|M) P(M) / P(S) = (1/2 * 1/50.000) / (1/20) = 20/100.000 = 1/5000$. Dat is een stuk minder dan je zou verwachten op grond van de correlatie S-M.*

- b. Gegeven is de volgende ‘fuzzy’ definitie van het begrip ‘rijk’. Laat zien in de tekening hoe je met behulp van een ‘**hedge**’ het begrip ‘een beetje rijk’ kunt uitdrukken.

Een ‘hedge’ is een functie die een begrip kan veranderen in een ander begrip. In het college (slide 42) stond hoe ‘very’ de fuzzy definitie naar rechts schoof. De hedge ‘een beetje’ schuift het begrip juist naar links.

