

Antwoorden Tentamen Representatie & Zoeken
15 Januari 1996

Vraag 1

Vraag 1a,b,c

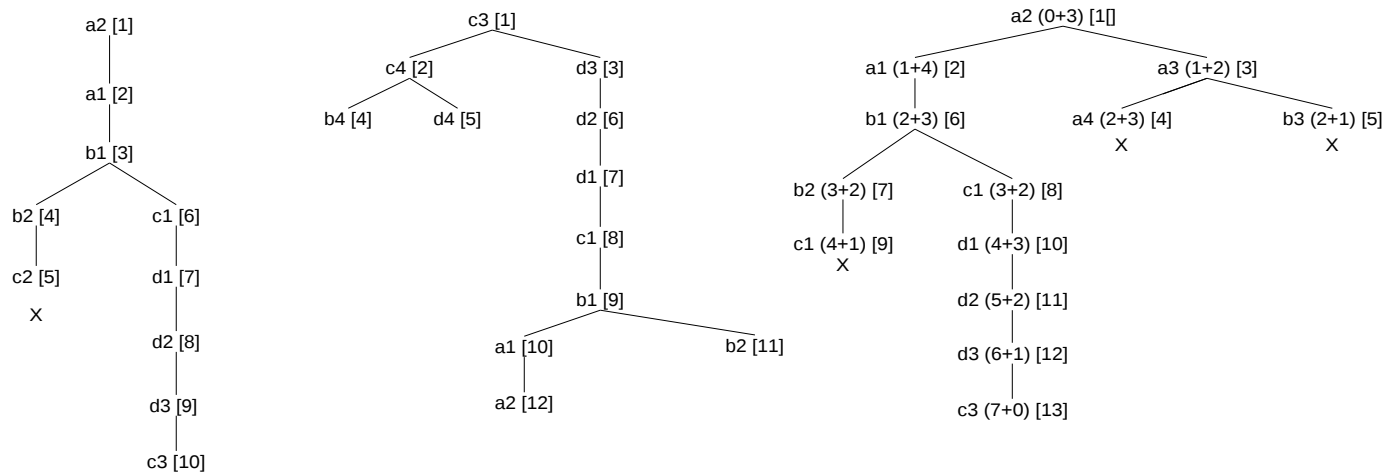


Figure 1: Vraag 1a,b,c

Vraag 1d

1. : a2
2. : a2, a1
3. : a2, a1, b1
4. : a2, a1, b1, b2, c1

Vraag 2

Vraag 2a

$\neg a(X_1, Y_1) \vee b(X_1)$
 $c(X_2) \vee \neg b(X_2) \vee \neg d(X_2)$
 $\neg c(X_3) \vee b(X_3) \vee \neg d(X_3)$
 $a(2, 1)$
 $d(2)$

Vraag 2b

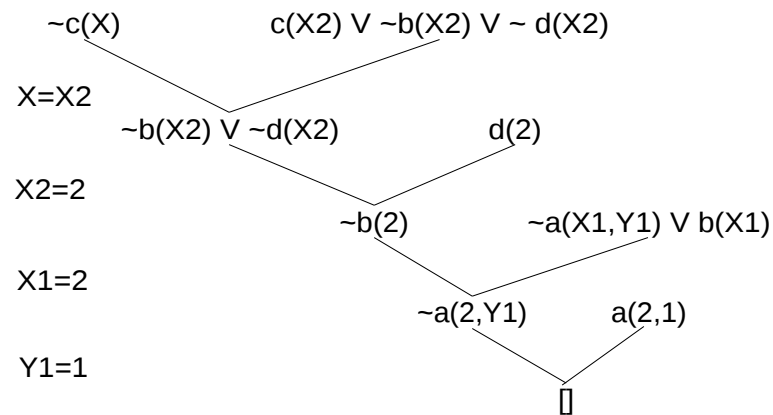


Figure 2: Vraag 2b

Vraag 3

Vraag 3a

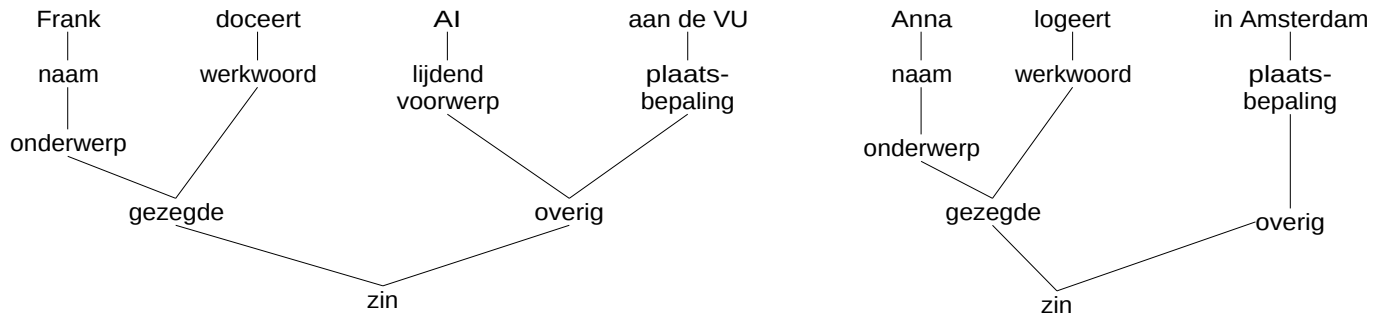


Figure 3: Vraag 3a

Vraag 3b

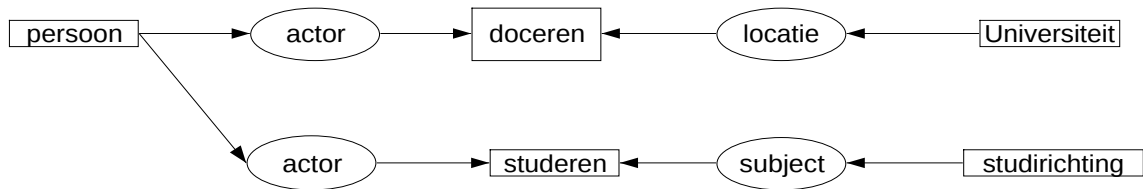


Figure 4: Vraag 3b

Een voorbeeld van een semantisch onzinnige maar syntactisch korrekte zin is:

“Anna logeert AI aan de VU”

Vraag 3c

Voeg de volgende regel toe aan de grammatica:

`zin = gezegde overig en zin`

Vraag 3d

getal	=	1-9 10-19 20-99 100-999
1-9	=	een 2-9
2-9	=	twee drie ... negen
10-19	=	tien elf ... negentien
20-99	=	1-9 en tiental
20-99	=	tiental
tiental	=	twintig dertig ... negentig
100-999	=	hondertal
100-999	=	hondertal 1-9
100-999	=	hondertal 10-19
100-999	=	hondertal 20-99
hondertal	=	honderd
hondertal	=	2-9 honderd

Vraag 4

Vraag 4a

De waarden zijn $p = 1$ en $q = 0$.

Vraag 4b

Kies $\delta = 0.1$. Na toepassing van de δ -regel hebben alle links naar p waarde 0.1, alle links naar q hebben waarde 0.3, behalve de link $d - -q$, die nog steeds 0.2 als waarde heeft.

Vraag 5

Invoer: Het Version-Space algoritme neemt als invoer een partieel geordende ruimte van concept-beschrijvingen (geordend naar algemeenheid van de beschreven concepten), plus een serie positieve en negatieve voorbeelden van het te leren concept.

Uitvoer: De uitvoer van het algoritme bestaat uit twee concept-beschrijvingen: een zo algemeen mogelijk en een zo specifiek mogelijk concept dat alle positieve voorbeelden beschrijft, en geen van de negatieve voorbeelden. In het ideale geval vallen deze concepten samen. Het kan ook voorkomen dat geen enkel concept kan worden gevonden dat aan deze eisen voldoet.

Stappen: Het algoritme begint met het meest algemene en meest specifieke concept als kandidaten. Bij aanbieding van een positief voorbeeld dat niet onder de huidige meest specifieke kandidaat valt, wordt deze kandidaat zo min mogelijk gegeneraliseerd om te zorgen dat dit wel het geval is, zonder oude negatieve voorbeelden te gaan beschrijven. Evenzo, bij aanbieding van een negatief voorbeeld dat wel onder de huidige meest algemene kandidaat valt, wordt deze kandidaat zo min mogelijk specifieker gemaakt, zonder oude positieve voorbeelden niet meer te beschrijven.

Terminatie: Het algoritme stopt als de verzameling voorbeelden is uitgeput, of als de meest algemene kandidaat specifieker geworden is dan de meest specifieke kandidaat. In dat geval bestaat er geen geldige concept beschrijving die aan de eisen voldoet.

Vraag 6

Invoer: De invoer van het ID3 algoritme is een verzameling voorbeelden, die elk bestaan uit een aantal attributen met de waardes voor die attributen, plus de klasse waartoe het voorbeeld behoort.

Uitvoer: De uitvoer is een beslis-boom waarin iedere knoop een gegeven instantie naar een van zijn kinderen stuurt op basis van een test op een bepaalde attribuut-waarde. Bij de bladeren staat aangegeven tot welke klasse de gegeven instantie behoort als het bij dat blad is aangekomen.

Stappen: Het ID3 algoritme kiest als top-knoop van de te construeren boom de attribuut-test die de voorbeeld-verzameling zo efficient mogelijk in delen opdeelt. Elke resulterende deelverzameling van de voorbeelden wordt nu bij de bijbehorende kind-knoop recursief weer zo efficient mogelijk opgedeeld op basis van een ander attribuut, net zo lang tot alle elementen van een deelverzameling tot nog slechts één klasse behoren. Die klasse is dan de label van het bijbehorende blad in de boom. De keuze van de attribuut-test om een verzameling voorbeelden zo efficient mogelijk op te delen werkt als volgt. ID3 splitst de

verzameling op basis van het attribuut met de hoogste informatie-winst. Informatie-winst van een attribuut-opsplitsing is gedefinieerd als het verschil tussen (i) de informatie-waarde van dat attribuut in de voorbeeldverzameling (gemeten in aantallen bits volgens de formule van Shannon) en (ii) de informatie die nog nodig is om de resterende deelverzamelingen verder op te splitsen.

Terminatie: Het ID3 algoritme stopt op het moment dat het, bij het recursief opsplitsen van de invoerverzameling in deelverzamelingen is aangekomen bij een deelverzameling die in zijn geheel tot één klasse behoort.