

Institutt for datateknologi og informatikk

Eksamensoppgave i TDT4137 Kognitive Arkitekturer

Faglig kontakt under eksamen: **Asbjørn Thomassen**

Tlf.: **90145710**

Eksamensdato: 15 desember 2017

Eksamenstid (fra-til): 0900-1300

Hjelpemiddelkode/Tillatte hjelpemidler: D/ Kalkulator

Annen informasjon:

Målform/språk: bokmål

Antall sider (uten forside): 6

Antall sider vedlegg: 0

Informasjon om trykking av eksamensoppgave

Originalen er:

1-sidig **2-sidig**

sort/hvit **farger**

skal ha flervalgskjema

Kontrollert av:

5/12-2017

Dato

Sign

Oppgave 1 (20%)

- a) Kognitive systemer kan deles opp i subklasser relatert til kognisjonsbegrepet (cognition). Gi navn på tre kognitive arkitekturer og plasser disse i tilhørende subklasse. Hvor hører systemer basert på nevralt nett hjemme? Vis deretter hvilke fagområde(r) hver av arkitekturerne i hovedsak er basert på.

- b) Hvilke sub-systemer med tilhørende prosessorer har vi i MHP (Model Human Processor)?

Forklar kort tre viktige basisoperasjoner i MHP som alle utføres av den kognitive prosessor og tar en syklus (cycle).

Et romersk totaltall (II) kommer opp på en skjerm. Like etterpå dukker tallet 2 også opp på skjermen:



Symbolet II kommer først opp på skjermen (romertall for 2)



Deretter kommer 2 opp ved siden av romertallet

Hvor lang tid i MHP går det fra det andre symbolet, 2, kommer opp på skjermen og til en gjenkjenner det som samme tall?

- c) Hicks lov (P7 i MHP) for n objekter med lik sannsynlighet er beskrevet ved

$$T = I_c H \quad \text{hvor} \quad H = \log_2(n+1).$$

Hva er T , I_c og H og hva uttrykker loven?

- d) Hva er analogi-resonnering?

Hvordan representerer Gentner kunnskap for å kunne mappe target og base i analogi-resonneringen?

Forklar kort "The Systematicity Principle".

Oppgave 2 (20%)

- a) Forklar virkemåten til et kunstig nevron og analogien med et biologisk nevron med dendritter, soma, synapser og axoner.
- b) Skriv lærings-algoritmen for et perseptron i pseudo-kode.
Deltaregelen er:

$$\Delta w_i(p) = \alpha x_i(p) e_i(p)$$

Funksjonen $\text{step}(X)$ er definert: 0 hvis $X < 0$ og 1 ellers.

Hva er rollen til α ?

Anta at du har et datasett med to kategorier 0 og 1. Hvert data-punktene er på formen (x_1, x_2, y_d) hvor y_d er ønsket ut-verdi (kategori) for x_1 og x_2 . Det er totalt bare seks punkter i settet:

$$\begin{matrix} (0.2, 0.3, 1) & (0.4, 0.9, 0) & (0.7, 0.9, 1) \\ (0.8, 0.9, 1) & (0.4, 0.3, 0) & (0.7, 0.3, 0) \end{matrix}$$

Kan du ut fra å betrakte det oppgitte datasettet over se om perseptronet vil være i stand til å lære funksjonen (skille de to kategoriene 0 og 1)? Begrunn svaret.

Tips: Tegn opp datasettet i et 2D-aksesystem med x_1 og x_2 .

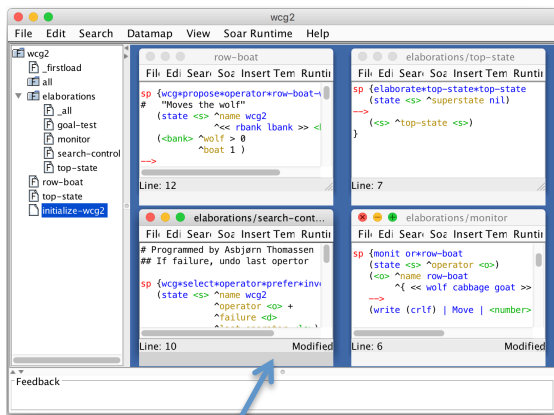
Bruk lærings-algoritmen i b) til å kjøre en iterasjon på første data-punkt i data-settet over.

Anta at vi har startet treningen av perseptronet med $\theta = 0.2$ og $\alpha = 0.1$ og har følgende verdier på vektene : $w_1 = 0.2$ og $w_2 = 0.1$.

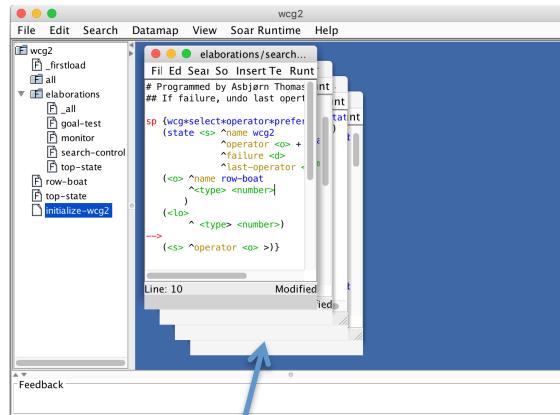
- c) Hva er tanken bak bruken av lokale reseptive felter (**local receptive fields**) i CNN (Convolutional Neural Networks)? Hva oppnår man ved å gjøre sub-sampling (pooling) etter et konvolusjons-lag (convolutional layer)?

Oppgave 3 (20%)

Når en programmerer produksjonsregler i Soar med VisualSoar, har en mulighet til å velge om en skal se fil-vinduene, dvs. koden, utlagt som "fliser" (tiled) eller i kaskade (cascade). Figuren på neste side viser de to alternativene:



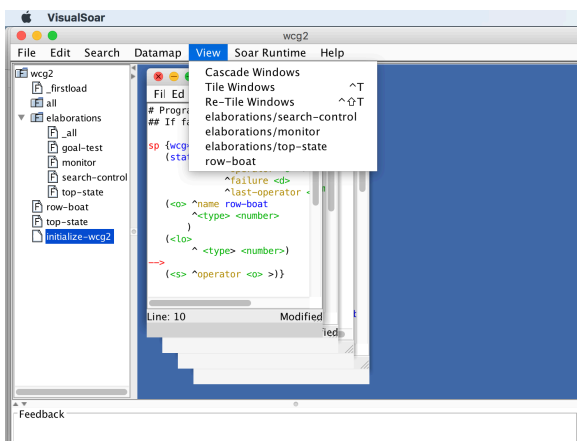
Alternativ 1: "Tile view" av fire vinduer med Soar-kode



Alternativ 2: "Cascade view" av de samme vinduene

For å se fil-vinduene som **Alternativ 1** kan en velge meny kommandoen `Tile Windows` under `View` i menylinja, eller bare bruke kortkommando `cmd t` ved å trykke ned **cmd**-tasten etterfulgt av tasten **t**.

Dersom en vil se fil-vinduene i kaskade, **Alternativ 2**, er det av ukjent grunn bare mulig å bruke menyen `Cascade Windows`. Figuren nedenfor viser `View`-menyen som bla. inneholder de to nevnte vindus-kommandoene:



- Modeller målet `selectSoarFileView` i NGOMSL som dekker de to nevnte vindus-alternativene. Generaliser meny metoden. Tegn målhierarkiet. Hva kan et dypt målhierarki tyde på?
- Beregn utførselstiden (execution time) og vis utførsel-trace med operatortider (du kan her notere på koden du skrev i a)). Hva uttrykker hovedleddene i utførselstid-formelen som du brukte?
- Hvilken type oppgaver kan ikke modelleres i GOMS? Hvorfor lar det seg gjøre å parallellisere noen av operatorene du brukte i a)? Gi et eksempel. Angi hvordan tiden beregnes når to eller flere step slås sammen.

Oppgave 4 (20%)

EMU er en emosjonell personlig agent som hjelper deg med å spille musikk fra ditt musikkbibliotek. EMUs grad av lykke vises frem som et ansikt og er basert på tiden siden du sist bad den spille noe for deg, `sinceTime`, og hvor lenge du lot den spille musikk, spilletidstreden `playtime` (gjennomsnittlig spilletid) . Nedenfor ser du to av de mange ansiktene EMU kan vise frem:



sad

-----andre ansikter i mellom-----

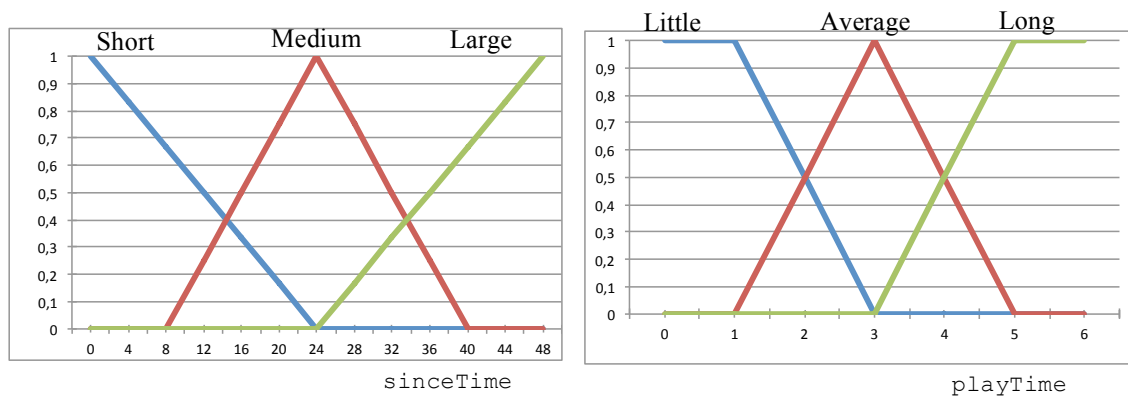


happy

Hver gang du spør EMU om å spille, blir den lykkeligere siden `sinceTime` (i timer) reduseres og `playTime` (i minutter) muligens øker. Men, EMU blir ulykkelig når `sinceTime` og/eller `playTime` minker.

EMUs ansiktsuttrykk er gitt ved `emotion` som er et tall i området 0-100. Området representerer/dekker tre ”sinnstilstander”: Sad (nedre del av skalaen), OK (rundt midten), og Happy (øvre del).

Verdien til ut-variablen `emotion` bestemmes av fuzzy-resonnering med to inn-variable: `sinceTime` og `playTime`. Variabelen `sinceTime` har settene: **Short**, **Medium** og **Large**, og `playTime` har: **Little**, **Average** og **Long**.



a) Anta `sinceTime = 14`, `playTime = 3.5` og følgende Sugeno-regler med tre singeltons:

If (playTime is Little) then emotion is 20

If (playTime is Average OR sinceTime is Medium) then emotion is 50

If (playTime is Long AND sinceTime is NOT Short) then emotion is 80

Hvor lykkelig er EMU (gitt ved `emotion`)?

Anta her Sugeno-resonnering og vis trinnene i hvordan du kommer frem til svaret. Du kan tegne og skrive direkte på figurene/reglene i oppgaveteksten.

Hva skiller Mamadani fra Sugeno?

Hvorfor bruker en hedger i Fuzzy-resonnering?
Gi et eksempel.

b) Skriv om reglene i a) til Mamdani og tegn opp ditt valg av ut-sett.

Hvilken emosjon vil EMU-agenten vise når du bruker Mamdani-resonnering.
Du kan her bruke tilnærmede verdier fra figuren din i beregningen.

Oppgave 5 (20%)

a) Mange AI-systemer bygger opp en intern modell av verden som kan brukes i resonneringen. Hvordan er dette i "Brooks subsumption-arkitektur"?

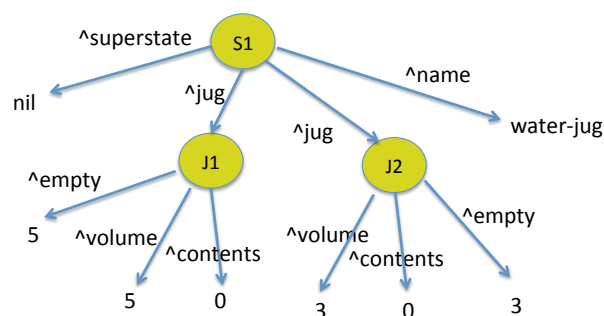
Forklar kort hoved-ideen bak arkitekturen.

b) Icarus kan sees på som en lagdelt arkitektur, med distinkte moduler som er integrert i en kaskade, hvor lav-nivå moduler leverer til høy-nivå moduler. Hvilke moduler har vi? Hva er oppgaven til den underste modulen?

Hva leder til en *impasse* i Icarus og hva gjør arkitekturen for å løse problemet.

c) Hvordan er Soar relatert til "The Heuristic Search Hypothesis"?

I det kjente "Water-jug"-problemet er det to mugger med kapasitet på 3 og 5 liter vann. Problemet som skal løses er å få 4 liter i en av muggene ved å fylle opp, helle over fra den ene til den andre, eller tømme ut vann. Anta at dette problemet initielt er representert ved:



Attributtet `^volume` er vannkapasiteten i liter, `^contents` angir hvor mange liter vann som er i mugga, og `^empty` angir ledig plass.

Skriv en Soar regel som tester om problemet er løst, dvs. om en av muggene inneholder 4 liter, og som avslutter agenten gjennom (`halt`) Hva slags type regel er dette?

Skriv nå kode for å fylle opp ei mugge helt opp ved først å sjekke om det er plass. **Hvordan** spiller Soars arkitektur inn på hvordan du programmer dette (i antall og type regler)?

Litt kodehjelp: Nedenfor viser en skisse på hvordan du endrer en verdi for et attributt `^attr` ved først å opprette ny attributt-verdi, her 4, og deretter slette den gamle:

```
sp {exampleRuleName
  (state <s> .... )
  ....
  (<j> ^attr <attr>)
--> .....
  (<j> ^attr 4          // make new attribute-value
   ^attr <attr> -)} // delete old
```

Du kan teste om et attributt er større enn en verdi, her 22, med konstruksjonen:

```
(<i> ^attr > 22)
```

- d) Ved en enkel analyse av EEG-data er det mulig å skille mellom 8 ulike positive og negative emosjoner: Joyful, Angry, Protected, Sad, Surprised, Fear, Satisfied and Unconcerned. Dette kan gjøres ved å se på de fire EEG-frekvensbåndene: Alpha (α), Beta (β), Theta (θ) and Delta (δ).

Hvordan kan en fra et EEG-signal få tak i disse båndene og bruke dem til å lage en digital kode for å klassifisere emosjonene?

Digital koding kan føre til generering av **samme** kode for mer enn en emosjon (gjelder for Surprised og Fear).

Hva kan en gjøre for å løse denne tvetydigheten?