NTNU

Norwegian University of
Science and Technology

Department of Computer and Information Science

Examination paper for TDT4137 Cognitive Architectures

Academic contact during examination: Asbjørn Thomassen

Phone: 73591839

Examination date: 5 desember 2016

Examination time (from-to): 0900-1300

Permitted examination support material: D/ Calculator

Other information:

Language: English

Number of pages (front page excluded): 5

Number of pages enclosed: 1

Informasjon om trykking av eksamensoppgave

Originalen er:

1-sidig □ 2-sidig □

sort/hvit □ farger □

skal ha flervalgskjema □

Checked by:

Date Signature

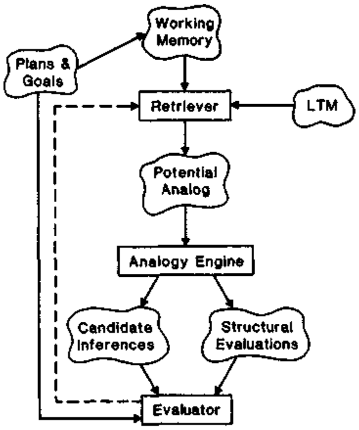
Question 1 (20%)

a) We have 4 important aspects of modelleing cognitive systems (Vernon).
Give a brief explanation of each of them.

b) What do the ”Physical Symbol Systems Hyphothesis” and the ”Heuristic Search Hyphothesis” claim?

c) Define different types of similarity based on domain knowledge represented as objects, attributes and relations (Gentner)?

d) The figure below shows an architecture for analogy reasoning.



Explain how the Analogy Engine works.

e) Explain briefly the MAC/FAC-implementation and how it finds analogies.

Question 2 (20%)

We want to model in NGOMSL the procedural knowledge of making triangles in Powerpoint. We limit ourselves to just look at how to draw a single triangle by using **copy-and–paste**. This can be done in two ways:

Alternative 1: First, you **mark** (select) an existing triangle on the screen.

Marking a triangle can either be done by moving the cursor to a position inside the triangle and then doing a mouse click, or by using a marquee rectangle, see Figure 1 .
To use a marquee rectangle you must first press down the mouse button in a start position **outside** the object (triangle). A rectangle with a fixed corner in the start position will now follow the cursor as shown in Figure 1. When the triangle is totally covered by the rectangle and the mouse button is released, the triangle will be marked as shown in Figure 2.

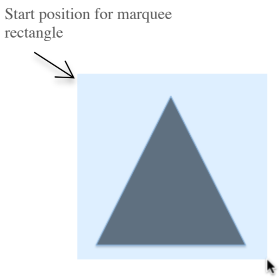


Figure 1. Use of rectangle to mark the triangle

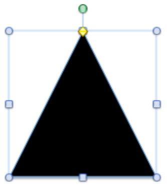
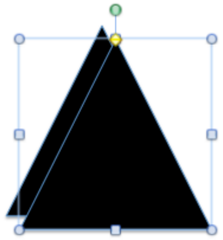


Figure 2. Marked triangle ready for copying

After the triangle has been marked, a **copy-and-paste** can be done by pressing the **cmd**-key together with **c** (copy) and afterwards together with **v** (paste). Notice that the new triangle is marked and is placed on top of the source triangle, so you have to drag it into wanted position on the slide:



Alternative 2: You may hold down the Option-key and press down the mouse button on an existing triangle. A marked triangle copy will be attached to the cursor, and this triangle can now be dragged into desired position where the mouse button is released.

a) Model the task `createTriangleByCopyAndPaste` in NGOMSL.
Notice that you are **not** asked to model the unit task level nor the use of WM/LTM

b) Compute execution time for making a new triangle using the marquee rectangle method.
(CP = 1.2, B=0.1, H=0.4, K=0.2, M=1.2, P = 1.1 - all in seconds)

c) Consistency can be defined as reuse of procedural knowledge. How does consistency appear in the goal graph and in the NGOMSL model?
Which operators do we have for memory storage and retrieval?
In your model, give examples of how you can utilize parallelism?
What is the difference between if-statements in selection rules and methods?

Question 3 (20%)

a) What is a recurrent neural network? How do you train an autoencoder (supervised learning)? What does it mean that a neural network is deep? Which class of cognitive systems do neural networks belong to? What is the main idea behind the delta rule?

b) We want to train a feed-forward-network using the backpropagation algorithm. The delta rule for training output layer is given by:

$$\Delta w_{jk}(p) = \alpha\, y_j(p)\, \delta_k(p)$$
$$\delta_k(p) = y_k(p)\,(1 - y_k(p))\, e_k(p)$$

and for hidden layer:

$$\Delta w_{ij}(p) = \alpha\, x_i(p)\, \delta_j(p)$$
$$\delta_j(p) = y_j(p)\,(1 - y_j(p)) \sum_k \delta_k(p)\, w_{jk}(p)$$

The activation function is:

k

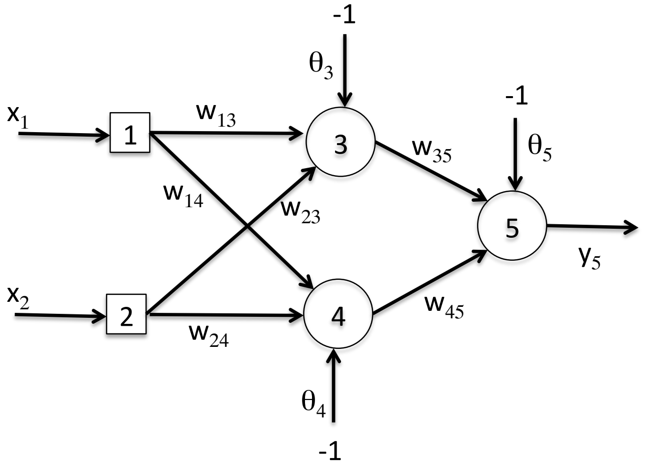
$$y_j = \frac{1}{1+e^{-X_j}}$$

where X_j is net input to a neuron.

What is α? Define X_j and e_k. Describe the steps of the backpropagation algorithm.

c) Given the network below with the following values after iteration p:

$x_1=1$, $x_2=0$, $y_d=1$, $\alpha=0.1$
$w_{13}= 0.5$, $w_{14}=0.9$, $w_{23}=0.4$, $w_{24}=1.0$, $w_{35}=-1.2$, $w_{45}=1.1$, $\theta_3=0.8$, $\theta_4=-0.1$, $\theta_5 = 0.3$



Show how you use the algorithm in b) to calculate changes in the weights. Notice that you shall only do one iteration (from p to p+1).

Question 4 (20%)

You are at a restaurant with some friends and it´s time to pay the bill. To solve the tipping problem, you apply your ”intelligent tipping assistant” running on your phone. It uses Fuzzy reasoning with two input variables: `service` and `food`. The variable `service` operates on the sets: **Poor**, **Good** and **Execellent**, and `food` has four sets: **Tasteless**, **Average**, **Delicious** and **Gourmet**. The output variable `tip` has the sets: **Cheap**, **Average** and **Generous**.
All sets and rules are described in Enclosure 1.

a) Assume Mamdani-reasoning and input values for `service` = 4 and `food` = 8.
What tipping advice gives the agent (show the steps) ?

You may write and draw on Enclosure 1. For simplicity, you can also use 3 as step size in the integration (a rough approximation).

b) What is the difference between Mamdani and Sugeno?
Rewrite the rules in Enclosure 1 to Sugeno by choosing singeltons in 3, 15 og 27 for `tip`. Show how the agent now comes up with a tipping advice.

How are hedges used in Fuzzy reasoning?

Question 5 (20%)

a) ACT-R can be classified as a hybrid cognitive system. Why?

b) Base-level activation is defined by:

$$B_i = \ln(\sum_j t_j^{-d})$$

What does this equation express?

c) The code below in Soar is from the well-known “Water Jug”-problem where we have two jars with capasity 3 and 5 liters. The attribute `^contents` is the water capasity in liters and `^empty` is the empty space left in a jar.

```
sp {water-jug*propose*pour
   (state <s> ^name water-jug
              ^jug <i> { <> <i> <j> })
   (<i> ^contents > 0 )
   (<j> ^empty > 0)
-->
   (<s> ^operator <o> + =)
   (<o> ^name pour
        ^empty-jug <i>
        ^fill-jug <j>)}
```

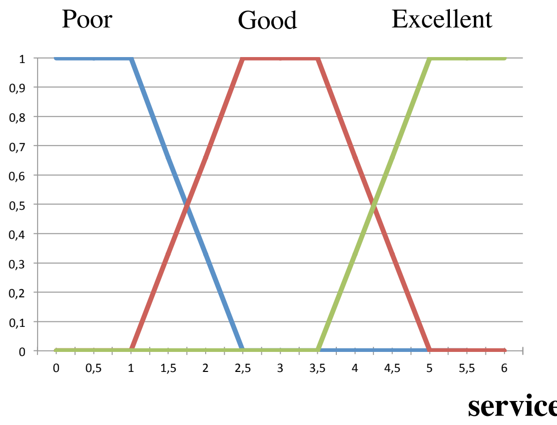
What is the code doing? Why is there no **application** of pour in the code? What does it mean to remove **+** = from the line `(<s> ^operator <o> + =)` in the code?

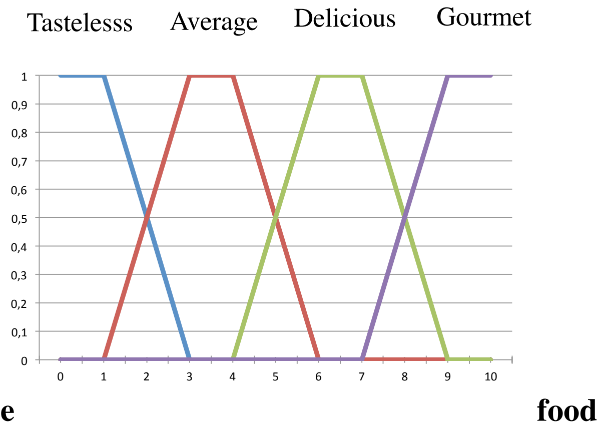
d) What is the purpose of the elaboration phase in Soar?
Assume that all operators have equal preference ”+”. What does Soar do to deal with the situation?

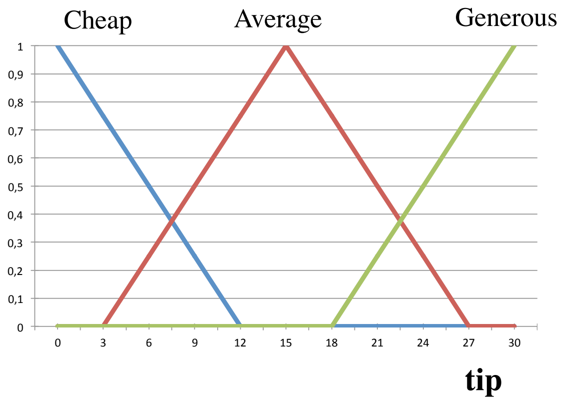
Enclosure 1

Kandidatnr/Candidate no:………………………
Dato/Date:……………… Side/Page: …………

Fuzzy sets and rules:







If (service is Poor) OR (Food is Tasteless) then (tip is Cheap)

If (service is Good) then (tip is Average)

If (service is Excellent) AND (food is Delicious OR food is Gourmet) then (tip is Generous)