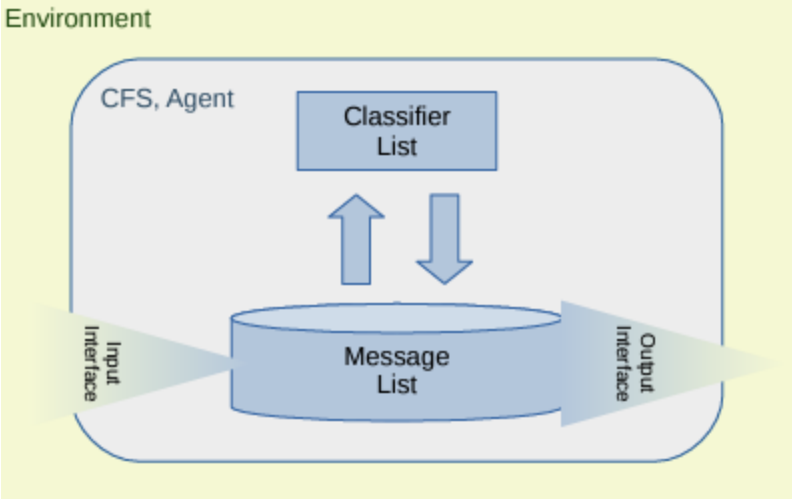
Lecture 11 - Classifier systems

- A cognitive architecture can mean:
  - a theory about the structure of the human mind
  - an implementation of such a theory (used in AI) - a cognitive system or agent
- Possible criteria for a cognitive architecture:
  - flexible behaviour
  - real time operation
  - rationality
  - large knowledge base
  - learning
  - development
  - adaptation
  - modularity
  - linguistic abilities
  - self awareness
- Competencies and behaviours demonstrated by such systems include:
  - perception
  - memory
  - attention
  - actuation
  - social interaction
  - planning
  - motivation

- emotion
- development
- using knowledge efficiently to perform new tasks

Architecture of a classifier system



The architecture of a classifier system

- The input interface generates messages that are written to the message list
- These messages are matched against the classifier rules to find out which actions need to be triggered
- The message list is then emptied and the encoded actions (that are also in form of messages) are stored in the message list
- Then the output interface checks the message list for actions
- The most basic language of CFS is {0,1,#}. In a real application we would use the natural language
- A message matches a condition if all its 0's and 1's are in the same positions as in the condition string.
- A negated condition is satisfied if no message in the message list matches it.
- In CFS actions are also fixed length strings from the same alphabet. Their length is usually the same as that of messages.

- An action message is build using the following procedure:
  - 0's and 1's in the action string are simply copied in the action message
  - #'s are substituted by the corresponding characters in the message that matches the first condition if the condition part.
- The output interface needs to recognize which messages are input messages posted by the input interface, which are internal messages, and which are messages meant to be the output messages. This is obtained by using tags usually consisting of two additional bits in the messages that are interpreted in a special way. Example:

Tag	Interpretation
01	Internal Message
11	Internal Message
00	Output Message
10	Input Message

Main cycle

1. Activate the input interface and post the input messages it generates to the message list
2. Perform the matching of all the conditions of all classifiers against the message list
3. Add the messages generated by the classifiers' actions
4. Activate the output interface, i.e. remove the output messages from the message list and perform the actions they describe, go to 1.

- Classifiers' rulesets can be non-overlapping, i.e. they generate only one action message per one input message, or they can be organized in a hierarchy.

Learning Classifier Systems (LCS)

- There are two ways a classifier system can "learn":

- Adaptation by credit assignment - changing the way existing classifiers are used
- Adaptation by rule discovery - introducing new classifiers in the system

The Bucket Brigade algorithm

- If there is a reward (or punishment) add it to the strength of all classifiers active in the current cycle
- Make each active classifier pay its bid to the classifiers that prepared the stage for it (i.e. posted messages matched by its conditions)
- We can calculate the bid by:

$$bid = k \cdot strength \cdot specificity$$

- Where k is constant (around 0.1) strength is the equivalent of money of the classifier and specificity is $\frac{length - \text{number of hashtags}}{length}$.
- The strength of the classifier in the next step can be calculated like this:

$$S(t+1) = S(t) + R(t) - T(t) - B(t)$$

- Where S is strength, R is reward, T is taxes and B is bid paid. The bid can also be a constant multiplied by the strength, without specificity.

Adaptation by rule discovery

- Evolutionary algorithm can be used for optimizing LCS's in the following ways:
  - Pittsburgh approach - consider the classifier list as a single individual
  - Michigan approach - considering each classifier as a separate individual