

Latent Learning and Action Planning in Anticipatory Classifier Systems

Wolfgang Stolzmann & Martin Butz

University of Wuerzburg, Institute for Psychology III
{stolzmann, butz}@psychologie.uni-wuerzburg.de

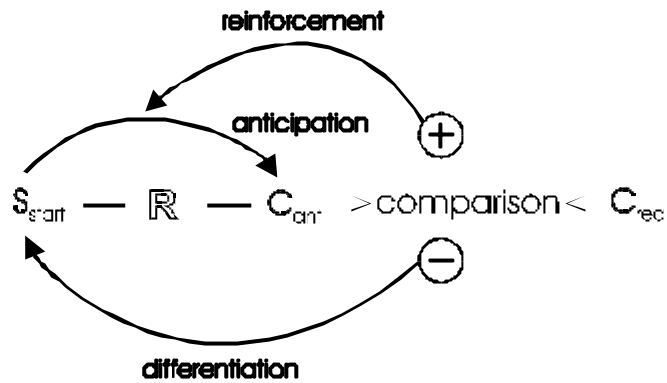
Wolfgang Stolzmann, University of Wuerzburg, Institute for Psychology III, Roentgenring 11, D-97070 Wuerzburg, Germany



Now Martin Butz and me will talk about anticipatory classifier systems. At first I will give you an introduction to ACS. Then I will show you how ACS can be used for latent learning in a mobil robot and in the third part of our talk Martin will explain how action planning can be combined with latent learning.

ACS are Learning Classifier Systems that learn by using anticipatory behavioral control.

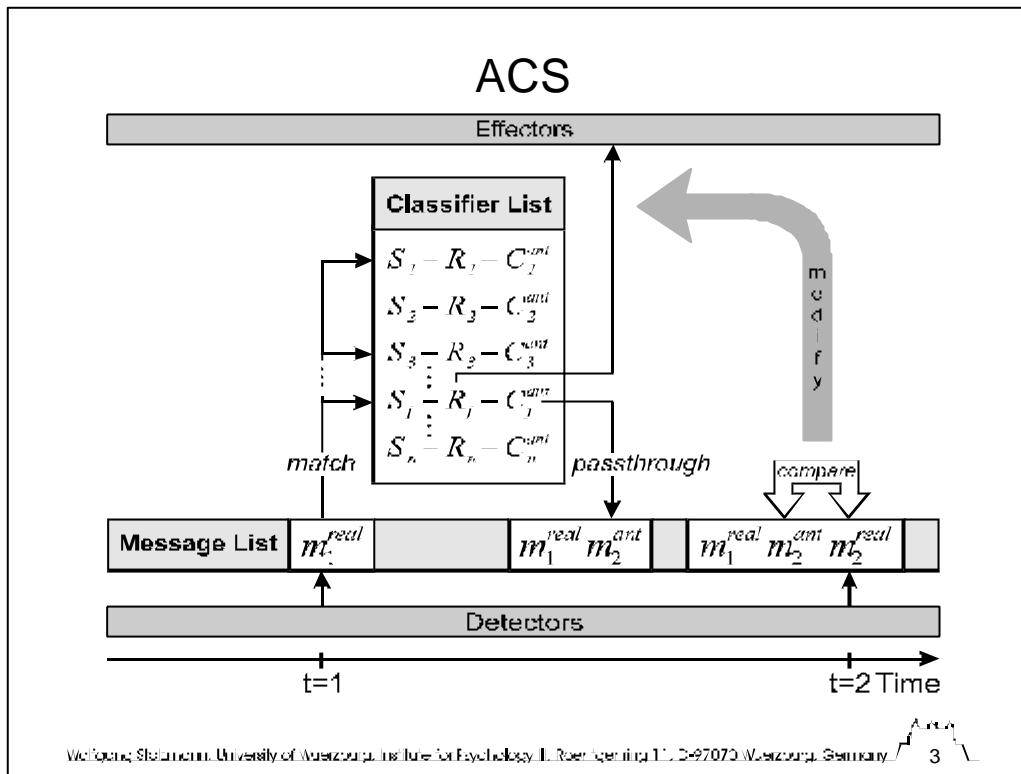
Anticipatory Behavioral Control



according to Hoffmann, Joachim (1993), Vorhersage und Erkenntnis.
Göttingen: Hogrefe, p. 44 (own translation)

Wolfgang Steinhilber, University of Würzburg, Institute for Psychology, II, Roentgenring 11, D-97070 Würzburg, Germany

Anticipatory behavioral control is a psychological learning mechanism that was formulated by Joachim Hoffman in 1992. He starts from the assumption that a decisive factor for purposive behavior is the assumption of the behavior consequences. Behavior consequences usually depend on the situation in which the behavior is executed. So it is necessary to learn in which situation S which behavior R leads to which behavior consequences C_{ant} . R stands for reaction. These S - R - C_{ant} units are learned by comparing the anticipated behavior consequences C_{ant} with the real behavior consequences C_{real} . If the anticipated consequences C_{ant} agree with the real consequences C_{real} , then reinforcement takes place. Otherwise the starting situation should be differentiated.



In anticipatory classifier systems we use classifiers that consist of an S -part, that is equal to the condition part of a 1-condition classifier system, an R -part that is responsible for motor actions of the system and a C^{ant} -part that describes the anticipated behavioral consequences. ‘Don’t care’-symbols can be used in the S -parts and the C^{ant} -parts.

At moment T_1 the detectors send the current perception of the environmental state to the message list, so that the message list contains an internal representation of the current environmental state. This message m_1^{real} is compared with the S -parts of all classifiers of the classifier list. Classifiers that match with m_1^{real} are called satisfied classifiers. A strength-based competition takes place and the C^{ant} -part of the winner adds a new message m_2^{ant} to the message list by using the passthrough process. This is the anticipated environmental state immediately after the motor action of the system.

The winner sends its R -part R_j to the effectors and causes a motor action of the system.

At moment T_2 the detectors send the new internal representation of the environmental state m_2^{real} to the message list. The system assumes that m_2^{real} is the result of R_j and compares m_2^{real} with m_2^{ant} .

Depending on the result of this comparison the classifier list is modified: Either the winner is reinforced or unreinforced, or a new classifier is added to the classifier list.

m_1^{real} and m_2^{ant} are deleted and m_2^{real} starts a new behavioral act.

Selection of a classifier

All satisfied Classifiers perform a competition.

- Either a strength-based competition ...

A satisfied Classifier c_j is selected with a probability of

$$\frac{b_{c_j}(t)}{\sum_{\text{satisfied } c} b_c(t)} \quad \text{with } b_c(t) = s_c(t) \cdot s_c^{\text{ant}}(t) \cdot 2^{\text{spec}(c)}$$

- ... or, with a probability of p_{explor} , a random competition:

A satisfied Classifier is selected with a probability of

$$\frac{1}{|\text{satisfied } c|}$$

Wolfgang Stalmann, University of Wuerzburg, Institute for Psychology II, Roentgenring 11, D-97070 Wuerzburg, Germany  4

In our current version of ACS the selection of a classifier is different from that one reported in my paper.

In the paper there is only a strength-based competition. Now we also have, with a probability of p_{explor} , a random competition, that forces exploration.

Classifier List at time t=0.

Let's consider an ACS with 5 detectors and one effector that enables it to do the following motor actions:

l, to make a 90 degree left turn,
r, to make a 90 degree right turn and
f, to go forward.

This ACS starts with the classifier list

```
##### - f - #####  
##### - l - #####  
##### - r - #####
```

Wolfgang Steilmann, University of Wuerzburg, Institute for Psychology, II, Roentgenring 11, D-97070 Wuerzburg, Germany



Before I start to talk about learning in ACS, please note that an ACS usually starts with such a classifier list.

The *S*-parts and the *C^{ant}*-parts contain only 'don't care'-symbols. The *R*-parts consist of the basic behaviors.

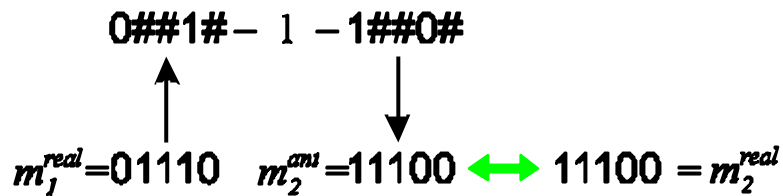
For example: If an ACS is able to do the following motor actions:

l, to make a 90 degree left turn,
r, to make a 90 degree right turn and
f, to go forward

then this is the classifier list, the ACS starts with.

Latent Learning in ACS

- ☞ expected case
- useless case
- correctable case
- not correctable case



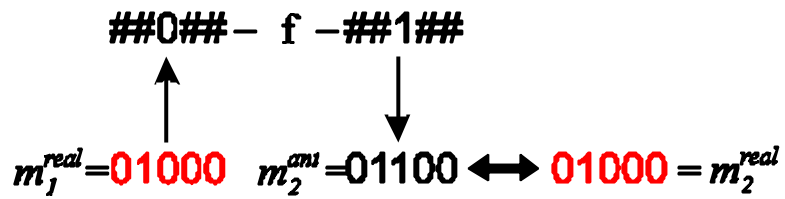
$$s_c^{ant}(t+1) = (1 - b^{ant}) \cdot s_c^{ant}(t) + b^{ant}$$

Let's talk about learning in ACS. We distinguish 4 cases: the expected case the useless case, the correctable case and the not correctable case.

If m_2^{ant} matches m_2^{real} and if the behavioral act changes the environment, then the anticipation-strength of the winning classifier is increased.

Latent Learning in ACS

- expected case
- ☞ useless case
- correctable case
- not correctable case



$$s_c^{ant}(t+1) = (1 - b^{ant}) \cdot s_c^{ant}(t)$$

Wolfgang Starmann, University of Wuerzburg, Institute for Psychology, II, Roentgenring 11, D-97070 Wuerzburg, Germany 7

If the behavioral act doesn't changes the environment, then the anticipation-strength of the winning classifier is decreased.

Latent Learning in ACS

- expected case
- useless case
- correctable case
- ☞ not correctable case



Latent learning in ACS

Specification of unchanging components

- Each Classifier gets a mark when applied in a *not expected case*.
- The mark describes the states in which the classifier did not work correctly.
- If a marked classifier is applied in the *expected case*, and if at least one component of the current state does not match the mark, then one of these components is selected to be specified in the classifier.

Wolfgang Steilmann, University of Wuerzburg, Institute for Psychology, II, Roentgenring 11, D-97070 Wuerzburg, Germany



Each Classifier gets a mark when applied in a *not expected case*.

The mark describes the states in which the classifier did not work correctly.

If a marked classifier is applied in the *expected case*, and if at least one component of the current state does not match the mark, then one of these components is selected to be specified in the classifier.

This is a rough idea of *specification of unchanging components*. Martin will give you a more precise description and an example.

Non-Markov states

Criterion for Non-Markov states:

If the specification of unchanging components fails, then the ACS assumes that the current environmental state is a Non-Markov state.

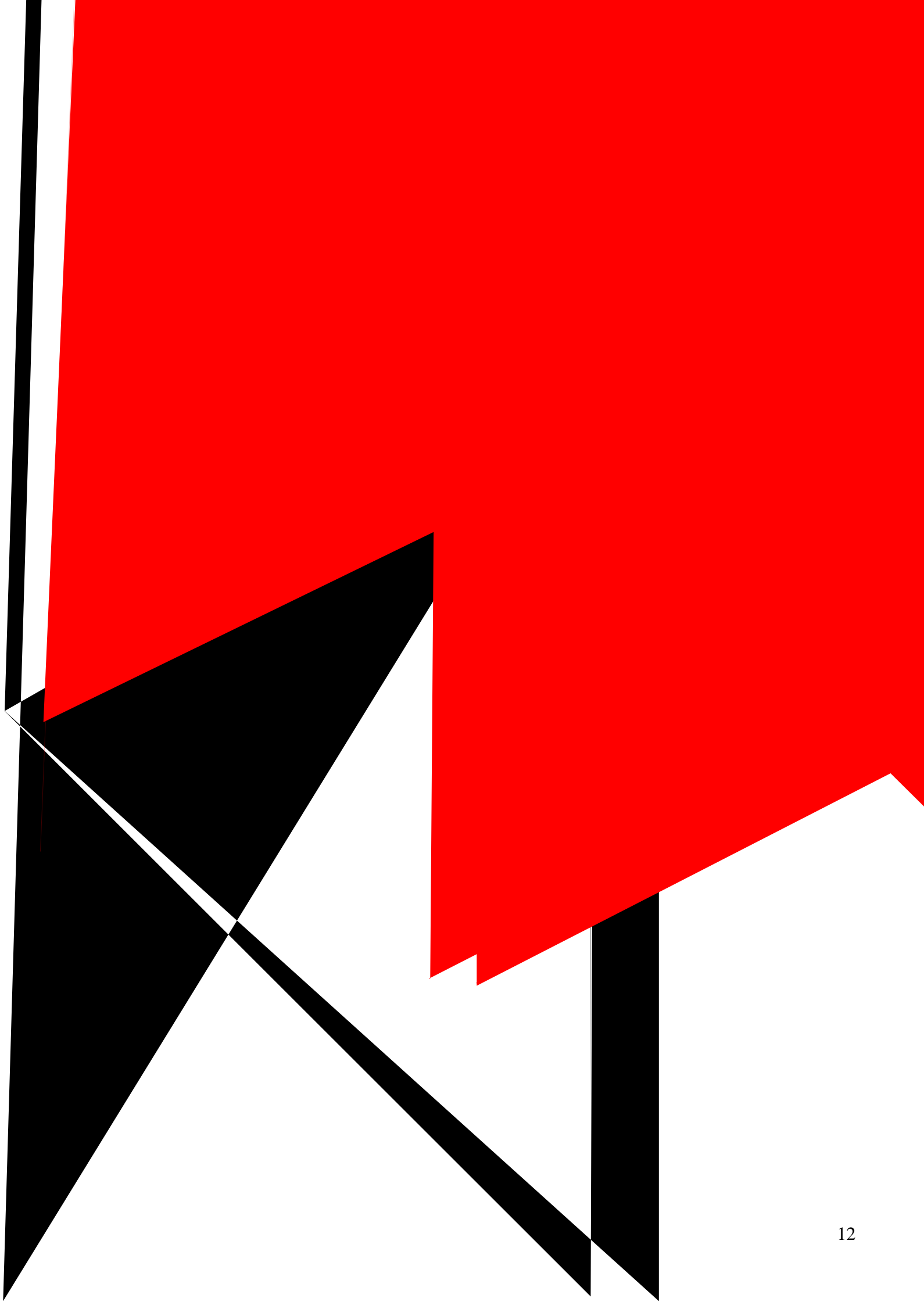
Wolfgang Stelmann, University of Würzburg, Institute for Psychology, II, Robert-Koch-Str. 11, D-97070 Würzburg, Germany  11

Usually a mobile robot can observe its environment only partially, so that the Markov property is not necessarily given. So we have to deal with Non-Markov states. At first we need a criterion to identify Non-Markov states. This can be done in conjunction with *specification of unchanging components*.

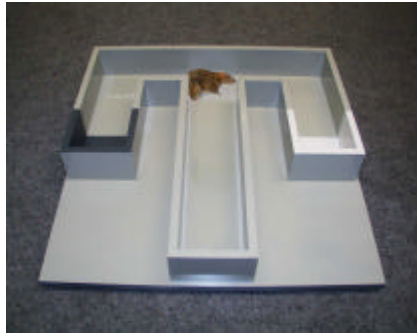
Our criterion for Non-Markov states is:

If the specification of unchanging components fails, then the ACS assumes that the current environmental state is a Non-Markov state.

This assumption may be wrong in some cases, but nevertheless it's a criterion that works very good.



Experiment about latent learning



Results

- *Main experiment*
28 of 32 tested rats demonstrate latent learning by going to the good-goal box.
- *Control experiment*
without an exploration-phase:
27 of 55 rats go to the good-goal box.

Wolfgang Steinhilber, University of Würzburg, Institute for Psychology, II, Robert-Koch-Str. 11, D-97070 Würzburg, Germany



Now I want to use an ACS for a robot-simulation of an experiment about latent learning in rats. Latent learning is defined as learning in the absence of environmental reinforcement.

The experiment was developed by John Seward in 1949. During an exploration phase a rat was put into a T-maze and explored it without getting any reward. After that the rat was not fed for 24 hours, placed directly in one of the goal boxes with food in it and allowed to eat for one minute. Then the rat is placed here (show S). 28 of 32 tested rats went directly to the good-goal box. During a control experiment without an exploration-phase only about 50% of the rats go to the good-goal box. This experiment proves, that the rats learned latently during the exploration phase.

Some of you know this experiment because it is the same one I talked about in Madison during GP98 last year. But this time it's much harder for the ACS, because the exploration-phase is not divided into different trials, and the ACS has to deal with Non-Markov states.

Replication of the experiment with a Khepera robot

It is possible to define the following 5 detectors and 1 effector for an ACS by using the 8 sensors and 2 motors of a Khepera-robot:

- d_1 : 0, if there is nothing in front of the robot,
1, if there is a wall in front of the robot,
 - d_2 : as d_1 , but on the left of the robot,
 - d_3 : as d_1 , but behind the robot,
 - d_4 : as d_1 , but on the right of the robot,
 - d_5 : 0, if there is no infra-red light near to the robot,
1, if there is a infra-red light near to the robot
- e_1 : l, to make a 90 degree left turn,
r, to make a 90 degree right turn,
f, to go forward.

Wolfgang Steilmann, University of Wuerzburg, Institute for Psychology, II, Roentgenring 11, D-97070 Wuerzburg, Germany



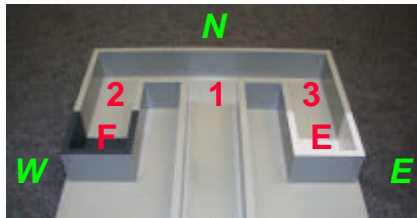
For a robot-simulation of this experiment we used a small Khepera-robot. The 8 sensors and the 2 motors of the robot can be used to define the following 5 detectors and 1 effector for an ACS:

- d_1 : 0, if there is nothing in front of the robot,
1, if there is a wall in front of the robot,
- d_2 : as d_1 , but on the left of the robot,
- d_2 : as d_1 , but on the left of the robot,
- d_3 : as d_1 , but behind the robot,
- d_4 : as d_1 , but on the right of the robot,
- d_5 : 0, if there is no infra-red light near to the robot,
1, if there is a infra-red light near to the robot.

We used such a light to distinguish the both goal-boxes.

- e_1 : l, to make a 90 degree left turn,
- r, to make a 90 degree right turn,
- f, to go forward.

Replication of the experiment with a Khepera robot



	S	1	2	3	F	E
N	□	—	┌	└		□
W	□		┌	└	■	□
S	□			┌	■	┐
				└	■	└

Action planning

Definition of sure Classifiers:

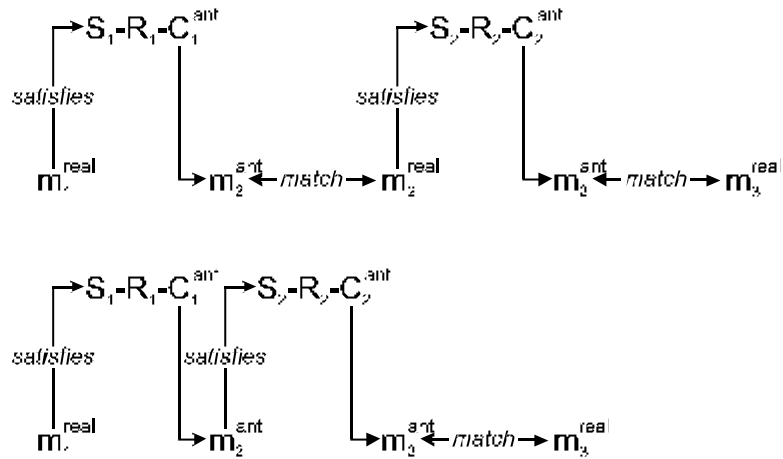
A classifier c is called sure, if $s_c^{ant}(t) \geq 1 - e^{sure}$,
where $e^{sure} \in [0,1]$ is a small constant.

Action planning is a possibility for ACS to behave goal-directed.

Let's start with a definition:

A classifier is called sure, if its anticipation strength is very close to 1.

Action planning



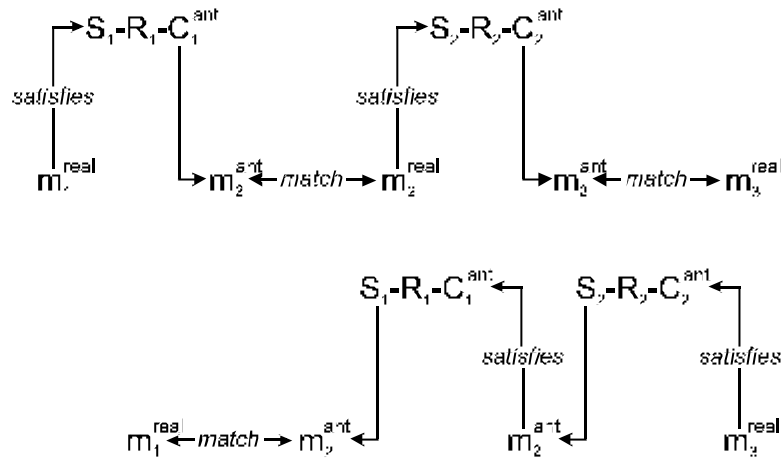
Wolfgang Steilmann, University of Wuerzburg, Institute for Psychology, II, Roentgenring 11, D-97070 Wuerzburg, Germany


17

Sure classifiers can be used to build behavioral sequences. Let us assume that we have two behavioral acts as shown here and let us assume that the classifiers are sure. That means that the probability that m_2^{ant} matches m_2^{real} is very high and we can combine these two classifiers as shown below.

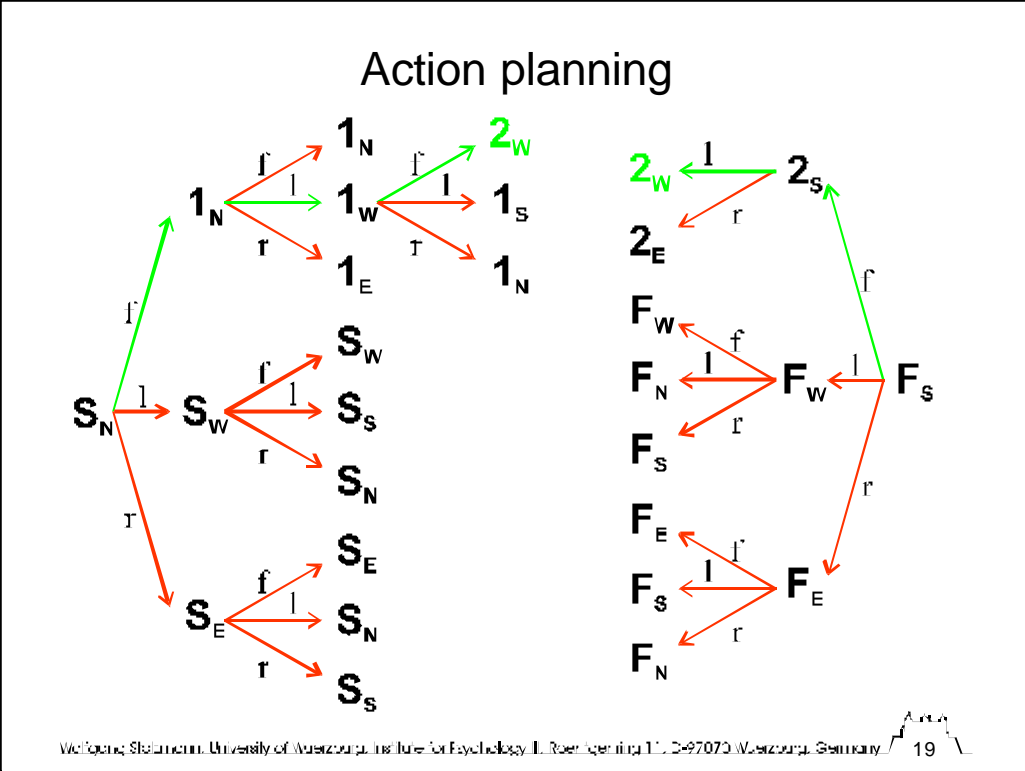
It's not necessary to perceive the environmental state m_2^{real} . If the R -parts of the classifiers are not executed, then sure classifiers can be used for *forwards-directed* action planning.

Action planning



Wolfgang Steilmann, University of Würzburg, Institute for Psychology, II, Robert-Koch-Str. 11, D-97070 Würzburg, Germany.  18

If there is a goal, for example m_3^{real} , then *backwards-directed* action planning can be used. This can be done by exchanging the meaning of the S -parts and the C^{ant} -parts of the classifiers.



We can combine *forwards-directed* action planning and *backwards-directed* action planning and use a *bidirectional search*.

In this example the robot is located in the position S. Its goal is to reach the left goal box F.

If the *forwards-directed* search and the *backwards-directed* search reach a common state, here position 2 in the maze, then the action planning is successful and the ACS knows the actions that are necessary to reach the goal, here flflf.

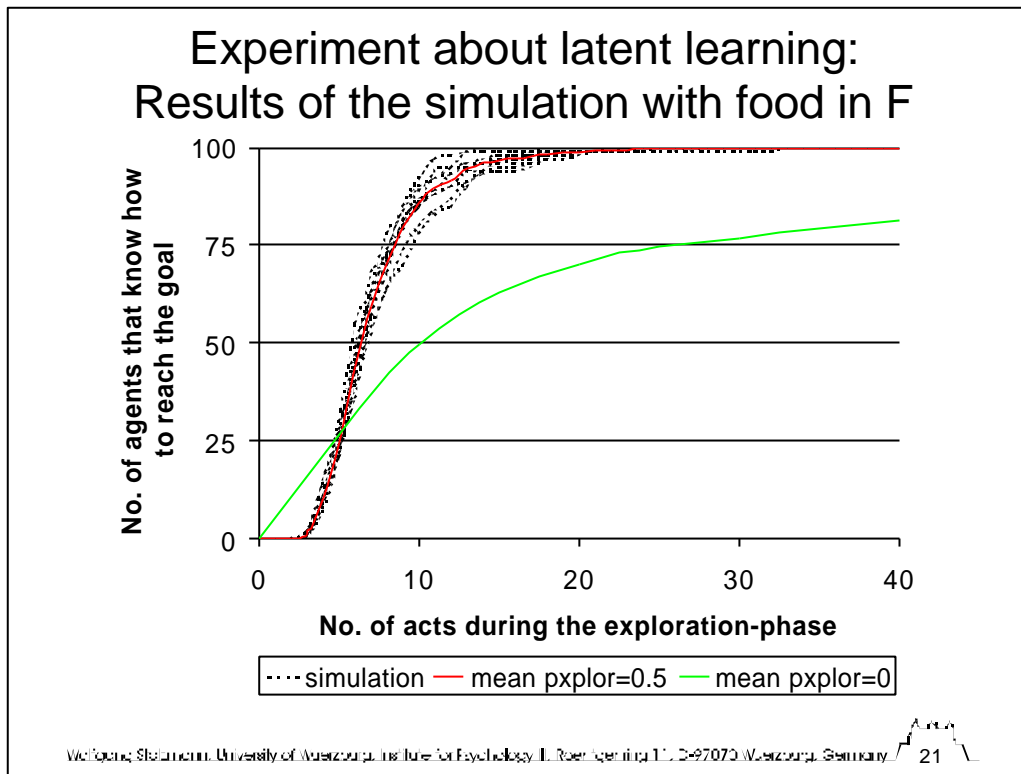
Experiment about latent learning: Results

In 100 runs the classifier list consists of
190 / 145 (maximum / mean) classifiers.
54 / 32 classifiers are sure.

Wolfgang Stalmeier, University of Wuerzburg, Institute for Psychology II, Roentgenring 11, D-97070 Wuerzburg, Germany  20

Results!:

In 100 runs the classifier list consists of 190 classifiers at most. The mean is 145. 54 or 32 classifiers are sure.



The green curve shows the results I reported in the paper. The robot-simulation was done with a software-simulation of a Khepera-robot. The abscissa contains the number of behavioral acts during the exploration-phase and the ordinate the number of agents out of 100 that are able to do action-planning successfully if their current position is the choice-point of the maze and if their goal is to reach the left goal box.

If we use exploration (You remember! I talked about a new algorithm for the selection of a classifier at the beginning of my talk), then we can improve the learning of the ACS.

The red curve shows our results for $p_{\text{xplor}}=0.5$.

Thank you so far! Now Martin will continue.