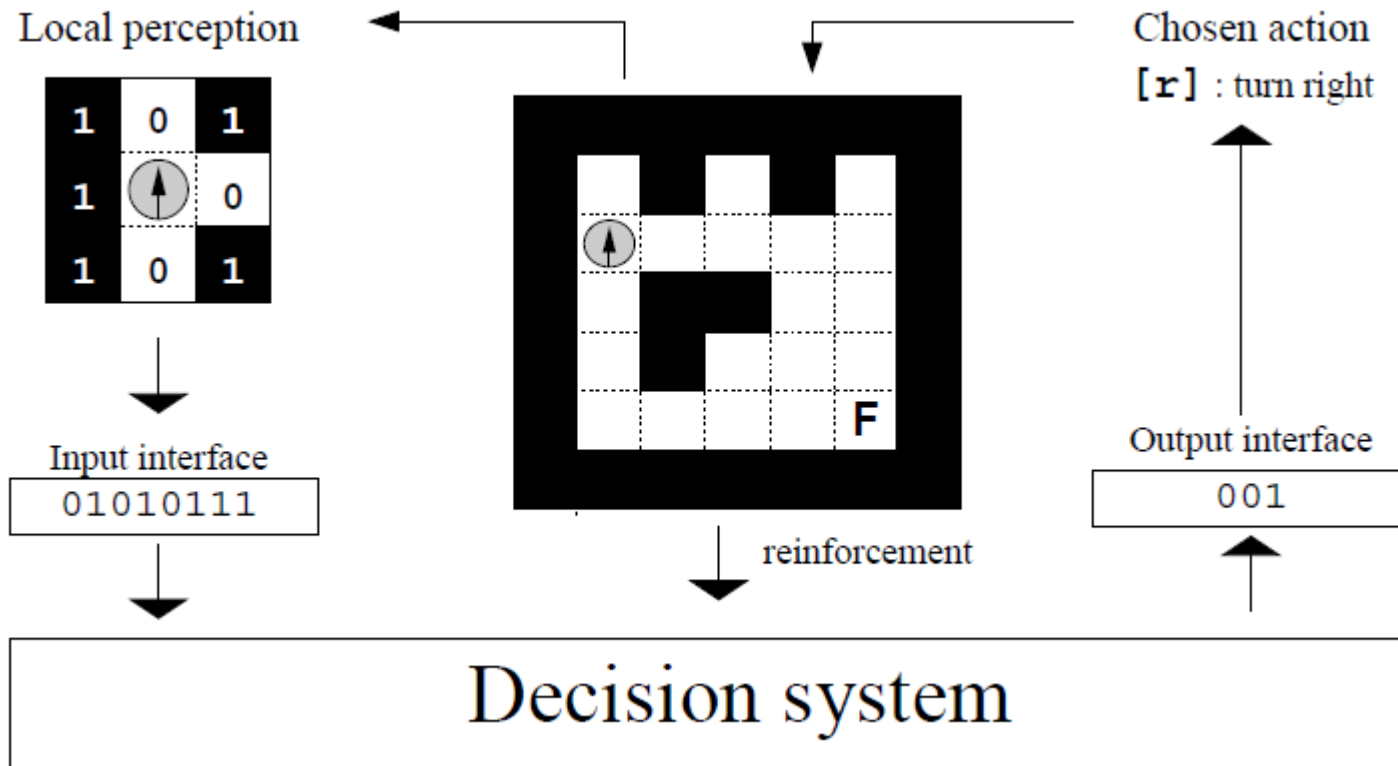

Algoritmy pre AI robotiku, V. Diel

LCS, NEAT

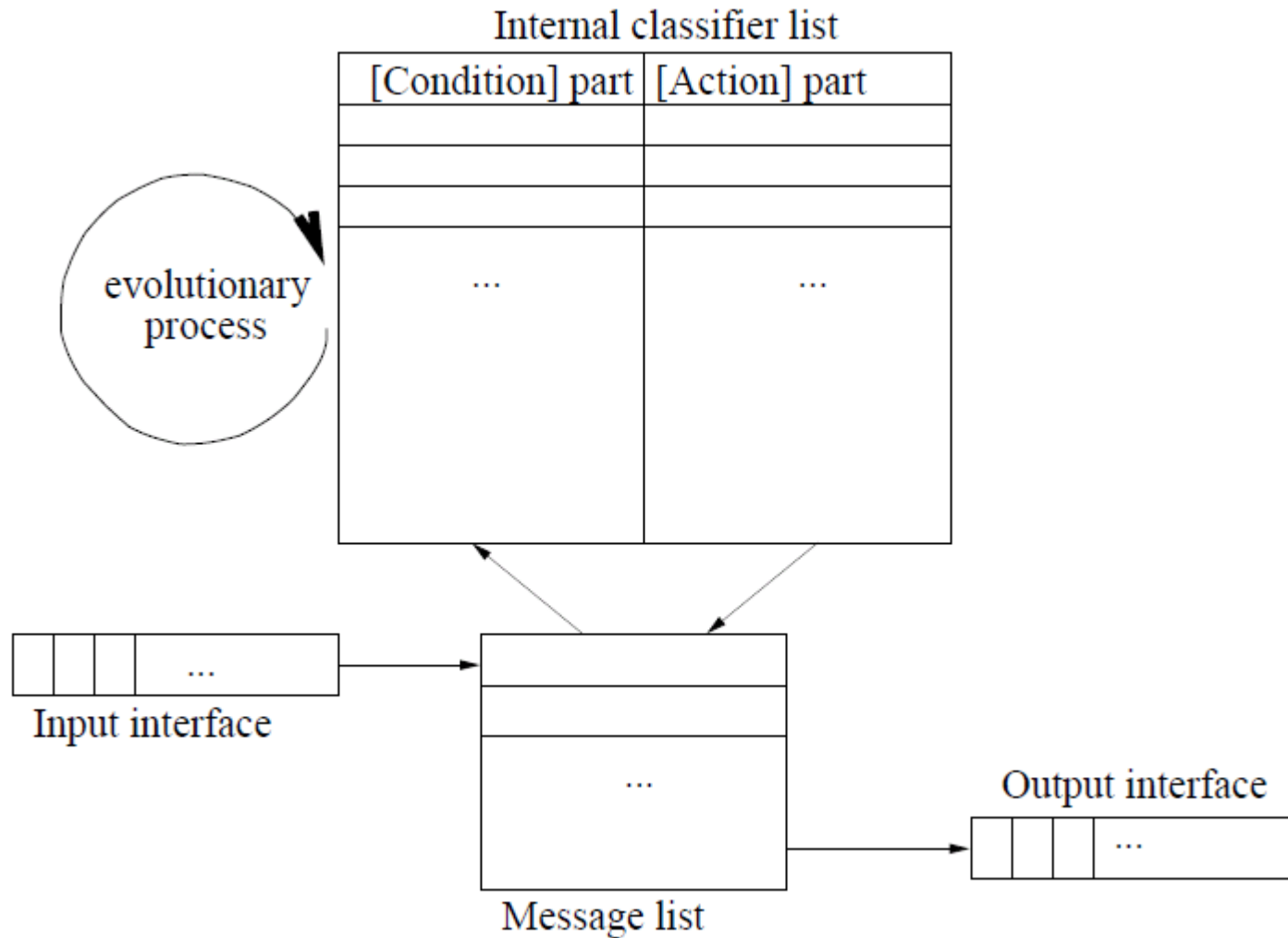
Pavel Petrovič
ppetrovic@acm.org

RL + GA = LCS



From animals to animats...

LCS - CS1



LCS - flavours

- Bucket brigade vs. Q-learning
- No message list, added strength to the rules -> ZCS, strength-based ZCS
- predict reward in addition to select action -> accuracy-based LCS (XCS)
- anticipatory-based LCS -> ACS

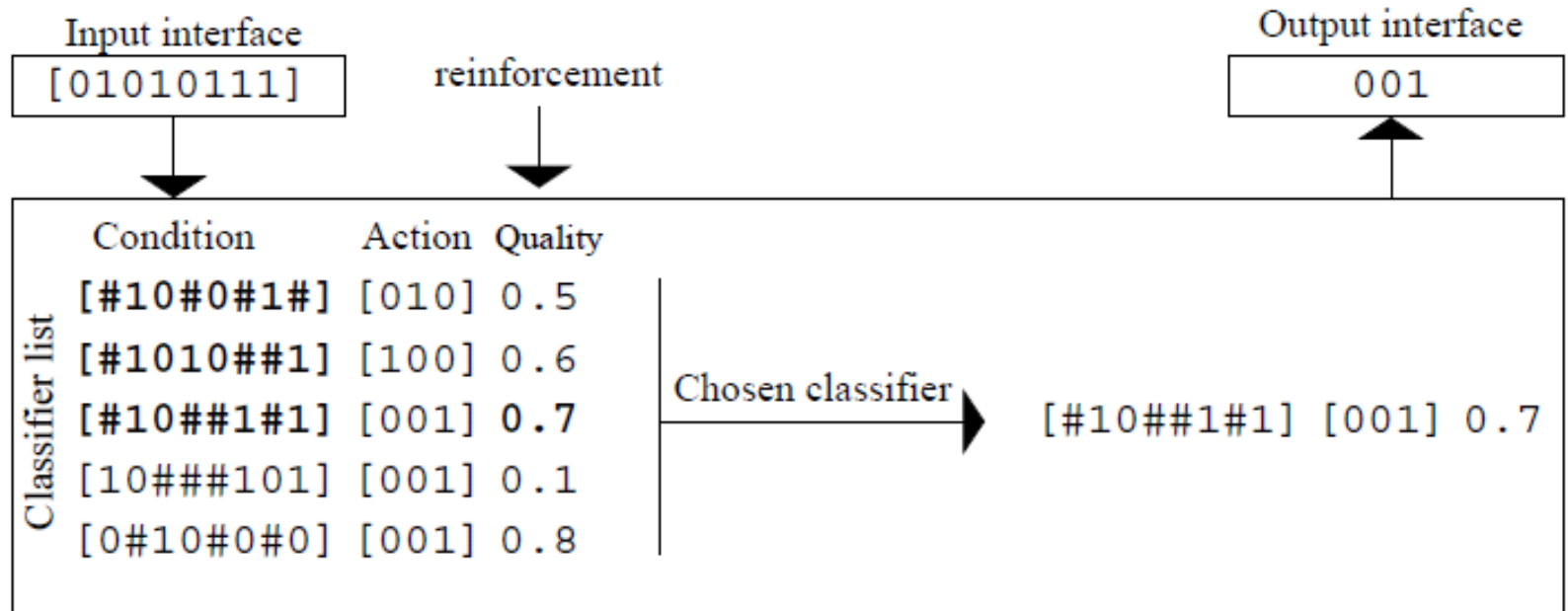
LCS - formalism

- classifier: $\langle c, a, p \rangle$
 - c: condition
 - a: action
 - p: prediction of expected accumulated reward

Condition – test on attribute values, e.g.
(0, 1, 1, #, 1, #) # = don't care
alternately: integers, real numbers, S-expressions

Action – multiple can be selected
-> set M (match set),
one is chosen – a (wrt. Exploration/exploitation)
-> [a] set – rules that advocated for this action a

LCS – strength-based: ZCS



Quality – both fitness and reward estimation
Action selection – roulette wheel
Classifier evaluation – three step mechanism

LCS – strength-based: ZCS

Classifier evaluation – three steps mechanism:

1. all classifiers that advocated for action $a(t-1)$ share equally a fraction $\alpha \cdot \gamma$ of the sum of the values of classifiers in $[a]$
2. all classifiers in $[a]$ share a fraction α of the reward $r(t)$
Received for executing $a(t)$
3. the value of all classifiers in $M - A$ is reduced with a tax τ

LCS – strength-based: ZCS

Classifier creation/deletion

- at each time step, run GA with probability p
- select two (roulette), recombine (1-point crossover), and mutate, initial fitness: average of parent classifiers, replace by inverse roulette
- each time M is empty or contains too weak classifiers, use „covering operator“ that generates new rule, using # with 33% probability and random action, fitness equal to average fitness of all classifiers

LCS – strength-based: ZCS

Typical parameters

Population size: 400

Initial fitness: 20

Learning rate (α): 0.2

Discount factor (γ): 0.71

Tax (τ): 0.1

GA firing rate p : 0.25

Crossover rate P_c : 0.5

Mutation rate P_m : 0.002

Covering operator firing rate: 0.5

NEAT – evolving RNN

See article in materials at the webpage of the course