The Animat Path to Artificial General Intelligence

Claes Strannegård
Chalmers University of Technology
University of Gothenburg, Sweden
claes.strannegard@chalmers.se

Nils Svangård
University of Gothenburg, Sweden
nils.svangard@gu.se

David Lindström
University of Gothenburg, Sweden
david.lindstrom@student.gu.se

Joscha Bach
Harvard University, USA
bach@fas.harvard.edu

Bas Steunebrink
NNAISENSE, Switzerland
bas@nnaisense.com

Abstract

Stewart Wilson introduced the term *animat* for artificial animals and outlined the *animat path* to artificial intelligence. In this paper the animat path to artificial general intelligence is explored. A general computational model is proposed for animats living in dynamic block worlds, e.g. in the Minecraft environment. The model uses mechanisms for learning and decision-making that are common to all animats. Each animat has its own sets of needs, sensors, and motors. It also has its own memory structure that undergoes continuous development and constitutes the basis for decision-making. The goal of the decision-making is always to keep the needs as satisfied as possible for as long as possible. The learning mechanisms are of two kinds: (i) structural learning that adds and removes nodes and connections of the memory structure; (ii) a local version of multi-objective Q-learning. The animats of the model are autonomous and able to adapt to arbitrary previously unseen block worlds without any need for seed knowledge. They adapt by learning basic skills such as foraging, locomotion, navigation, and pattern recognition.

1 General intelligence

In psychology, the term *general intelligence* refers to the way that a person's performance on one psychometric task tends to correlate with her performance on other tasks [Spearman, 1904]. In artificial intelligence, the term tends to be used more broadly for versatile and autonomous agents [Legg, 2008]. A survey of performance measures relating to general intelligence in humans and artificial systems can be found in [Hernández-Orallo, 2016].

According to the physicist Pieter van Heerden [Heerden, 1968]:

> Intelligent behavior is to be repeatedly successful at satisfying one's psychological needs in diverse, observably different, situations on the basis of past experience.

Interpreted broadly, van Heerden's characterization of intelligent behavior takes all types of needs in Dörner's taxonomy into account: physiological, social, and cognitive [Dörner, 2001]. Note that van Heerden's characterization applies to all animal species, not just humans. It is also general in the sense that it does not rely on human judgement, like the Turing test does; or on human artifacts, like standard IQ tests do.

Moreover, van Heerden's characterization harmonizes with an embodied view of intelligence, whether or not the distinction between body and mind is maintained. In fact, the ability of an animal to satisfy its needs depends on the body, the control system of the body, and the interplay between the two, to the extent that those notions can be meaningfully separated in the first place.

The goal of *artificial general intelligence* is to take the step from "narrow" AI programs that are tailored for specific tasks or problem domains to general AI programs with intelligence "at the human level and beyond" [Pennachin and Goertzel, 2007].

Reinforcement learning and in particular Q-learning has been used in agents where the goal is to accumulate reward over time [Sutton and Barto, 1998]. In standard reinforcement learning, the reward signal is one-dimensional; in multi-objective reinforcement learning, it is multidimensional [Roijers *et al.*, 2013]. Sometimes it is straightforward to reduce multidimensional reward signals into scalars, e.g. money of several currencies can be converted into money of a single currency. Sometimes it is harder, e.g. in the case of an animal that receives a reward signal with an energy and a water component. No amount of energy can compensate for a lack of water and vice versa.

Certain agents have a set of needs and receive a (multi-dimensional) reward signal that measures changes in the status of those needs. Such *homeostatic* agents strive to keep several internal signals in certain intervals [Konidaris and Barto, 2006; Yoshida, 2017]. For homeostatic agents, the above-mentioned characterization of intelligent behavior by van Heerden essentially coincides with the so-called *reinforcement learning hypothesis* [Lettman, 2006]:

> Intelligent behavior arises from the actions of an individual seeking to maximize its received reward

signals in a complex and changing world.

Deep Q-learning combines deep networks with reinforcement learning [LeCun *et al.*, 2015; Schmidhuber, 2015]. One of the most prominent examples of this method in the direction of general intelligence is the generic Atari-game player that learned to play 31 Atari games at super-human level [Mnih and others, 2015]. Although deep Q-learning has been groundbreaking, several issues remain problematic: avoiding catastrophic forgetting; enabling lifelong, one-shot, and transfer learning; reducing the need for large training volumes; and supporting logical reasoning [Harrigan, 2016]. For a discussion of some theoretical problems associated with deep Q-learning, see [Wang and Li, 2016].

Graph structures that develop gradually have been studied, e.g. in finite automata learning [Angluin, 1980], cascade correlation networks [Fahlman and Lebiere, 1990], and deep network cascades [Angelova *et al.*, 2015].

Cognitive architectures, e.g. Soar [Laird, 2012], ACT-R [Anderson *et al.*, 2004], and MicroPsi [Bach, 2015], are computer systems that attempt to model aspects of the human mind, including general intelligence. *Agent architectures* reflect a wider notion that includes systems for artificial intelligence that do not necessarily aim for biological realism, e.g. OpenCog [Goertzel *et al.*, 2014], AERA [Nivel *et al.*, 2013], and NARS [Wang and Hammer, 2015].

Animal intelligence has been studied extensively: e.g., in comparative psychology and artificial life. The objects of study in artificial life include artificial evolution, cellular automata, and particle swarm optimization [Langton, 1997; Tuci *et al.*, 2016].

Stewart Wilson introduced the term *animat* for artificial animals via the following postulates, quoting from [Wilson, 1986]:

- The animal exists in a sea of sensory signals. At any moment, only some signals are significant; the rest are irrelevant.
- The animal is capable of actions (e.g., movement) which change these signals.
- Certain signals (e.g., those attendant on consumption of food) or their absence (e.g., those relating to pain) have special status.
- He acts, both externally and internally, so as approximately to optimize the rate of occurrence of the special signals.

Wilson also outlined the *animat path to AI*, which seeks to create artificial intelligence by modeling animal intelligence [Wilson, 1990].

In this paper we explore the animat path to artificial general intelligence. Section 2 describes our strategy for constructing a general and autonomous computational model. Section 3 describes our constructed model. Section 4 presents the prototype implementation *Generic Animat* of the model and gives examples of how it learns and makes decisions in the context of foraging, locomotion, navigation, and concept formation. Section 5 discusses the scalability of the model and Section 6, finally, draws some conclusions.

Figure 1: A Minecraft world with blocks of type “water”, “grass”, “sand”, etc.

The proposed computational model is partly a continuation of our previous work [Bach, 2015; Nivel *et al.*, 2013; Strannegård *et al.*, 2015; Strannegård and Nizamani, 2016]. The mechanisms for local Q-learning and structural learning are novel to the best of our knowledge.

2 Strategy

Our approach to general intelligence is based on the idea that radically different nervous systems can be formed by the same underlying biological mechanisms, starting with different bodies and experiencing different sensory data. We model the following generic mechanisms for learning and decision-making:

1. Decision-making that aims for the satisfaction of multiple physiological needs [Roijers *et al.*, 2013].
2. Reinforcement learning that strengthens/weakens behavior associated with reward/punishment [Niv, 2009].
3. Hebbian learning, captured in the popular phrase “cells that fire together, wire together” [Baars and Gage, 2010].
4. Sequence learning, which is Hebbian learning with signal delay taken into account [Bear *et al.*, 2015].
5. Forgetting, as expressed in the phrase “use it or lose it” [Wixted, 2004].

In the present setting, the mechanisms 1-5 will only be used as an inspiration for our computational model. Nevertheless it is interesting to note that they seem to be ubiquitous in the animal kingdom [Roijers *et al.*, 2013; Niv, 2009; Baars and Gage, 2010; Bear *et al.*, 2015; Wixted, 2004].

We will define animats by specifying their sets of needs, sensors, and motors. They will then develop automatically by means of computational versions of the above-mentioned generic mechanisms for learning and decision-making. To model the environments of the animats, we use block worlds, e.g. worlds in the Minecraft computer game environment [Johnson *et al.*, 2016]. The animats can be put into the “skins” of Minecraft animals such as sheep, rabbits, and wolves. For instance, we can put the animats into the world shown in Figure 1 and study them as they strive to satisfy their needs for company, grass, and drinking water.

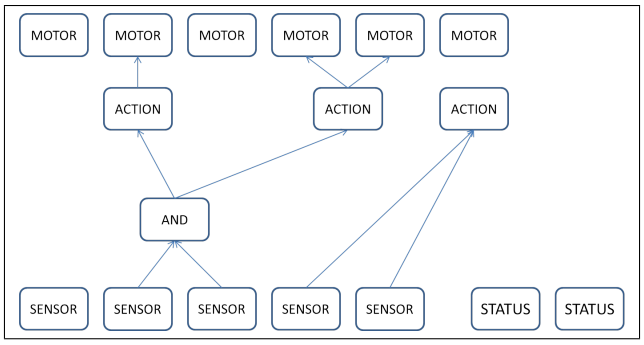

Figure 2: An example of a dynamic graph with some annotation on the arrows omitted. Note that ACTION nodes may be connected to 0, 1, 2, or more MOTOR nodes. The ACTION node that is not connected to any MOTOR node represents passivity.

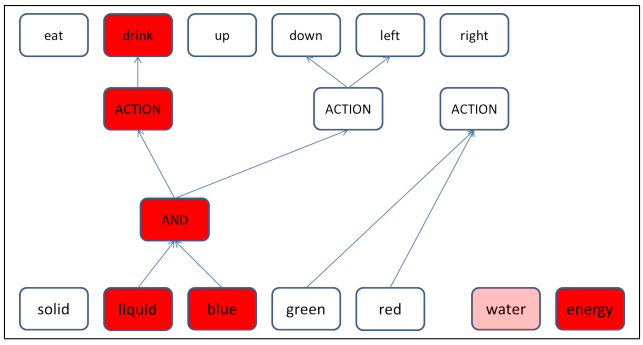

Figure 3: An example of an activity pattern on a graph. This is the same graph as in Figure 2, but with the optional node names displayed. Shades of red represent activity levels in $[0, 1]$, with white representing activity 0 and red activity 1.

Using Minecraft as an observatory enables us to study how different designs of the generic mechanisms affect the survival of the animats in different environments. Our goal is to design powerful generic mechanisms that enable a broad range of animats to survive in a broad range of worlds. Thus we may approach the challenge of constructing programs with general intelligence in a gradual fashion.

3 Computational model

This section presents the animats and their environments.

3.1 Worlds

Definition 1 *A* world *is a set of blocks. A* block *consists of:*

- *A block type (a natural number).*
- *A block position (a point in three-dimensional space $\mathbb{Z}^3$).*

3.2 Dynamic graphs

To model memory structures of animats, we use labeled graphs. The nodes of the graphs can be identified with formulas of temporal logic [Gabbay *et al.*, 1994]. In particular we use the binary modal operator SEQ that enables the construction of sequences. The formula p SEQ q is true at time t if p is true at $t-1$ and q is true at t.

Definition 2 *A* dynamic graph *consists of:*

- *A set of* nodes *labeled SENSOR, STATUS, MOTOR, AND, OR, NOT, SEQ, or ACTION and optionally given a name.*
- *A set of* arrows*, i.e. a binary relation on the set of nodes. Arrows pointing to ACTION nodes are labeled with two real values: local Q-values and R-values, as will be explained in subsection 3.6.*

Figure 2 shows a dynamic graph.

3.3 Activity

Definition 3 *An* activity *of dynamic graph G is an assignment of values in $[0, 1]$ to the nodes of G, subject to the restriction that non-STATUS nodes must be assigned values in $\{0, 1\}$.*

Figure 3 shows an activity. Time is modeled in discrete time steps or *ticks*. Input activity is transmitted from the environment to the SENSOR and STATUS nodes. Activity propagates to the other nodes as expected, except in the case of the ACTION nodes. The activity of ACTION nodes is specified by the policy given in Definition 12.

3.4 Animats

Definition 4 *An* animat *consists of:*

- *A dynamic graph G.*
- *An activity of G.*
- *A position: a point in the space $\mathbb{Z}^6$.*

The position system used here specifies the six degrees of freedom for positioning a rigid body in three-dimensional space: i.e. the three spatial coordinates together with the angles for pitch, roll, and yaw. This enables orienting animats towards e.g. a food source.

3.5 Top activity

Definition 5 (Perception nodes) *A node labeled SENSOR, AND, OR, NOT, or SEQ is called a* perception node*.*

The following notion plays a key role in both decision-making and learning:

Definition 6 (Top-active node) *Node $b \in G$ is top active at time t if:*

- *b is a perception node.*
- *b is active at t.*
- *There is no blue arrow that starts in b and ends in some other perception node b' that is also active at t.*

We use the notation $TA(t)$ for the set of top active nodes at t.

Figure 3 offers an example, where the red AND node is the only top-active node. In general, many nodes can be top-active at the same time. Intuitively, the top-active nodes together constitute a description of the present situation in terms of the given memory structure, at the maximum possible level of detail.

3.6 Local Q-learning

We work in the Multi-Objective Reinforcement Learning framework and define a local variant of Q-learning that also takes estimates of Q-value reliability into account.

Definition 7 (Status) *The* status *of the STATUS node i of the animat A at time t, $x_{i,t}$, is defined as the input to i at t.*

An animat with STATUS nodes water and energy could have $x_{water,t} = 0.8$ and $x_{energy,t} = 0.6$. The following measure reflects the overall well-being of an animat at a given moment.

Definition 8 (Vitality) *The* vitality *of the animat A at time t is defined as*

$$\min_{i \in STATUS} x_{i,t}.$$

An animat with $x_{water,t} = 0.8$ and $x_{energy,t} = 0.6$ has vitality 0.6 at t. If the vitality reaches 0, we say that the animat *dies*. The learning and decision-making mechanisms of the generic animat were designed with long-term vitality as the one and only goal.

Definition 9 (Rewards) *The* reward *of the animat A at time $t+1$ with respect to the STATUS node i is defined as $r_{i,t+1} = x_{i,t+1} - x_{i,t}$.*

Definition 10 (Reliability) *The* reliability *of the finite data set D is defined as $Rel(D) = 1/(SD(D)+1)$. Here SD is the standard deviation.*

We write a_t for the action that is performed at time t. Now we shall define the *local Q-values* $Q_{i,t}(b,a)$ and the *local reliability values* $R_{i,t}(b,a)$.

Definition 11 (Q-values and R-values) *At $t = 0$ we proceed as follows. Let*

$$Q_{i,0}(b,a) = 0 \text{ and } R_{i,0}(b,a) = 1$$

for all perception nodes b, ACTION nodes a, and STATUS nodes i.

At $t+1$ we proceed as follows. If $b \notin TA(t)$ or $a \neq a_t$, then we let $Q_{i,t+1}(b,a) = Q_{i,t}(b,a)$. If $b \in TA(t)$, then we let

$$Q_{i,t+1}(b,a_t) = Q_{i,t}(b,a_t) + \alpha \cdot (r_{i,t+1} + \gamma \cdot \Delta),$$

where Δ is

$$\max_{a \in Actions} \left[\frac{\sum_{b' \in TA(t+1)} Q_{i,t}(b',a) \cdot R_{i,t}(b',a)}{\sum_{b' \in TA(t+1)} R_{i,t}(b',a)} \right] - Q_{i,t}(b,a_t).$$

Here α and γ are parameters for learning rate and discount rate, respectively. Also let $R_{i,t+1}(b,a)$ be

$$Rel(\{Q_{i,t'}(b,a) : t' \leq t+1, a = a_{t'} \text{ and } b \in TA(t')\}).$$

Definition 12 (Policy) *Fix a real number λ and let*

$$\pi(t) = \operatorname*{argmax}_{a \in Actions} \left[\min_{i \in STATUS} (x_{i,t} + \lambda \cdot \Omega) \right],$$

where Ω is

$$\frac{\sum_{b \in TA(t)} Q_{i,t}(b,a) \cdot R_{i,t}(b,a)}{\sum_{b \in TA(t)} R_{i,t}(b,a))}.$$

The policy π selects actions aimed at keeping the vitality of the animat as high as possible, for as long as possible. It weighs up the animat's present status with expected status changes in the future. These expectations are in turn weighted by their estimated reliability. An animat with the two needs energy and water will be likelier to drink if water is its most urgent need. On the other hand, if its experience indicates that it would lose large quantities of energy by doing so, it might nevertheless refrain from drinking. Thus π is different from policies that first select the most urgent need and then look for actions that can satisfy that particular need without taking the other needs into account.

The decision-making algorithm is ε-greedy, where $\varepsilon \in [0,1]$. With probability ε it explores by activating a random set of MOTOR nodes (with higher probability for smaller sets) and with probability $1-\varepsilon$ it exploits by following the policy $\pi(t)$.

3.7 Structural learning

Definition 13 (Surprise) *The* surprise *of a perception node b at time $t+1$ w.r.t. the STATUS node i is defined as follows:*

$$z_{i,t+1}(b) = |Q_{i,t+1}(b,a_t) - Q_{i,t}(b,a_t)|$$

Definition 14 (Surprised) *An animat is* surprised *at time $t+1$ if $z_{i,t+1}(b) > Z$, for some STATUS node i and perception node b such that $R_{i,t}(b,a) > R$. Here Z and R are parameters regulating concept formation.*

When the animat is surprised, a new node will be added to the graph. The surprise indicates that the animat needs a more fine-grained ontology to be able to identify similar situations in the future.

Definition 15 (Node candidate) *A* node candidate *is an expression of the form*

- *b AND b', where $b, b' \in G$ are perception nodes and b AND $b' \notin G$, or*
- *b SEQ b', where $b, b' \in G$ are perception nodes and b SEQ $b' \notin G$.*

The node candidates do not belong to the graph, but they have local Q-values and R-values associated with different actions that are initiated and updated just like the local values of the perception nodes of the graph.

Suppose the animat gets surprised at $t+1$. Then the learning algorithm will consider the possibility of adding a new node. Let i be a randomly selected STATUS node subject to surprise at $t+1$. First, the algorithm explores the benefit of adding an AND node. To that end it searches for a node candidate b AND b' such that (i) both b and b' were top-active at t, and (ii) the prediction error

$$|Q_{i,t+1}(b \; AND \; b', a_t) - Q_{i,t}(b \; AND \; b', a_t)|$$

is minimal. If this prediction error is below a given threshold, the node b AND b' is added to the graph.

Second, if no AND node is added, the algorithm proceeds by exploring the benefit of adding a SEQ node. To that end it searches for a node candidate b SEQ b' such that (i) b was top-active at $t-1$, (ii) b was top-active at t, and (iii) the prediction error

$$|Q_{i,t+1}(b \; SEQ \; b', a_t) - Q_{i,t}(b \; SEQ \; b', a_t)|$$

is minimal. If this prediction error is below a given threshold, the node b SEQ b' is added to the graph. Whenever a new node is added to the graph, new node candidates are formed (by Definition 15).

4 Results

We have implemented the prototype system *Generic Animat*, which is available at `https://github.com/ni1s/animats`. The system is a simplification of the model described in the previous section. It is integrated with Minecraft via the Malmo interface [Johnson *et al.*, 2016]. To define an animat an initial animat must be specified. Any animat will do. For instance, its graph can be a *tabula rasa* with STATUS, SENSOR and MOTOR nodes only. It can also be arbitrarily complex and include, e.g. reflexes that connect directly from perception nodes to MOTOR-nodes (without passing via ACTION nodes). For simplicity we sometimes omit the MOTOR nodes from the diagrams and draw the ACTION nodes only.

Once it has been initialized, the animat can be put into an arbitrary world, where it will learn and make decisions autonomously. To measure the performance of an animat in a given world, we consider its *vitality curve* that maps vitality against time. The typical shape of the vitality curve is the square-root sign: The animat starts with high vitality, e.g. as a result of high vitality at birth. Then its vitality declines while it explores the world, learns, and consumes resources due to metabolism. As the animat begins to learn how to replenish its resources, the vitality curve turns up again. Then it may stay high if the environment so permits or decline gradually if the resources are being depleted. Next we will give several examples illustrating how the animats can learn how to eat, drink, move, navigate, and conceptualize their worlds.

4.1 Foraging

First let us consider a sheep animat that learns how to eat and drink. We assume that the sheep has two needs: energy and water. Its world is shown in Figure 4 and its initial memory in Figure 5. The interaction dynamics between animat and

Figure 4: A world containing three blocks: grass, sand, and water.

world are described in tables 1, 2, and 3. We ran a simulation with the sheep animat. Figure 6 shows the memory after convergence and Figure 7 shows its vitality curve.

	Grass	Sand	Water
red	0	1	0
green	1	0	0
blue	0	0	1

Table 1: Perception of the sheep animat.

4.2 Locomotion

Here we show a frog animat and a toad animat that learn how to move. First, let us consider a frog that lives in the world

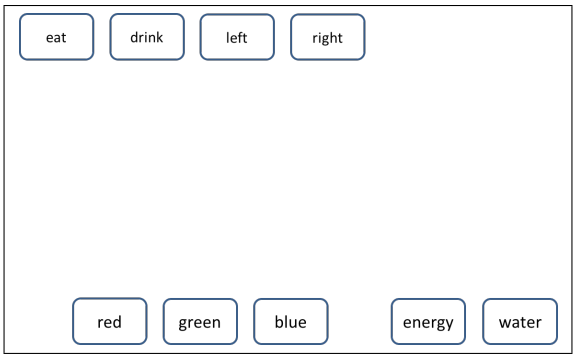

Figure 5: The memory of the sheep animat at the start. It has two STATUS nodes: "energy" and "water"; three SENSOR nodes: "red", "blue" and "green"; and four ACTION nodes.

energy	Grass	Sand	Water
eat	0.1	-0.05	-0.05
drink	-0.05	-0.05	-0.001
left	-0.001	-0.001	-0.001
right	-0.001	-0.001	-0.001

Table 2: Status changes for the STATUS node "energy".

water	Grass	Sand	Water
eat	-0.001	-0.05	-0.05
drink	-0.05	-0.05	0.1
left	-0.001	-0.001	-0.001
right	-0.001	-0.001	-0.001

Table 3: Status changes for the STATUS node "water".

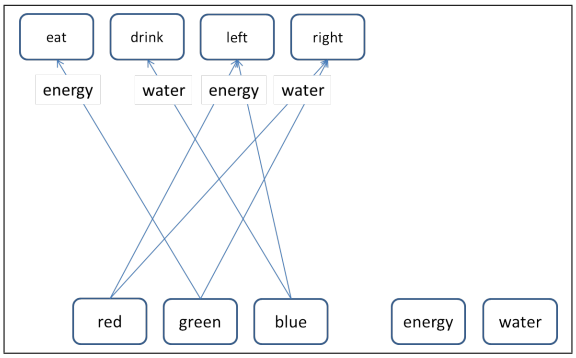

Figure 6: The memory of the sheep animat after convergence at time step 20. The displayed arrows indicate preferred actions in response to each need.

shown in Figure 8. Figure 9 shows the initial memory of the frog.

The frog consumes energy from metabolism at each time tick. It can only gain energy by moving to new blocks and ingesting the food that is available there. It can only move to a new block by jumping, i.e. by extending both hind legs simultaneously. The result of a 100-tick simulation is shown in Figure 10 and Figure 11.

Next, let us consider a toad that lives in the same world as

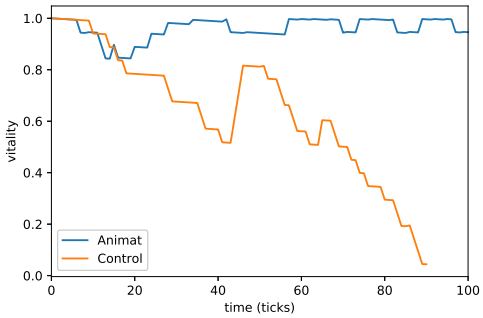

Figure 7: Vitality curve for the sheep animat, where vitality is the minimum of "energy" and "water". "Control" shows the same animat performing random actions. Multi-objective reinforcement learning helps the animat to learn a strategy that lets it survive: alternating between eating grass and drinking water depending on the most pressing need. "Control" quickly declines and dies.

Figure 8: The frog world. A one-dimensional world of green blocks.

the frog. The toad cannot jump; it can only move forward by alternately extending its hind legs. It can only extend legs that are folded and is equipped with proprioception sensors that indicate which hind legs are folded. The initial memory of the toad is shown in Figure 12. The result of a 100-tick simulation is shown in Figure 13 and Figure 14.

4.3 Navigation

Here we show that the generic animat can learn to navigate like a Braitenberg vehicle. We model a bee that navigates in a landscape with scenting flowers.

Consider a bee animat living in the world shown in Figure 15. When the bee visits a flower it collects the nectar, transforming the flower into grass. Each flower diffuses a scent into its surroundings. The intensity of the scent from a flower at a given distance follows the inverse-square law ($intensity \propto 1/distance^2$). The attractive and repulsive flowers have, respectively, positive and negative scents. The initial memory of the bee is shown in Figure 16.

The energy level of the bee changes depending on three factors: metabolism, scent intensity and whether nectar is collected.

We ran a 200-tick simulation of the bee in the bee world. Figure 17 shows a mid-simulation plot of how the bee has moved in the world so far.

The result of the simulation is shown in Figure 18 and Figure 19.

4.4 Spatial concept formation

Here we illustrate the benefit of structural learning in the case of AND node addition. Consider a sheep animat that lives in the world shown in Figure 20. Figures 5 and 21 show its memory at start and after convergence, respectively. Figure 22 shows how its vitality develops over time.

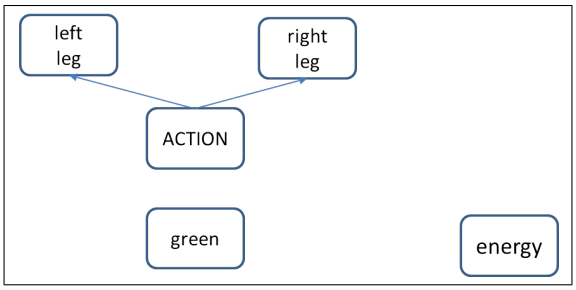

Figure 9: The initial memory of the frog. The frog has three actions: it can extend its left hind leg only, its right hind leg only, or both its hind legs (jump).

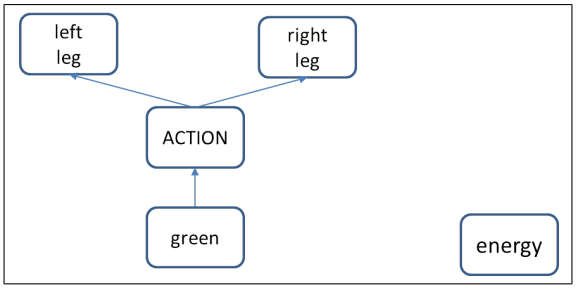

Figure 10: The memory of the frog after convergence. Convergence happens at time 3 when the frog has learned to prefer jumping.

4.5 Temporal concept formation

Here we illustrate the benefit of structural learning in the case of SEQ node addition. Again we consider a sheep that drinks and grazes, but this time the sheep lives in a world that contains both good water and bad, poisonous water. The problem is that the sheep cannot differentiate directly between the good and the bad water with its sensors. By learning that the bad water always appears in a certain context, in this case close to sand, the animat can learn to avoid drinking it.

The world is shown in Figure 23. The animat starts with the same memory as in the previous example (Figure 21). It adds the node "red SEQ blue" the first time a red block is encountered (one-shot learning). Figure 24 shows its vitality curve.

5 Scalability

Our model was constructed with the goal of combining full generality with scalability. In the interest of scalability, it avoids explicit representation of subsets of sensors (and sequences of such subsets) in favor of top-active nodes that represent partially defined states. In addition, the model was designed so that it only adds nodes reluctantly when it gets sufficiently surprised with respect to reward or punishment. To control the size of the network further, two additional mechanisms can be added to the present model: forgetting and compression.

5.1 Forgetting

Since the only nodes that can be added are AND nodes and SEQ nodes, it is sufficient to define forgetting rules for those

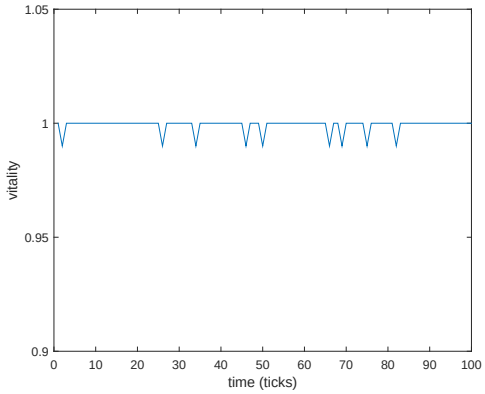

Figure 11: Vitality curve of the frog. The frog has learned how to jump after three ticks. The sporadic dips in the vitality curve are due to exploration of non-optimal actions.

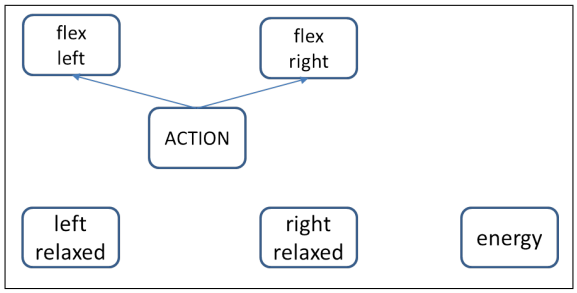

Figure 12: The initial memory of the toad. The toad has two sensors for proprioception and the same actions as the frog.

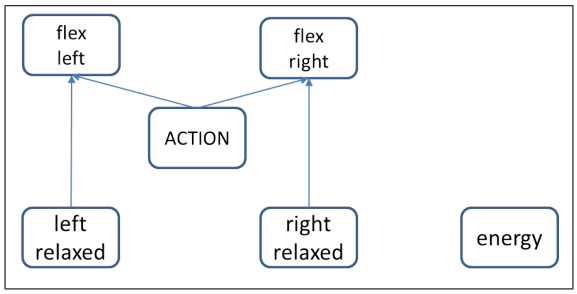

Figure 13: The memory of the toad after convergence. It has learned how to crawl.

nodes only. When a node is forgotten, it is removed from the network along with all perception nodes that are above it. Therefore the forgetting criteria must also take the nodes above into account. To determine whether a node b should be removed, two factors can be considered: (i) how often is b active and (ii) how similar are the Q-vectors of b and the nodes above b to those of the two predecessors of b. If b is rarely active and if the Q-vectors mentioned are similar to each other, those are indications that b should be removed. A rudimentary form of this strategy was developed in [Strannegård *et al.*, 2015].

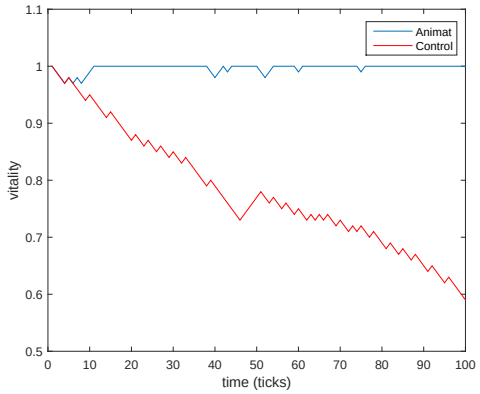

Figure 14: Vitality curve. "Animat" is the toad with proprioception sensors and "Control" is similar but with no proprioception sensors.

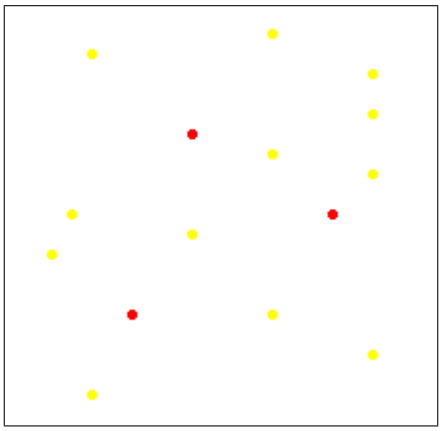

Figure 15: The bee world. This world is a two-dimensional 20 $\times$ 20 array of blocks with a torus topology. There are three types of blocks: attractive flowers, repulsive flowers, and grass. There are 12 attractive flowers (yellow dots) and 3 repulsive flowers (red dots).

5.2 Compression

On a standard neural network, it is relatively hard to define compression operations. On Boolean circuits, on the other hand, it is relatively easy. For instance, given a Boolean formula A, finding a logically equivalent formula A' of minimum size can be done by using a SAT-solver. Compression in the context of dynamic graphs is slightly different, but the problem is again reducible to a problem that can be handled by a SAT-solver. This compression can either be lossy (in the interest of generalization or to comply with size limits) or lossless, depending on which size limits are imposed. Propositional reasoning with bounded cognitive resources was considered in [Strannegård *et al.*, 2010].

6 Conclusion

We proposed a general computational model for animat learning and decision-making. Our model is inspired by the idea that many animals use the same fundamental principles for learning and decision-making although they have different

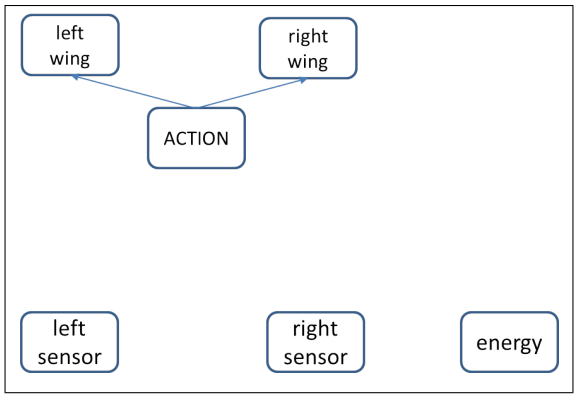

Figure 16: The initial memory of the bee. The bee has two sensors sensitive to the direction of the scent gradient of nectar from flowers in its 9x9 blocks neighborhood. One is active if the scent gradient is to the left of the animat or within 22.5 degrees to the front-right, and the other is active if the scent gradient is to the right of the animat or within 22.5 degrees to the front-left. Thus, there is an overlap of the sensitivity of the left and right sensors and they will both be active if the scent vector is within +/- 22.5 degrees of the forward direction of the animat.

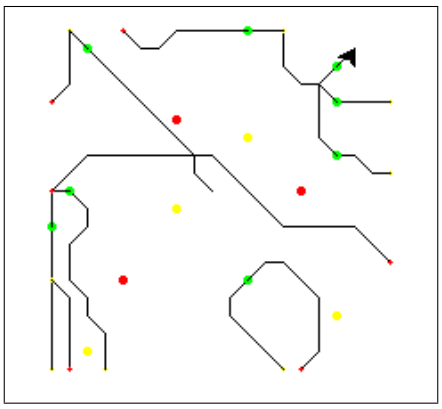

Figure 17: A mid-simulation snapshot of the bee's trajectory. The arrowhead represents the current position and heading of the bee. The black line shows the trajectory of the bee from the start. The yellow dots represent attractive flowers. Green dots represent flowers from which the bee has collected nectar. Red dots represent repulsive flowers.

bodies and live in different environments. The model is primarily based on multi-objective reinforcement learning combined with reward signals that reflect variations in the degree of need satisfaction. We also use a few "homegrown" ingredients: dynamic graphs for memory representation; top activity for perception; reliability and local Q-values for decision-making; and surprised-based structural learning for memory development.

Our animats are capable of starting with an arbitrary dynamic graph – e.g. a blank slate – and gradually build a memory structure that helps them keep their needs satisfied and survive. The animats are autonomous and fully general in the sense that they can adapt to arbitrary block worlds.

Our *Generic Animat* system is still in a prototype phase

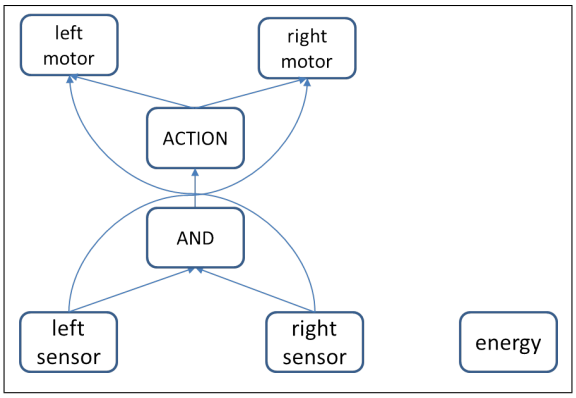

Figure 18: The memory of the bee after convergence. The bee has learned to turn left when the "left" sensor is active and right when the "right" sensor is active. An AND node was added and the bee has learned to move forward when this node is active.

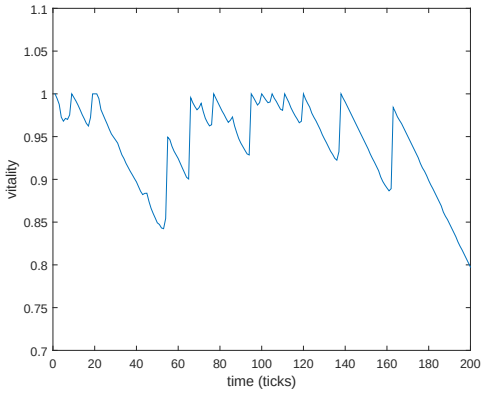

Figure 19: Vitality curve of the bee. An AND node is added to the action graph at tick 7. The first five times nectar is collected are at ticks 7, 17, 53, 64 and 75. The last available nectar is collected at tick 161.

Figure 20: The green block represents grass that is good to eat and the blue block water that is good to drink. The middle block represents a swamp where eating or drinking leads to vomiting and thus to decreased water and energy levels.

and much work remains to be done in terms of improving the learning and decision-making mechanisms as well as testing the system with respect to adaptability and scalability.

Acknowledgement

This research was supported by The Swedish Research Council, grants 2012-1000 and 2013-4873. C.S. is grateful to Martin Nowak for enabling a research visit to the Evolutionary Dynamics program at Harvard University.

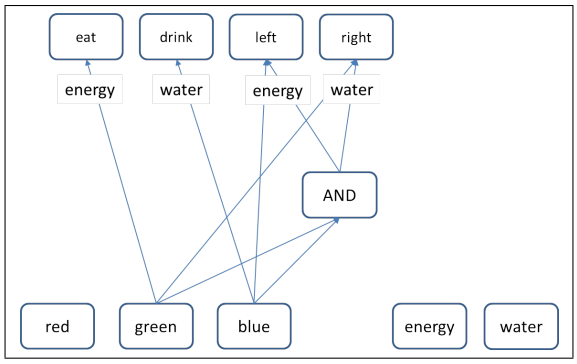

Figure 21: The memory after convergence (at time 25). The labels on the arrows indicate the preferred actions for the different needs when the lower node is top active. The AND node that was added automatically enables the animat to survive.

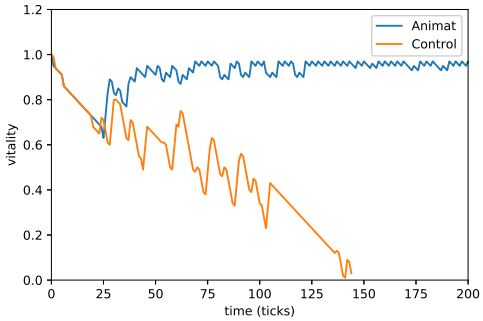

Figure 22: "Animat" is the sheep animat. "Control" is similar, but its dynamic concept formation is switched off. They both start out with a blank slate. "Animat" adds an AND node at time step 25. It manages to survive, while "Control" dies.

Figure 23: This world is a long path that begins with Water and Grass blocks, where the animat can learn to eat and drink. Then come the Poison blocks for the first time. Each Poison block has a Sand block to its left. This enables animats that are capable of sequence learning to differentiate between Water and Poison.

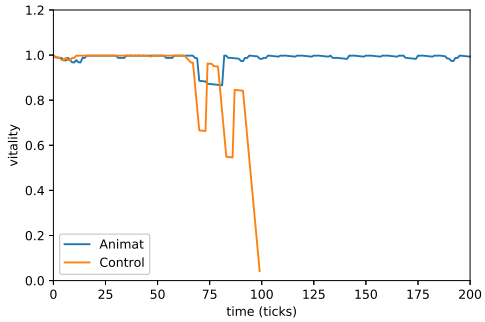

Figure 24: "Animat" is the sheep animat. "Control" is similar, but it has its capacity to add SEQ nodes switched off. The animats start with a blank slate. "Animat" adds a SEQ node at time step 75. It survives, while "Control", unable to learn sequences and contexts, dies at time step 100.

References

[Anderson *et al.*, 2004] J.R. Anderson, D. Bothell, M.D. Byrne, S. Douglass, C. Lebiere, and Y. Qin. An integrated theory of the mind. *Psychological review*, 111(4):1036, 2004.

[Angelova *et al.*, 2015] Anelia Angelova, Alex Krizhevsky, Vincent Vanhoucke, Abhijit S Ogale, and Dave Ferguson. Real-time pedestrian detection with deep network cascades. In *BMVC*, pages 32.1–32.12, 2015.

[Angluin, 1980] Dana Angluin. Finding patterns common to a set of strings. *Journal of Computer and System Sciences*, 21(1):46–62, 1980.

[Baars and Gage, 2010] B.J. Baars and N.M. Gage. *Cognition, brain, and consciousness: Introduction to cognitive neuroscience*. Academic Press, 2010.

[Bach, 2015] Joscha Bach. Modeling motivation in MicroPsi 2. In *International Conference on Artificial General Intelligence*, pages 3–13. Springer, 2015.

[Bear *et al.*, 2015] Mark F Bear, Barry W Connors, and Michael A Paradiso. *Neuroscience*. Wolters Kluwer, 2015.

[Dörner, 2001] Dietrich Dörner. *Bauplan für eine Seele*. Rowohlt Tb., 2001.

[Fahlman and Lebiere, 1990] Scott E Fahlman and Christian Lebiere. The cascade-correlation learning architecture. In *Advances in neural information processing systems*, pages 524–532, 1990.

[Gabbay *et al.*, 1994] Dov M Gabbay, Ian Hodkinson, and Mark Reynolds. *Temporal logic (vol. 1): mathematical foundations and computational aspects*. Oxford University Press, Inc., 1994.

[Goertzel *et al.*, 2014] Ben Goertzel, Cassio Pennachin, and Nil Geisweiller. The OpenCog Framework. In *Engineering General Intelligence, Part 2*, pages 3–29. Springer, 2014.

[Harrigan, 2016] Cosmo Harrigan. Deep learning for artificial general intelligence: Survey of recent developments. Presentation at the International Conference on Artificial General Intelligence, New York City, July 2016.

[Heerden, 1968] Pieter Jacobus van Heerden. *The foundation of empirical knowledge: with a theory of artificial intelligence*. Wistik, The Netherlands, 1968.

[Hernández-Orallo, 2016] José Hernández-Orallo. The measure of all minds: evaluating natural and artificial intelligence, 2016.

[Johnson *et al.*, 2016] Matthew Johnson, Katja Hofmann, Tim Hutton, and David Bignell. The malmo platform for artificial intelligence experimentation. In *International joint conference on artificial intelligence (IJCAI)*, page 4246, 2016.

[Konidaris and Barto, 2006] George Konidaris and Andrew Barto. An adaptive robot motivational system. In *SAB*, pages 346–356. Springer, 2006.

[Laird, 2012] John Laird. *The Soar cognitive architecture*. MIT Press, 2012.

[Langton, 1997] Christopher G Langton. *Artificial life: An overview*. MIT Press, 1997.

[LeCun *et al.*, 2015] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[Legg, 2008] Shane Legg. *Machine super intelligence*. PhD thesis, Università della Svizzera italiana, 2008.

[Lettman, 2006] Michael Lettman. Lecture 20: Reinforcement learning. https://www.cs.rutgers.edu/ mlittman/courses/cs442-06/lectures/20RL.pdf, 2006. Retrieved on June 26, 2107.

[Mnih and others, 2015] Volodymyr Mnih et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[Niv, 2009] Yael Niv. Reinforcement learning in the brain. *Journal of Mathematical Psychology*, 53(3):139–154, 2009.

[Nivel *et al.*, 2013] Eric Nivel, Kristinn R Thórisson, Bas R Steunebrink, Haris Dindo, Giovanni Pezzulo, M Rodriguez, C Hernandez, Dimitri Ognibene, Jürgen Schmidhuber, Ricardo Sanz, et al. Bounded recursive self-improvement. *arXiv preprint arXiv:1312.6764*, 2013.

[Pennachin and Goertzel, 2007] Cassio Pennachin and Ben Goertzel. Contemporary approaches to artificial general intelligence. *Artificial general intelligence*, pages 1–30, 2007.

[Roijers *et al.*, 2013] Diederik M Roijers, Peter Vamplew, Shimon Whiteson, Richard Dazeley, et al. A survey of multi-objective sequential decision-making. *J. Artif. Intell. Res.(JAIR)*, 48:67–113, 2013.

[Schmidhuber, 2015] J. Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, 2015.

[Spearman, 1904] Charles Spearman. "General Intelligence," objectively determined and measured. *The American Journal of Psychology*, 15(2):201–292, 1904.

[Strannegård and Nizamani, 2016] Claes Strannegård and Abdul Rahim Nizamani. Integrating symbolic and sub-symbolic reasoning. In *International Conference on Artificial General Intelligence*, pages 171–180. Springer, 2016.

[Strannegård *et al.*, 2010] Claes Strannegård, Simon Ulfsbäcker, David Hedqvist, and Tommy Gärling. Reasoning Processes in Propositional Logic. *Journal of Logic, Language and Information*, 19(3):283–314, 2010.

[Strannegård *et al.*, 2015] Claes Strannegård, Simone Cirillo, and Johan Wessberg. Emotional Concept Formation. In *Proceedings of the Eighth Conference on Artificial General Intelligence*, pages 166–176. Springer, 2015.

[Sutton and Barto, 1998] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 1998.

[Tuci *et al.*, 2016] Elio Tuci, Alexandros Giagkos, Myra Wilson, and John Hallam, editors. *From Animals to Animats. 1st International Conference on the Simulation of Adaptive Behavior*. Springer, 2016.

[Wang and Hammer, 2015] Pei Wang and Patrick Hammer. Assumptions of decision-making models in agi. In *Artificial General Intelligence*, pages 197–207. Springer, 2015.

[Wang and Li, 2016] Pei Wang and Xiang Li. Different conceptions of learning: Function approximation vs. self-organization. In *International Conference on Artificial General Intelligence*, pages 140–149. Springer, 2016.

[Wilson, 1986] Stewart W Wilson. Knowledge growth in an artificial animal. In *Adaptive and Learning Systems*, pages 255–264. Springer, 1986.

[Wilson, 1990] Stewart W. Wilson. The animat path to ai. In *Proceedings of the First International Conference on Simulation of Adaptive Behavior on From Animals to Animats*, pages 15–21, Cambridge, MA, USA, 1990. MIT Press.

[Wixted, 2004] J.T. Wixted. The psychology and neuroscience of forgetting. *Annu. Rev. Psychol.*, 55:235–269, 2004.

[Yoshida, 2017] Naoto Yoshida. Homeostatic agent for general environment. *Journal of Artificial General Intelligence*, 2017. DOI: https://doi.org/10.1515/jagi-2017-0001.