[2]**Institute of Solid State Physics**
Technische Universität Darmstadt
Hochschulstr. 6
64289 Darmstadt
Germany

Information Maximizing Agents in a Structured Environment

Informationsmaximierende Agenten in einer strukturierten Welt

Bachelor Thesis

Marco Möller

April 2008

Supervision:
Dr. Daniel Polani[1]
Prof. Dr. Barbara Drossel[2]

[1]Adaptive Systems Research Group, School of Computer Science,
University of Hertfordshire, UK

Abstract

This work analyzes representations of the world attained through an infomax principle by agents acting in a simple environment. The representations obtained by different agents differ to some extent from each other. This causes ambiguities in the way the environment is represented. An information-theoretic formalism was developed, able to extract a "common conceptualization" of the world for a group of agents. It turns out that the common conceptualization intuitively seems to capture much higher regularities or symmetries of the environment than the individual conceptualizations.

Notions for identification of these “weak” symmetries in the environment were formalized, with respect to "extrinsic" operations on the environment as well as with respect to "intrinsic" operations, i.e. the reconfiguration of the agent’s embodiment. In particular, using the latter formalism, an agent can be re-written to conform to the highly symmetric common conceptualization to a much higher degree than an unrefined agent; and that without having to re-optimize the agent from scratch. In other words, we can "re-educate" an agent to conform to the de-individualized common language of the agent group with comparatively little effort.

Zusammenfassung

Diese Arbeit analysiert Repräsentationen einer einfachen Welt, die durch Informationsmaximierung von in ihr agierenden Agenten gewonnen wurden. Von unterschiedlichen Agenten generierte Repräsentationen unterscheiden sich in einem gewissen Grad voneinander, was zu einer Doppeldeutigkeit der Repräsentation der Welt führt. Hierzu wurde ein informationstheoretischer Formalismus entwickelt, der in der Lage ist, eine “gemeinsame Klassifizierung” der Welt von einer Gruppe von Agenten zu bestimmen. Es stellt sich heraus, dass diese gemeinsamen Konzepte intuitiv eine höhere Regelmäßigkeit oder Symmetrie der Welt erfassen als die individuellen Konzepte.

Zur Identifizierung dieser “schwachen” Symmetrien wurden Begrifflichkeiten formalisiert, sowohl bzgl. “extrinsischer” Operationen auf der Welt als auch bzgl. “intrinsischer” Operationen, d.h. Abwandlungen des Körpers des Agenten. Speziell können wir mit letzterem Formalismus einen Agenten so abändern, dass er einem hoch symmetrischen gemeinsamen Konzept besser entspricht als ein unveränderter Agent, und das ohne ihn neu zu optimieren. Anders gesagt können wir diesen Agenten mit verhältnismäßig geringem Aufwand “umerziehen”, so dass er einem de-individualisierten gemeinsamen Konzept einer Gruppe von Agenten entspricht.

Contents

1 Introduction **6**

2 Background **7**
2.1 Notation . . . 7
2.2 Information . . . 7
2.3 Graphs . . . 10
2.4 Causal Bayesian Network . . . 11
2.5 Perception-Action Loop . . . 11
2.6 Initial Position Capturing . . . 16

3 Simulated Annealing **21**
3.1 Problem Description . . . 21
3.2 Algorithm . . . 21
3.3 Deterministic Mappings as Problem Type . . . 23
3.4 Experiments with Different Temperature Functions . . . 24

4 Common Concepts **26**
4.1 Concept . . . 26
4.2 Objective Common Concept . . . 27
4.3 Subjective Common Concept . . . 29
4.4 Neighbourhood of Symbols inside of a Concept . . . 30
4.5 Optimal Size for Common Concept . . . 34
4.6 Discussion . . . 36

5 Symmetries **39**
5.1 Extrinsic Symmetries . . . 40
5.1.1 Definitions . . . 40
5.1.2 Experiments . . . 42
5.2 Intrinsic Symmetries . . . 46
5.2.1 Definition . . . 46
5.2.2 Experiments . . . 48
5.3 Discussion . . . 56

6 Conclusion **57**

7 Future Work **58**

8 Acknowledgements **59**

A Appendix **60**

A.1 Superstition in Agent Groups 60
A.2 Additional Figures 62

B Index **68**

C List of Figures **70**

D List of Algorithms **71**

E References **72**

1 Introduction

We[1] are interested in how agents can model their environment based on informational considerations. Using infomax principles (Vergassola et al. (2007)), one obtains a classification or concept of a given environment for a given agent (Klyubin et al. (2007)). Our agent model is described in detail in Sec. 2.6 on page 16.

In such an infomax process, it turns out that different agents develop in general different representations of the environment. Even in very simple and highly symmetric scenarios, the resulting concept of the environment can considerably vary from agent to agent as a result of the infomax optimization (Sec. 3 on page 21).

This raises the issue of what the different concepts of those agents have in common, i.e. is it possible to develop a mutually compatible set of concepts (see e.g. Philipona and O'Regan 2006; Steels 1997; i Cancho and Solé 2003)? If so, what properties of the environment or the agents do such common concepts capture? How do they relate to the individual agents' concepts?

One hypothesis is that, while individual agents may suffer from symmetry breaking, common concepts will reveal symmetries (or other regularities) of the whole agent/environment system. We may now ask as well under which conditions the individual agents can relate to this expected higher regularity of the common language.

The technical challenges arising from these issues are manifold. We aim to find a description that is consistent with a fundamentally information theoretical picture of the agents and their environment. For this, one needs to suitably formulate the development of a common language of a set of agents. Furthermore, one needs to model the concept of regularity or weak symmetry in a suitable way.

This work settles on the results and techniques of Klyubin et al. (2007) presented in Sec. 2 on the facing page where we also reproduce some of those results in different environments. Section 3 on page 21 describes the optimization algorithm which will be used for all maximizations resp. minimizations. All used formulas have unique results, their computation is not part of this work. However, those techniques and algorithms are discussed in Möller (2008).

[1] I will use the first person plural throughout this text. Nevertheless, the experiments, results and writing of this thesis have been performed by myself.

2 Background

2.1 Notation

In the following, we introduce the basic notations.

Definition 2.1 Notation

Sets are denoted as calligraphic capital letters:
e.g. $\mathcal{A}, \mathcal{B}, \mathcal{C}, \ldots$

Elements of sets are denoted as small letters. They are contained in the according set:
e.g. $a \in \mathcal{A}, b \in \mathcal{B}, c \in \mathcal{C}, \ldots$

Size of sets is the number of elements in a set and denoted by $|.|$:
e.g. $|\mathcal{A}|, |\mathcal{B}|, |\mathcal{C}|, \ldots$

Power set of a given set $\mathcal{A}$ is the set of all subsets of $\mathcal{A}$ and denoted by
$2^{\mathcal{A}} = \{\mathcal{A}' | \mathcal{A}' \subseteq \mathcal{A}\}$

Random variables are represented by capital letters. They base on the accordingly sample space. A is a random variable assigning a probability $\Pr(A = a)$ to each value $a \in \mathcal{A}$.

Probabilities are short denoted like:
$p(a, b, \ldots) := \Pr(A = a, B = b, \ldots)$

Sums over elements of a set are written like:
$\sum_a f(a) := \sum_{a \in \mathcal{A}} f(a)$

Logarithms are always to the basis of 2:
$\log(a) := \log_2(a)$

2.2 Information

As explained before, we will use information theory to make statements about agents resp. dynamical systems. There are three advantages in doing so.

Information is an universal unit that is applicable on everything that can be described by probability theory. There is e.g. no difference between information of the memory and of the external world of an agent. It can be applied to both in the same way and the values are directly comparable.

Furthermore, information is independent of the concrete realization of the agent, i.e. it is architecture independent. For example, it can be applied as well to a biological system like a fly as to an artificial system like the aibo-robot.

Considering for example memory and the information coded inside, we do not impose any particular structure on the content representation. This is analogue to a "coordinate-free" view on the system like suggested in Klyubin et al. (2007). The choice of an appropriate basis for a vector space depends on the given problem. Polynomial interpolation is easily done in a Lagrange polynomial basis in contrast to the canonical basis while it is easier to use the latter to evaluate the polynomial. This example shows that there is not always a generic optimal basis. By using information theory, we avoid imposing a "coordinate system".

Information theoretic techniques do not allow (directly) to distinguish permutation of states inside of a random variable or between them. This aspect is another part of the coordinate-free view on the system.

Definition 2.2 Entropy related quantities

Let $A, B, \ldots$ be given random variables with a joint probability distribution. All following values have *bit* as unit.

Entropy (Shannon entropy) is a measurement for the amount of uncertainty:

$$H(A) := -\sum_{a} p(a) \cdot \log(p(a)) \tag{2.1}$$

$$H(A, B, \ldots) := -\sum_{a,b,\ldots} p(a, b, \ldots) \cdot \log(p(a, b, \ldots)) \tag{2.2}$$

$0 \leq H(A) \leq \log|\mathcal{A}|$ becomes maximal if A is uniformally distributed.
By adding random variables, the entropy $H(A, B) \geq H(A)$ can not shrink.

Conditional entropy is the amount of remaining entropy in A if the knowledge about B is completely used:

$$H(A|B) := H(A, B) - H(B) \tag{2.3}$$

$0 \leq H(A|B)$ vanishes if A is deterministically dependent of B.

Information distance is a mathematical metric on the space of random variables so that the triangle inequality holds (Crutchfield (1990)). It describes the amount of information which is not shared by two variables A and B:

$$D(A, B) := H(A|B) + H(B|A) \tag{2.4}$$

For a visualization of these quantities see Fig. 2.1 on page 10. One way to understand them is by applicating them on compression. Consider a sequence $a_1, a_2, \ldots$ of symbols $a_i \in \mathcal{A}$ coming out of a probabilistic source. This can be modeled by a chain of random variables $A_1, A_2, \ldots$ which are identically independently distributed (i.i.d.) with the same distribution as the random variable A. The entropy $H(A)$ is the minimal average number

of binary symbols/bits $\in \{0,1\}$ to encode one symbol of the output sequences of this source.

The conditional entropy can be understood in topic of channel coding as well. Imagine two scientists observing the same global nature phenomenon A (e.g. sunspots) and some local effects B of this (e.g. average outdoor temperature). To build a generic compression scheme, each scientist has to encode only a part of the local effects because the other one measured A as well and can use this knowledge of the possible effect from A on B to reconstruct the rest. In this case, it also becomes clear that $H(B|A) = 0$ vanishes if B is deterministically depending on A.

Definition 2.3 Information related quantities

Let $A, B, C, \ldots$ be given random variables with a joint probability distribution. All following values have *bit* as unit.

Mutual information between variable A and B is the amount of uncertainty which can be reduced in A if B is known and vice versa:

$$I(A;B) := H(A) - H(A|B) \tag{2.5}$$
$$= H(B) - H(B|A) \tag{2.6}$$
$$= H(A) + H(B) - H(A,B) \tag{2.7}$$

Conditional mutual information is the amount of information shared by A and B which is left if C is known:

$$I(A;B|C) = H(A|C) + H(B|C) - H(A,B|C) \tag{2.8}$$

Multiinformation is the amount of information shared by more than two variables $A_1, \ldots, A_n$

$$I(A_1;\ldots;A_n) := \left[\sum_{i=1}^{n} H(A_i)\right] - H(A_1,\ldots,A_n) \tag{2.9}$$

Conditional multiinformation is the amount of information shared by $A_1, \ldots, A_n$ which is left if C is known:

$$I(A_1;\ldots;A_n|C) = \left[\sum_{i=1}^{n} H(A_i|C)\right] - H(A_1,\ldots,A_n|C) \tag{2.10}$$

To understand the definition of mutual information, let us consider a noisy channel with an input distribution A (distribution of the symbols sent into the channel) and an output distribution B (distribution of the symbols leaving the channel). The average number of bits which can be transported without error over this channel is at most $I(A;B)$.

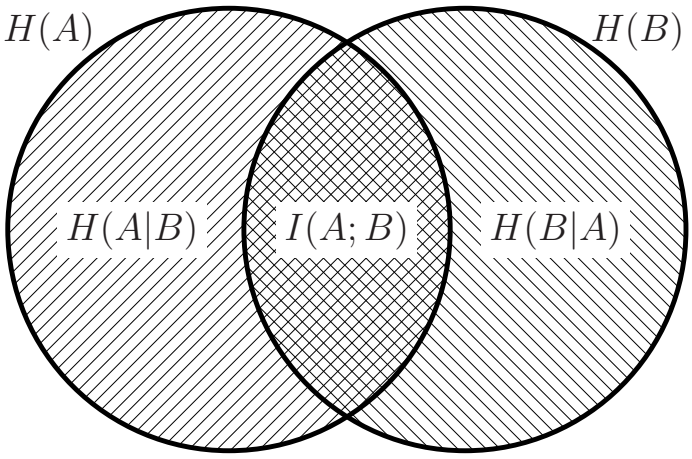

Figure 2.1: Relationship between entropy and information related quantities

The multiinformation is one possibility to generalize the mutual information to more than two variables. Let $A_1, \ldots, A_n$ be sources of symbol streams. If we compress every stream separately, we need in total $\sum_i H(A_i)$ bits for each tuple of symbols in average over time. If we compress tuples with one symbol of every source instead, we need $H(A_1, \ldots, A_n)$ bits for every tuple. So $I(A_1, \ldots, A_n)$ is the number of bits that we loose if we compress every source separately and do not use the correlations between them.[2]

2.3 Graphs

Definition 2.4 Directed graph

A *directed graph* $G := (N, E)$ is an ordered pair of two sets. The elements of N are called *nodes*. $E \subseteq N^2$ contains ordered pairs of nodes called *edges*.

In the common graphical notation elements of N are represented as boxes or circles. An edge $e = (n_1, n_2) \in E$ is represented by an arrow from node n_1 to node n_2.

Definition 2.5 Parents

The function

$$\text{parents} : N \to 2^N \tag{2.11}$$

is defined as

$$\text{parents}(n) \mapsto \left\{ n' \middle| \left(n', n\right) \in E \right\} \tag{2.12}$$

and gives for a node n the set of nodes where an edge to n starts from.

[2] This corresponds to the Kullback–Leibler divergence $D_{KL}\left(p(a_1, \ldots, a_n) \;||\; \prod_{i=1}^{n} p(a_i) \right)$

Definition 2.6 Path

A *path* $(n_1, \ldots, n_m)$ in a graph $G = (N, E)$ is an ordered tuple of nodes $n_i \in N$ connected by edges:

$$\forall i : (n_i, n_{i+1}) \in E \tag{2.13}$$

Definition 2.7 Cycle

A *cycle* $(n_1, \ldots, n_m, n_1)$ is a path in a graph with equal initial and end node.

Definition 2.8 Acyclic directed graph

A *acyclic directed graph* G is a graph which does not contain any cycles.

2.4 Causal Bayesian Network

In the following, we will use the technique of causal Bayesian networks to describe models of dynamical systems resp. agents like in Klyubin et al. (2007). The advantage of this method is the possibility of denoting dependencies between random variables graphically in a precise but intuitive way. Furthermore, this tool works perfectly together with information theoretical techniques since only random variables are used.

Definition 2.9 Causal Bayesian network

A *causal Bayesian network* is a directed acyclic graph whose nodes represent random variables, and whose edges represent conditional dependencies between the variables. The joint distribution of the whole network with nodes $X_1, \ldots, X_n$ is given by

$$p(x_1, \ldots, x_n) = \prod_{i=1}^{n} p(x_i | \text{parents}(x_i)), \tag{2.14}$$

where $p(x_i|\emptyset) := p(x_i)$ are unconditional probabilities.
The causal Bayesian network gives a causal interpretation to probability distributions which is not contained in the distribution itself. In difference to the bayesian network, the edges of the causal bayesian network do model a cause-and-effect relationship between the random variables.

2.5 Perception-Action Loop

The following definition is a generic model for an agent interacting with the world through a reading (sensor) and a writing (actuator) channel. To clarify the idea, some details about our special realization are given after each subsection.

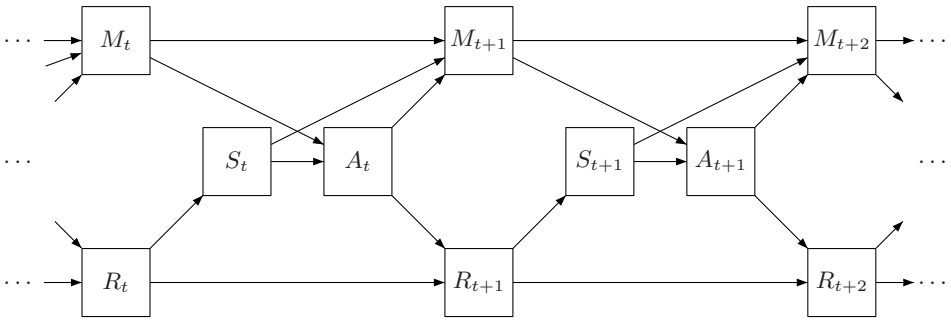

Figure 2.2: Perception-action loop as a causal Bayesian network

Definition 2.10 Perception-action loop

To define the *perception-action loop* and its components, we follow the presentation of Klyubin et al. (2007). To model the interaction between a single *agent* and his "surrounding" world, we use a *causal Bayesian model of the perception-action loop*. This model contains 4 *components* which will be precised in the following definitions:

- M - the *memory* of the agent (resp. *internal state*)
- A - the *action* performed by the agent
- R - the *world* where the agent is embedded in (resp. *rest of the system*)
- S - the *sensor* which provides the agent's access to the world

The relations between these components are visualized as causal Bayesian network in Fig. 2.2.

Definition 2.11 World

The *world* is the abstraction for a storage of information outside of the agent's mind.

In this work, we use a two dimensional infinite grid world and code the agent's position in the according random variable R, so the sample space is $\mathcal{R} = \mathbb{Z}^2$.

Definition 2.12 Time evolution

The simulation of the agent and his world R starts at $t = 0$, with a given initial agent memory M_0 and starting conditions in the world R_0. Then the calculation cycle starts and is repeated for every time step with the same conditional probability transition.

As long as not annotated differently, we use a deterministic memory initialized by $M_0 = 0$ and an uniformally distributed world initialized by $\mathcal{R}_0 = \{-5, \ldots, 5\}^2$. In our simulations, we use an infinite world so the agent is able to move out of this initial area.

Definition 2.13 Sensor

Depending on the state of the world R_t, the *sensor* value S_t will be calculated by using a conditional probability distribution

$$p(s|r). \tag{2.15}$$

Depending on the context, this conditional probability distribution will also be called sensor.

In the following experiments, we will use 2 different sensors, both with $\mathcal{S} = \{0 = \downarrow, 1 = \leftarrow, 2 = \uparrow, 3 = \rightarrow\}$. Consider for the first sensor, called "sensor s+" ($+ \approx$ possible sensor directions), a world with a "pheromone" source in the center $(0, 0)$ where the sensor always points to. At positions where 2 or 4 equally good sensor results exist (diagonals resp. center), all values are possible results with $p = \frac{1}{2}$ resp. $p = \frac{1}{4}$. A visualization for sensor s+ is shown in Fig. 2.3 on the following page. The second sensor is similar to the first one and called "sensor sq" (q $\approx$ quadro/4 pheromone sources). Its world contains not only a single source of pheromone but four of them at the positions $\{-5, 5\}^2$, compare Fig. 2.4 on the next page. Because of the bigger structure, experiments with this sensor use $\mathcal{R}_0 = \{-10, \ldots, 10\}^2$ as world initialization.

Definition 2.14 Action

Another element of the cycle is the *action*. It is calculated from the current sensor and memory values and described by a conditional probability distribution

$$p(a|m, s). \tag{2.16}$$

This probability distribution is comparable to a decision process of the agent "what to do next in this situation".

Definition 2.15 Actuator

An action A can now be applied on the current world state R and transform it. This *actuator* is described by

$$p(r_{t+1}|a_t, r_t). \tag{2.17}$$

An action can be imagined as a command to an actuator.

Our experiments used an actuator called "actuator a+". It has 4 different actions $\mathcal{A} = \{0 = \downarrow, 1 = \leftarrow, 2 = \uparrow, 3 = \rightarrow\}$ which move an agent to one of his next neighbour fields. The 4 possible actions of this actuator are visualized in Fig. 2.5 on page 15.

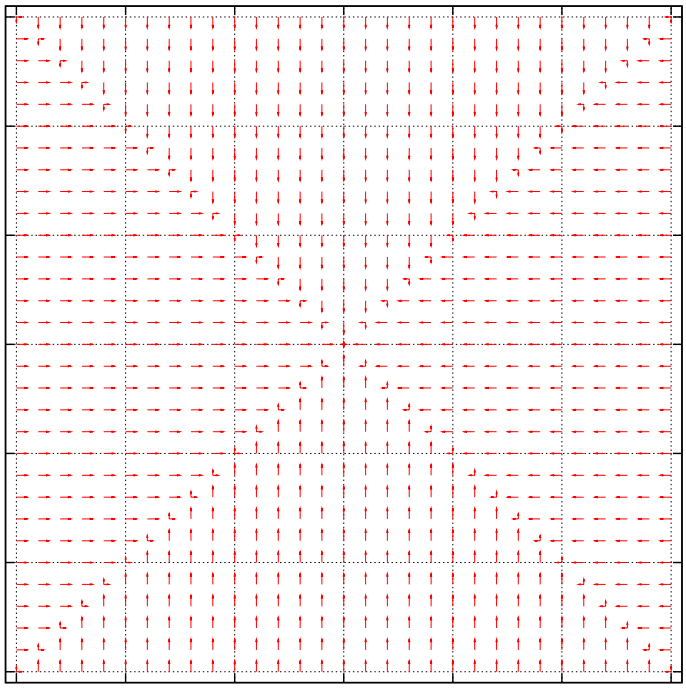

Figure 2.3: Sensor s+
The length of the arrows corresponds to $\frac{1}{2}$·probability of the sensor value on this position.

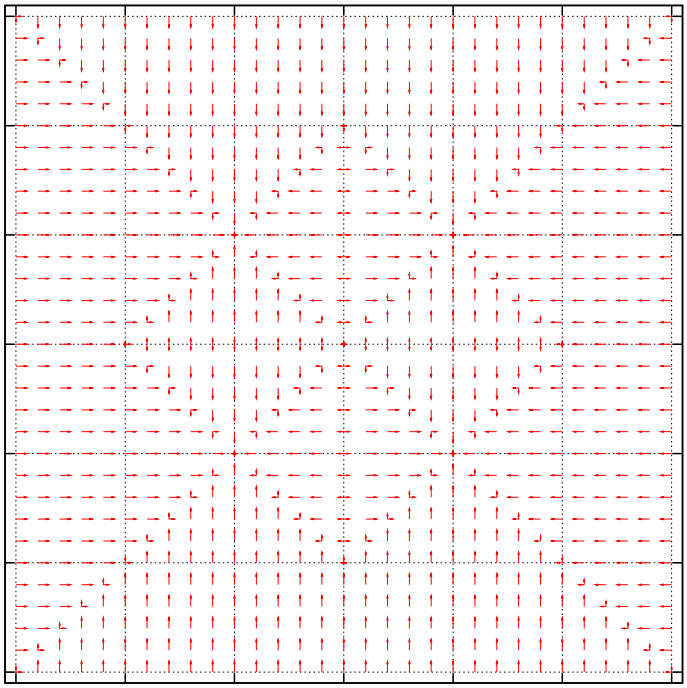

Figure 2.4: Sensor sq
The length of the arrows corresponds to $\frac{1}{2}$·probability of the sensor value on this position.

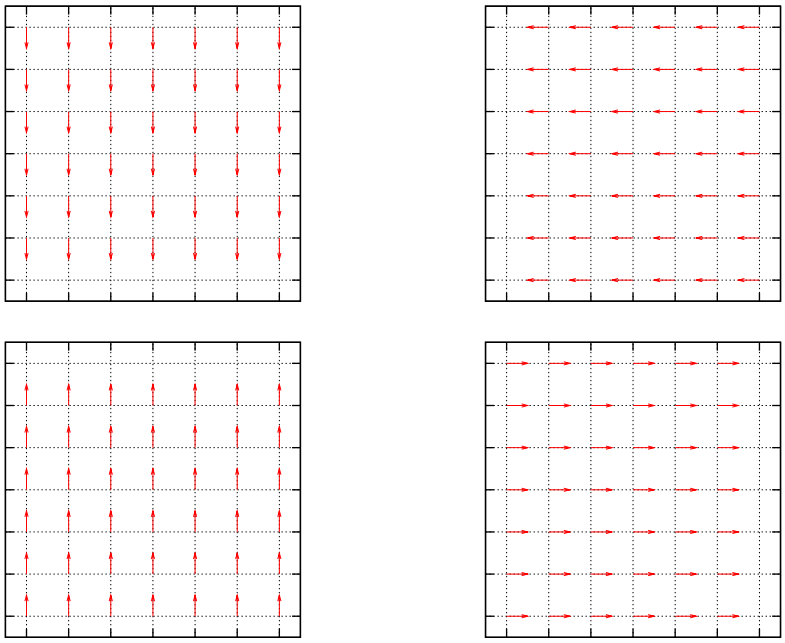

Figure 2.5: Actuator a+
The length of the arrows corresponds to $\frac{1}{2} \cdot p\left(r_{t+1}|r_t, a_t\right)$, half the probability for the action if the action command is given (each sub-figure is one command a_t). The arrows are pointing to the target field of the movement.

Definition 2.16 Memory

The *memory* M_t is a representation of the state of the agent's mind at time $T = t$. The next memory M_{t+1} will be calculated from the current memory M_t, the current sensor value S_t and the current chosen action A_t

$$p\left(m_{t+1}|m_t, s_t, a_t\right). \tag{2.18}$$

Definition 2.17 Controller

The Eq. 2.16 on page 13 and Eq. 2.18 form together the agent's *controller* which is the total abstraction of the "decision process in the agent's mind".

Definition 2.18 Embodiment

The sensor S and the actuator A form the agent's body. A concrete choice of sensor and actuator is called *embodiment*.

On the first view, this model looks a little bit "simple" because there is just one sensor, one actuator and one memory. Indeed that makes the model much more generic! For example, a model with two sensors $S^{(1)}$ and $S^{(2)}$ can be simulated by our model with $S = S^{(1)} \otimes S^{(2)}$. This is applicable to memory and actuator in the same way.

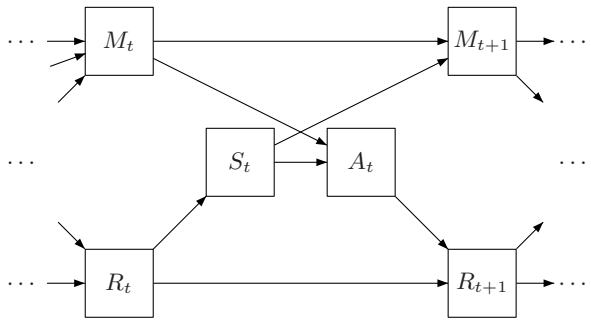

Figure 2.6: Perception-action loop with deterministic controller as a causal Bayesian network

2.6 Initial Position Capturing

In this subsection, we presented the basic model (suggested by Klyubin et al. (2007)) where our later experiments settle on. In the following we describe a reduced model of the perception action loop which is better computable.

Definition 2.19 Deterministic controller

A *deterministic controller* is a deterministic function

$$(m_{t+1}, a_t) = f(m_t, s_t) \tag{2.19}$$

that replaces Eq. 2.16 on page 13 and Eq. 2.18 on the preceding page in the perception-action loop. The corresponding causal Bayesian network is shown in Fig. 2.6.

A deterministic controller is equivalent to a Mealy machine[3]. For each state (memory value) and each input value (senor value), there is a following state (Eq. 2.18 on the preceding page) and a corresponding action (Eq. 2.16 on page 13), compare Fig. 2.8 on page 19.

Now we need a criteria for a good controller resp. a task for our agent. The following task was suggested by Klyubin et al. (2007) and we will use it as basic task throughout this work.

[3] A Mealy machine is a deterministic finite state machine that has output values (actions) chosen deterministically according to the last memory value and the input value (sensor). So it can be represented as a directed graph with the memory values as nodes and tuples of sensor and actions values denoted on the edges.

Definition 2.20 Initial position capturing

Given a perception-action loop with fixed sensor and actuator. The goal of the *initial position capturing* is to maximize

$$\max \stackrel{!}{=} I\left(R_0; M_t\right) \tag{2.20}$$

by varying the deterministic controller where t is a fix constant representing the "end of the run".

An good initial position capturer is informally spoken an agent who walks through the world and tries to collect as much information as possible about his initial position and memorizes it. By applying simulate annealing to this problem statement, we find easily a solution for a good agent. In Fig. 2.7 on the next page, some typical good solutions are shown.

In Klyubin et al. (2007), it was suggested that deterministic controllers were not a significant restriction. This choice is very intuitive because otherwise, we would only bring additional randomness into our system what would worsen $I\left(R_0; M_t\right)$ i.e. the performance of learning the initial position into the memory.

What are the deeper qualities of a good solution? The different memory values at the end of the run represent different classes of "similar" starting positions. These classes (let us say *symbols*, see Def. 4.1 on page 26) are not random, they are structured by the agent's ability of interaction with the world. For example the agent with a memory size of 4 has created the *symbols* for "m=0→west-triangle", "m=1→south-triangle", "m=2→north-triangle", "m=3→east-triangle". These symbols together form the *concept* (see Def. 4.1 on page 26) of this agent. In a bad solution (for example see Fig. 2.9 on page 19), the symbols are not structured that well.

How "similar" is a found good solution to an optimal solution? And how "close" are two solutions that have a similar utility value? Those questions are important and a possible answer to the latter will be discussed in Sec. 4 on page 26. For the first question, take a look at the utility value. The quality of a solution can be estimated by the upper bound of the utility function

$$I\left(R_0; M_t\right) = H\left(M_t\right) - H\left(M_t|R_0\right) \leq H\left(M_t\right) \leq \log|\mathcal{M}| \tag{2.21}$$

$$I\left(R_0; M_t\right) = H\left(R_0\right) - H\left(R_0|M_t\right) \leq H\left(R_0\right) \leq \log|\mathcal{R}| \tag{2.22}$$

$$I\left(R_0; M_t\right) \leq \min\left[\log|\mathcal{M}|, \log|\mathcal{R}|\right]. \tag{2.23}$$

If $R_0 \to M_t$ is a deterministic mapping, $H\left(M_t|R_0\right)$ vanishes. If additionally the same amount of values of R_0 are mapped to each m_t (let us say this mapping is uniformally distributed), M_t would be uniformally distributed as well since we chose R_0 uniformally

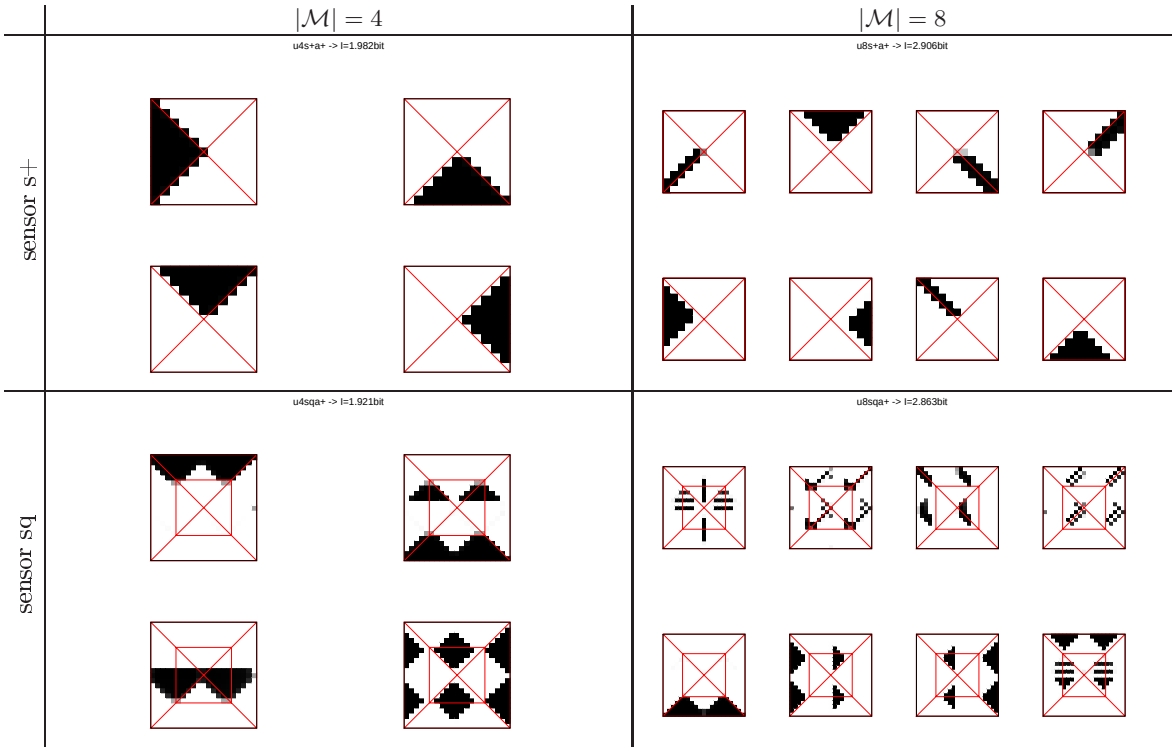

Figure 2.7: Solution of initial position capturing
Given an uniformally distributed $\mathcal{R}_0 = \{-5, \ldots, 5\}^2$ and $t = 15$. An agent with actuator a+ and sensor s+ (first row) and sq (second row) is used. Memory size is $|\mathcal{M}| = 4$ resp. 8 with utility value $I(R_0; M_t)$ written in the single titles.
Each picture represents $c_{m_t} \cdot p(r_0|m_t)$ in different grey levels (0=white, 1=black) with $m_t = \{1, \ldots, |\mathcal{M}|\}$ where c_{m_t} is a scaling constant so that the darkest grey in every picture becomes black. From now on, we will use this method to visualize conditional probabilities.
These results look very representative for the given problem as further investigations in Sec. 4 on page 26 will show. For sensor s+, they are equal to those of Klyubin et al. (2007).

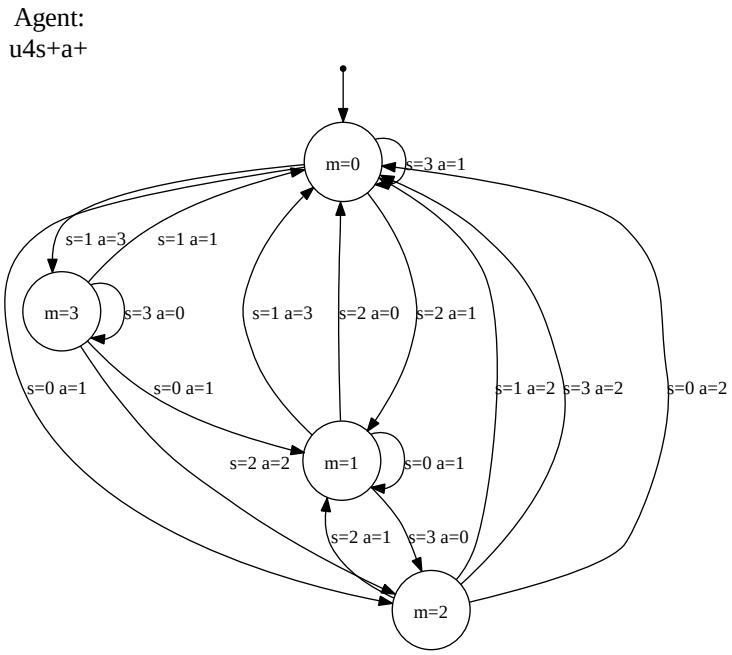

Figure 2.8: Example for a Mealy machine representation of a deterministic controller
This agent has $|\mathcal{M}| = 4$ and uses sensor s+ and actuator a+. $m \in \{0, \ldots, 3\}$ are the different possible values for the memory, $s \in \mathcal{S} = \{0 = \downarrow, 1 = \leftarrow, 2 = \uparrow, 3 = \rightarrow\}$ for the sensor and $a \in \mathcal{A} = \{0 = \downarrow, 1 = \leftarrow, 2 = \uparrow, 3 = \rightarrow\}$ for the actuator. He belongs to the upper left result in Fig. 2.7 on the preceding page.

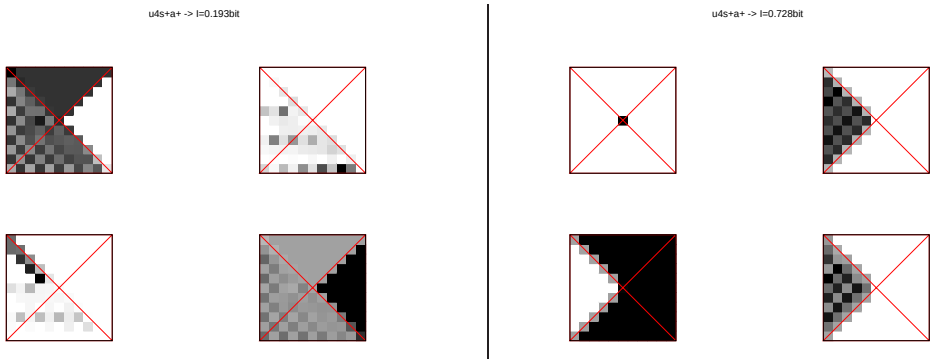

Figure 2.9: Bad solution of initial position capturing
The pictures show two bad solutions of an initial position capturer with actuator a+, sensor s+ and $|\mathcal{M}| = 4$. The left one is a random solution with $I(R_0; M_{15}) = 0.193$ bit and the right one is a Simulated Annealing solution after 29 iterations with $I(R_0; M_{15}) = 0.728$ bit.

distributed[4]. The uniformally distribution of this mapping is only possible if $\frac{|\mathcal{R}|}{|\mathcal{M}|} \in \mathbb{N}$. Since we use[5]

$$|\mathcal{M}| \ll |\mathcal{R}| \tag{2.24}$$

this is at least approximately

$$\frac{\left\lfloor \frac{|\mathcal{R}|}{|\mathcal{M}|} \right\rfloor}{\left\lceil \frac{|\mathcal{R}|}{|\mathcal{M}|} \right\rceil} \approx 1 \tag{2.25}$$

true. In this case, all inequalities in Eq. 2.21 on page 17 become equalities

$$I\left(R_0; M_t\right) = H\left(M_t\right) = \log|\mathcal{M}|\,. \tag{2.26}$$

That means that every agent who is producing a (almost) deterministic, uniformally distributed mapping $R_0 \to M_t$ is (almost) optimal.

In a visualization like Fig. 2.7 on page 18, this means that the size of the initial position for every final memory is equal (uniformaly distributed mapping) and there are no grey areas (deterministic mapping). Since the only information the memory can get about the world is filtered by the sensor, these mappings are somehow similar to the sensor. This fact is further investigated in Sec. 5 on page 39.

In Fig. 2.7 on page 18, we reached in case of $|\mathcal{M}| = 4$ the utility values 1.982 bit (sensor s+) resp. 1.921 bit (sensor sq). That is both very close to the bound of 2 bit. In case of $|\mathcal{M}| = 8$, the utility values of 2.906 bit (sensor s+) resp. 2.863 bit (sensor sq) are still good but not that close to the bound of 3 bit. By increasing the memory size, the distance to the boundary is increasing as well until $|\mathcal{M}| \geq |\mathcal{R}|$. Then, the new boundary will be $\log|\mathcal{R}|$.

[4] Compare below Def. 2.12 on page 12.

[5] If $|\mathcal{M}| \gtrapprox |\mathcal{R}|$, there is no need to find "concepts" in the world. Each position $r \in \mathcal{R}$ could be an own concept.

3 Simulated Annealing

To find a good solution for a given problem, we use in this thesis a variant of Simulated Annealing. Subject of this section is to define what is meant by "a good solution" and to explain how to find it.

3.1 Problem Description

Definition 3.1 Problem and optimal solution

Consider a set $\mathcal{P}$ called *problem space* and a *utility function* $f : \mathcal{P} \to \mathbb{R}$. A *neighborhood function* $nb : \mathcal{P} \to 2^{\mathcal{P}}$ must be defined as well. The goal is to find an *optimal solution* $p^\star \in \mathcal{P}$ with maximal utility value

$$\forall p \in \mathcal{P} : \ f(p) \leq f(p^\star). \tag{3.1}$$

To find a solution $\tilde{p}$ with a minimal utility function value $f(\tilde{p})$, we just have to maximize $\tilde{f} := -f$.

If the problem space $\mathcal{P}$ is large, it is not always possible to find the optimal solution $p^\star$ in acceptable computing time. The following variant of *Simulated Annealing* will return an approximation p' of $p^\star$ so that $|f(p') - f(p*)| \leq \epsilon$ is "small". In the following, such a p' is called *good solution*. Indeed, boundaries for the utility function in most of our experiments are easy to find, so there is a way to estimate the value of ϵ. But there are also cases where we do not have (good) boundaries, so we tried different runtimes to get a "feeling" for the quality of the solutions. In some cases, it is intuitively clear what an optimal solution looks like, so we can use them as well as test cases for the algorithm.

3.2 Algorithm

In Alg. 3.1 on the next page, a pseudo code listing of the used algorithm is shown. The idea of Simulated Annealing is to start with a solution and replace it iteratively in some cases by a neighbour solution:

1. If the neighbour solution has a higher utility value, it replaces the solution.
2. If 1. did not hold, the solution is replaced by the neighbour solution with a probability $p = e^{-\frac{\text{difference of the objective}}{\text{temperature}}}$.

The temperature decreases during the search for an optimal solution and becomes very small at the end. So 2. is less and less applied towards the end of the run.

For the choosing the temperature during the computing steps, we use the following function:

$$\text{temperature} = T_{min} \cdot \frac{\text{total interations}}{\text{current iteration number}} \tag{3.2}$$

Algorithm 3.1 Used variant of Simulated Annealing in pseudo code

```
p' ∈ P simulatedAnnealing(p_start ∈ P, speedBase∈ ℝ){
    T_min = 1/|nb(p_start)|;
    n = speedbase·log_{|nb(p_start)|}(|P|);
    p' = p_start;
    p_tmp = p_start;
    c' = f(p_tmp);
    c_tmp = f(p_tmp);
    for (i = 1; i ≤ n; i++){
        p_next =getRandomlyAnElement(nb(p_tmp));
        c_next = f(p_next);
        if (c_next > c_tmp){
            c_tmp = c_next;
            c' = c_next;
            p_tmp = p_next;
            p' = p_next;
        } else {
            T = T_min · n/i;
            if (getRandomNumberInInterval01() ≤ e^{(c_next − c_tmp)/T}){
                c_tmp = c_next;
                p_tmp = p_next;
            }
        }
    }
    return p';
}
```

The value of T_{min} is semi-empirically chosen as

$$T_{min} = \frac{1}{|nb(p_{start})|}. \tag{3.3}$$

This is necessary to assure that only better solutions (i.e. with a higher utility value) are taken at the end of the run. For a larger neighborhood, more solutions have to be tested before a better one is found. So the temperature has to be lowered to avoid that one of the worse solutions is taken.

The total number of iteration steps is calculated by

$$\text{total iterations} = \text{speedbase} \cdot \log_{|nb(p_{start})|}(|\mathcal{P}|) \tag{3.4}$$

which is on the one hand manually scalable to get a higher precision and on the other hand automatically adaptive to the given problem. $\log_{|nb(p_{start})|}(|\mathcal{P}|)$ is approximately proportional to the minimal number of steps to get from one solution to any other: $|\mathcal{P}| = |nb(p_{start})|^{\text{min. steps}}$. The parameter speedbase is set in the most experiments to our

default value 1000. We use this parameter to trade-off between fast and inaccurate (small numbers) computation and slow and very accurate (large numbers) computation in terms of expected distance between the utility values of the found good solution p' and the optimal solution $p*$.

3.3 Deterministic Mappings as Problem Type

In this work, the problems we search good solutions for are all from a certain type. In general, they are deterministic mappings

$$\mathcal{X}_1 \times \cdots \times \mathcal{X}_n \to \mathcal{Y}_1 \times \cdots \times \mathcal{Y}_m \tag{3.5}$$

where $\mathcal{X}_1, \ldots, \mathcal{X}_n, \mathcal{Y}_1, \ldots, \mathcal{Y}_m$ are finite sets. The deterministic mappings used in this thesis are[6]:

- Deterministic controller $\mathcal{M}_t \times \mathcal{S}_t \to \mathcal{M}_{t+1} \times \mathcal{A}_t$ from Eq. 2.19 on page 16.
- Objective common concept $\mathcal{X} \to \mathcal{Y}_*$ from Def. 4.3 on page 27.
- Subjective common concept $\mathcal{Y}_1 \times \ldots \times \mathcal{Y}_m \to \mathcal{Y}_*$ from Def. 4.4 on page 29.

The problem space $\mathcal{P}$ is the set of all mappings according to Eq. 3.5. The size[7] of this problem space is

$$|\mathcal{P}| = |\mathcal{X}_1 \times \cdots \times \mathcal{X}_n \to \mathcal{Y}_1 \times \cdots \times \mathcal{Y}_m| \tag{3.6}$$

$$= \left(\prod_{i=1}^{m} \mathcal{Y}_i \right)^{\prod_{j=1}^{n} \mathcal{X}_j} . \tag{3.7}$$

For a given mapping $p : \mathcal{X}_1 \times \cdots \times \mathcal{X}_n \to \mathcal{Y}_1 \times \cdots \times \mathcal{Y}_m$, a mapping p' is a neighbour if there is exactly one tuple $(x_1, \ldots, x_n) \in \mathcal{X}_1 \times \cdots \times \mathcal{X}_n$ so that $p'|_{\mathcal{X}_1 \times \cdots \times \mathcal{X}_n \setminus \{(x_1,\ldots,x_n)\}} = p|_{\mathcal{X}_1 \times \cdots \times \mathcal{X}_n \setminus \{(x_1,\ldots,x_n)\}}$ and $p'(x_1, \ldots, x_n) = (y_1, \ldots, \tilde{y}_i, \ldots, y_m)$ differs in exactly one place $\tilde{y}_i \neq y_i$ from $p(x_1, \ldots, x_n) = (y_1, \ldots, y_i, \ldots, y_m)$.

This neighbourhood has the following size[8]:

$$\forall p \in \mathcal{P} : |nb(p)| = \left(\prod_{j=1}^{n} \mathcal{X}_j \right) \cdot \left(\sum_{i=1}^{m} (\mathcal{Y}_i - 1) \right). \tag{3.8}$$

Our implementation picks the neighbour randomly and takes care that every neighbour has the same probability to be selected.

[6] In Möller (2008), we also used others.

[7] In our experiments, the search space size varies over a large range from 2^{32} (Sec. 2.6 on page 16) to about 2^{4000} (Sec. 4.3 on page 29).

[8] In our experiments, the neighbourhood size varies between 2^6 (Sec. 2.6 on page 16) and 2^{14} (Sec. 4.3 on page 29).

3.4 Experiments with Different Temperature Functions

To compare different temperature curves, we used an experimental setting from Sec. 4.2 on page 27 and Fig. 4.3 on page 31, but the details are not necessary to understand the following. For the given utility function, we tested the temperature from Eq. 3.2 on page 21 which is a function of the current iteration number:

$$T = \text{temperature}\,(\text{iteration}\#) \tag{3.9}$$

We compared the result with those of $T_1 = 10 \cdot T$, $T_2 = 0.1 \cdot T$, $T_3 = 0.01 \cdot T$ and $T_4 = 0$. They are shown in Fig. 3.1 on the next page. The second figure in the first row conforms to T. Even if this seems not to be the best temperature curve for this problem, it is indeed: T_4 simulates "hill-climbing" what means that only better solutions are picked and never a worse one. In this experiment, T_4 leads to the same solution as T but in a little shorter time. But T is more stable than T_4 as we can see at T_3: Shortly after the beginning of its run, the temperature is so low that only better (or equal) solutions are taken, so it behaves like T_4. But it "freezes" to early into a suboptimal solution. So T_3 shows us that for another random number sequence (used by the neighborhood function) or for another starting solution the hill-climbing method might not be optimal. In total, our optimization method seems to be better than hill-climbing.

To have more evidence that we got an almost optimal solution, we can not only look at one annealing curve but we have to compare different runs or use additional knowledge about the problem.

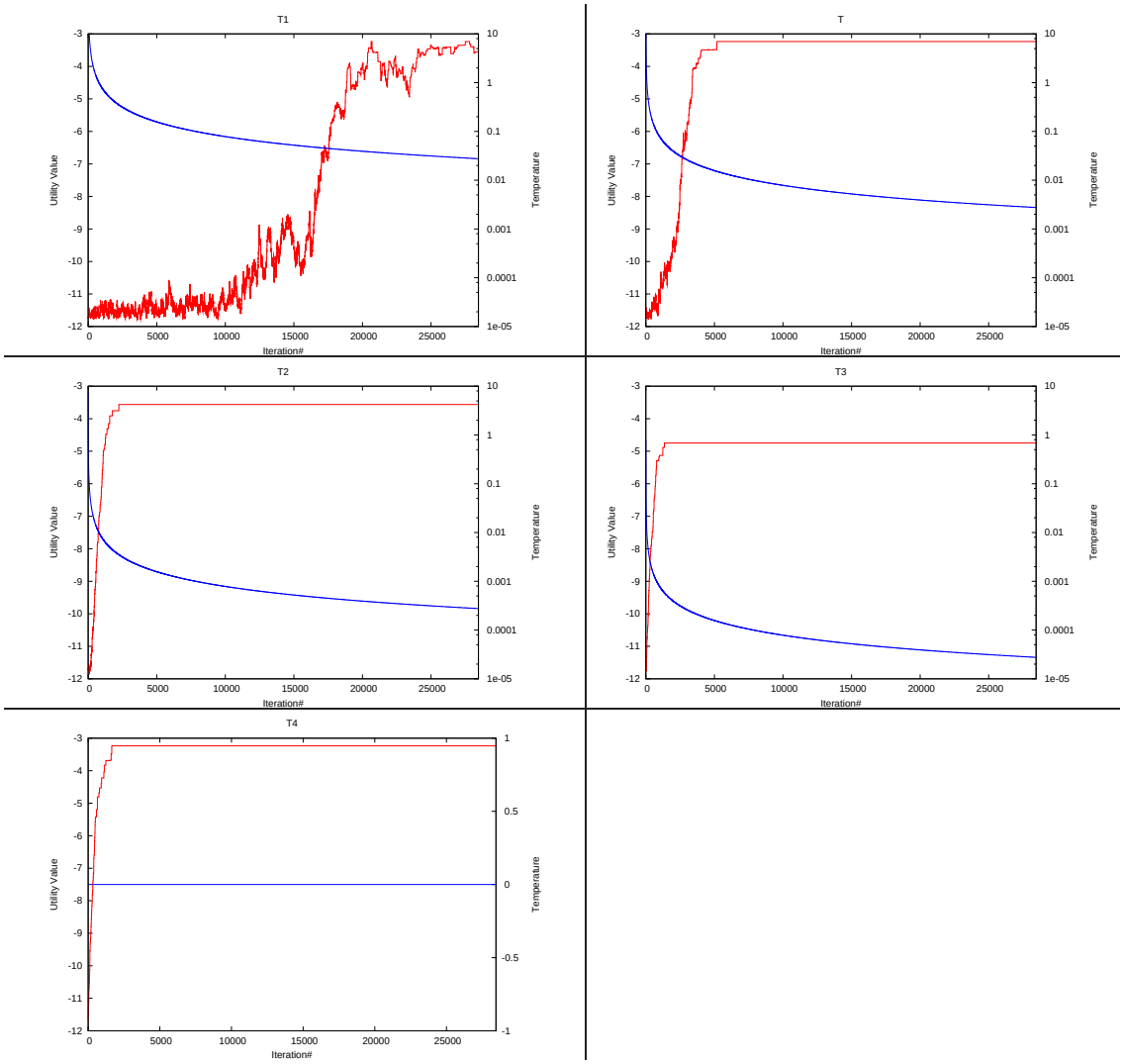

Figure 3.1: Comparison of different temperature functions in Simulated Annealing
Each picture shows a graph of the utility value (left y-axis and red line) and the temperature curve (right y-axis and blue line) during the annealing process (x-axis = number of iteration) .
The best reached utility values are: T_1 →-3.236, T →-3.236, T_2 →-3.563, T_3 →-4.747 and T_4 →-3.236.

4 Common Concepts

4.1 Concept

In Sec. 2.6 on page 16, we consider only one (more or less randomly chosen) solution for each problem. Testing manually different starting conditions and experimental settings, the solutions seem to be all very similar. But is there a way to render this "similar looking" more precisely? In this section, we suggest a way how to do it and describe many fascinating derived values that we get out easily.

Definition 4.1 Concept

A pair of two random variables (X, Y) with a joint distribution $p(x, y)$ is called a *concept* if Y represents X in some way:

$$I(X;Y) > 0 \tag{4.1}$$

We say as well that Y *is a concept for* X or that Y *is a description of* X. The elements $y \in \mathcal{Y}$ are called *symbols* and represent an *object* $x \in \mathcal{X}$ with probability

$$p(x|y) = \frac{p(x, y)}{p(y)}. \tag{4.2}$$

The number of symbols $|\mathcal{Y}|$ is called the size of the concept.

In the following, there are two different concepts we will discuss. Firstly, the pair of a world R and a sensor S

$$(X, Y) = (R, S) \tag{4.3}$$

can be understood as a concept. Secondly, we discuss concepts created by an initial position capturing agent, see Sec. 2.6 on page 16, with $X = R_0$ and $Y = M_t$. The typical concept generated by agents with $|\mathcal{M}| = 4$ has symbols for north, east, south and west.

Definition 4.2 Common concept

Given a set of concepts $\{(X, Y_1), \ldots, (X, Y_n)\}$ with a joint distribution $p(x, y_1, \ldots, y_n)$.
If Y_* is a description of X that is "similar" to all Y_i, we call the concept (X, Y_*) a *common concept* for $\{(X, Y_1), \ldots, (X, Y_n)\}$. The latter are called *input concepts*.

What we mean by "similar descriptions" is defined in the following in different ways.

The common concept can be interpreted as an inverse multivariate information bottleneck (Tishby et al. (1999)) or an inverse factorization (Möller (2008)): While a concept is split into several concepts in a factorization, we try to fuse them to a common one.

4.2 Objective Common Concept

Definition 4.3 Objective common concept

An *objective common concept* is a common concept $\left(X, Y_*^{obj}\right)$ for the input concepts $\{(X,Y_1),\ldots,(X,Y_n)\}$ where Y_*^{obj} is given by a deterministic mapping $X \to Y_*^{obj}$ that maximizes

$$\max \stackrel{!}{=} \sum_i \left(I\left(Y_*^{obj};Y_i\right) - \alpha \cdot I\left(X;Y_*^{obj}\right)\right) \tag{4.4}$$

$$= \left(\sum_i I\left(Y_*^{obj};Y_i\right)\right) - \alpha \cdot n \cdot I\left(X;Y_*^{obj}\right), \tag{4.5}$$

where α is a model parameter.
The dependence between the random variables $X, Y_1, \ldots, Y_n, Y_*^{obj}$ is defined by the Bayesian network in Fig. 4.1.

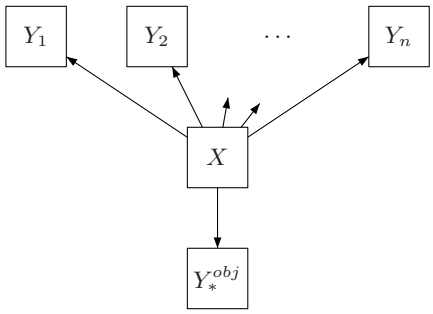

Figure 4.1: The Bayesian network for an objective common concept

This definition does not lead to a unique optimal objective common concept. For example, it is not possible to distinguish permutations in Y_*^{obj} like in (almost) all information theoretic quantities. Additionally, we find for this problem only good solutions instead of an optimal one because of the very big search space stated in Sec. 3 on page 21.

To motivate this definition, we take a look at its components. $\max \stackrel{!}{=} I\left(Y_*^{obj};Y_i\right)$ says that we want to have a common concept Y_*^{obj} as similar to the Y_i as possible without taking care of symbol permutations. All input concepts are similar weighted with 1 because there are no preferential ones.

There are no restrictions on the sizes of the individual concepts. If the common concept would be sufficiently large

$$\left|\mathcal{Y}_*^{obj}\right| \geq \prod_i \left|\mathcal{Y}_i\right|, \tag{4.6}$$

we could include the cross product of all individual concepts

$$Y_*^{obj} = Y_1 \times \ldots \times Y_n \tag{4.7}$$

so that all the input concepts would fit in the common one. Since we do not want the common concept to be this trivial one,

$$\min \stackrel{!}{=} \alpha \cdot n \cdot I\left(X; Y_*^{obj}\right) \tag{4.8}$$

penalizes this behavior of being too accurate in learning X i.e. the cross product of the input concepts. In our experiments, we set $\alpha = 0.2$. In some experiments, this value turned out to be a good choice to get a good but not too high accuracy.

This common concept is called *objective* because Y_*^{obj} is constructed directly with knowledge about X. What it means for a concept not to be objective will become clear by regarding the subjective common concept in Sec. 4.3 on the facing page.

To test this definition, we worked with the initial position capturing from Sec. 2.6 on page 16 and setting $X = R_0$ and $Y_i = (M_t)_i$ where $(M_t)_i$ is the result M_t from the i-th run of the optimization algorithm. Each run is started with a different initialization (seed) for the pseudo random number generator. We do not allow two similar initializations for run k and run j because $(M_t)_k = (M_t)_j$ would occur trivially.

In Fig. 4.3 on page 31, some experimental results for a set of 5 input concepts with 4 symbols are shown. The generated objective common concept has almost the "same kind" of symbols as the input concept, namely: "north", "east", "south" and "west". To see that almost every input concept is well represented in the common concept, we considered the mutual information between the common concept and the i-th input concept and calculated the distance between that mutual information and its upper bound[9]:

$$\min \stackrel{!}{=} I\left(X; Y_i\right) - I\left(Y_*^{obj}; Y_i\right) \geq 0 \tag{4.9}$$

We got for $i = 1, ..., 5$ the values $(0.002, 0.002, 0.701, 0.193, 0.351)$ bit. In this case, the third and fifth concept are not perfectly similar to the common concept. The reason for that is the different naming of the diagonals i.e. the diagonals are assigned to different symbols. For example, the upper right diagonal is assigned to the same symbol as the right triangle for $i = 1, 2, 4$, but not for $i = 3, 5$.

In another experiment in Fig. 4.4 on page 32, we used 4 input concepts with 6 symbols. We tried to find a common concept with 8 symbols. Here, we got essentially new phenomena. Through the common concept, we can find some completely new structures in the world which none of the input concepts is able to "see": We get 4 diagonals as well as 4 triangles.

[9] The bound is derived from the data processing inequality.

In Sec. 4.5 on page 34, we study systematically the influence of the common concept's size and whether there is a preferred size.

In some further experiments in Fig. A.4 on page 64, we also tried to find common concepts for different numbers of symbols in each input concept which is also possible. The results are similar to the two above.

4.3 Subjective Common Concept

If we want to generate a common concept with Def. 4.3 on page 27, we have to know X explicitly. Another problem of the objective common concept is the fact of loosing *superstition*: If an input concept uses different symbols $y \in \mathcal{Y}$ for one $x \in \mathcal{X}$, we call this behaviour superstition. A good measurement for superstition is the conditional entropy $H(Y|X)$ that vanishes for the objective common concept since we choose $X \to Y_*^{obj}$ deterministically.

Therefore, we suggest a method to construct a common concept that is able to adopt superstition from the input concepts. Additionally, it works without explicit knowledge of X.

Another kind of superstition is discussed in Sec. A.1 on page 60.

Definition 4.4 Subjective common concept

A *subjective common concept* is a common concept $\left(X, Y_*^{subj}\right)$ for the input concepts $\{(X,Y_1),\ldots,(X,Y_n)\}$ where Y_*^{subj} is given by a deterministic mapping $Y_1 \times Y_2 \times \ldots \times Y_n \to Y_*^{subj}$ that minimizes

$$\min \stackrel{!}{=} I\left(Y_1; Y_2; \ldots; Y_n | Y_*^{subj}\right). \tag{4.10}$$

The dependence between the random variables $X, Y_1, \ldots, Y_n, Y_*^{subj}$ is defined by the Bayesian network in Fig. 4.2.

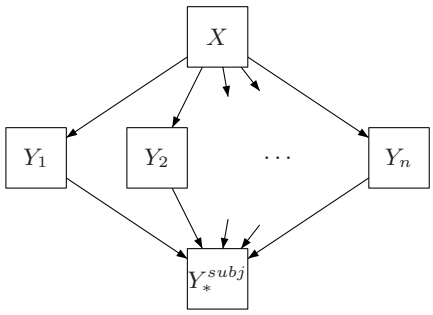

Figure 4.2: The Bayesian network for a subjective common concept

Analogue to the objective common concept, we do not get a unique optimal subjective common concept and we approximate the optimal solution by a good solution in the following.

With this definition, the common concept tries to collect all information shared by the input concepts i.e. information that is encoded in just one of the individual concepts about X is disregarded in Y_*^{subj}.

In Fig. 4.3 on the next page and Fig. 4.4 on page 32, we show a subjective common concept compared with an objective one. To have an objective and information theoretic measurement of the distance between two arbitrary concepts, we use the information distance as defined in Def. 2.3 on page 9. In this two figures, the information distance is with 0.0006 bit resp. 0.38 bit (very) small, and in every other tested setting as well. Because the observed differences were very small, we used in further experiments always the objective common concept. That is necessary especially for larger systems since the calculation time[10] for the subjective common concept grows exponentially with the number of input concepts n and the mean size of these concepts $\frac{1}{n}\sum_i \left|\mathcal{Y}_i^{subj}\right|$.

As annotated before, the objective common concept has by definition no superstition because Y_*^{obj} depends deterministically on X. But also the tested subjective common concepts, like in Fig. 4.4 on page 32, have a very low superstition. Together with the low information distance between the different common concepts, the superstition value of 0.03 bit is another reason not to make the effort of calculating the subjective common concept.

Note that we have to be careful in comparing different common concepts where the sets of input concepts are not equal. In general, this can not be done by comparing the utility values from Eq. 4.4 on page 27 in objective resp. from Eq. 4.10 on the preceding page in subjective common concepts. This is because adding additional concepts to the set of input concepts moves the bound for the utility function and so the utility values can not be compared directly.

In contrast to that, we can compare the utility values of common concepts if the set of input concepts stays equal and only the common concepts' sizes are increased like we did in Sec. 4.5 on page 34. This is unproblematic because the original common concept can be interpreted as a bigger one that does not use the additional symbols. In both interpretations, this common concept has the same utility value. The system just gets more degrees of freedom: It can use the additional symbols to find a better a solution.

4.4 Neighbourhood of Symbols inside of a Concept

Taking a look on the translation matrices of a common concept as in Fig. 4.4 on page 32, we can see that the translation $Y_i \leftrightarrow Y_*$ (for one particular i) is not deterministic (and

[10] Both number of search steps and calculation time per step are growing exponentially.

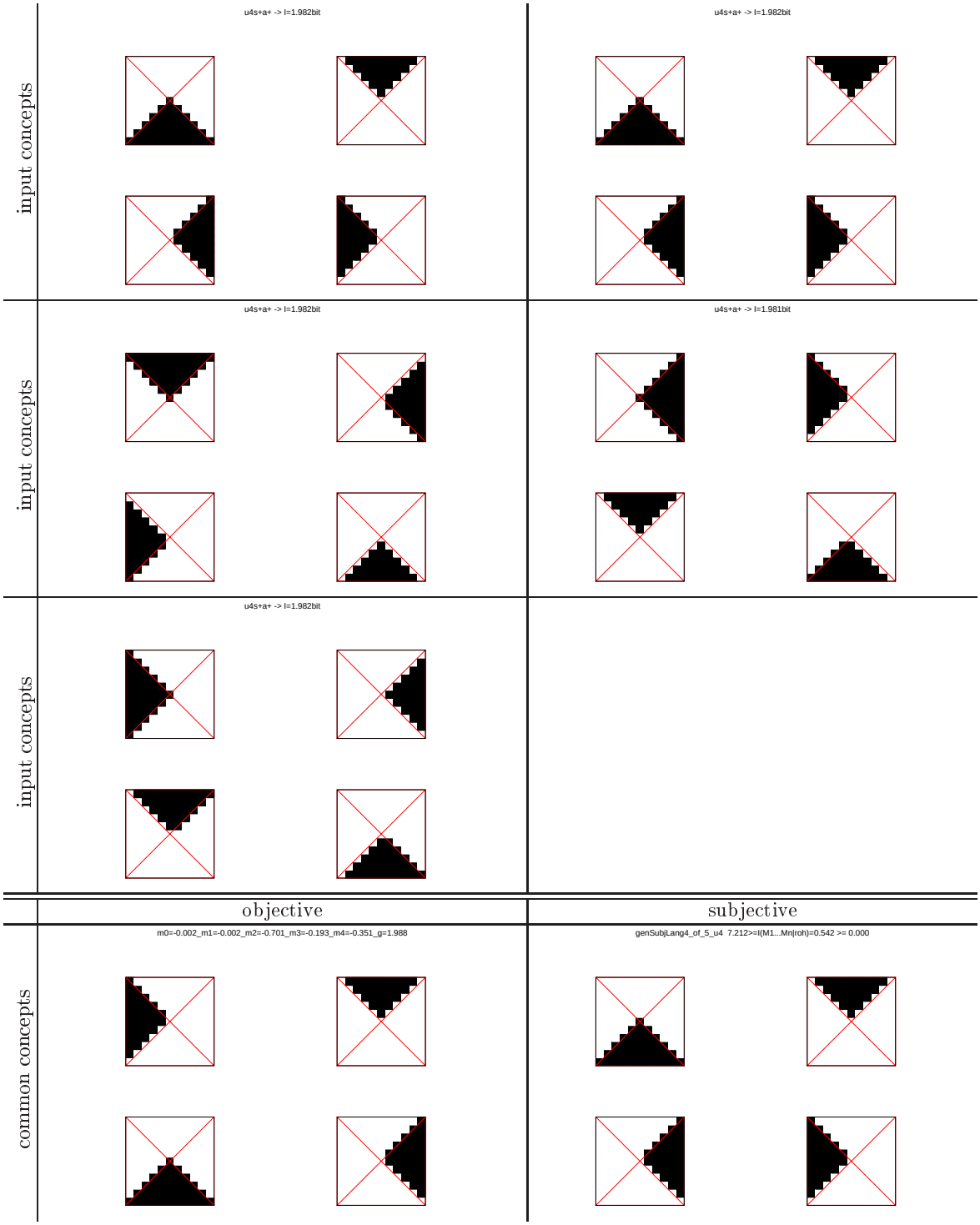

Figure 4.3: Common concepts
Above the parting line, the five input concepts are shown. Out of them we calculated the objective (left) and subjective (right) common concept. The information distance between these concepts is $6.3 \cdot 10^{-4}$ bit, so they are almost the same. The superstition of the subjective common concept is with $H(M_*|R) = 2.8 \cdot 10^{-3}$ bit almost disappearing.

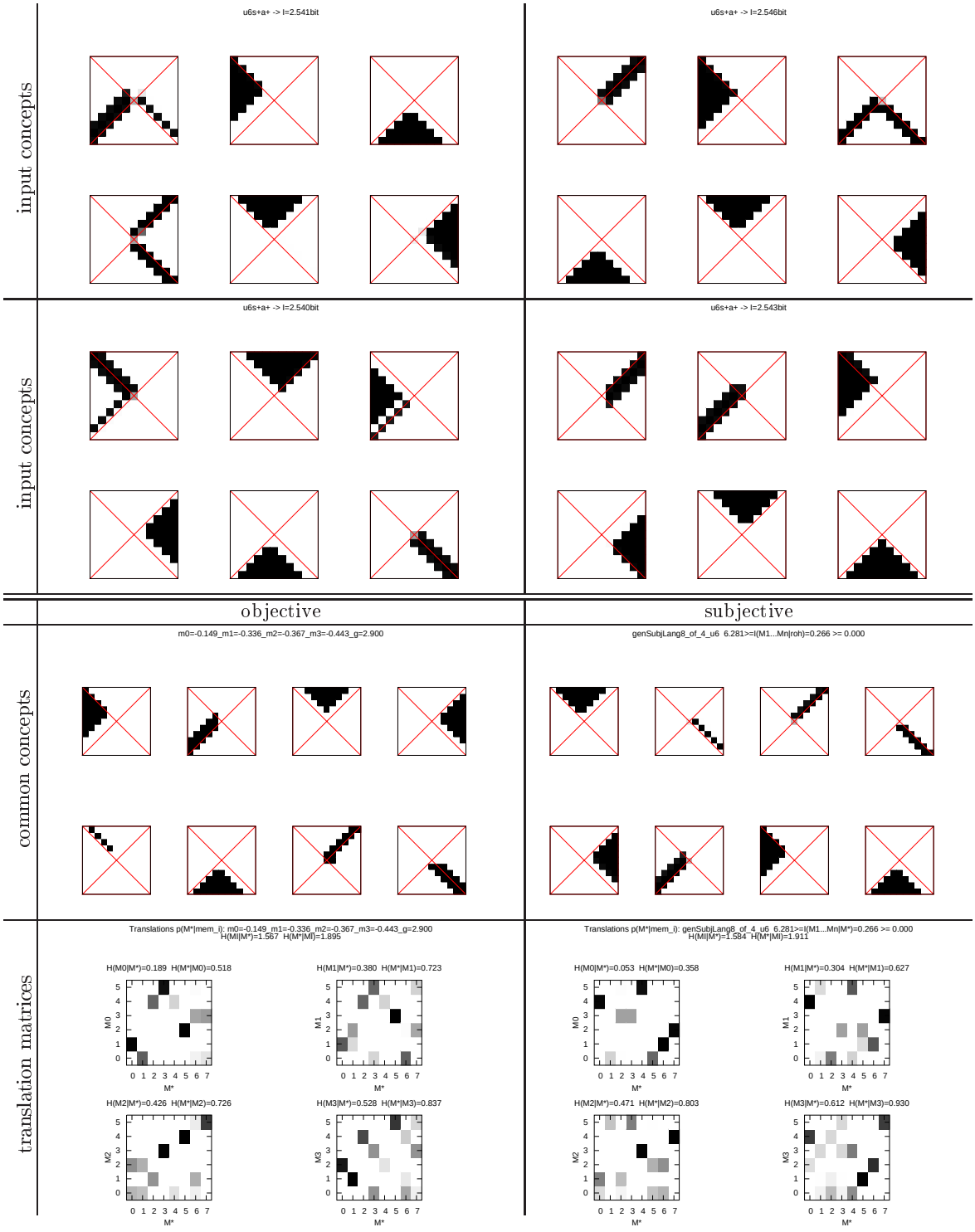

Figure 4.4: Common concepts with sizes bigger than input concepts' sizes
The four input concepts are shown above the parting line. Out of them we calculated the objective (left) and subjective (right) common concept shown in the first row below the input concepts. The information distance between these common concepts is 0.38 bit. So they are similar, but not completely equal. The superstition of the subjective common concept is with $H(M_*|R) = 2.9 \cdot 10^{-2}$ bit almost disappearing.
The *translation matrices* are given in the last row. They plot the conditional probability $p(m_*|m_i)$ in greyscales in a separate sub-picture for every i.

bijective) as expected. This effect increases if $|\mathcal{Y}_*| < |\mathcal{Y}_i|$ so that the probabilistic mapping $Y_* \to Y_i$ has in general more than one y_i to assign to each y_*. We will use this ambiguity to define a fuzzy neighbourhood on the symbols of the concept (X, Y_i).

The approach is principally based on the fact that the generation of the input concepts does not lead to a unique result. Furthermore, if they are derived e.g. from a initial position capturer, they are often not quite optimal. All subsets of objects of X that are associated to a symbol in Y_i but to more then one symbol in Y_j with $j \neq i$ are the basis in this method and the key to a neighbourhood. For example in Fig. 4.3 on page 31, the two "south" diagonals ("southeast" resp. "southwest") are both associated to the symbol "south" in the first input concept , in contrast to the third input concept where they are separated from "south" and attached to the symbols "west" resp. "east". The degree of neighbourhood depends on the amount and the variability of these ambivalent places.

Definition 4.5 Fuzzy neighbourhood of symbols

Given a set of input concepts $\{(X, Y_1), \ldots, (X, Y_n)\}$ and a derived common concept (X, Y_*). The degree of *fuzzy neighbourhood* of a symbol $a \in \mathcal{Y}_i$ to a symbol $b \in \mathcal{Y}_i$ is given by the probability to start with a and get b after a translation to Y_* and back to Y_i :

$$p(b|a) = \sum_{c \in \mathcal{Y}_*} p(b|c) \cdot p(c|a). \tag{4.11}$$

This probability value is an asymmetric kind of neighbourhood measurement.

The considered input concepts created by the initial position capturing have the special property that the mapping $R_0 \to M_t$ $(X \to Y)$ is mostly deterministic. Thus by translating a memory state m_t to a set of initial positions r_0 and back to memory m'_t, we will get $m_t = m'_t$ with a very high probability. So it makes sense not to use a translation to X but to Y_* to figure out a kind of neighbourhood. So more than one input concept is needed to have any neighbours because otherwise, the common concept is too close to the input concept. Increasing the number of input concepts, the degree of neighbourhood should increase as well.

Since we can understand a concept as a "language" of an agent and a common concept as a "group language", there might be missunderstandings due to translation into the group language and back. If we assume that these translation errors are not random but a change to a neighboured symbol, then this definition is useful. It depends on the kind of input concepts if this is the case. Our tested input concepts from the initial position capturing seem to have this property.

In Fig. 4.5 on the following page, we visualized Eq. 4.11. In some cases, e.g. in the top/left input concept shown as neighbourhood matrice, the translation is quite unique and there are not many possibilities to derive a neighbourhood relation. In other cases, like the second-row/right concept, the input and the common concept do not fit together that perfectly. Here, the diagonals are not according to the same symbols in the common

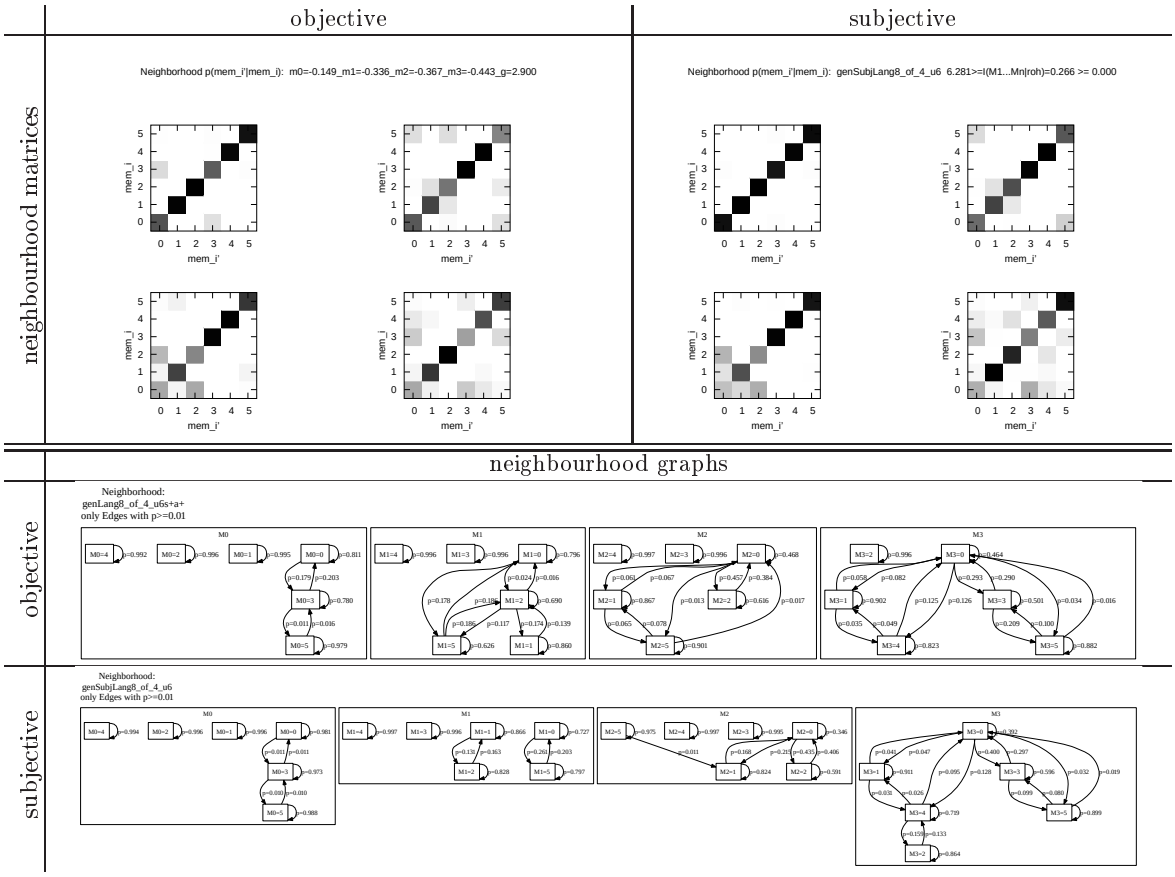

Figure 4.5: Neighbourhood inside of a concept
These figures extend the results of Fig. 4.4 on page 32. In the first row, the conditional probabilities $p(m'|m)$ (compare Def. 4.5 on the preceding page) are shown in greyscale in a separate sub-picture for every input concept, on the left for the objective and on the right for the subjective common concept. The set of 4 graphs in the second (objective) and third (subjective) row are visualizing the neighborhood structure on every input concept. An arc from node a to node b is drawn if $p(b|a) \geq 0.01$.

concept and the individual concept. One of the most common neighbourhoods to a triangle is an adjacent diagonal.

An example with a much more meshed neighbourhood is shown in Fig. 4.6 on the facing page, created by shrinking the size of the common concept from 8 to 4.

Unfortunately, not all "intuitive" neighbours are found by this method. However, note that this method uses only information and probability techniques and is not aware of geometry for neighbourhood.

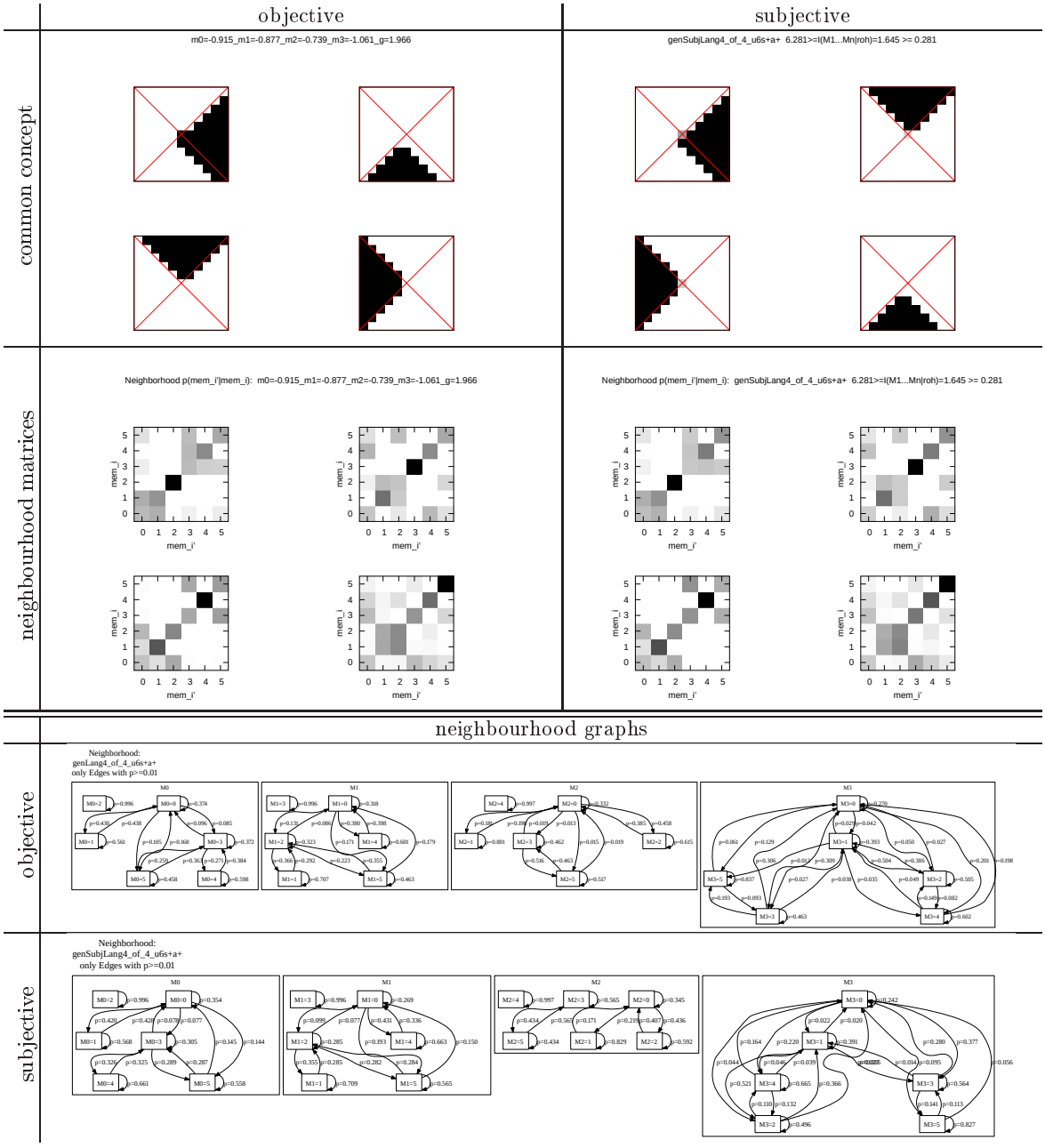

Figure 4.6: Neighbourhood inside of a concept - more related

These common concepts are based on the same input concepts as shown in Fig. 4.4 on page 32.
In the first row, the common objective (left) and subjective (right) concept are shown. In the second row, their translation matrices are shown presenting the conditional probabilities $p(m'|m)$ (compare Def. 4.5 on page 33) in greyscale in a separate sub-picture for every input concept.
The set of 4 graphs in the third (objective) and fourth (subjective) row are visualizing the neighborhood structure on every input concept. An arc from node a to node b is drawn if $p(b|a) \geq 0.01$.

4.5 Optimal Size for Common Concept

The last question we discuss is the optimal size of the common concept. Fig. 4.7 on page 37 plots the utility function f from Eq. 4.4 on page 27 as function of the common concept's size $f(|\mathcal{M}_*|)$ as well as the second derivative with respect to the logarithmic scale of $|\mathcal{M}_*|$:

$$\frac{\frac{f(|\mathcal{M}_*|+1)-f(|\mathcal{M}_*|)}{\log(|\mathcal{M}_*|+1)-\log(|\mathcal{M}_*|)} - \frac{f(|\mathcal{M}_*|)-f(|\mathcal{M}_*|-1)}{\log(|\mathcal{M}_*|)-\log(|\mathcal{M}_*|-1)}}{\log(|\mathcal{M}_*|+1)-\log(|\mathcal{M}_*|)} \tag{4.12}$$

The points where the latter is very negative are interesting and candidates for a good common concept's size: The additional utility value for enlarging the memory size $|\mathcal{M}_*|$ is shrinking.

By setting the size of the common concept to 8 (the size of the input concepts), most of their information can be covered. The utility function ceases to increase for $|\mathcal{M}_*| \gtrapprox 17$. This provides a way to estimate a useful size of the common concept with an appropriate trade-off between size and covering most of the input concepts' information.

For this experiment, we started the calculation with $|\mathcal{M}_*| = 2$ and increased it to 64. In every step (except the first one), we used the best solution from the smaller common concept as starting solution for Simulated Annealing. This assures the (theoretically given) monotony of our experimental curve

$$f(|\mathcal{M}_*|+1) \geq f(|\mathcal{M}_*|)\,. \tag{4.13}$$

We also tried a random starting solution for every step. In this case, the result becomes a bit noisier and monotony is not assured anymore. But the result stays principally the same.

4.6 Discussion

In this section, we gave a definition of the concept (resp. language) of e.g. an agent. It is a set of symbols representing different circumstances in a world which have different probabilities. The benefit of formulating an abstract definition is that it also applies on sensors. So we can treat sensors, which are also a kind of representation of the world, similarly to an internal representation of agents. We state that those concepts can vary from agent to agent even if the latter are similar in terms of utility value.

We have shown how to extract common concepts (resp. language) out of the individual concepts of an agents' group. Two different techniques were presented, the objective and the subjective common concept. There is evidence that both objective and subjective methods are similar if, as in our case, the input concepts are mostly deterministic. So we can save computation resources by calculating only the objective one concerning the

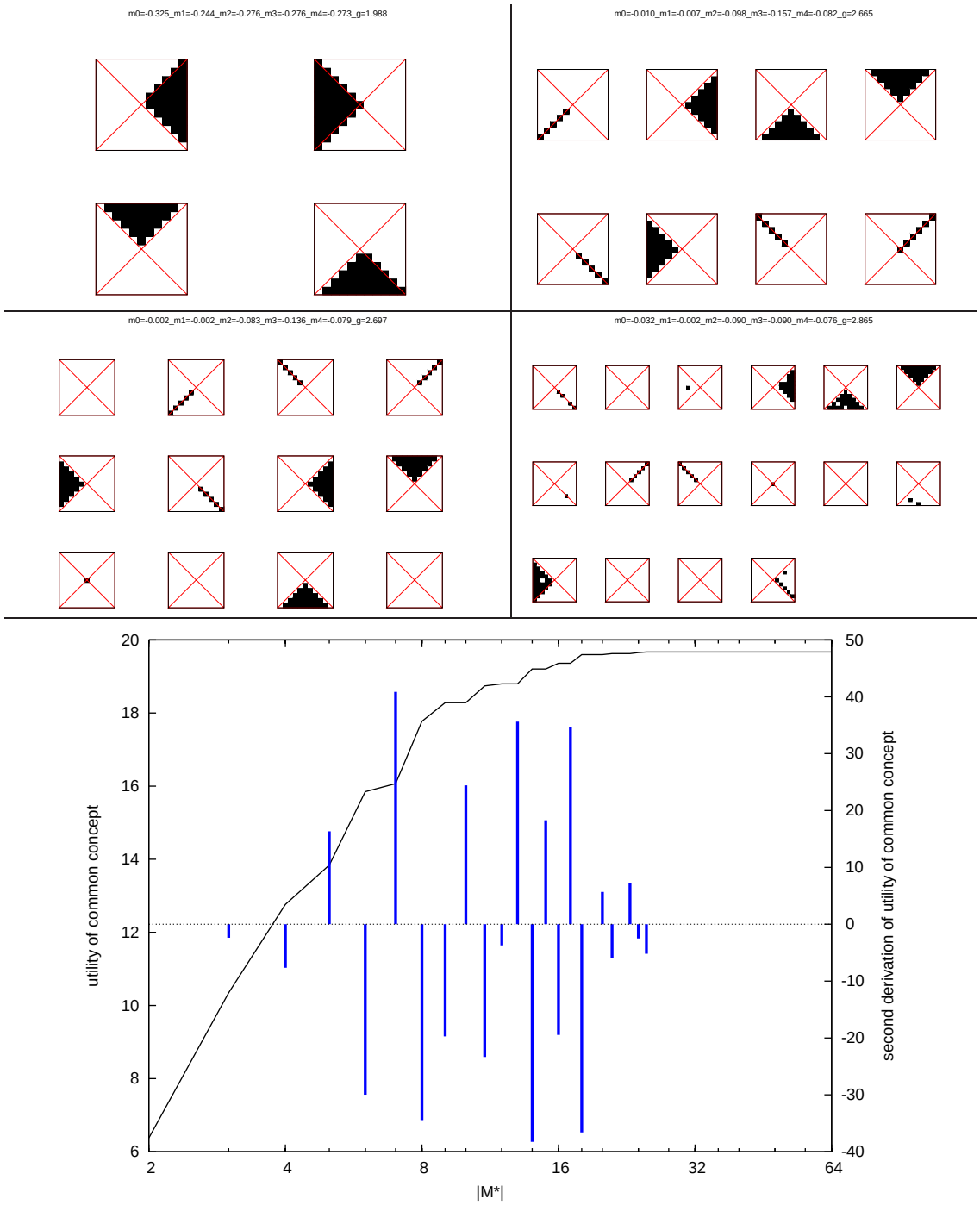

Figure 4.7: Quality of different common concept sizes
We use in this experiment 10 input concepts of size 8 shown in Fig. A.2 on page 62. The first two rows show 4 objective common concepts with different sizes. The entirety of all common concepts generated in this experiment is shown in Fig. A.3 on page 63.
The main diagram shows the utility functions' dependency on the common concept's size. The blue pulses are the derivative (Eq. 4.12 on the preceding page) of the utility function.

exponential growing effort for the subjective one. Both methods are based on the fact that for some locations in the world the agents have a disagreement about how to group them into symbols (in our example e.g. the 4 diagonals). The common concept methods are able to identify these regions and assign new symbols to them. If we would enlarge the memory size for every agent in the group, they may also find individually the diagonals but with much lower accuracy than the common perspective. Especially the symbol for the "center of the world" (Fig. 4.7 on the preceding page - 9th symbol in the common concept with 12 symbols) was never found by an individual agent in our experiments. Thus, a new level of structuring the world emerged by considering a whole group of agents. Instead of building an additional hierarchical layer between agents, we got similar results resp. concepts in Möller (2008) by putting a hierarchy inside of a single agent (hierarchical initial position capturing).

For the objective common concept, we introduced a parameter α to counter the behavior of just building the cross-product concept of all input concepts. The existence of this parameter or at least the special choice of its value are not very satisfyingly. Especially for small sizes of the common concept, it is not really necessary.

The optimal size of a common concept was also attended. Here, we presented a method to figure out the trade-off between size and quality in matching the input concepts very accurate. So there is no absolute answer for a good size but a tool to identify interesting sizes and their quality.

One additional application of the common concepts and the "translation-mistakes" was the definition of a neighbourhood on a concept's symbols. It turned out that the number of neighbourhood relations is heavily depending on the relation between the individual concepts' sizes and the common concept's size. If the common concept is smaller than the considered individual one, we have many neighbours, otherwise almost none. But even in experiments where we were able to create many neighbour relations, they are only partly related to the geometrical neighbourhood in the grid world.

5 Symmetries

During the analysis of the agents' concepts (see Def. 4.1 on page 26), we recognized that they have certain symmetries which fit to world's symmetries. This motivated us to define a utility function for "symmetry operations" and to test them on both the world (sensor) and the generated concepts. We recognized as well that the common language are in the majority of cases more "symmetric" than each single input language.

To study these symmetries systematically, we consider two different approaches. In one of them, symmetry operates from "outside" of the system (*extrinsic symmetries*) and in the other one, symmetry is applied "inside" (*intrinsic symmetries*). In latter, we explore the implicit knowledge of the agent about symmetries.

Both methods have the following idea in common: They transform a concept and compare it to another concept by measuring the mutual information between them. The analyse of symmetries is done by these steps:

1. Define a set of operations on a concept $\rightarrow$ *symmetry operation*.
2. Define a utility function for each of those symmetry operations, e.g.:
    a) Define another concept where all "operated concepts" are compared to.
    b) Define how this comparison will be measured, for example by mutual information.

5.1 Extrinsic Symmetries

Consider a concept with $(X, Y) = (R, Y)$ settled on the two dimensional grid world $\mathcal{R} = \mathbb{Z}^2$. Extrinsic symmetries is a method to analyze this concept after applying mathematical symmetry operations on $\mathbb{Z}^2$, i.e. rotation (in 90° steps), mirroring and translation. Since we do not analyze the agent's internal structure but only the resulting concept, we "have a look from outside".

5.1.1 Definitions

Definition 5.1 Extrinsic symmetry operation

An *extrinsic symmetry operation* ξ in a two dimensional grid world is a combination of rotation, mirroring and translation:

$$\xi^{\theta,\varphi,x_0,y_0} : \mathbb{Z}^2 \to \mathbb{Z}^2 \tag{5.1}$$
$$\xi^{\theta,\varphi,x_0,y_0} := \xi^{x_0,y_0}_{\text{trans}} \circ \xi^{\varphi}_{\text{rot}} \circ \xi^{\theta}_{\text{mir}} \tag{5.2}$$

where the mirroring (around the y-axis) is described by $\theta \in \{+1, -1\}$

$$\xi^{+1}_{\text{mir}}(x, y) := (x, y) \tag{5.3}$$
$$\mathrm{s}^{-1}_{\text{mir}}(x, y) := (-x, y), \tag{5.4}$$

the rotation (in 90° steps counterclockwise) by $\varphi \in \{0°, 90°, 180°, 270°\}$

$$\xi^{0°}_{\text{rot}}(x, y) := (x, y) \tag{5.5}$$
$$\xi^{90°}_{\text{rot}}(x, y) := (-y, x) \tag{5.6}$$
$$\xi^{180°}_{\text{rot}}(x, y) := (-x, -y) \tag{5.7}$$
$$\xi^{270°}_{\text{rot}}(x, y) := (y, -x) \tag{5.8}$$

and the translation by $(x_0, y_0) \in \mathbb{Z}^2$

$$\xi^{x_0,y_0}_{\text{trans}}(x, y) := (x + x_0, y + y_0). \tag{5.9}$$

In our experiments, we apply only translations of maximal $\pm r_0$ in x, y-direction so the number of extrinsic operations to compare is

$$\#\text{Sym.Operations}_{ext} = (2r_0 + 1)^2 \cdot 8. \tag{5.10}$$

Definition 5.2 Extrinsic symmetry utility

The quality of an extrinsic symmetry operation is measured by a continuous utility function defined in the following. Consider a concept (R, Y) with $\mathcal{R} = \mathbb{Z}^2$. By applying an extrinsic symmetry operation $\xi^{\varphi,\theta,x_0,y_0}$ (Def. 5.1 on the facing page), we get two new probabilistic mappings $R \to \xi R$ and $\xi R \to Y^\xi$ (Fig. 5.1) via

$$\Pr\left(\xi R = r' | R = r\right) := \delta_{r',\xi(r)} \tag{5.11}$$

$$= \begin{cases} 1 & r' = \xi\left(r\right) \\ 0 & \text{else} \end{cases}. \tag{5.12}$$

We choose the probabilistic mapping $\xi R \to Y^\xi$ as a "copy" of $R \to Y$:

$$\Pr\left(Y^\xi = y | \xi R = r\right) := \Pr\left(Y = y | R = r\right) \tag{5.13}$$

The *utility function for extrinsic symmetry operation* $\xi^{\varphi,\theta,x_0,y_0}$ for the concept (R, Y) is defined as

$$I\left(Y; Y^\xi\right) \tag{5.14}$$

where a higher utility value is a better one.

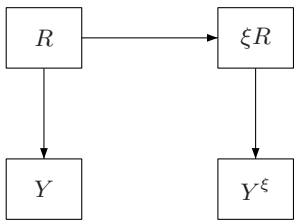

Figure 5.1: Bayesian network for calculating extrinsic symmetry utility

There are two methods to visualize results of symmetry experiments:

A Spectrum of symmetry operations shows the number of symmetries (y-axis) for a given value of $x = \frac{I\left(R;Y^{transformed}\right)}{\max I\left(R;Y^{transformed}\right)}$. Additional to these peaks, a smoothing curve

$$\sum_i y_i e^{-40000(x-x_i)^2} \tag{5.15}$$

is added to the spectrum, whereby y_i is the number of symmetry operations with relative utility value x_i. Also a cumulation of the peaks, i.e. the number of symmetry operations with $x \leq \frac{I\left(R;Y^{transformed}\right)}{\max I\left(R;Y^{transformed}\right)}$, is shown.

A Symmetry Matrix is a visualization of a matrix containing utility values of the symmetry operations. Each value is represented by a shade of grey. The values' arrangement in the matrix depends on the method (extrinsic resp. intrinsic).

We made as well some experiments with information distance $D\left(Y, Y^{\xi}\right)$ as utility function which should be minimal for the identity transformation

$$\xi = \mathrm{id} := (\theta = +1, \varphi = 0°, x_0 = 0, y_0 = 0)\,. \tag{5.16}$$

It is bounded by

$$0 \leq D\left(Y, Y^{\xi}\right) \leq 2 \cdot H\left(Y, Y^{\xi}\right) \tag{5.17}$$

but the real upper bound depends very much on the given concept (R, Y). Therefore, it is difficult to compare different spectra. In case of using mutual information $I\left(Y; Y^{\xi}\right)$, ξ =id gives us a very good (empiric) approximation for a close generic bound:

$$0 \leq I\left(Y; Y^{\xi}\right) \lessapprox I\left(Y; Y^{\mathrm{id}}\right) \tag{5.18}$$

So using $I\left(Y; Y^{\xi}\right)$ instead of $D\left(Y, Y^{\xi}\right)$ makes comparison much easier.

5.1.2 Experiments

In the experiments, we analyze the symmetries of a sensor s+ / sq S by interpreting $R \rightarrow S$ as a concept (R, S). In comparison to this, we analyze the symmetries of the concept (R_0, M_{15}) generated by an initial position capturing agent who uses the particular sensor. The question is how much symmetry of the world "provided" by the sensor will the agent conserve? In Fig. 5.3 on page 44, we show the results for an agent with $|\mathcal{M}| = 4$.

It turned out that the best operations in terms of symmetry were those we had expected. In most experiments, the descending order of their quality corresponds to the order of the following list:

Identity with operation$(\theta = +1, \varphi = 0°, x_0 = 0, y_0 = 0)$

Rotations with operation $(\theta = +1, \varphi = 0°/90°/180°/270°, x_0 = 0, y_0 = 0)$

Other operations become in general more complicated. Therefore, we visualized the translation dependency on the symmetry utility as 8 symmetry matrices for each possible rotation/mirroring value. Depending on the compared concept, we see that for example diagonal translations are better than orthogonal ones (sensor s+). For sensor sq, we see indeed some very weak long distance translation symmetries (distance of the 4 gradient sources). In the spectrum, they are to weak to separate them from noise. This weakness may be caused by the non-repetition of the pattern (only 2 repetitions of the gradient pattern in each direction instead of an infinitive number).

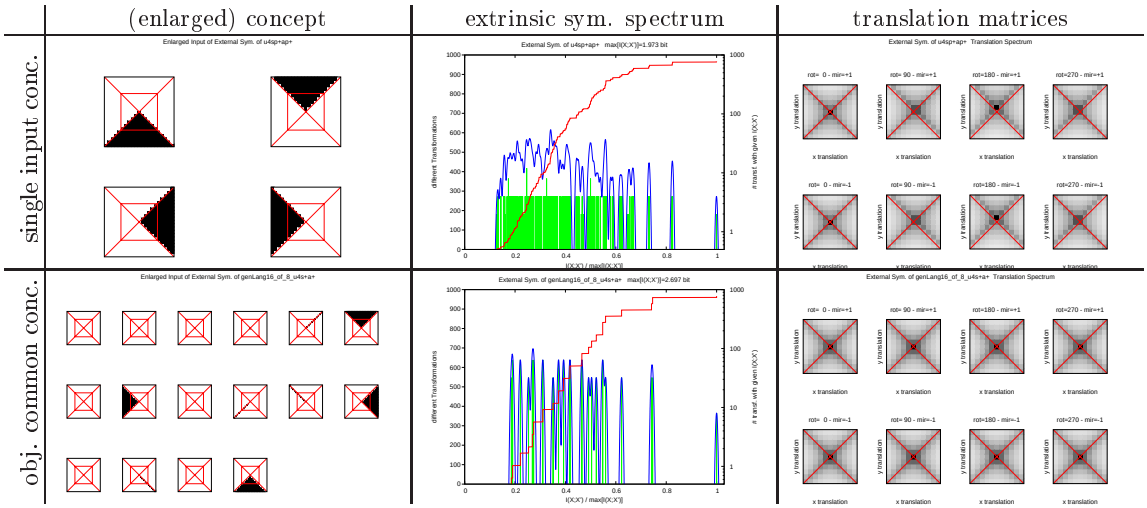

Figure 5.2: Extrinsic symmetries of objective common concept
An initial position capturing agent with sensor s+, actuator a+ and $|\mathcal{M}| = 4$ was used. The first picture shows the concept generated by this agent. The picture below shows an objective common concept for this case. The second column shows the according extrinsic symmetry spectrum. The third column shows the according symmetry matrices for the different rotation/mirroring values.
Both concepts are investigated further in Fig. 5.7 on page 52. The input concepts for the common concept are shown in Fig. A.5 on page 65.

Different as expected, the identity symmetry is not necessarily the best one. In Fig. 5.3 on the next page ((R_0, M_{15}) for sensor s+), the best symmetries are $(\theta = \pm 1, b = 180°, x_0 = 0, y_0 = 1)$ which are about 0.1% better than identity. These and all other transformations are shown in Fig. 5.4 on page 45. By analyzing the data in detail, it turned out that the diagonals are not as deterministic as they seem: Every symbol has a small but non-zero probability for the diagonals even if it looks like zero on the plot. By rotating and translating the concept, it looks like before but the diagonals are not identical. So the mapping $M_{15} \to M_{15}^{\xi}$ is more deterministic than the not transformed one $M_{15} \to M_{15}^{\text{id}}$ what means that identity is not the best operation.

There are some similarities between the symmetry spectrum of the sensor and the resulting concept but they are not that expressive as we can see by comparing the two scenarios with the different sensors.

In Fig. 5.2, we compared the symmetry spectrum of an individual concept and a common concept. While the individual concept seems to be very symmetric, it has only 4 best symmetries. The common concept in contrast has 8 best symmetries, all rotations and mirrorings.

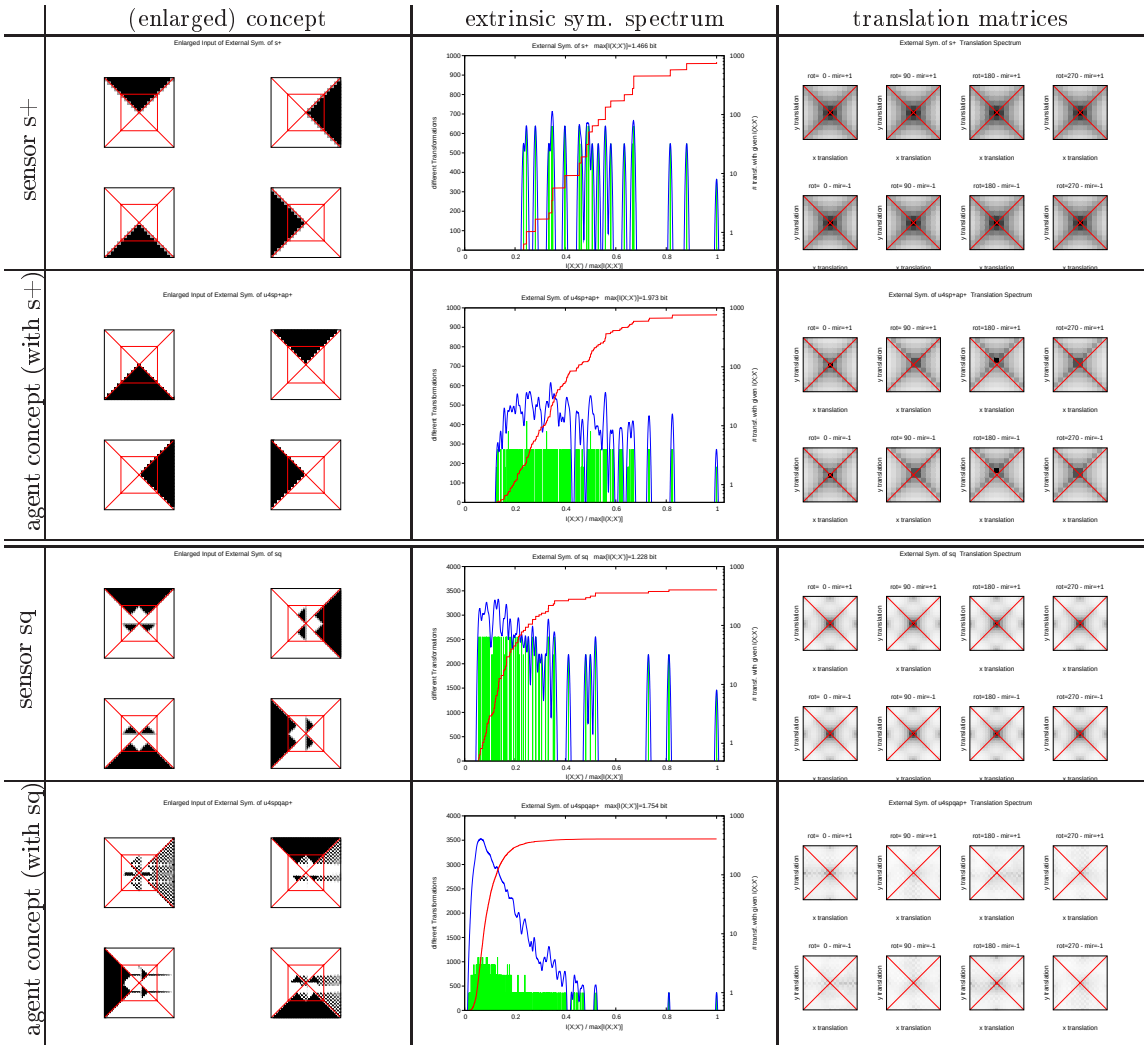

Figure 5.3: Extrinsic symmetries

An initial position capturing agent with actuator a+ and sensor s+ (upper half) / sq (lower half) was used.

In the first column, the according sensor interpreted as concept is shown in the first row, the agent's concept is shown in the second row. In each of those pictures, a set of positions $\mathcal{R}_0$ is surrounded by a red square box. This set has several usages: The agent was optimized for equal distributed starting positions $\mathcal{R}_0$. Furthermore, only those operations are calculated which leave $\mathcal{R}_0$ inside of the "big box" (the entirety of shown positions). That restricts only the translations. Additionally, $\mathcal{R}_0$ gives us the area in which the transformed concept is compared to the original one.

In the second column, the spectrum (drawn in green, right y-axis), the cumulated spectrum (drawn in red, left y-axis) and the smoothed spectrum (drawn in blue, right y-axis) of the extrinsic symmetry are shown. Latter is smoothed with Gaussian $\sum_i y_i \exp\left(-40000 \cdot (x - x_i)^2\right)$ where (x_i, y_i) accords to the coordinates of the spectrum's i-th peak.

In the third column, the symmetry matrices are plotted for the 8 mirroring/rotation operations.

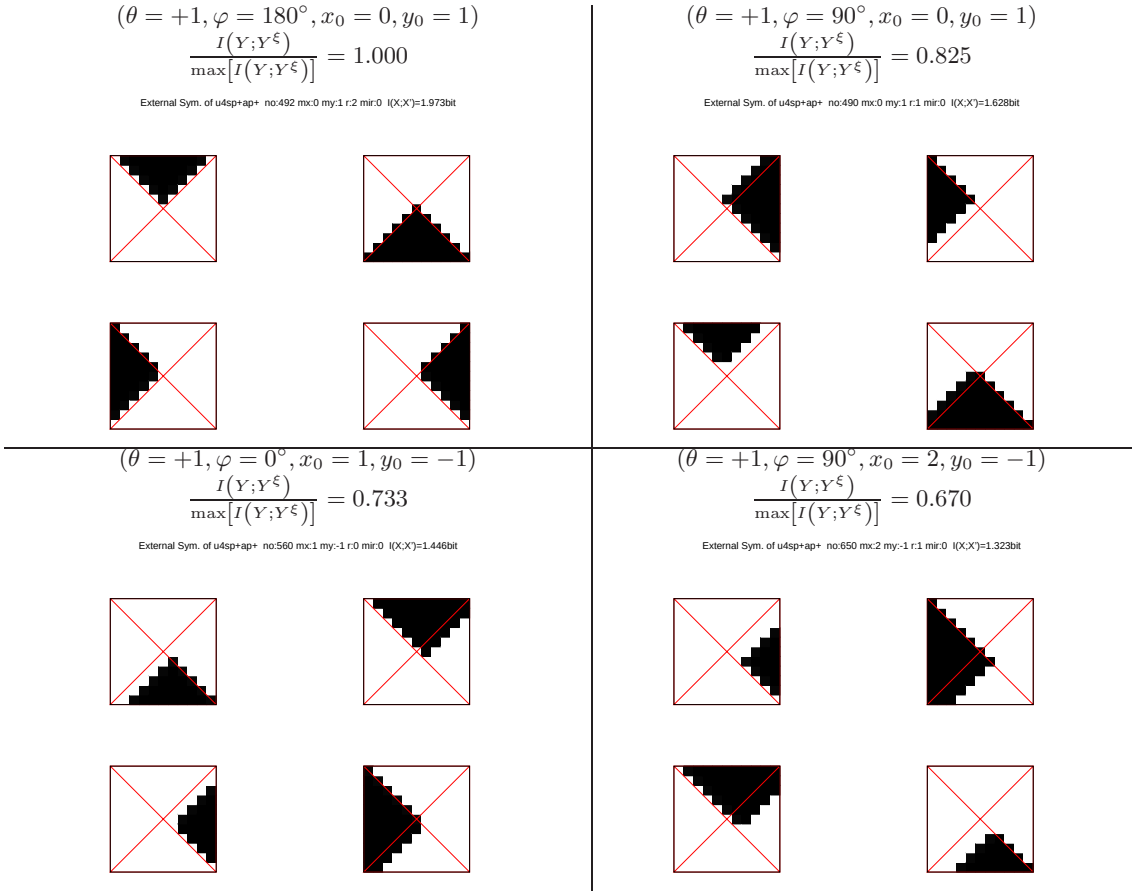

Figure 5.4: Application of extrinsic symmetries on concept
Here are some transformed concepts (R_0, M_{15}) for an initial position capturing agent with actuator a+ and sensor s+ shown. The untransformed agent is shown in Fig. 5.3 on the facing page. From each of the 4 last peaks of the symmetry spectrum, one representative operation is shown.

5.2 Intrinsic Symmetries

Concerning intrinsic symmetries, we do not have just transformations on R. Instead we apply a transformation on the embodiment of the agent. For example a "rotation" of the agent by 90° is equivalent to a particular permutation of the sensor and actuator states of the agent. So except for the translation, all operations are equivalent to a special permutation of the agents' sensor and action values. But additionally, there are other permutations of sensor and actuator values which can not be interpreted as extrinsic symmetry operation. The intrinsic symmetry transformations are formalized in Def. 5.3.

The question is how to rate the quality of an intrinsic symmetry operation. Here we do not just compare the individual concept to its transformed one but compare the latter to a common concept generated by many agents. The benefit is that also variations of the concept are allowed as long as they are similar to a typical good solution for this problem. The utility function is formalized in Def. 5.4 on the facing page.

This rating method could also be used in extrinsic symmetry if the concept is not induced by a sensor[11]. But with this restriction, we are not able to compare the results directly since we can not apply the same method on agent and sensor.

5.2.1 Definition

Definition 5.3 Intrinsic symmetry operation

An *intrinsic symmetry operation* on an agent's embodiment is described by replacing the sensor and actuator node in the perception-action loop (like shown in Fig. 5.5 on the next page): Node S is replaced by $S^\pi \to S^{orig}$ and node A by $A^{orig} \to A^\pi$. Each intrinsic symmetry operation $\pi = (\pi_S, \pi_A)$ is equivalent to two deterministic, bijective (state permutation) mappings

$$\pi_S : S^\pi \to S^{orig} \tag{5.19}$$

$$\pi_A : A^{orig} \to A^\pi. \tag{5.20}$$

In total, there are

$$\#\text{Sym.Operations}_{int} = |\mathcal{S}|! \cdot |\mathcal{A}|! \tag{5.21}$$

of these operations.

[11] Differences between an original and a permuted sensor are not directly detectable by information theoretic quantities.

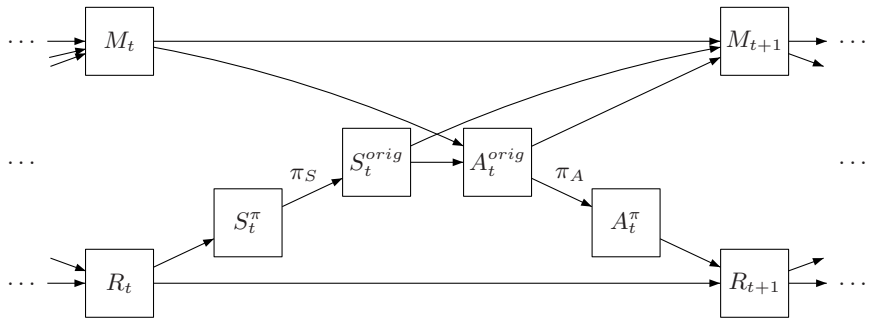

Figure 5.5: Perception-action loop with modifications to calculate intrinsic symmetry utility function

Definition 5.4 Intrinsic symmetry utility

To analyze an intrinsic symmetry operation $\pi = (\pi_S, \pi_A)$, we generate at first a whole set of concepts $\{(X, Y_1), \ldots, (X, Y_n)\}$ according to good solutions of a given experiment with the (unmodified) perception action loop (equivalent to applying the symmetry operation $(\mathrm{id}, \mathrm{id})$). This set is used as set of input concepts and a common concept (X, Y_*) is computed.
To evaluate π, we apply it on the perception-action loop[a]. The concept according to this modified perception-action loop is called (X, Y^{π}).
The utility function of π is given by

$$I\left(Y_*; Y^{\pi}\right) \tag{5.22}$$

where a higher utility value is a better one (compare Def. 5.2 on page 41).
Parameters of this definition are the number of input concepts n for the common concept (X, Y_*) and the size of the common concept $|\mathcal{Y}_*|$.

[a] We do not change the controller, i.e. we do not optimize the quality of the according agent.

In difference to Def. 5.2 on page 41, we are not able to compute sensor symmetries with this method. But it is more generic in terms of the concept types because we also allow concepts with $X \neq R$.

An application of an intrinsic symmetry operation on an agent's embodiment is equivalent to a change of the agent's controller. To optimize an agent, the controller is manipulated as well. So there is a risk of misunderstanding: The analyzing of the intrinsic symmetries of an agent is not a re-optimization with the goal to approximate the common concept because we only see a vanishing small part of the search space:

$$\frac{\#\text{Sym.Operations}_{int}}{\#\text{Constroller}} = \frac{|\mathcal{S}|! \cdot |\mathcal{A}|!}{(|\mathcal{M}| \cdot |\mathcal{A}|)^{|\mathcal{M}| \cdot |\mathcal{S}|}} \tag{5.23}$$

In case of $|\mathcal{S}| = |\mathcal{A}| = |\mathcal{M}| = 4$, the quotient is

$$\frac{4! \cdot 4!}{(4 \cdot 4)^{4 \cdot 4}} \approx 3.1 \cdot 10^{-17}$$

and decreases exponentially with growing $|\mathcal{M}|$. That gives us evidence that the found symmetries are not just an accident, especially because the number of found symmetries seems to be independent of the given memory size after a certain threshold.

5.2.2 Experiments

Intrinsic Symmetry Experiments were repeated in Fig. 5.7 on page 52 for the same setup as the extrinsic experiments from Fig. 5.3 on page 44. For sensor s+ and sq, there are more best symmetries in case of intrinsic symmetries than in the extrinsic case. In case of "second best" symmetries, the comparing of the methods is more difficult: It depends on the considered sensor (s+ / sq) which method detects more symmetries. For further analysis, we have to mention that these spectra are not totally comparable over the whole x-range because we apply different operations (translations are not included in the intrinsic model, torsion[12] is not included in the extrinsic model). This explains furthermore why there are different total numbers of symmetries. Nevertheless, we compare at least the number of "best" symmetries to analyze this two symmetry calculation methods.

Additionally to the spectrum, we plotted the symmetry matrix (possible π_S are enumerated on the x-axis, possible π_A on the y-axis) to figure out what combinations of sensor and actuator permutations (π_S, π_A) are the most symmetric ones. On the diagonal, we have the so called *synchronous permutations*[13] where the actuator and sensor states are permuted in the same way: $\pi_S = \pi_A$. There are $|\mathcal{S}|! = |\mathcal{A}|! = 4! = 24$ of them, 8 are intuitively clear: There are the 8 rotations and mirroring combinations that occur in the extrinsic symmetry operations as well. Additionally, there are some cases of embodiment's torsion where actuator and sensor are twisted in the same way. Outside of the diagonal, this coupling of sensor and actuator permutations is not given: $\pi_S \neq \pi_A$. In Fig. 5.7 on page 52, we see that for sensor sq all 4 best symmetries are outside of this easily interpretable region. The transformed concepts (R_0, M_{15}^{π}) are shown in Fig. 5.8 on page 53. In comparison to this, some bad intrinsic symmetries are shown in Fig. 5.10 on page 55.

[12] Torsion of a sensor is e.g. "switching north and west, without touching south and east".

[13] It makes only sense to speak from synchronous permutations with $\pi_S = \pi_A$ if $|\mathcal{S}| = |\mathcal{A}|$ and the different sensor and actuator values are ordered in the same way in their variables. This needs a method to identify sensor and actuator values with each other. In our examples, this is easily done. The sensor is always pointing to one of the 4 adjacent cells. It is identified with the action which moves the agent there.

The Structure of the Best Permutations in Fig. 5.8 on page 53 is not very obvious. If we measure the utility function $I\left(R_0; M_{15}^{\pi}\right)$ for which the original concept $(R_0; M_{15})$ was optimized, the 4 symmetric concepts are about 15% worse than the untransformed one. But intuitively, they should have a higher symmetry than the original concept $(R_0; M_{15})$ from which they are "derived". This results from the fact that the "variance" of a good solution (e.g. for the initial position capturing task) causes an asymmetry in the corresponding concept. By generating many of those good solutions, the derived common concept smoothes this variance out so that it has a much higher symmetry. From another perspective, these intrinsic highly symmetric concepts are those which are similar to the common concept and "inherit" its symmetries.

Comparing the 4 best symmetries, we see that their concepts can be transformed into each other by applying a mirroring on the x-axis or/and y-axis. In opposite to the concepts, the embodiments could not be transformed that simple into each other. These transformations of embodiment are different for sensor and actuator, but each of them can be interpreted as mixture of extrinsic rotation and mirroring: $\pi_{S/A} \approx \xi_{\text{rot}}^{\varphi} \circ \xi_{\text{mir}}^{\theta}$. For further analysis of the structure of these 4 best permutations, we take a look at the "relative permutations" between them. Therefore, we define the set of all possible permutations of a set $\mathcal{X}$:

$$\Pi_{\mathcal{X}} = \{f : \mathcal{X} \to \mathcal{X} | f \text{ is bijective}\}. \tag{5.24}$$

Every intrinsic symmetry operation is now element of the following set:

$$(\pi_S, \pi_A) \in \Pi_{\mathcal{S}} \times \Pi_{\mathcal{A}} \tag{5.25}$$

Let us denote the set of all found best intrinsic symmetry operations as $K \subseteq \Pi_{\mathcal{S}} \times \Pi_{\mathcal{A}}$. We define the set of the operations from K to K (automorphisms) as

$$\text{Aut}\,(K) := \{g \in \Pi_{\mathcal{S}} \times \Pi_{\mathcal{A}} | gK = K\} \subseteq \Pi_{\mathcal{S}} \times \Pi_{\mathcal{A}}. \tag{5.26}$$

For the experiment with K as shown in Fig. 5.8 on page 53, this set contains mirroring around the x and/or y-axis[14]:

$$\begin{aligned}
\text{Aut}\,(K) = \{&\text{id}, && (5.27)\\
&(\rightarrow_S \leftrightarrow \leftarrow_S, \uparrow_A \leftrightarrow \downarrow_A), && (5.28)\\
&(\uparrow_S \leftrightarrow \downarrow_S, \rightarrow_A \leftrightarrow \leftarrow_A), && (5.29)\\
&(\rightarrow_S \leftrightarrow \leftarrow_S, \uparrow_S \leftrightarrow \downarrow_S, \rightarrow_A \leftrightarrow \leftarrow_A, \uparrow_A \leftrightarrow \downarrow_A)\} && (5.30)
\end{aligned}$$

[14] For clarity, we denote only elements which are not mapped to themselves. $a_X \leftrightarrow b_X$ means that this permutation permutes only $a, b \in \mathcal{X}$ and does not influence the positions of the other states in X.

This set of permutations is complete i.e. $\mathrm{Aut}(K) = \langle \mathrm{Aut}(K) \rangle$ and builds a subgroup of $\Pi_{\mathcal{S}} \times \Pi_{\mathcal{A}}$ (automorphism group). To mention an interesting fact, this group is also a product of two independent subgroups

$$\mathrm{Aut}(K)_1 = \{\mathrm{id}, (\rightarrow_S \leftrightarrow \leftarrow_S, \uparrow_A \leftrightarrow \downarrow_A)\} \tag{5.31}$$

$$\mathrm{Aut}(K)_2 = \{\mathrm{id}, (\uparrow_S \leftrightarrow \downarrow_S, \rightarrow_A \leftrightarrow \leftarrow_A)\} \tag{5.32}$$

$$\mathrm{Aut}(K) = \mathrm{Aut}(K)_1 \cdot \mathrm{Aut}(K)_2 \tag{5.33}$$

$$:= \{k_1 k_2 \in \Pi_{\mathcal{S}} \times \Pi_{\mathcal{A}} | k_1 \in \mathrm{Aut}(K)_1, k_2 \in \mathrm{Aut}(K)_2\} \tag{5.34}$$

Those subgroups describe the mirroring around the two axes. On the first view, it is conspicuous that the sensor's permutation is a 90° rotation of the actuator's one. That leaves from the fact that in reality, we can only speak about pairs of sensor and actuator permutations and can not discuss them separately.

The Memory Size Dependency of symmetries for both methods (intrinsic / extrinsic) is shown in Fig. 5.9 on page 54. Please keep in mind that both methods will keep every operation perfectly symmetric for $|\mathcal{M}| = 1$. So we see that the number of found best symmetries decreases with increasing $|\mathcal{M}|$.

For $|\mathcal{M}| = 2$ we see a very clear structure in the extrinsic symmetries. The extrinsic peaks are distinguished and ordered by $|\lambda|$ where λ is a translation parameter: $(x_0, y_0) = (\lambda + \mu, \lambda - \mu)$. Operations with equal λ and different μ are equal in terms of symmetry utility value. In the intrinsic symmetry spectrum, this structure can not exist because of the missing translation freedom. But the matrix of the intrinsic symmetries shows some salient features. For every sensor permutation, there is a permutation next to it which is equal in terms of symmetry quality. Another similar one can be seen 4 units further on x/sensor-axis. Apparently, this fact is related to the structure of the concepts which is very simply for such a small memory size. Another consequence of a small memory size is the high number of good symmetries.

The case $|\mathcal{M}| = 4$ is shown in Fig. 5.7 on page 52 and was discussed before.

The last considered size is $|\mathcal{M}| = 8$ in Fig. 5.9 on page 54. In this case, the intrinsic approach detects much more symmetries. 6 of the 8 best symmetries in the intrinsic case are synchronous permutations $\pi_S = \pi_A$. The other ones are complementary to each other: The sensor and actuator permutation are swapped, $\pi = (\pi_1, \pi_2)$ and $\pi' = (\pi_2, \pi_1)$. In difference to the experiment with $|\mathcal{M}| = 4$, the identity permutation is one of the best symmetries. By enlarging the memory size any further, the number and kind of symmetries seems not to change much.

In Fig. 5.6 on the facing page, we plotted the number of best extrinsic and intrinsic symmetries in dependence on the memory size $|\mathcal{M}|$. To give a criterion for a "good" symmetry,

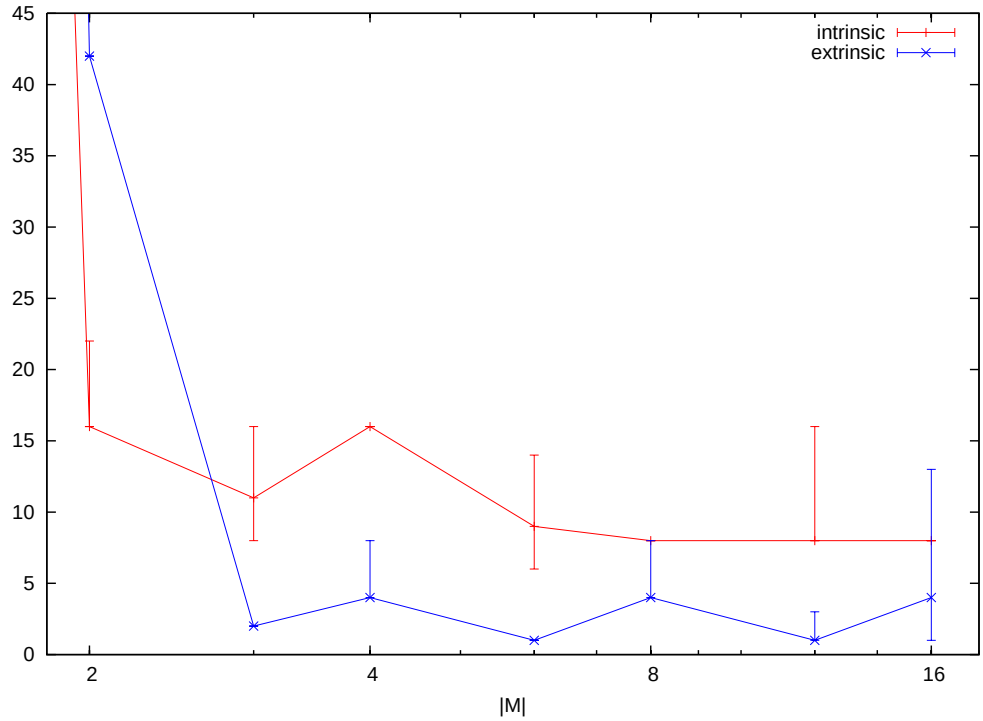

Figure 5.6: Number of good symmetries in dependency on memory size
Given an initial position capturing agent with sensor s+ and actuator a+. Each line shows the number of intrinsic (red) and extrinsic (blue) good symmetries. The according concepts are shown in Fig. A.7 on page 67. The y-axis describes the number of symmetries with a relative symmetry utility value $\geq 85\%$. The error bars show the number of symmetries with a relative symmetry utility value $\geq 82.5\%$ resp. 87.5%.

we use its relative utility value[15]. We count the number of symmetries with a relative utility value $\geq 85\%$. To make a statement about the quality of this threshold, we do the same for 82.5% and 87.5% as well and visualize that as error bars. If latter are big (which occurs), the cutoff for counting has big effect on the number of "good" solutions. As we see in the figure, the number of found symmetries decreases for both methods with growing $|\mathcal{M}|$. The intrinsic symmetries go down to 8, the extrinsic to 1 for a memory size $|\mathcal{M}| \gtrapprox 8$. A number of 8 "good" symmetries for the grid world corresponds to intuition, considering the 8 rotations and mirrorings.

[15] relative utility value $= \frac{\text{utility value}}{\text{maximal utility value}}$

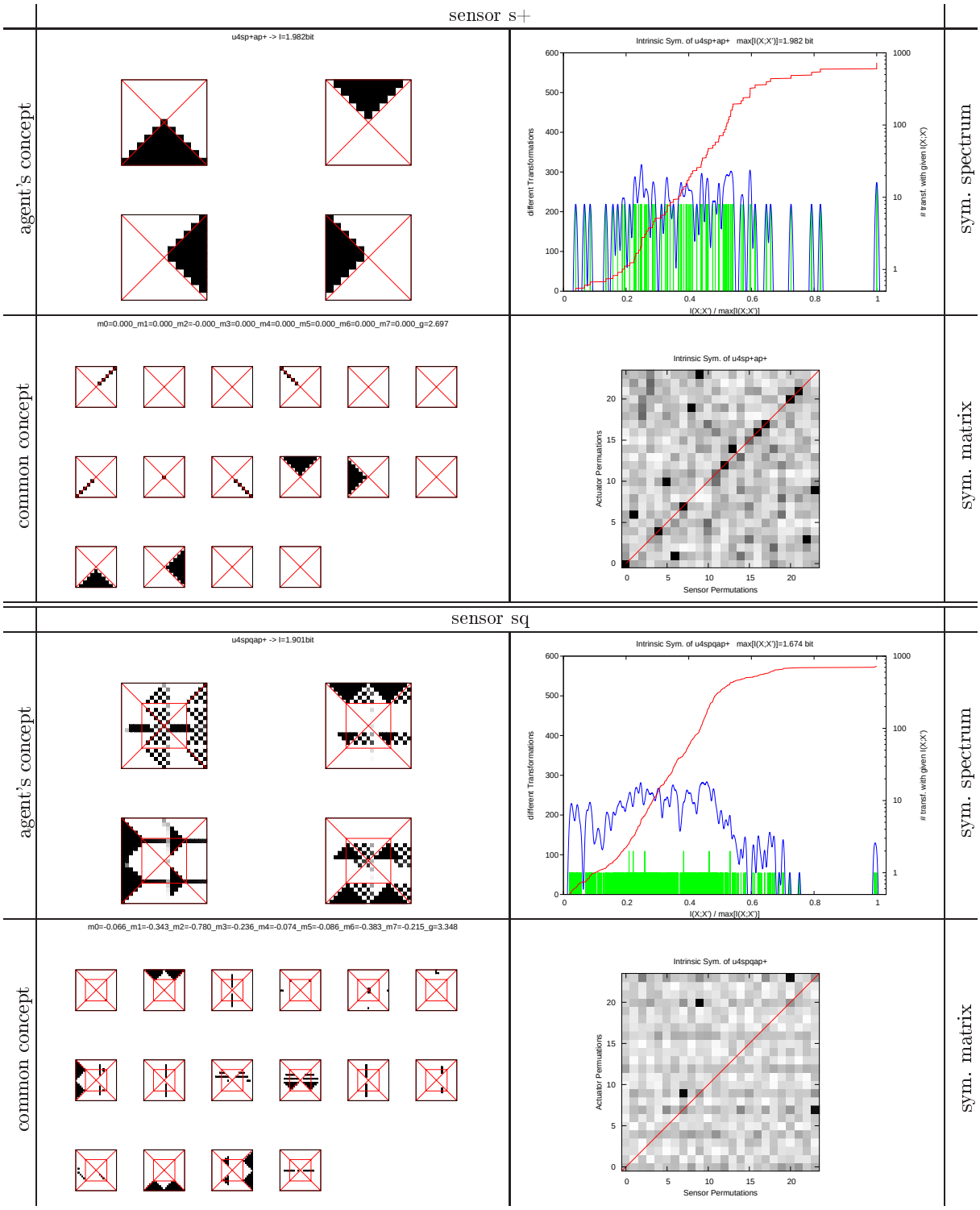

Figure 5.7: Intrinsic symmetries
The agents are identical to those from Fig. 5.3 on page 44 and use sensor s+ (upper half) / sq (lower half). The first picture is the input concept of which the symmetries are calculated (not enlarged like in Fig. 5.3). The second picture is the spectrum of the intrinsic symmetry utility values and the spectrum's derivative. The third picture shows an objective common concept calculated out of 8 other good solutions of the initial position capturing (Fig. A.5 on page 65). The forth picture shows the utility value of all intrinsic symmetries as greyscale (with enlarged black-white contrast for high values $x \mapsto e^x - 1$) in dependency on different sensor (y-axis) and actuator (x-axis) permutations. The diagonal line marks the synchronous permutations with $\pi_S = \pi_A$. The identity permutation is shown in the left lower corner of the picture. Extrinsic symmetries for the upper common concept are shown in Fig. 5.2 on page 43.

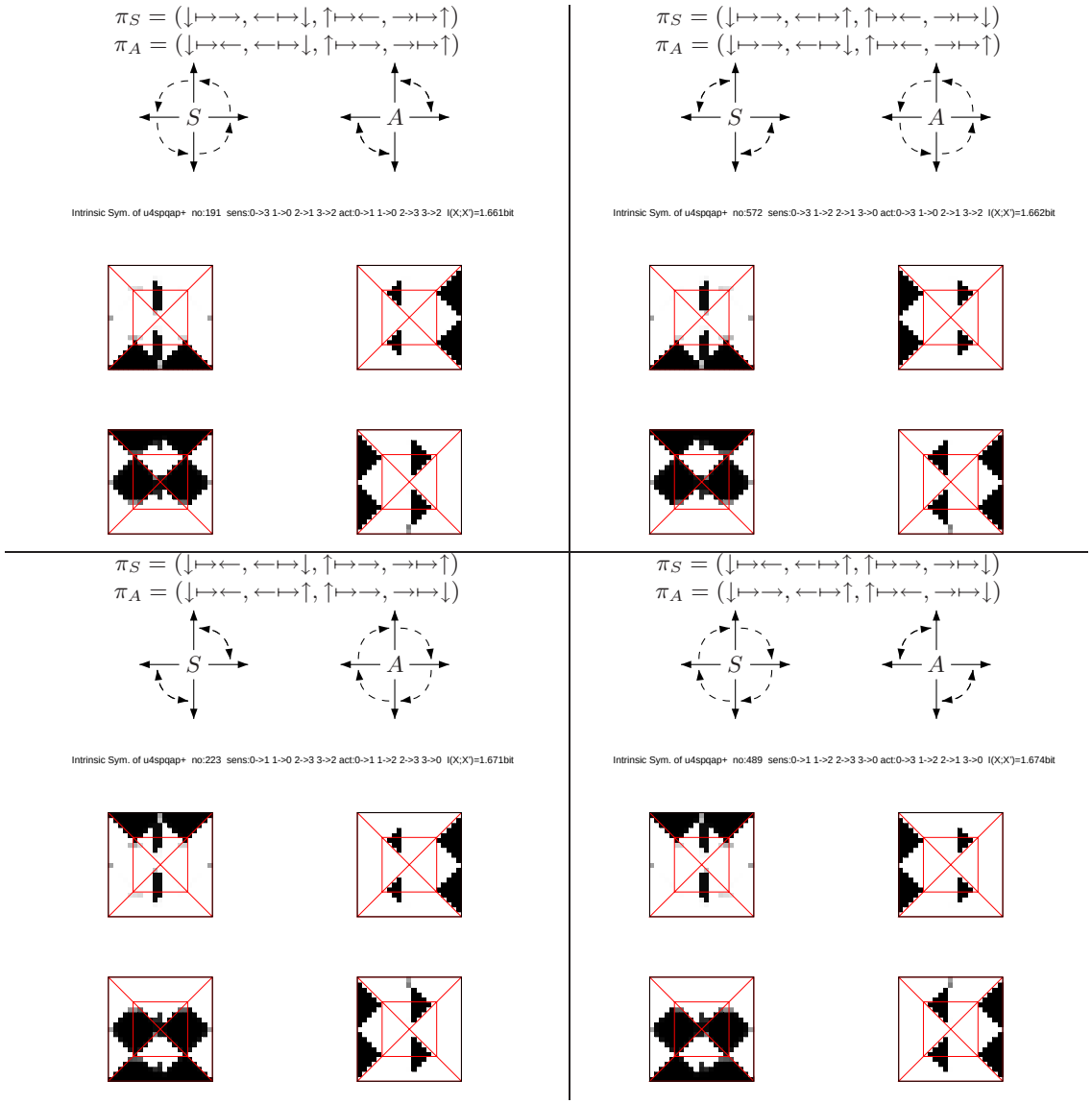

Figure 5.8: Intrinsic symmetries applied on concept

The 4 best ($\frac{I(Y;Y^\xi)}{\max[I(Y;Y^\xi)]} \approx 1$) transformed concepts (R_0, M_{15}) of an initial position capturing agent with actuator a+ and sensor sq are shown. The untransformed agent and other figures for this setup are shown in Fig. 5.7 on the preceding page. Above each concept, the corresponding intrinsic symmetry operation $\pi = (\pi_S, \pi_A)$ is shown. These permutations are also visualized by the two small graphs below the tuple description. They show how the 4 sensor resp. actuator values $\{\downarrow, \leftarrow, \uparrow, \rightarrow\}$ are permuted. All 4 symmetries have an initial position capturing utility value about $I(R_0; M_{15}) \approx 1.66$ bit.

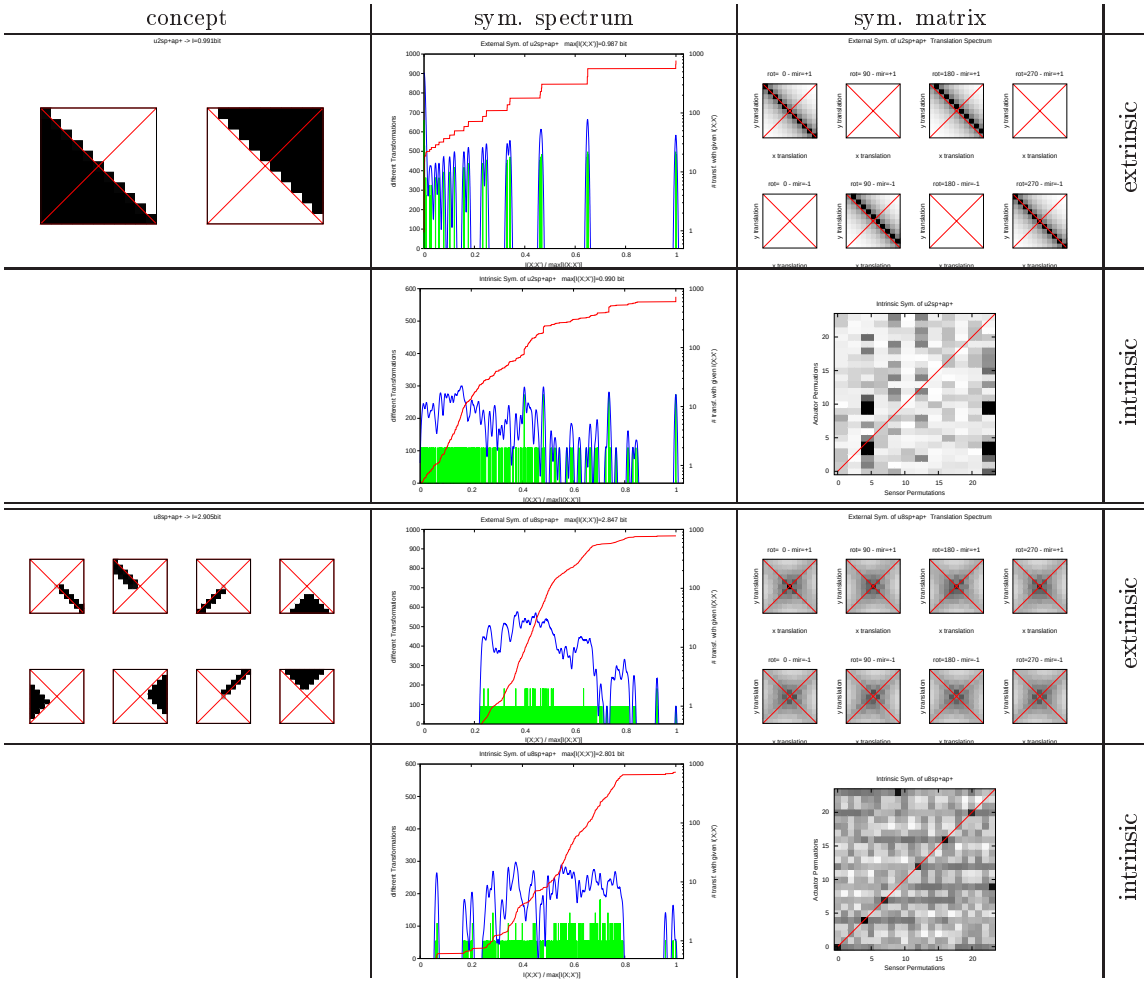

Figure 5.9: Intrinsic vs. extrinsic symmetries in memory size dependency
Agents with initial position capturing, actuator a+, sensor s+ and $|\mathcal{M}|$ =2 (upper half) / 8 (lower half) are used.
The first picture is the input concept (R_0, M_{15}) whose symmetries are calculated. The centered pictures show the spectrum of the extrinsic (upper) resp. intrinsic (lower) symmetry. On the right, the same (intrinsic/extrinsic) operations are displayed with their utility values as matrix. The according objective common concept and its input concepts are shown in Fig. A.6 on page 66.

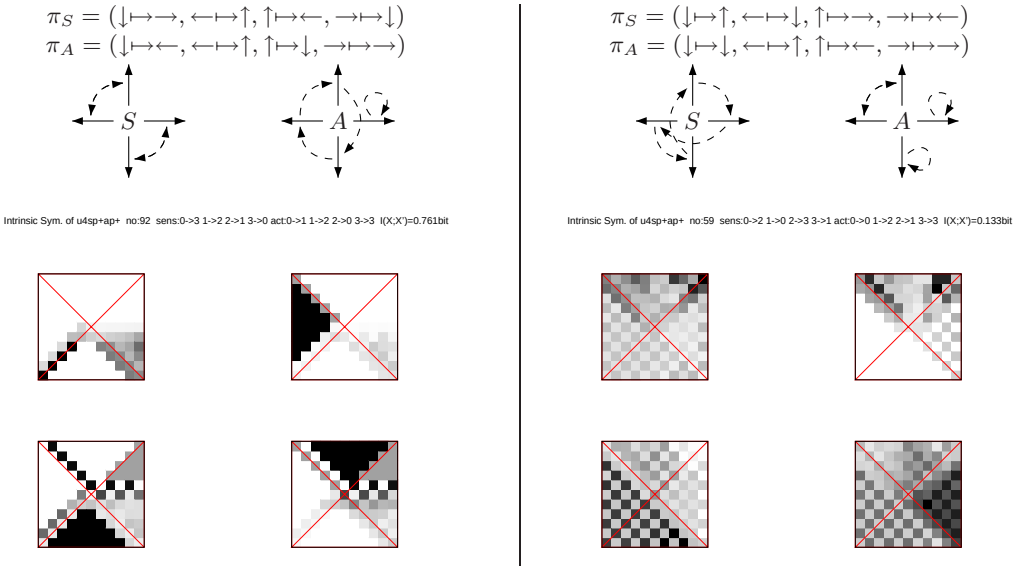

Figure 5.10: Bad examples of intrinsic symmetry
Used is an initial position capturing agent with sensor s+, actuator a+ and $|\mathcal{M}| = 4$. Two bad example of intrinsic symmetry with a relative symmetry utility value $\frac{I(Y;Y^{\xi})}{\max[I(Y;Y^{\xi})]}$ of 0.384 (left) resp. 0.067 (right) are shown.
Above each concept, the corresponding intrinsic symmetry operation $\pi = (\pi_S, \pi_A)$ is shown. These permutations are also visualized by the two small graphs below the tuple description. They show how the four sensor resp. actuator values $\{\downarrow, \leftarrow, \uparrow, \rightarrow\}$ are permuted. One of the best symmetries is the identity. Latter and its symmetry spectrum are shown in Fig. 5.7 on page 52.

5.3 Discussion

We have evidence that "good" agents' concepts and especially common concepts have a higher degree of "symmetry". That raises the question how we could study those symmetries which are strictly speaking not symmetrical in a mathematical way ("weak symmetries"). We presented two methods in this section. The extrinsic symmetry uses our (external) knowledge about the symmetry group operating on the world of the agents. Rotation and mirroring symmetries were found and are very symmetric. Translations were also available but much weaker. We were able to see some "long distance" similarities with sensor sq. They are stronger than neighboured translations but much weaker than small translations around the identity. This fact made them invisible in the symmetry spectrum, they were too weak and vanished in the "noise" of the other translations. But they can be seen in the matrix visualization of the translations.

As opposed to the extrinsic, the intrinsic symmetry is nothing an external observer knows about the world the agents are living in (Philipona et al. (2004)). One just observes which changes in the agents' interaction with the environment (actuator/sensor permutations) have no (bad) effect on their concept. This method is additionally improved in that we do not compare the permuted concept with the individual (original) concept but with a common one. This common concept is "free" of special decisions of individual agents and represents a more universal set of symbols for a task than any individual solution. The intrinsic operation forces the agents by "transplanting their brain into another body" to "speak" the common language without being optimized again. Searching for the best intrinsic operations is partly a re-optimization of the controller, but in total we only test a vanishingly small part of the search space. The meaning of intrinsic symmetry is not fully understood yet. Partly, they are identical to the extrinsic symmetries (rotation, mirroring) but there are many more operations. Furthermore, we could see that the number of best intrinsic symmetries as opposed to the extrinsic ones do not drop to 1 by increasing the agent memory size, what means that the intrinsic symmetries are not too sensitive for possible variations of the concept due to operations.

This raises another interesting idea concerning these symmetries. The intrinsic symmetry may give us a hint for an optimal memory size of an agent. With growing memory, the agents begin to "realize" that not every symmetry they "see" is really in the world. But this process stops at a certain $|\mathcal{M}|$. Such a memory size might be a good choice for an agent in the considered environment. Here, he "realizes" every "symmetry break" and is in terms of symmetry perfectly educated. One problem with this method is that the number of symmetries depends strongly on the applied threshold.

6 Conclusion

This thesis shows how to handle the uncertainty evolving from different optimization results for agents. The concepts of a group of agents can be fusioned into a single concept with two different methods presented in here: The objective common concept has explicit knowledge about the world and is in contrary to the subjective common concept highly efficient computable. But the subjective common concept is able to work without such explicit knowledge about the world. In our experiments, the resulting concepts for both methods are almost identical. These common concepts figure out features of the world that single agents do not find resp. not in such a high quality. So the common perspective seems to be a good "second hierarchical layer" to find interesting structures of the world.

Comparing concepts of individual agents, a concept of a good initial position capturing agent is more symmetric than a concept of a non-optimized one. Furthermore, common concepts seem to have even higher regularities resp. much higher symmetries than good individual concepts. But these symmetries are not always perfectly fulfilled. Therefore, we developed two information theoretical approaches for studying such weak symmetries: The extrinsic symmetry considers the mirroring, rotation and translation operations of the two-dimensional grid world and compares the transformed concept with the original one by applying mutual information. Here we see that this method is able to identify all intuitive symmetries and also tolerates small differences. In total, it is able to identify the symmetry of an individual concept but not of the world and therefore suffers of many unnecessary symmetry breaks. Therefore, we developed a second approach, the intrinsic symmetry. It performs permutations of the sensor and actuator states; so the embodiment of an (already optimized) agent is changed. Additionally, we compare the resulting concepts to a common concept to remove every individual decision of the agents. With this technique, we are able to find out what operations do not influence the ability of the agent to "speak a common language". Rotation and mirroring are only 8 of many possible intrinsic operations which are in total not always intuitive. Some of these contra-intuitive intrinsic operations are rated in our experiments as very symmetric what will need further investigation. Instead of looking directly onto the operations, we get more obvious results by looking at the set of automorphism operations between them.

7 Future Work

Some interesting questions arised while writing this thesis from which further investigations might start:

- For both common concepts and symmetries, it would be interesting to see how they work on more complex environments. Maybe these techniques have to be adapted to the new setting. (However, there could be new phenomena.)
- In common concepts, we could investigate more detailed the effects of the parameter α and if it may be replaced by something else.
- There should be further explorations in defining a best common concept's size since we designated only possible candidates.
- We might factorize the common concept somehow into common and not common symbols to have a part that all input concepts have in common. By plotting the probability spectrum of a common concept's symbols, we might find a gap between those two classes. Additional to this, there should be other more information theoretic approaches to distinguish them as well.
- In this work, we did not put much effort in computing the best solutions efficiently. Especially for the common concept, there should be a much more efficient way to compute good solutions. Therefore, start with a solution $Y^*_{obj} = X$ for the objective resp. $Y^*_{subj} = Y_1 \times \ldots \times Y_n$ for the subjective common concept and identify values of $y \in \mathcal{Y}^*$ which can be combined to a single symbol without loss of utility value. The remaining symbols should be combined iteratively until the number of symbols in the common concept follows the restriction of the experiment. This iterational grouping might use a greedy strategy.
- There might be an analogon to translation symmetries in the intrinsic case in form of shifting the initial memory state.
- To investigate the intrinsic operations' meaning further, we should put more effort in examining the structure of the controller search space: We would like to ensure that the appearance of a symmetric solution has indeed a low probability. The role of the automorphisms between the good symmetries is a candidate for further investigations as well.
- It would be interesting to see if there is a possibility to separate symmetric and asymmetric features of the concepts. A possible starting point could be the techniques presented in Franco and Polani (2003).

8 Acknowledgements

This research was conducted during a visit to the University of Hertfordshire, UK, and is the product of many people's contributions. I would like to thank Daniel Polani for the guidance and supervision that he made available to me at all hours of the day and night, Barbara Drossel for her insightful suggestions and helpful ideas and my friends who helped me to revise and proofread this text: David Scheffler, Martin E. Schauer and Ursula Goltz. I would especially like to thank Claudia Möller whose patience, diligence and countless hours of revising have helped transform this text into its current form.

A Appendix

A.1 Superstition in Agent Groups

In this thesis, we defined superstition in Sec. 4.3 on page 29 as a behaviour of a concept to use different symbols $y \in \mathcal{Y}$ for one $x \in \mathcal{X}$. Our initial interest was directed to a different phenomenon which was unfortunately not covered by the modelling our experiments base on.

Consider a group of agents with a joint distribution of their concepts

$$p(x, y_1, \ldots, y_n) \tag{A.1}$$

which is not (exclusively) connected through a shared knowledge about X. Here, we have conditional multiinformation

$$I(Y_1; Y_2; \ldots; Y_n | X) \tag{A.2}$$

between the concepts which is not depending on X.

If this conditional multiinformation is non-zero, the agents "are talking about something not related to the world". So we might intuitively call this agent group superstitious (which was our initial intention to use this term). In our modelling, these phenomena can not appear: Equation A.1 is achieved by a combination of

$$p(x, y_1), \ldots, p(x, y_n) \tag{A.3}$$

into[16]

$$p(x, y_1, \ldots, y_n) = p(x) \cdot \prod_{i=1}^{n} p(y_i | x) \tag{A.5}$$

$$= p(y_1, x) \cdot \prod_{i=2}^{n} \frac{p(y_i, x)}{\sum_{y_i} p(y_i, x)} \tag{A.6}$$

which results from the causal Bayesian network in Fig. A.1 on the facing page (compare Fig. 4.1 on page 27 resp. Fig. 4.2 on page 29). This leads to the fact that $Y_1, \ldots, Y_n$ are independent once X is known. So the conditional multiinformation between them vanishes:

$$I(Y_1; Y_2; \ldots; Y_n | X) = 0 \tag{A.7}$$

[16] Therefore, we demand that every concept has a distribution equal in x:

$$p(x) = \sum_{y_1} p(y_1, x) = \ldots = \sum_{y_n} p(y_n, x) \tag{A.4}$$

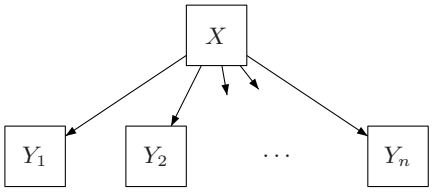

Figure A.1: Bayesian network for common concept

Nevertheless, the subjective common concept from Def. 4.4 on page 29 is in principle able (after small adaptions) to handle experiments including this kind of superstition.

A.2 Additional Figures

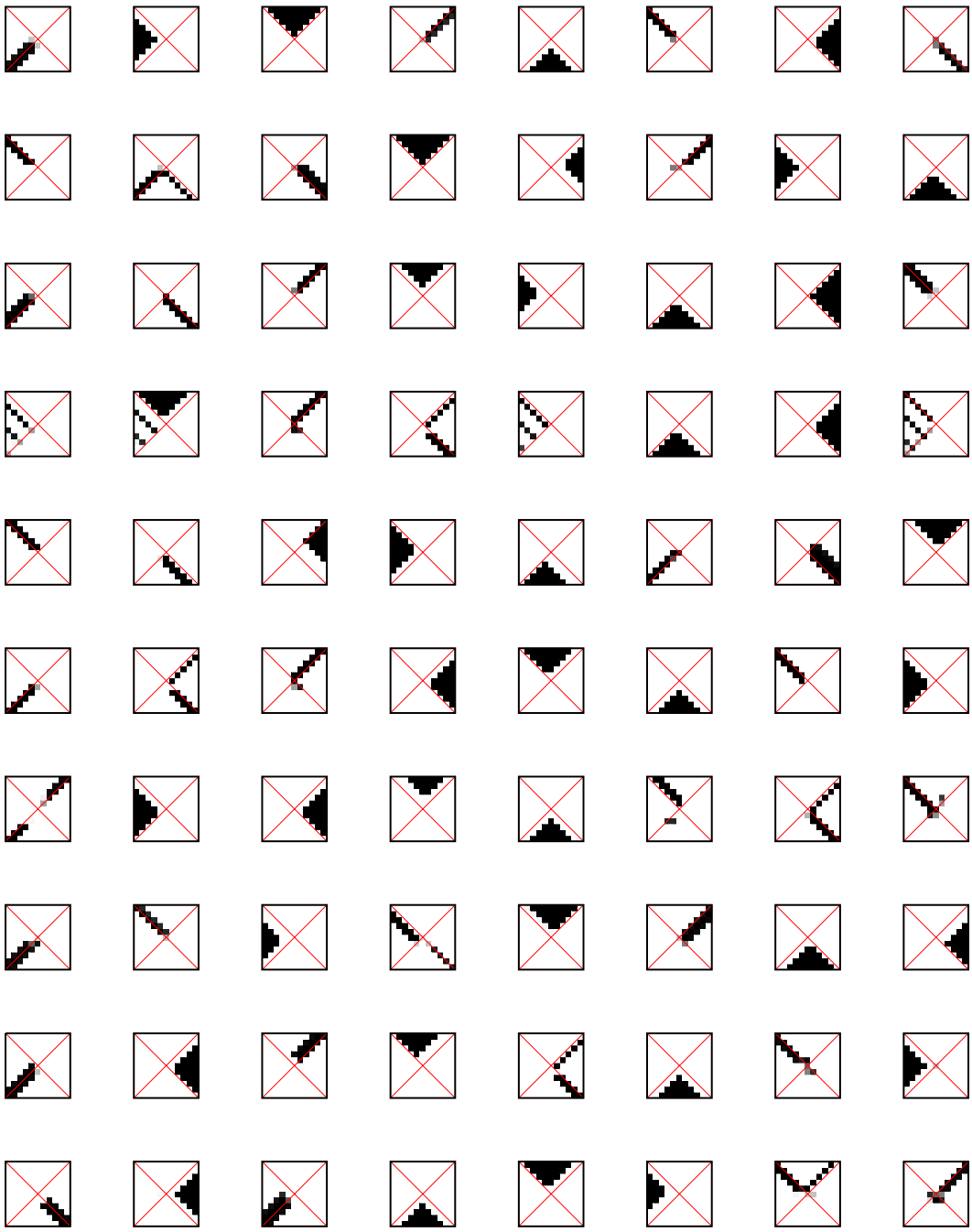

Figure A.2: Input concepts for optimal common concept's size experiment
Every line contains one input concept for the experiment in Fig. 4.7 on page 37. In total, 10 results of an initial position capturing agent with actuator a+, sensor s+ and $|\mathcal{M}| = 8$ are shown.

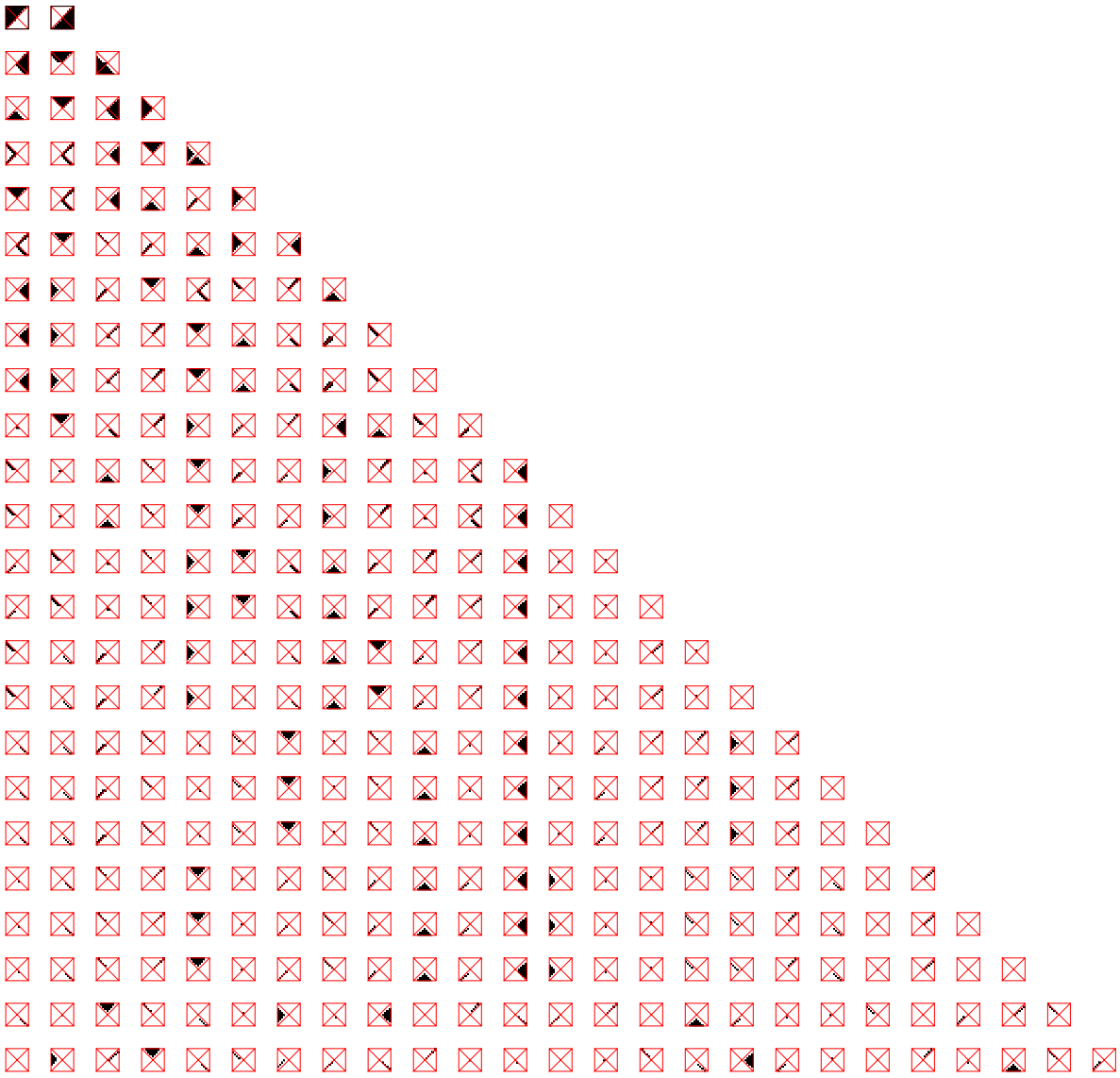

Figure A.3: All common concepts for optimal common concept's size experiment
Every line contains one common concept for the experiment in Fig. 4.7 on page 37. The maximal common concept's size we show is 25, bigger common concepts are equal to this.

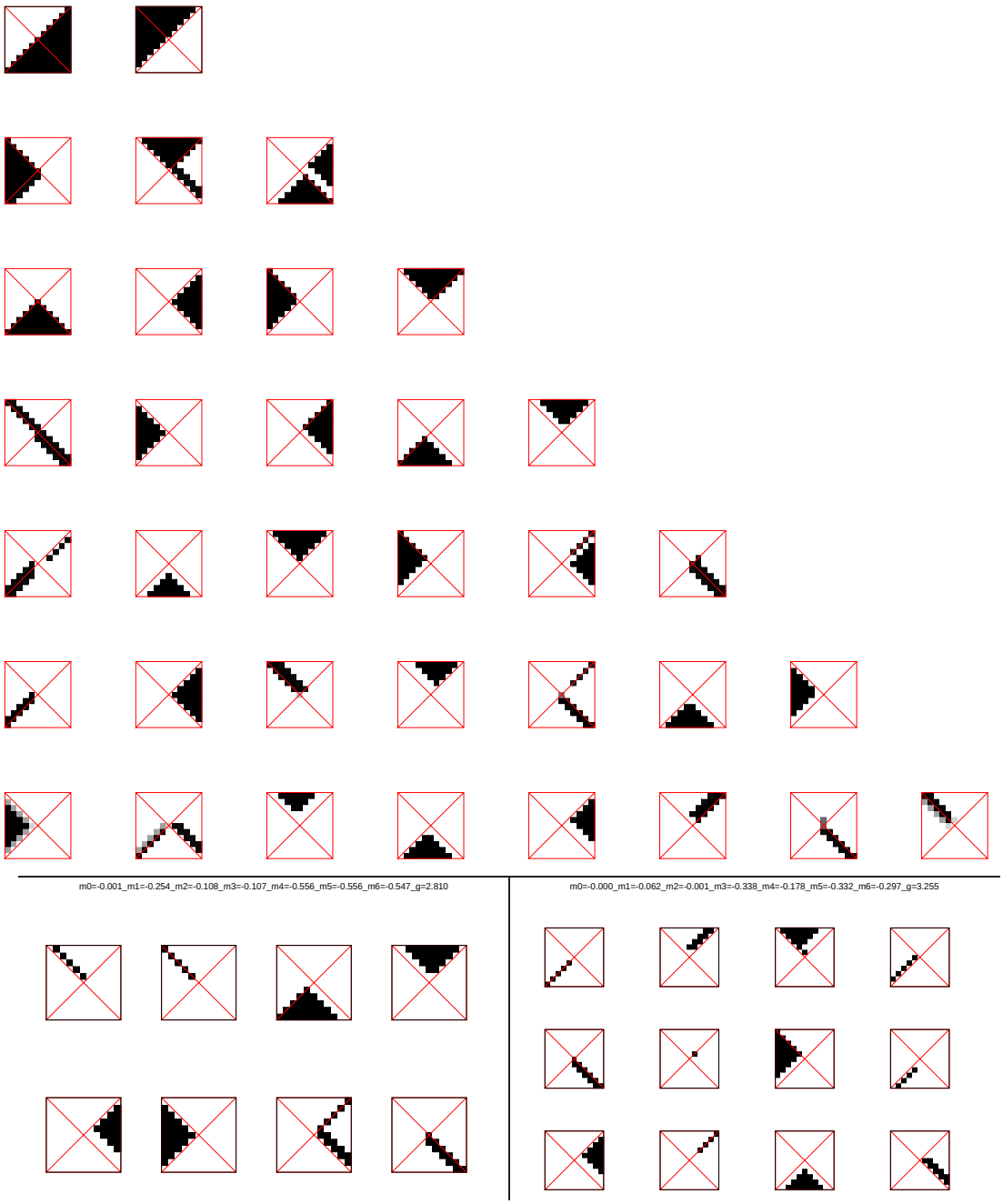

Figure A.4: Common concept generated from input concepts with different sizes
In the upper half, the results (one result each line) of an initial position capturing agent with actuator a+, sensor s+ and $|\mathcal{M}| = 2, \ldots, 8$ are shown. In the lower half, two objective common concepts with different sizes are represented.

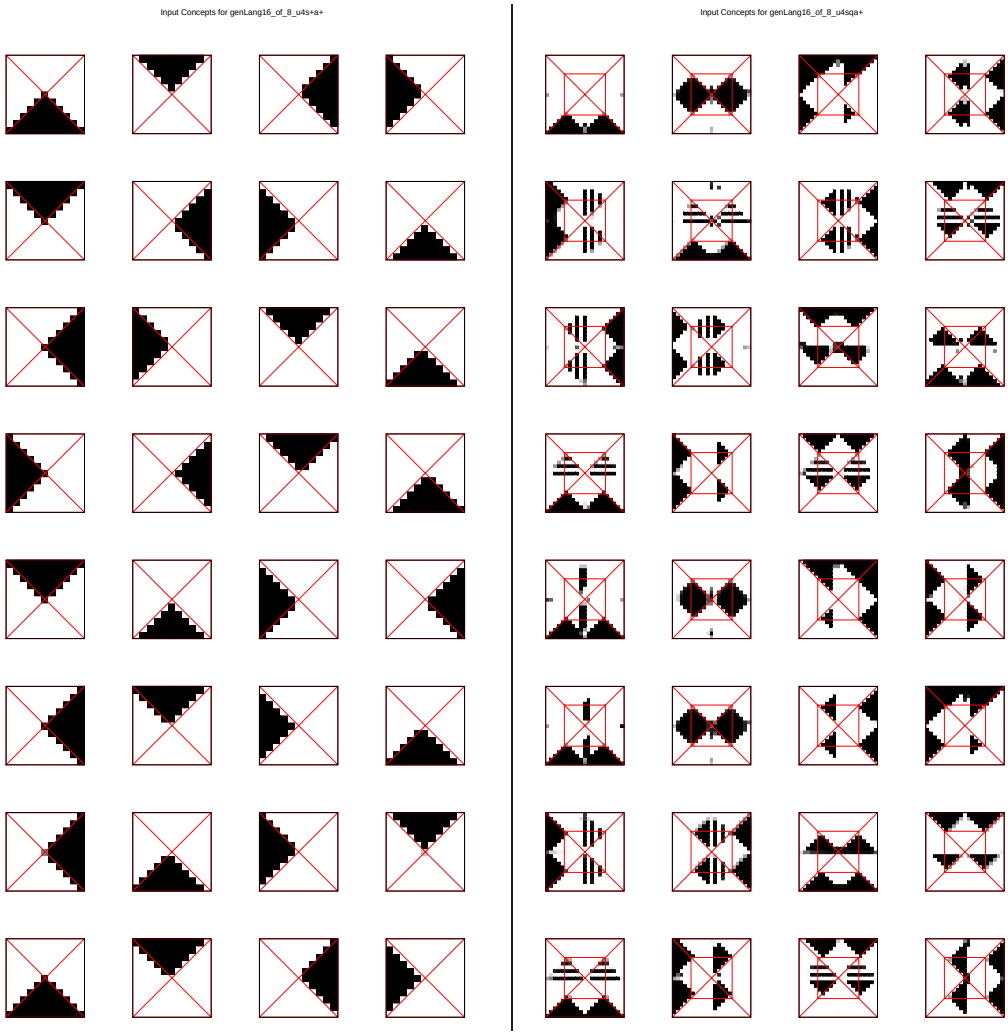

Figure A.5: Input concepts for common concept used in intrinsic symmetry
These are the 8 input concepts for the common concept in Fig. 5.7 on page 52, on the left with sensor s+ and on the right with sensor sq.

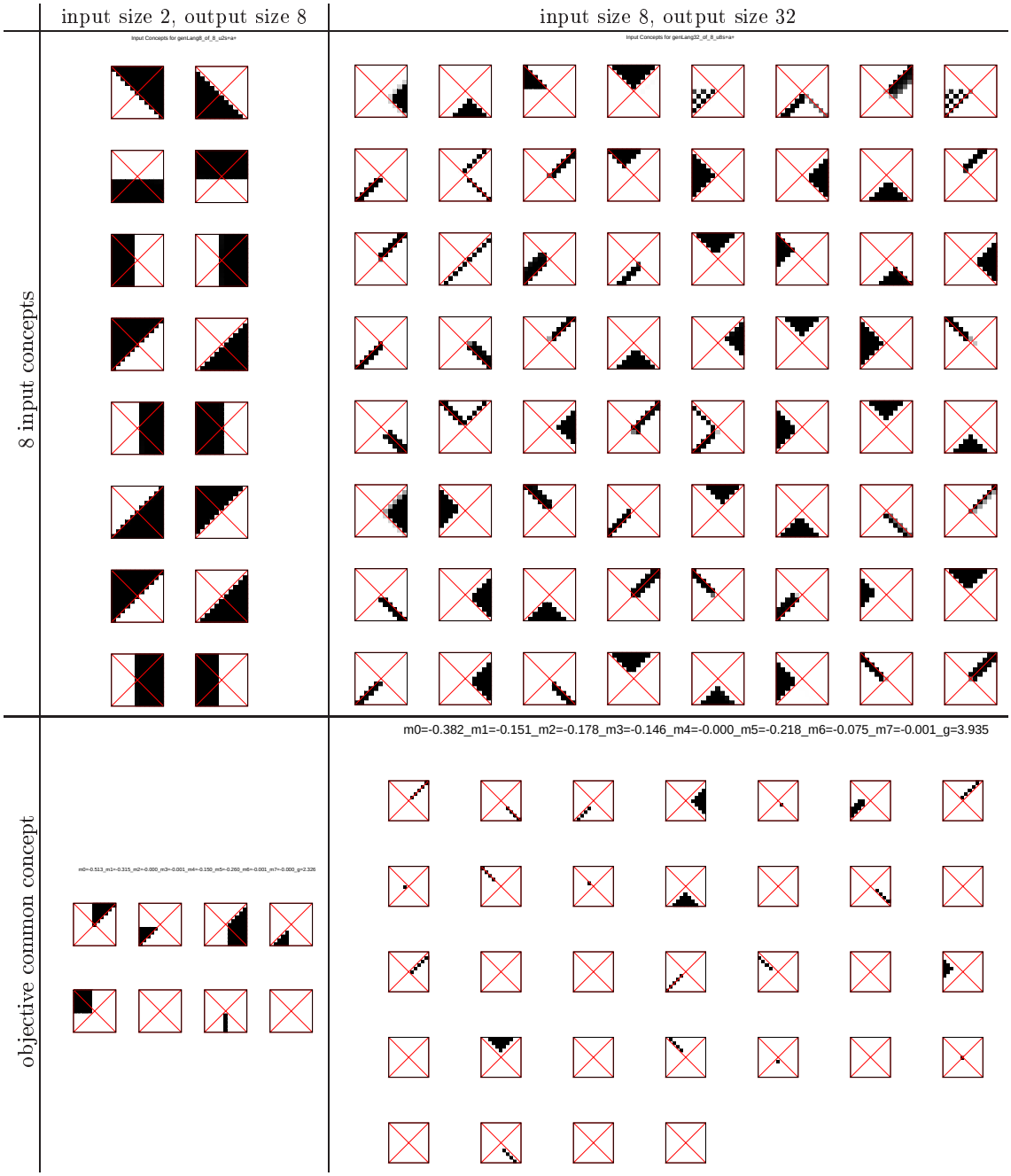

Figure A.6: Input concepts and common concept used in intrinsic symmetry
The lower half shows an objective common concept, the upper half the 8 according input concepts (left $|\mathcal{M}| = 2$, right $|\mathcal{M}| = 8$). They are used to calculate the intrinsic symmetry in Fig. 5.9 on page 54.

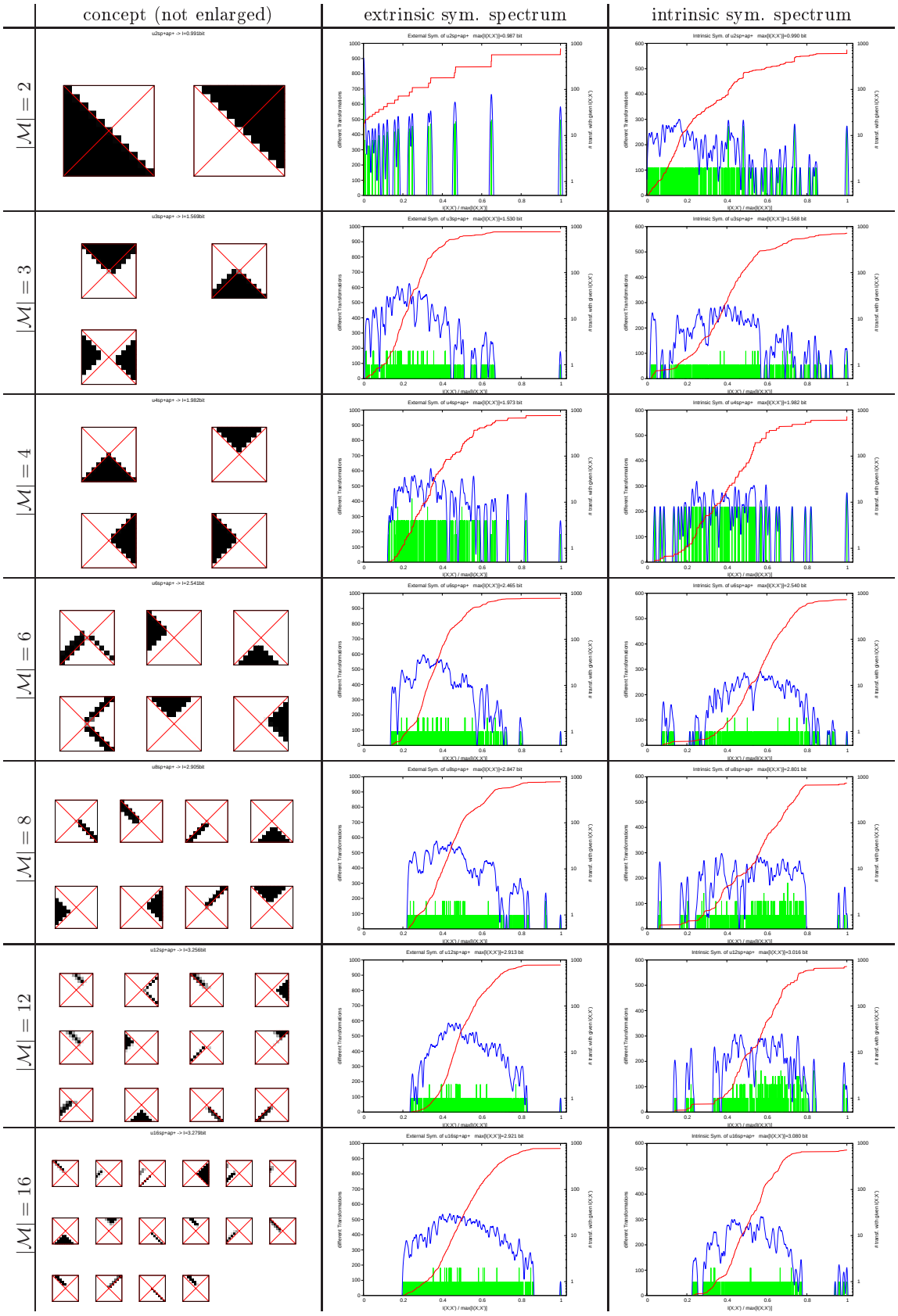

Figure A.7: All intrinsic and extrinsic symmetries used in memory size dependency
These are the concepts (R_0, M_{15}) of initial position capturing agents with sensor s+ and actuator a+, each row with a different memory size $|\mathcal{M}| \in \{2, 3, 4, 6, 8, 12, 16\}$. The concept is shown in the first, the extrinsic symmetries' spectrum in the second and the intrinsic symmetries' spectrum in the third column. They are used in Fig. 5.6 on page 51.

B Index

Action, 13
Actuator, 13
Acyclic directed graph, 11
Agent, 12

Bayesian network, 11

Causal Bayesian network, 11
Common Concept
    Objective, 27
    Subjective, 29
Common concept, 26
components, 12
Concept, 17, 26
    Common, 26
Conditional entropy, 8
Controller, 15
    Deterministic, 16
Coordinate free, 8
Cycle, 11

Deterministic controller, 16
Deterministic Mapping, 23
Directed graph, 10

Edge, 10
Embodiment, 15
Entropy, 8
    Conditional, 8
Entropy related quantities, 8
Extrinsic symmetry, 40
Extrinsic symmetry operation, 40
Extrinsic symmetry utility, 41

Fuzzy neighbourhood of symbols, 33

Graph, 10

Information, 7
    Distance, 8
    Multiinformation, 9
    Mutual, 9
Information related quantities, 9
Initial position capturing, 17
Intrinsic symmetry, 46
Intrinsic symmetry operation, 46
Intrinsic symmetry utility, 47

Mealy machine, 16
Memory, 15
Multiinformation, 9
Mutual information, 9

Neighborhood function, 21
Node, 10
Notation, 7

Object, 26
Objective common concept, 27

Parents, 10
Path, 11
Perception-action loop, 12
Power set, 7
Probability, 7
Problem and optimal solution, 21
Problem space, 21

Random variable, 7

Sensor, 13
    s+, 13
    sq, 13
Set, 7
Shannon Entropy, 8
Simulated Annealing, 21
Solution, 21
    Good, 21
    Optimal, 21
Subjective common concept, 29

Symbol, 17, 26
Symmetry, 39
Symmetry operation
    Extrinsic, 40
    Intrinsic, 46

Temperature, 21
Time evolution, 12
Trosion, 48

Utility function, 21

World, 12

C List of Figures

2.1 Relationship between entropy and information related quantities 10
2.2 Perception-action loop as a causal Bayesian network 12
2.3 Sensor s+ 14
2.4 Sensor sq 14
2.5 Actuator a+ 15
2.6 Perception-action loop with deterministic controller as a causal Bayesian network 16
2.7 Solution of initial position capturing 18
2.8 Example for a Mealy machine representation of a deterministic controller . . 19
2.9 Bad solution of initial position capturing 19
3.1 Comparison of different temperature functions in Simulated Annealing . . . 25
4.1 The Bayesian network for an objective common concept 27
4.2 The Bayesian network for a subjective common concept 29
4.3 Common concepts 31
4.4 Common concepts with sizes bigger than input concepts' sizes 32
4.5 Neighbourhood inside of a concept 34
4.6 Neighbourhood inside of a concept - more related 35
4.7 Quality of different common concept sizes 37
5.1 Bayesian network for calculating extrinsic symmetry utility 41
5.2 Extrinsic symmetries of objective common concept 43
5.3 Extrinsic symmetries 44
5.4 Application of extrinsic symmetries on concept 45
5.5 Perception-action loop with modifications to calculate intrinsic symmetry utility function 47
5.6 Number of good symmetries in dependency on memory size 51
5.7 Intrinsic symmetries 52
5.8 Intrinsic symmetries applied on concept 53
5.9 Intrinsic vs. extrinsic symmetries in memory size dependency 54
5.10 Bad examples of intrinsic symmetry 55
A.1 Bayesian network for common concept 61
A.2 Input concepts for optimal common concept's size experiment 62
A.3 All common concepts for optimal common concept's size experiment 63
A.4 Common concept generated from input concepts with different sizes 64
A.5 Input concepts for common concept used in intrinsic symmetry 65
A.6 Input concepts and common concept used in intrinsic symmetry 66
A.7 All intrinsic and extrinsic symmetries used in memory size dependency . . . 67

D List of Algorithms

3.1 Used variant of Simulated Annealing in pseudo code 22

E References

Crutchfield, J. (1990). Information and its Metric. In Lam, L. and Morris, H., editors, *Nonlinear Structures in Physical Systems – Pattern Formation, Chaos and Waves*, pages 119–130. Springer Verlag.

Franco, S. and Polani, D. (2003). Bold hearts 2003: Skill learning via information-theoretical decomposition of behavior features. Technical Report 399, University of Hertfordshire.

i Cancho, R. F. and Solé, R. V. (2003). Least effort and the origins of scaling in human language. *PNAS 788-791*, 100(3):788–791.

Klyubin, A. S., Polani, D., and Nehaniv, C. L. (2007). Representations of space and time in the maximization of information flow in the perception-action loop.

Möller, M. (2008). Information theoretical models for hierarchical structured agents and their computation. Bachelor's thesis, Department of Computer Science, Technische Universität Darmstadt.

Philipona, D. and O'Regan, J. (2006). Color naming, unique hues, and hue cancellation predicted from singularities in reflection properties. *Visual Neuroscience*, 23(3-4):331–339.

Philipona, D., O'Regan, J., Nadal, J.-P., and Coenen, O. (2004). Perception of the structure of the physical world using unknown multimodal sensors and effectors. In Thrun, S., Saul, L., and Schölkopf, B., editors, *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA.

Steels, L. (1997). The synthetic modeling of language origins. *Evolution of Communication*, 1(1):1–34.

Tishby, N., Pereira, F., and Bialek, W. (1999). The information bottleneck method. In *Proceedings of the 37-th Annual Allerton Conference on Communication, Control and Computing*, pages 368–377.

Vergassola, M., Villermaux, E., and Shraiman, B. I. (2007). 'infotaxis' as a strategy for searching without gradients. *Nature*, 455:406–409.