A theory of interaction semantics

Johannes Reich, johannes.reich@sap.com

I dedicate this article to Bernd Finkbeiner

February 26, 2022

Abstract

The aim of this article is to delineate a theory of interaction semantics and thereby provide a proper understanding of the "meaning" of the exchanged characters within an interaction.

I propose 7 criteria any theory of interaction semantics has to fulfill at least. It should

1. be consistent with the model of information transport and processing;
2. clarify the relationship to models of formal semantics;
3. account for the fact that the meaning of a character is invariant against appropriate renaming;
4. locate the concept of meaning in our usual technical description of interactions;
5. define when two different characters have the same meaning;
6. define what is "interpretation" and what is an "interpretation context"; and
7. explain under which conditions meaning is compositional, and when it is not.

The key idea is to approach the semantics of an interaction as we do for a formal language. This approach consists of two steps: first to assign values to variables and second to provide meaning by an interpretation function.

A natural choice for the variables are the state functions of the interacting systems, assigning values at each time step. Thereby the description of a system's behaviour with an input/output-transition system (I/O-TS) becomes a representation of the variable-to-value assignments.

To identify the interpretation function I propose to model the interaction of systems based on Shannon's theory of information with the protocol concept, complemented by decisions to form a "game in interactive form (GIF)". Decisions in this sense determine the transition relation and thereby create a transition function. Then the natural choice for the interpretation function is the transition function of the GIF. In this sense, the interpretation of the interaction becomes its execution. Now we can say that the interpretation of the characters during the GIF's execution results in their meaning, the result of the mapping. Equivalent meaning is based on resulting equivalent states of the world.

Interestingly, based on the decisions we can partition any GIF into a deterministic traditional automaton where the states represent equivalance classes of GIF-states related to a single decision. Except for the utility function, this automaton is equivalent to a traditional game in extensive form. Except for the utility function, this is equivalent to a traditional game in extensive form, which allows the statement that from the view of the proposed theory of interaction semantics, traditional game theory actually abstracts from interactions and deals with the meaning of decisions.

The fuzzyness of meaning can be traced back to the fuzzyness of the consequences of decisions and interactions – as finally everything depends somehow on everything else in a virtually infinite interaction network and we have to make simplifying assumptions to limit the the section of the world we want to comprehend. In fact, we get a simple criterion how we can truly simplify our world of interactions by guaranteeing true freedom of decisions at least in the interacting subjects.

All in all, I show that the proposed theory of interaction semantics does fulfill all proposed criteria.

1 Introduction

In our normal live, the concepts of understanding and meaning have proven to be extremely powerful. Intuitively, we say that the signs exchanged in a conversation have a meaning that is to be understood by the participants. What do we mean by that? What are the benefits to talk this way? This problem, posed this way or another, has puzzled philosophers and scientists since ancient times.

Meaning is a key concept for our understanding of our natural language. It was the late Ludwig Wittgenstein [Wit53] who stressed the function of language as a tool for interaction with his famous remark "For a large class of cases of the employment of the word 'meaning' — though not for all — this word can be explained in this way: the meaning of a word is its use in the language" (paragraph 43)

Based on his impression of the interactive nature of language, David Lewis [Lew69] was the first to introduce game theory to analyze social conventions and in particular to analyze the conventional use of language expressions. He viewed mutual understanding in an exchange of characters as a coordination problem and introduced signaling games as an analytical instrument. In a signaling game, a sender sends some message as a function of its state such that the receiver gains knowledge about this state and becomes capable to choose an action that is beneficial for both. Then, a convention is a solution of such a coordination problem which contains at least two proper coordination equilibria. Karl Warneryd [Wä93] showed that such conventions arises naturally in evolutionary settings, based on the concept of evlutionary stable strategies as developed by John Maynard Smith and George R. Price [SP73].

However, using game theory to analyze the conventional character of language and determine any conception of meaning already requires a very elaborate conceptual framework. We have to know what a game is, what decisions are, how they relate to our physical world, whether a signal is an action or something else, etc. And, even more importantly, traditional games, neither in their strategic nor in their extensive form account for any information exchange, that is interaction. They are only about states, decision and utility (or any other form of preference evaluation). Although, David Lewis tried to fill this gap with signaling games, it is still a rather ad hoc extension.

I therefore propose to start my considerations of meaning with the interactive aspect. This interactive character of meaning was strongly emphasized by Herbert P. Grice [Gri89] by noting that to understand an utterance is to understand what the utterer intended to convey - and that what has been traditionally understood as its "meaning" is only loosely related to that. Quite recently, K.M. Jaszczolt proposed that to understand the concept of meaning one has to investigate "not the language system and its relation to context but principally the context of interaction itself, with all its means of conveying information" ([Jas16] pp.12-13).

Generally speaking, I understand our natural language as a facilitation mechanism for inter-subjective interactions in a social context. In this sense it is a pragmatic solution to the circular or "chicken-or-the-egg"-problem that on the one hand a purposeful interaction requires mutual understanding and on the other hand establishing a mutual understanding requires purposeful interaction. Another well described and increasingly technically well solved circular problem of this kind is that of "simultaneous localization and mapping (SLAM)" problem (e.g. [TBF05]). To determine my own position in a terrain I need a map and to determine the map I need to know my own position. Following the SLAM acronym, I propose to speak of the "simultaneous interaction and understanding (SIAU)" problem. The obvious solution to such problems is iterative: an internal model is increasingly improved by empirically collecting data. The intelligence of the solution then rests in the way, the relevant aspects of the external world are internally represented and in the update mechanism.

In fact, we already do have a well accepted approach, to talk about semantics from a technical perspective, namely about the semantics of formal languages. This approach, to define the semantics of a formal language is by itself not formal in the sense as a formal language is, but is an "ordinary" mathematical concept that consists of two steps: first to assign values to variables and second to provide meaning by an interpretation function. The key idea of this contribution is to apply this approach also to interactions in the line of thought of Gerard Holzmann [Hol91] who already noted the similarities between protocols and natural language. Thus, with the same intuition as we talk about semantics of a formal language, we can talk about semantics of an interaction. And interestingly we will be required to introduce a decision concept and thereby derive the concept of a game as a transition system that relates to the meaning of decisions and, in its traditional form, abstracts from all interactions. Thus, I make a proposal how to describe interactions, understanding and meaning from a technical perspective so that, perhaps, in a future step, some iterative technical approach might solve the SIAU problem.

From an engineering perspective we are talking of at least three different languages when dealing with interactions: there are at least two languages we use to talk about the interaction. These are our normal (engineering) language and perhaps some formal programming languages. And there is the interaction itself which I also view as a language. Actually, this is our language of interest. To distinguish it from the others, I name it "interaction language"[1].

In my opinion, a theory of interaction semantics or the semantics of the interaction language should fulfill certain requirements. It should at least

1. be consistent with the model of information transport and processing;
2. clarify the relationship to models of formal semantics.
3. account for the fact that the meaning of a character is invariant against appropriate renaming in our engineering language;
4. locate the concept of meaning in our usual technical description of interactions;
5. define when two different characters have the same meaning;
6. define what is "interpretation" and what is an "interpretation context"; and
7. explain under which conditions meaning is compositional, and when it is not.

These criteria also provide a good base to compare my approach with other approaches to define interaction semantics.

The structure of the article is as following: In section 2, I set the stage by recapitulating the concepts of information and of formal semantics. The key idea then is to proceed in four steps: In section 3, I describe the interaction (between discrete systems) by a mechanism that depends on information exchange, that is, on the identical naming of the "exchanged" characters — a protocol. Next, I look for a decisive property of such a protocol, namely its consistency, that depends on the particular choice of its sets of characters, also named its alphabets. Then, in section 4, I first introduce the decision concept to make the protocol executable in a functional sense, despite its nondeterministic character. Secondly, I define a fulfillment relation where the assignment of the set of alphabets to a protocol makes it consistent or inconsistent. This approach requires the definition of an interpretation function of the protocol and its constituents,

[1] [BDR16] abbreviated it as "I-language". However, This term is already used as "I"- versus "E"-language by Noam Chomsky in an attempt to distinguish between the "I"(intensional)-language, referring to the internal linguistic knowledge, and the "E"(extensional)-language, referring to the observable language people actually produce [Ara17].

namely its characters. This interpretation is the execution of the protocol and the interpretation of the characters during execution results in their meaning. Section 5 provides a brief overview of the relevant work of others. I conclude this article in the final section 6 with a summarizing discussion and some speculations.

2 Information and meaning

The first cornerstone of my theory of interaction semantics is its relation to information theory as it was invented by Claude E. Shannon [Sha48, Sha49] and others. His major invention was the introduction of two new concepts.

First, instead of describing the world by physical quantities like pressure, force, etc. he focused exclusively on the distinguishability of the state values. To continue to be able to talk about the same states as before, he had to introduce an additional alphabet with a character for each distinguishable physical state value into the engineering language: information emerged. A character in the sense of information theory is a unique name in our engineeering language for a physical state value that can be distinguished from all the other state values this state can take.

Secondly, equipped with this concept of information, he could now describe the essence of what happens between two "communicating" systems. With this I mean that a "receiver" system somehow reproduces a distinguishable difference in one of its state that had been a distinguishable difference in one of the states of a "sender" system beforehand and besides that, both systems remain structurally invariant. To speak about "information transport" it is therefore necessary to agree beforehand on using the same characters in our engineering language to denote these distinguishable state values in our description of both, the sender and the receiver system.

Thereby information theory separates information transport from information processing: information transport is about reproducing distinguishable state values *between* systems – enforcing the discussed naming conventions for the necessary characters in our engineering language description of multiple systems. And information processing is about relating arbitrary state values to other arbitrary state values on the base of these naming conventions, happening *within* systems.

Remarkably, Claude E. Shannon already related information processing to the field of semantics, and valued the meaning of the transported information as being irrelevant for solving the engineering problem how best to describe communication. He wrote in the introduction to [Sha48]: *"The fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point. Frequently the messages have meaning; that is they refer to or are correlated according to some system with certain physical or conceptual entities. These semantic aspects of communication are irrelevant to the engineering problem."*

So, if we want to talk about meaning in accordance with information theory,

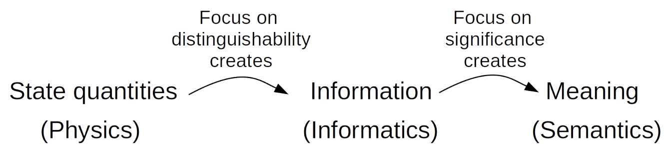

Figure 1: The relation of the concepts of physical state quantities, information and meaning.

it has to emerge from processing of the transported information. It also follows that the meaning of the exchanged information of the interaction language must be invariant against renaming them within our engineering language (assuming no naming conflicts).

An additional, perhaps a bit trivial insight is that not everything that is distinguishable - i.e. every piece of information - in an interaction is also of equal relevance or significance. Actually the ancient Greek word "σημαντικός" (semantikos) is usually translated as "significant" or "meaningful". I think that the concept of meaning in its core is about distinguishing the relevant from the irrelevant — which shows the importance, but also the indeterminacy of this concept. What determines the significance of an exchanged information? Intuitively speaking, the consequences resulting from its processing, and again, we arrive at the transformational behavior of the systems where we have to look for meaning and thereby at, what I would call, a local theory of interaction semantics. Fig. 1 illustrates the dependency relation between the concepts of physical states, information and meaning.

As per theory only information can be transported, it follows immediately that – in the frame of information theory – meaning cannot be transported in the same sense, but is somehow attributed by processing. This makes "meaning" in a way magical, being transmitted while not being transported. In my opinion, magical imagination is essentially an erroneous attribution of meaning: An amulet is attributed an effect that it does not have; spells are recited which are effect free; objects are thought to transfer properties on their owners that they do not have; etc. (see also [Grü10]). In his model of cognitive development of children the developmental psychologist Jean Piaget describes how actually every child in the so-called "pre-operational" phase from 3-7 years thinks magically. Clouds rain because they are sad, the monsters wait in the dark cellar, ... Only in the course of a healthy cognitive development do we humans develop realistic thinking — which allows us to fully understand a local model of attributing meaning by processing and its consequences.

In fact, from the point of Claude E. Shannon's information theory, all models that directly assign meaning to the characters themselves are archaic in a certain sense as they naively mix the language we use to describe what we are doing (to which the character symbol belongs) with the language we use to interact and ignore the relevance of the local processing of the transported information.

2.1 The concept of compositionality

The concept of compositionality will play a major role in my train of thoughts. Mathematically, composition[2] means making one out of two or more mathematical objects with the help of a mathematical mapping. For example, we can take two functions $f, g : \mathbb{N} \to \mathbb{N}$, which map the natural numbers onto themselves, and we can define a concatenated function $h = f \circ g$ by $h(n) = f(g(n))$.

If we apply this notion to arbitrary mathematical entities, we can define their composition into a composed entity by means of a corresponding composition operator as a partial function[3] on these entities:

Definition 1. *Be $\mathcal{S}_1 \in S_1, \ldots, \mathcal{S}_n \in S_n$ some arbitrary mathematical entities and $comp_{\mathcal{S}} : S_1 \times \cdots \times S_n \to S'$ is a partial function, such that*

$$\mathcal{S}_{tot} = comp_{\mathcal{S}}(\mathcal{S}_1, \ldots, \mathcal{S}_n). \tag{1}$$

Then I name this partial function a "composition". If $\mathcal{S}_1, \ldots, \mathcal{S}_n$ and $\mathcal{S}_{tot}$ are of the same sort, that is $\mathcal{S}_1 = \cdots = \mathcal{S}_n = \mathcal{S}'$, then I name this composition "homogeneous", otherwise I name it "inhomogeneous" or "emergent"[4].

Applying this definition to properties of the entities, where a property $\alpha : S \to A$ of an entity $\mathcal{S} \in S$ is a partial function which attributes values of some attribute value set A to an entity $\mathcal{S} \in S$, we can now classify the properties of the composed entities into homogeneous ones, that arise comparatively simply from the same properties of the subentities and emergent ones, that arise from other properties of the subsystems:

Definition 2. *A property $\alpha : S \to A$ of an entity $\mathcal{S} \in S$ is a partial function which attributes values of some attribute set A to an entity $\mathcal{S} \in S$. I call a property α of a composed entity $\mathcal{S}_{tot}$ "homogeneously compositional (or homogenous)" with respect to the composition $comp_{\mathcal{S}}$, if there exists an operator $comp_\alpha$ such that $\alpha(\mathcal{S}_{tot})$ results as $comp_\alpha(\alpha(\mathcal{S}_1), \ldots, \alpha(\mathcal{S}_n))$, thus, it holds:*

$$\alpha(comp_{\mathcal{S}}(\mathcal{S}_1, \ldots, \mathcal{S}_n)) = comp_\alpha(\alpha(\mathcal{S}_1), \ldots, \alpha(\mathcal{S}_n)) \tag{2}$$

Otherwise I call this property "emergently compositional (or emergent)".

From a mathematical point of view, in the case that α is homogenously compositional, α is a homomorphism mapping the mathematical structure $(comp_{\mathcal{S}}, S)$ onto $(comp_\alpha, A)$.

A simple example of a homogeneous composition is to create the physical system of an oscillating circuit from the physical systems of a capacitor and a coil. A homogeneous property of the oscillator would be its mass, as it is just the

[2]It was mainly Arend Rensink in his talk "Compositionality huh?" at the Dagstuhl Workshop "Divide and Conquer: the Quest for Compositional Design and Analysis" in December 2012, who inspired me to these thoughts and to distinguish between composition of systems and the property of being compositional for the properties of the systems.

[3]"Partial" means that this function doesn't need to be defined for all possible entities

[4]For a more thorough discussion of emergence, see [Rei01]

sum of the masses of its parts. A simple example of an emergent property is its resonance frequency, as neither the coil nor the capacitor has such a resonance frequency, but only an inductance and and a capacity.

2.2 The concept of meaning in formal languages

To create the concept of interaction semantics, the central idea of this work is to apply the procedure model of formal semantics to a suitable interaction model. I therefore now present this model of semantics of formal languages in more detail.

Based on Alfred Tarski [Tar35], formal languages are structured according to a certain scheme. First, the syntax is defined, consisting of a set of allowed characters together with a set of rules describing which expressions can be formed. Then, in a second step, the semantics is defined by an interpretation function determining the meaning of the allowed expressions by mapping them to entities that are legitimately assumed to exist and about we can talk in our normal language.

I illustrate this briefly with the example of the propositional calculus. For this we assume that we already know what a proposition is, that a proposition can be either true or false and that we can state at least some elementary propositions. Then, the calculus describes how one can obtain further statements from elementary statements by *and*, *or* and *negation* operations. In order to ensure the distinction between the expressions attributed to the calculus and those attributed to our normal language, I put all calculus expressions in quotation marks.

The colloquial expressions that we use to formulate the rules of syntax and semantics deserve special attention. To formulate propositional logic, we have to use so called "propositional forms". Syntactically, propositional forms correspond to calculus expressions, but they belong to our engineering language as we use them to define calculs expressins with special variables as placeholders for real calculus expressions. I write down propositional forms like calculus expressions in quotation marks, but symbolize the special variables with a prefixed $ sign, in order to be able to distinguish them reliably from the variables which are part of the calculus.

The allowed characters of the propositional calculus are determined by the alphabet $\{''w'', ''f''\}$, the set of operator characters $\{''\vee'', ''\wedge'', ''\neg''\}$, as well as the set of characters representing variables for propositions $V = \{''p'', ''q'', etc.\}$.

The syntax rules for building propositions are:

1. $''w''$ and $''f''$ are propositions;
2. Each variable is a proposition;
3. Are $''\$a''$ and $''\$b''$ propositions, then $''\neg\$a''$, $''\$a \vee \$b''$ and $''\$a \wedge \$b''$ are also propositions.

The interpretation of a proposition $''\$a''$, $\mathcal{I}_b(''\$a'')$, provides its meaning and consists of

1. an assignment of truth values to all variables: $b : V \to \{true, false\}$, where $true$ and $false$ are expressions of our colloquial engineering language we hopefully fully comprehend.

2. a recursive rule that determines the meaning of the proposition:

   (a) $\mathcal{I}_b(''w'') = true$; $\mathcal{I}_b(''f'') = false$;

   (b) $\mathcal{I}_b(''p'') = b(''p'')$;

   (c) $\mathcal{I}_b(''\neg\$a'') = true[false]$ if $\mathcal{I}_b(\$a) = false[true]$;

   (d) $\mathcal{I}_b(''\$a \vee \$b'') = true$, if $\mathcal{I}_b(\$a) = true\ or\ \mathcal{I}_b(\$b) = true$;

   (e) $\mathcal{I}_b(''\$a \wedge \$b'') = true$ if $\mathcal{I}_b(\$a) = true\ and\ \mathcal{I}_b(\$b) = true.$

Because the interpretation maps each syntactically correct formula to truth values, we can also define a "fulfillment"-relation $\models$ where an assignment b fulfills a formula $''\$a''$, or, symbolically, $(b, ''\$a'') \in \models$, or, in the usual infix notation, $b \models'' \$a''$ iff $\mathcal{I}_b(''\$a'') = true.$

One interesting aspect of the semantics of a formal language is its homogenously compositional character, that is that the meaning of composed terms results exclusively from the meaning of its parts.

3 The description of systems and their interactions

To describe the interaction of systems, I follow the approach of [Rei20]. A system separates an inner state from the state of the rest of the world, the environment. A state in this sense is a time dependent function, taking a single out of a set of possible values, the alphabet A, at a given time[5] [IECff]. I prefer to speak of "state function" and "state value".

The key idea is how this separation occurs. These time-varying values are not independent, but some of them are somehow related. If this relation is a function, then based on the uniqueness property of a function, this function – the so called "system function" – allows the identification of a system by identifying the state functions of a system and attributing them their input-, output-, or inner character. Such a functional relation logically implies causality and a time scale.

Depending on the class of system function or time, different classes of systems can be identified. For the purpose of our investigation, resting on information theory, I will focus on discrete systems. I describe the behavior of a (possible projection of a) discrete system by input/output transition systems (I/O-TSs) of the following form[6]:

[5] This state concept actually captures the perspective of classical physics. In quantum physics, a state is a more complex concept.

[6] Another name in the literature is "transducer" [Sak09], because this machine translates a stream of incoming characters into a stream of outgoing characters.

Definition 3. *An input/output transition system (I/O-TS) $\mathcal{A}$ is given by the tuple $\mathcal{A} = (I, O, Q, (q_0, o_0), \Delta)$ with I and O are the possibly empty input and output alphabets and Q is the non empty set of internal state values, (q_0, o_0) are the initial values of the internal state and output and $\Delta_{\mathcal{A}} \subseteq I^\epsilon \times O^\epsilon \times Q \times Q$ is the transition relation describing the behavior of a discrete system.*

The timing behaviour of the system is implicitely contained in the transition relation by attributing the input character and the start state to time k and the output character and the target state to time $k + 1$. Instead of writing $(i_k, o_{k+1}, p_k, p_{k+1}) \in \Delta$, I also write $p_k \stackrel{i_k/o_{k+1}}{\rightarrow} p_{k+1}$.

The system's (projected) state is determined at any time k by the triple (i_k, o_k, p_k). In the arrow notation, a single state-determining 3-tuple at time k has to be written as $\stackrel{/o_k}{\rightarrow} p_k \stackrel{i_k/}{\rightarrow}$.

An *execution fragment* of an I/O-TS is a sequence of 3-tuples[7], listing the values that the input, output and state functions of the corresponding system have at the considered times: $(i_0, o_0, p_0), (i_1, o_1, p_1), \dots, (i_n, o_n, p_n)$. Or, in the arrow notation, it becomes $\stackrel{/o_0}{\rightarrow} p_0 \stackrel{i_0/o_1}{\rightarrow} p_1 \stackrel{i_1/o_2}{\rightarrow} \dots \stackrel{i_{n-1}/o_n}{\rightarrow} p_n \stackrel{i_n/}{\rightarrow}$.

I call an execution fragment which starts with an initial state a *"run"*. With a *"path"* I denote only the state values of a run and not the I/O-characters.

3.1 System interactions

In our model, interaction simply means that information is transmitted. Accordingly, the description of interaction is based on the use of equal characters in the sending and receiving systems such that the state values of an output component of a transition of a "sender" system are reproduced in the input component of the "receiver" system and serve there as input of a further transition (see Fig. 2).

I call such a state function that serves as output as well as input of two systems a "Shannon state function" or "Shannon state". It is an idealized Shannon channel as it has per definition no noise and no delays.

I name an execution fragment that is started by a spontaneous transition at time t and goes on until the output becomes empty after m-steps an "*interaction chain*".

3.2 Protocols

In the following we focus on systems which interact with multiple other systems in a stateful and nondeterministic way. In the literature there have been many names coined for these kind of systems like "processes" (e.g. [MPW92]), "reactive systems" (e.g. [HP85]), "agents" (e.g. [Pos07]) or "interactive systems" (e.g.

[7]This is a bit different to ordinary automata theory of the "standard" three-tuple automata with a single transition label, where an execution fragment is usually defined as a sequence of consecutive transitions. This difference is due to the fact, that in the case of I/O-TS, we can attribute the labels of the transitions – the I/O-characters – to the times of the state values

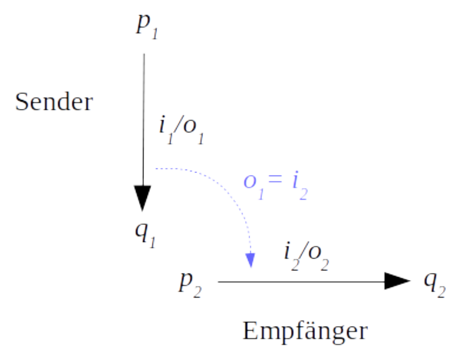

Figure 2: Interaction between two systems in which the output character of a "sender" system is used as the input character of a "receiver" system. Interaction therefore means the coupling of the two transition systems of sender and receiver based on the "exchanged" character.

[Bro10]). I think the term "interactive systems" is most appropriate. Their interactions are described by protocols [Hol91].

While in deterministic interactions a natural "purpose" in the sense of a consequence of composition is simply the creation of the resulting super-system, in nondeterministic interactions, things are different. We need an additional criterion for success, the so-called acceptance condition represented by an additional component Acc. We thus get from I/O-TSs to I/O automata (I/O-Aa) by adding a "success model", represented by an additional acceptance component Acc. For finite calculations with some desired final state value, Acc_{finite} consists of the set of all final state values. For infinite calculations of a finite automaton there are differently structured success models. One of them is the so-called Muller acceptance, where the acceptance component is a set of subsets of the state value set Q, i.e. $Acc_{Muller} \subseteq \wp(Q)$. An execution (see below) is considered to be successful whose finite set of infinitely often traversed state values is an element of the acceptance component (e.g. [Far01]).

If our I/O-Aa now represent interacting systems which are coupled by information transport, based on Shannon states, we have to add the information which output and input components of our alphabets actually represent the same Shannon state function. Thus we have to complement the simple product automaton of all I/O-Aa with this information and specify the execution rule accordingly. Because of this extra component, the resulting automaton is no longer an I/O-A, although it is quite similar. Its execution rule pays special attention to the character exchange unknown to an I/O-A. This coupling actually restricts the transition relation compared to the uncoupled product I/O-A.

If this coupling is complete or closed in the sense, that all output states are connected to some input state and no further input state is left, then I call it a "protocol". Thereby a protocol is a self-contained or closed interaction in the sense that it has neither any additional external inputs nor does it output any character towards somewhere else. Its "inner" structure represents *external* interactions.

To simplify our further considerations, I assume any character of any I/O-A to have at most one component unequal the empty character ϵ.

Definition 4. *A protocol $\mathcal{P} = (A, C)$ is a pair of a set $A = \{\mathcal{A}_1, \ldots, \mathcal{A}_n\}$ of I/O-As, and a set of index pairs C indicating which output component relates to which input component of A in the sense of a Shannon state. I name the $\mathcal{A}_i$ the "roles" of the protocol.*

The execution rules are as following with the convention that the current values of i, o and q are indicated by a $*$ and the values calculated in the current step by a $+$:

1. **Initialization (time $j = 0$):** $(q^*, o^*) = (q_0, o_0)_{\mathcal{P}}$.
2. **Loop:** *Determine for the current state q^* the set T^* of all possible transitions. If T^* is empty, end the calculation.*
3. **Determine input character i^*:** *Proceed in the following sequence:*
   (a) *If the current output character $o^* \in O_{\mathcal{P}}$ has the value $v \neq \epsilon$ in its k-th component, i.e. $o^* = \epsilon[v, k]$, and o^* is part of a feedback signal $c = (k, l)$ to the input component $1 \leq l \leq n_I$, then set $i^* = \epsilon[v, l]$.*
   (b) *Otherwise, select one of the possible input characters as i^* for q^* (if some element of T^* is a spontaneous transition, the selection $i^* = \epsilon$ is also allowed).*
   (c) *otherwise end the calculation.*
4. **Transition:** *With q^* as current state value and i^* as current input character select a transition $t = (i^*, o^+, q^*, q^+) \in T^*$ and so determine o^+ and q^+. If there is no possible transition at this point, terminate the calculation with an error.*
5. **Repetition:** *Set $q^* = q^+$ and $o^* = o^+$ and jump back to 2*

A protocol can therefore also be viewed as a product I/O-A where the I/O-coupling imposes a restriction on the transition relation.

As can be seen from the error condition in the execution rule, a protocol must fulfill certain consistency conditions to make sense. It has to be "well-formed" in the sense that for each transition with a sent character o unequal to ϵ in at least one component, a corresponding receiving transition must exist. Is must not contain infinite chains of interaction, i.e. it must be "interruptible". And for each run, the acceptance condition has to be fulfilled. So, I define:

Definition 5. *A protocol is named ...*

1. *... "well formed" if each input character determined in step 2 can be processed in step 3.*

2. ... *"interruptible" if each interaction chain remains finite.*

3. ... *"accepting" if for each run the acceptance condition is fulfilled.*

A protocol that is well formed, interruptible, and accepting is named "consistent".

Keeping a protocol's consistency is also a condition that a renaming of I/O-characters in our engineering language has to fullfil. The consistency of a protocol essentially depends on the equally named input and output characters. Thus any renaming has to leave the consistency of the protocol invariant. This fits nicely to the "consistency management" Johann van Bentham refers to [vB+08].

3.2.1 Example: The single-track railway bridge

To illustrate the protocol concept I give the simple example of a single-track railway bridge drawn from [Alu15]. As is shown in Fig. 3, two trains, $\mathcal{Z}_1$ and $\mathcal{Z}_2$, must share the common resource of a single-track railway bridge. For this purpose, both trains interact with a common controller $\mathcal{C}$, which must ensure that there is no more than one train on the bridge at any one time.

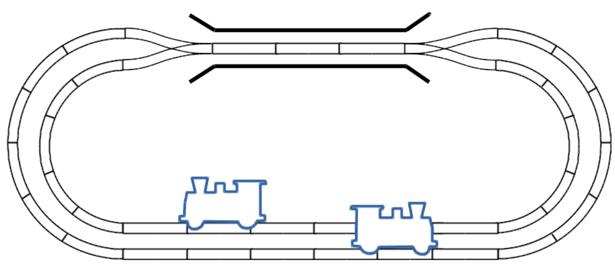

Figure 3: A single-track railway bridge crossed by two trains. To avoid a collision on the bridge, both trains interact with a central controller.

The interaction between each train and the controller is described by a complete and consistent protocol. For this we need to describe both the train and the controller in terms of the role they play in the interaction. For both the train and the controller we choose a model of 3 state values, which we call $Q_{Z_{1,2}/C} = \{away, wait, bridge\}$ for each train as well as for the controller. The input alphabet of the trains $I_{\mathcal{Z}_{1,2}} = \{go\}$ is the output alphabet of the controller $O_{\mathcal{C}}$ and the output alphabet of the trains $O_{\mathcal{Z}_{1,2}} = \{arrived, left\}$ is the input alphabet of the controller $I_{\mathcal{C}}$.

In Fig. 4 the protocol is shown. It can be seen how the interaction by a Shannon state, or, as I can say, by "information transport", restricts the transition relation of the product automaton. The protocol between train and controller is indeed complete as no further external characters occur. It is well formed as for each sent character there is a processing transition at the right time. And finally, it is consistent as it is well formed, interruptible, and accepting.

Intuitively, the train's state values represent the state of the train as it is known to the train itself and the controller state values represent the train state

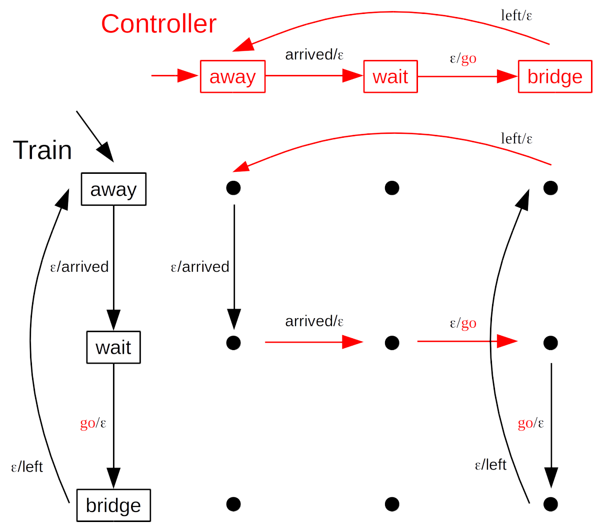

Figure 4: Presentation of the protocol between train and controller for the problem of the single-track railway bridge. Initially, both controller and train are in the *away* state. When a train arrives, it signals *arrived* to the controller. This sign must now be processed by the controller, the controller in turn changes to its *wait* state. The controller releases the track with *go* and the train signals the controller with *away* that it has left the bridge again. The interaction is successful when both the train and the controller go through their three states infinitely often.

as it is known to the controller. Please note that the correctness, we could also say the truth, of the representation of the state of the train in the controller depends on both, the correctness of the knowledge of the train of its own state as well as the correctness of the protocol.

4 Interaction semantics

Now we come to the semantics of an interaction. The idea is to use the mathematical concept of meaning with respect to formal languages also for interactions. As I have introduced in section 2.2, the mathematical concept of attributing meaning to a term of a formal language consists of two steps: First an assignment of values to the variables of the term and then the application of an interpretation function to all of its elements. To apply this concept of meaning to the interaction between systems, we therefore have to identify the possible variables and the possible interpretation function of the interaction.

Looking at our system + protocol concepts, we see that a state function can be viewed as assigning characters to a state variable at each point in time. Interestingly, with this understanding the representation of the system behavior by some I/O-TS becomes the representation of the variable-value-assigment. And the interpretation function? In the world of systems, we can view the system function as "interpreting" the assignment by mapping one assignment onto the "next" one. Thus, the "meaning" of the assigned values are again assigned values. This is a major difference to the application of the meaning concept in formal languages. There, the meaning of a formal term always had to be provided by some engineering-language denoted entity, like a truth value.

However, in the world of interactions, we still have an issue: the transition relation of a protocol is usually nondeterministic and therefore does not define a function that maps assignments to assignments. There would be the possibility to look at sets of state values instead of single state values, as every nondeterministic transition relation does define a function, mapping the input character and internal state value onto a set of pairs of possible output characters and new state values. But this would not be in sync with the conception we have developed within the system notion.

4.1 Decisions

The idea to stay within our interaction conception is to complete the nondeterministic transition relation of a protocol such that it becomes deterministic. I call the necessary additional input characters "decisions". In this sense we can say that decisions determine the behavior, that is, they determine those transitions which would otherwise be indeterminate (see [Rei20]).

As we will see, this approach reveals the very close relationship between protocols and games – which makes me confident to be on the right track.

According to the two mechanisms that give rise to nondeterminism of transitions, we can distinguish two classes of decisions: spontaneous decisions that determine the spontaneous transitions without input character and selection decisions that determine a selection if for a given input character and state value several transition could be selected.

Decisions in this sense are very similar to information and can be seen as a further, "inner" input alphabet D, which complements the input alphabet I of an I/O-TS $\mathcal{A}$ according to Def. 3 to $I' = I \times D$ such that a complemented transition relation Δ' becomes deterministic. They are enumerated by an alphabet and their names are relevant only for their distinction. In contrast to ordinary input characters, whose main characteristic is to appear in other output alphabets and that are allowed to appear in different transitions, we name all decisions of a corresponding transition system differently and different from all input and output characters and internal state values, so that we can be sure that they really do determine all transitions.

Definition 6. *Be $\mathcal{A}$ an I/O-TS and D an alphabet. The transition system $\mathcal{A}'$ is called a "decision system" to $\mathcal{A}$ and the elements of D "decisions", if $I \cap D = \emptyset$, $O \cap D = \emptyset$, $Q \cap D = \emptyset$, and $\Delta' \subseteq (I^{\epsilon}_{\mathcal{A}} \times D) \times O^{\epsilon}_{\mathcal{A}} \times Q_{\mathcal{A}} \times Q_{\mathcal{A}}$ with $((i,d),o,p,q) \in \Delta'$ if $(i,o,p,q) \in \Delta$ and for d applies:*

$$d = \begin{cases} \epsilon, & \textit{if there's no further transition } (i^*, o^*, p^*, q^*) \in \Delta \\ & \textit{with } (i,p) = (i^*, p^*). \\ \textit{so selected} & \textit{that } \Delta' \textit{ is deterministic, i.e. } \Delta' \textit{ determines the function} \\ & f' : I^\epsilon \times D^\epsilon \times Q \to O^\epsilon \times Q \textit{ with } (o,q) = f'(i,d,p) \textit{ such that} \\ & \textit{for two transitions } t_1', t_2' \in \Delta' \textit{ holds } t_1' \neq t_2' \Rightarrow (d_1 \neq d_2 \\ & \textit{or } d_1 = d_2 = \epsilon). \\ & \textit{Additionally, } \Delta' \textit{ is the smallest possible set.} \end{cases}$$

For a decision system we can modify the protocol execution rule 4 *Transition* such that the selection choice becomes determined by some possible decision.

Obviously, the set of decisions for an already deterministic I/O-A is empty. In Fig. 5 I illustrate the decision notion with the train-controller protocol. To determine the actions of train and controller three decisions are necessary. The train has to decide when to arrive and when to leave ("IArrive" and "ILeave") and the controller has to decide when it let the train go ("ILetYouGo").

I call the decision automaton to a consistent protocol also a "game in interactive form (GIF)"[8].

The transition relation $\Delta_{\mathcal{P}}$ of a GIF $\mathcal{P}$ defines per construction a transition function $\delta : I_{\mathcal{P}} \times Q_{\mathcal{P}} \to O_{\mathcal{P}} \times Q_{\mathcal{P}}$.

An execution fragment of a GIF is like the execution fragment of the protocol, but extended by the additional decisions: $\stackrel{/o_0}{\longrightarrow} p_0 \stackrel{(i_0,d_0)/o_1}{\longrightarrow} p_1 \stackrel{(i_1,d_1)/o_2}{\longrightarrow} \dots \stackrel{(i_{n-1},d_{n-1})/o_n}{\longrightarrow} p_n \stackrel{(i_n,fd_n)/}{\longrightarrow}$. As an interaction chain is started by an empty character and ends by an empty output, we now can say that an interaction chain is always triggered by a spontaneous decision.

A run r is then the result of an interpretation of the initial assignment, that is the initial state q_o and some input sequence of decisions seq: $r = \delta^*_{\mathcal{A}}(q_0, seq)$.

4.2 Meaning within an interaction

Now, with the transition function of the GIF, we have an interpretation function which defines the meaning of the input character i with respect to some start value p and possibly some decision d — which is the new state value q with the possibly generated output character o:

$$(o,q) = f((i,d),p) =: interp_{d,p}(i) \tag{3}$$

Viewing the execution of a GIF as its interpretation, we can now say that the result of the complete interpretation of a GIF under the assignment of its initial state is all its possible runs. Or we can extend the assignment to the decisions

[8] Actually this "game" still lacks the utility function, a game in the traditional game theoretic sense has. But such a utility function is just one way to introduce a certain concept to determine decisions, which is, in this case, based on non-hierarchical, consistent preferences. There are many others.

and say that the result of the interpretation of a GIF under the assignment of its initial state and a sequence of decisions is a specific run.

With this concept of interpretation we can define a fulfilment relation similar to that in section 2.2 where we relate what we assign, some initial state q_0 together with a sequence of decisions seq, to the behaviour of the structure with the variables, the GIF.

Definition 7. *Let $\mathcal{P}$ be a GIF. We say that some initial state q_0 together with a sequence of decisions seq fulfils $\mathcal{P}$, in symbolic notation $(q_0, seq) \models \mathcal{P}$, iff $interp_{q_0,seq}(\mathcal{P})$ is an accepted run of the GIF.*

4.3 The meaning of a decision

The interpretation function of a GIF defines the resulting state and output value as the meaning of a decision and an input character. However, if we are only interested in what happens by a decision until the next decision is taken, we can view all states that are reached by the ensuing interaction ping pong as being equivalent and lump them together by constructing the respective equivalence classes.

I have thereby introduced the equivalence class concept as *the* semantic concept for abstraction by subsuming ”equal meaning”. Thinking about ”equal meaning” we always have to specify ”equal with respect to what?” And this ”what” becomes the base for an equivalence class partitioning. In the case of the meaning of decisions this leads to the notion of ϵ-closure in the decision space and to construct a somehow ”reduced” decision automaton in a procedure similar to ϵ-elimination for determining a deterministic from a nondeterministic finite automaton.

I first define the ϵ-decision closure of a state value q as the set of all states that are accessible from q without further decisions including q itself:

Definition 8. *Let $\mathcal{P}$ be a GIF. Then $h_\epsilon(q) = \{p \in Q |\, p = q$ or: if there is a $p' \in h_\epsilon(q)$ and there exists $i \in I$ and $o \in O^\epsilon$ s.t. $(i, \epsilon, o, p', p) \in \Delta_{\mathcal{P}}\}$ is the ϵ-decision closure of the state value q.*

Please remember that if the decision is ϵ, such an input character always exists. We can now attribute every decision an ϵ-decision closure as a ”target state”, which I name the *”abstract meaning* of the decision in contrast to its concrete meaning as it is directly provided by the GIF's transition function. With these sets of state values we can construct a ”reduced” decision automaton as following:

Definition 9. *Let $\mathcal{P}$ be a GIF. Then I call the automaton $\mathcal{B}$, constructed by the following rules, the ”reduced decision automaton” or ”game in its decision form (GDF)” of $\mathcal{P}$:*

- *$Q_{\mathcal{B}}$ is the set of ϵ-decision-closures of $\mathcal{P}$ that partitions $Q_{\mathcal{P}}$.*
- *The input alphabet $I_{\mathcal{B}} = D_{\mathcal{P}}$ is the set of all decisions of $\mathcal{P}$.*

- *The transition relation $\Delta_{\mathcal{B}} \subseteq I_{\mathcal{B}} \times Q_{\mathcal{B}} \times Q_{\mathcal{B}}$ is defined by: $(d, p, q) \in \Delta_{\mathcal{B}}$ if and only if there exists a reachable $p' \in p$ and some $i \in I_{\mathcal{P}}$, $o \in O_{\mathcal{P}}$, and $q' \in q$ such that $(i, d, o, p', q') \in \Delta_{\mathcal{P}}$.*

- *The acceptance component $Acc_{\mathcal{B}}$ is defined by: $p \in Q_{\mathcal{B}}$ is an element of $Acc_{\mathcal{B}}$ if and only if at least one $p' \in p$ is an element of $Acc_{\mathcal{P}}$.*

- *The initial state $q_{0\mathcal{B}}$ is defined as $q_{0\mathcal{B}} = h_{\epsilon}(q_{0\mathcal{P}})$.*

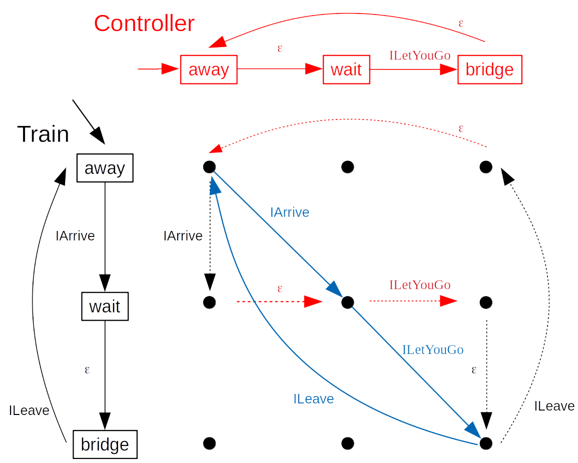

Figure 5: The transitions of the GDF of the train-controller GIF are shown in blue. The train decides that it arrives and leaves ("IArrive" and "ILeave") and the controller decides when it let the train go ("ILetYouGo")

The resulting reduced decision automaton or GDF is deterministic with D as its input alphabet. Please note, that it relates to the product state space of the GIF (or protocol). As the states of a GDF are the abstract meanings of the decisions of a GIF and a GDF is more or less a game in the sense of traditional game theory, I think that it is justified to say that, in our sense, game theory is the theory of the meaning of decisions.

It's interesting that the input- and output characters do no longer appear in the GDF. They only contribute implicitly as being part of the coupling mechanism between the interacting roles and thereby influence which state becomes part of the meaning of a decision.

I illustrate the transformation of a GIF to a GDF in Fig. 5 for the train-controller GIF.

We can go a step further and ask, which decisions have the same meaning? Which we translate in: which decisions make a GDF to transit into one concrete state value? The resulting equivalence classes of equivalent decisions correspond, in my opinion, to our intuitive understanding of decisions and their meaning. Intuitively, we judge decisions by their consequences rather independently of the initial situation. Whether a child hears a violin and decides "to play the violin", or whether it sees a picture and – we are already talking like this – also decides "to play the violin". Now, we can formulate this more precisely that both decisions are different but have the same meaning, as both result in the same state of the child to play the violin and are therefore, in this sense, equivalent.

4.4 The meaning of an exchanged character

Looking at our GIF, the significance of the exchanged characters is their role as a coupling mechanism which enables each participant to take the possible decisions according to the rules of the interaction-game.

As we have seen, the meaning of a concrete incoming character is determined by a selection decision (which could be ϵ) which occurs simultaneously with the incoming character. We can therefore say that for a selection decisions, an incoming character unfolds its relevance by enforcing this selection decision. So, while the concrete meaning of an incoming character is given be the interpretation function of a GIF according to equation 3, we can abstract by looking for incoming characters enforcing selection decisions with the same abstract meaning. I therefore define

Definition 10. *Let $\mathcal{P}$ be a GIF and let $i \in I$ occur in all transitions $(i, d_j, *, p, q_j) \in \Delta$ with $d_j \neq \epsilon$ for $j = 1, \ldots, n$. Then I name the selection decisions to be "enforced by i from p".*

Two characters i_1 and i_2 have the same abstract meaning if the abstract meaning of the associated selection decisions $(d_j)_1$ and $(d_j)_2$ are equal.

With the relation to the abstract meaning of the selection decisions we do have taken into account some ensuing character ping pong. But what if the incoming character did not enforce any selection decision and its triggered transition resulted in a state value where some spontaneous decisions could be taken? Wouldn't we say that this is also part of the relevance of the exchanged characters, that they put the participants in the position to also make their spontaneous decisions? Thus it appears to me that by defining the abstract meaning of a character we somehow aribitrarily have to restrict ourselves to some order of reachability, where definition 10 grasps the significance of a character only to "first order".

4.5 Composition of meaning in the interaction language

The question is, whether the meaning of characters and decisions as we have defined it in the previous sections is compositional in the sense of Def. 1.

Definition 11. *The meaning of two consecutive characters or decisions c_1, c_2 is compositional if an operator op exists such that for the interpretation function for two consecutive characters holds $interp(c_1.c_2) = op(interp(c_1), interp(c_2))$.*

In the case of characters we focus on the operation of concatenation and want to know whether the interaction semantics of consecutive occurring characters (or strings) can be deduced solely from the semantics of the characters itself.

The first thing we have to clarify is what we mean be "consecutively occurring". From the perspective of a single system, two incoming characters occur consecutively if one succeeds the other as input of the system. However, from an interaction perspective, two characters occur consecutively, if they are consecutively exchanged between two systems.

The next thing we have to clarify is what our operator *op* is supposed "to know". It knows actually only $(o_1, q_1) = interp(c_1)$ and $(o_2, q_2) = interp(c_2)$. If both interpretations represent consecutive transitions, then its clear that the output of our operator is just (o_2, q_2). But if they do not represent consecutive transitions, then our operator cannot calculate the composed meaning.

Additinally, it is clear that the compositionality of the meaning of characters and decisions depends on the well-definedness of the transition relation. In settings where the transition relation, i.e. the interaction context itself, becomes the object of consideration, compositionality of meaning is also lost.

5 Other work

As I have already indicated in the beginning, this article touches scientific, engineering as well as philosophical aspects.

5.1 Carolyn Talcott

Carolyn Talcott [Tal97] uses the term "interaction semantic" of a component to denote the set of sequences of interactions in the sense of input or output messages or silent steps in which it might participate. She composes her components of multisets of so called actors with unique addresses where the actor semantics could be either internal transitions as a combination of execution and message delivery steps or interaction steps with an exchange of messages. In her formalism she takes into account that the interaction semantics must be invariant against renaming of addresses, state and message values but she neither addresses any semantic fulfillment relation nor the concept of the "meaning" of a single exchanged character. In summary, her approach is very similar to the $\pi - calculus$ [MPW92] but her addresses refer to actors with state and not to stateless channels and the interactions are asynchronous.

5.2 Tizian Schröder and Christian Diedrich

Tizian Schröder and Christian Diedrich [SD20] published an approach that is quite similar to mine. They view the semantics of the exchanged characters within an interaction as being provided by its processing. However, they do not link their approach to the way, mathematics has dveloped to describe the semantics of formal languages. Like [Rei10] they use a discrete system with a system function f, mapping the two sets of input and internal state values onto a set of output and internal state values as processing model. Instead of using transition systems or automata to describe the behavior of these systems, they use the functional representation for both the system under consideration as well as for the environment. Both, the system as well as the environment receive additional (not considered or "rest") inputs resulting in nondeterministic, stateful and asynchronous behavior towards each other (although it remains unclear where this additional input is supposed to come from, possibly from some "unconsidered environment"). They define as the semantics Sem of a considered input character u_{cons} the set of all possible pairs of output characters and new internal state values provided by the system function, operating on u_{cons}, the current internal state value and any possible u_{rest}.

To select a unique result out of this set, they define the set of "decisions" Dec of some u_{rest} as the set of all possible pairs of output characters and new internal state values provided by the system function, operating on u_{rest}, the current internal state value and any possible u_{cons}. They view the internal state value x as the context of this decisions. Now, they claim that the (x, y) realized by the system in its internal state x in response to the input u_{cons} is the intersection $Sem_x(u_{cons}) \cap Dec_x(u_{rest})$. But this claim seems to depend on whether the system function is a bijection or not, as the the simple example in Tab. 1 shows:

u_{cons}	0	0	0	0	1	1	1	1
u_{rest}	0	0	1	1	0	0	1	1
x	0	1	0	1	0	1	0	1
x'	1	*	*	*	1	*	1	*
y'	0	*	*	*	1	*	0	*

Table 1: An example system function for a system as described in [SD20]. The values of the mapping which are irrelevant to the example have been marked with a '*'.

Just consider $u_{cons} = 1$ and $x = 0$ where we have $Sem_0(1) = \{(1, 1), (1, 0)\}$ and for $u_{rest} = 0$ and the same $x = 0$, we have $Dec_0(0) = \{(1, 1), (1, 0)\}$, resulting in $Sem_0(1) \cap Dec_0(0) = \{(1, 1), (1, 0)\}$ which has more than one element. So, in summary, their key proposal to use the system function to define the semantics of the exchanged character is very much aligned with the ideas of this article. However, according to my understanding, their decision concept is not consistent.

5.3 David Lewis signaling games

I already mentioned David Lewis [Lew69] and his signaling games in the introduction. A signaling game consists of a sender in somes state s, sending a signal σ to a receiver, which in turn responds by some action a. The mapping from s to σ and from σ to a is achieved by two functions F_c and F_a. The benefit for both depends on their choice of signal and action. If their preferences are aligned [CS82] then we have a coordination problem whose solution contains at least two proper coordination equilibria. Which one will be realized is, according to David Levis, due to convention. Reinterpreting the benefit as an evolutionary selection advantage, Karl Warneryd [Wä93] showed that such "conventions" arises naturally in evolutionary settings, based on the concept of evlutionary stable strategies as developed by John Maynard Smith and George R. Price [SP73].

David Lewis identified the signal's relation to either the state of the sender or to the action of the receiver as its "meaning". He named a signal with a predominant relation to the state of the sender a "signal-that" or an indicative signal and a signal with a predominant relation to the action of the receiver a "signal-to" or an imperative signal and a signal with both parts a "neutral" signal.

In this line of thought, Brian Skyrms [Sky10] identifies the information content of a signal in how it affects the probabilities of states and actions (p.34). He defines the quantity of information in a signal as the logarithm of the factor that changes the a priori probabilities of either sender state or receiver action. With respect to the sender's state it becomes the (logarithm of) the probability of the state s given that a signal σ was sent in relation to the unconditional probability of this state $log(pr(s|\sigma)/pr(s))$. And with respect to the receiver's action $log(pr(a|\sigma)/pr(a))$. Thereby Brian Skyrms arrives at different information contents of a single message, depending on its assumed relation.

What I think this approach to meaning of a character grasp correctly is the aspect of context dependency. So, I think that this "Information content" does reflect correctly *some* aspects of meaning. However, there are several issues with this approach. First it is quite ad hoc. What is the difference between a signal and an action? Probability is just one out of many concepts to deal with nondeterminism. What if we do cannot or just do not know any probabilities? I think the most important unsuitable element is the attempt to make meaning measurable, as "information content". I think that one of the the central ideas of the information concept is that any consideration in terms of "information" does lead to measurability, but not to meaning.

For example, assume that two different characters would have the identical Lewis/Skyrms-"signal-to"-Information content of 1 Bit with respect to some of my actions. The first would entail me to hit a button to get a cup of coffee, the other would entail me to hit a button to ignite a bomb. Obviously the meaning of both characters would be *very* different.

5.4 Iterative acquiring knowledge about interaction semantics

Then there exists extensive research where the iterative character of acquiring knowledge about interaction semantics is already investigated. This could be on an evolutionary time scale (e.g. [BEJvR11] for a brief overview) or on an online-timescale. An example for the latter is Sida I. Wang, Percy Liang and Christopher D. Manning [WLM16] who explore the idea of language games in a learning setting, which they call interactive learning through language game (ILLG). A human wishes to accomplish a certain configuration of blocks, but can only communicate with a computer, who performs the actual actions. The computer initially knows nothing about language and therefore must learn it from scratch through interaction, while the human adapts to the computer's capabilities. The objective is to transform a start state into a goal state, but the only action the human can take is entering an utterance.

5.5 Dialogical logic

Mathematicians have also addressed the relation between meaning, knowledge, and logic in the context of interactions in the sense of games or dialogues under the notions of "dialogical logic" [LL78] or "game-theoretical semantics" [HS97]. The former focuses more on real human discourse while the latter focus is more on model-oriented analysis of the logical meaning of linguistic sentences and its relation to certain rule-governed human activities. The basic idea of Hintikka's evaluation game is that as a proof, a Verifier tries to find a winning strategy in a two person game against a Falsifier such that a given first order formula ϕ is true in a given Model $\mathcal{M}$ under some assignment of the variables. Negation, conjunction and disjunction are translated into role switches and choice attributions.

5.6 Computational semantics

Also related is the field of computational semantics as it is concerned with computing approximations of the meanings of linguistic objects such as sentences, text fragments, and dialogue contributions (e.g. [BM99, Bol20]).

6 Discussion

The aim of this article was to delineate a theory of interaction semantics and en passant provide a concrete understanding of the meaning of characters within an interaction. Its key idea was to apply the mathematical concept of meaning, which was developed to provide meaning to formal expressions, to interactions. I therefore argue that with the same intuition as we talk about the semantics of a formal language we can talk about the meaning of an interaction. The state functions of the participants of an interaction became the assignments and to acquire the necessary interpretation function, we had to augment the nondeterministic transition relation of the interaction with an additional input alphabet, the decisions.

Just assume for a moment, that this approach is complete nonsense and thus would not contribute even the smallest grain to our comprehension of the meaning of meaning — Would it be irrelevant? For sure, we can assume that reading it would then leave the capacity of any esteemed reader to say something meaningful completely invariant. So, despite of being nonsense, it would enfold a certain significance by then serving as a good example for the delightful fact, that we do not have to comprehend the meaning of meaning correctly to say something else meaningful.

Have I met all the requirements I listed in my introduction any "good" theory of interaction semantics should fulfill?

First, it is not only consistent with the model of information transport and processing, but essentially depends on it. The idea to name the value of physical states by names whose only characteristic is to make them distinguishable is one of the key elements of my construct. It in fact paved the way to use the approach of formal semantics to define the semantics of an interaction in terms of a GIF.

Thereby the concept of meaning of an exchanged character could be quite naturally identified in our technical description of interactions and also the relation of equal meaning and compositional meaning of two such characters. The transition relation of the GIF became the "interpretation context".

The semantics of an interaction remains invariant against any renaming of the I/O-characters as long as their coupling function and thereby its consisteny is conserved. In fact, the characters are part of out description language we have to use to express our interaction model.

What could be possible consequences of this theory of interaction semantics? First it suggests a very tight conceptual connection between system theory with its delineations of borders between different parts of the world, information theory with its focus on distinguishability and its distinction between information transport and processing, my theory of interaction with its emphasis on state and nondeterminism, and the concepts of decisions leading to a theory of interaction semantics and further to game theory. We can say that by introducing decisions to our interactions we intentionally isolate the roles of all interacting systems of interest from the rest of the world, necessary to attribute meaning.

This approach suggests a special role of our decision making capabilities for us, attributing meaning within our interaction. In a well defined interaction, modelled by a consistent GIF, the receiver has the opportunity to decide in a predefined frame about the meaning of a received character. A GIF is consistent if all allowed decision sequences are accepted or fullfil the GIF. However, in real life with free decisions, we additionally have to use our decisions to stick to the rules of the game – or deviate from them. Hence, in natural interactions, we constantly have to asure ourselves whether our assumed context of our interpretation is still valid.

Additionally, we gain a deeper understanding of game theory. The traditional view is expressed by Roger B. Myerson [Mye91] who starts his book by defining game theory as "the study of mathematical models of conflict and cooperation between intelligent rational decision-makers". With our approach we now can say that game theory becomes the theory of the meaning of decisions. However, the notion of decision is a complex one, as I introduced it only as a fiction to fill the void left by the nondeterministic interaction. As this void could also be filled by other interactions a subject has to coordinate, decisions in this sense can be viewed as a concept to enable the isolated consideration of consistent interactions of a subject that is in fact only partially separable.

Only looking at a subject as a whole leverages the full potential of the decision notion, as only this holistic view leads to the important question, what might be a genuinely "free" or internally determined decision in contrast to an effectively externally determined one. Thus, the focus of game theory could shift from exploring strategies in individual interactions by optimizing ad hocly assumed utilities more to the problem of coordination of multiple interactions, for example how to preserve the freedom of decision while coordinating multiple interactions.

In the area of computer science, the enormous growth of the internet in the past was mainly due to semantically agnostic transport protocols for documents like HTTP, FTP, SMTP, etc., leaving the essential problems of semantic interoperability within the sphere of the human mind. Nowadays, however, technical information-processing systems are more and more integrated into interactions on a content-level with a certain degree of autonomy, greatly increasing the interest in clear and systematic concepts of semantic interoperability. Based on the ideas of this theory of interaction semantic, Tizian Schröder and myself [RS20] proposed a simple classification of interactions according to their information transport and processing characteristics, which allows for a sound layering of computer applications. In contrast to other interface theories as proposed for example by Luca de Alfaro and Thomas A. Henzinger, who wrote in [DAH01] that "*Nondeterminism in interfaces, however, seems unnecessary and is expensive ...*[9], the resulting interface notion emphasizes the importance of non-deterministic, game-like interfaces in the form of roles to achieve semantic interoperability in non-hierarchical interaction networks.

However, I think that the real power of the meaning concept in computer science will unfold its full potential if we start to tackle the "simultaneous interaction and understanding (SIAU)" problem with iterative algorithms, mentioned in the introduction. In analogy to the solution of the SLAM problem, this requires us to explore how to represent easy context identification and switching capabilities and language-expressible vague knowledge properly structured to improve it iteratively, based on the speaker's experience. And not in the sense of a "speech-collage" were a system learns how to formulate sentences in a way that it becomes difficult to distinguish them from those generated by a knowledgeable system by some less knowledgeable system — although this mechanism also seems to be not uncommon even among humans.

Within linguistics, there is a long tradition to distinguish between semantics and pragmatics. Semantics is viewed as the study of the relation between syntactical and real world entities in a sense of the literal meaning of language expressions, following by and large Gottlob Frege's principle of compositionality. While pragmatics is viewed as the theory of language use, dealing with the way context can influence our understanding of language expressions (e.g. [Sza09]). I think, that my explanations argues against such a distinction, but rather for a model that first emphasizes necessary local interpretation contexts in a given interaction, which might be hierarchically structured and which might

[9] Please note that Thomas Henzinger today does think that nondeterminism is an important aspect of interactions (personal communication)

be changed on the fly, and that secondly emphasizes the role of internal states, both for representing and for acting.

From a philosophical point of view, the theory of interaction semantics implies that without interpretation, the world is meaningless. Concretely it says (or means) that meaning is attributed by interpretation and without such a mapping which we declare as interpretation, there is no meaning. More abstractly it says that the notion of meaning depends necessarily on the notion of interpretation performed by some interpreter and there is no meaning in any absolute sense.

Based on the presented concept of meaning, one could speculate that the "flow of thought" we introspectively experience when we think abstractly is based on "anticipated interactions" which would bind our capability to think abstractly reciprocally and thereby tightly to our ability to express ourselves language-wise, just as our ability to imagine playing an instrument like a violin depends on the extend of practice this instrument.

One could further speculate that sense and sensibility are inseparable if we understand our sensibility as a mode of understanding. If we are calm or angry, if we hate or love, we essentially interpret our world differently. Our emotions modulate our intuition about what is relevant or not (see e.g. [SL18] for an overview, how emotion and cognition interact).

Actually, this theory of meaning relates state values to state values. There are other theories doing so, like physics. So, I think, the most important consequence of this theory of meaning is to show that talking about meaning is not something special, almost magical, or only philosophical, but it is just another way to talk (and think) about some phenomena, in this case our interactions. This implies some potential, namely to derive powerful concepts, but it also implies some limitations. We can talk about the physics of an asteroid impact on Earth or we can talk about the meaning of such an event for the existence of humanity. In the latter case it is us who interprets, that is, makes some distinctions about the relevance of a "physical" phenomenon by choosing a certain context. Do bacteria attribute meaning if they follow a gradient of some soluble indicator substance? Yes and no. No in the sense that they do not have a theory of meaning and can articulate what they are doing, but yes in the sense that we can describe what they do in the framework of our theory of meaning relating states to states, separating the relevant from the irrelevant in a chosen context.

A unifying understanding of interaction semantics and meaning could therefore provide a common conceptual framework such that scientists of natural sciences and humanities as well as engineers could understand each other more easily, especially with the advent of the cyber-physical systems that are just on the doorstep.

Acknowledgments: I specially thank Tizian Schröder and Christian Diedrich with whom I had many fruitful discussions on this topic.

References

[Alu15] Rajeev Alur. *Principles of Cyber-Physical Systems.* MIT Press, 2015.

[Ara17] Naoki Araki. Chomsky's I-language and E-language. *Hiroshima Institute of Technology Research*, pages 17–24, 2017.

[BDR16] Felix Bangemann, Christian Diedrich, and Johannes Reich. A model for the semantics of component interactions of cyber-physical systems. In *Industrial Electronics (ISIE), 2016 IEEE 25th International Symposium on*, pages 1042–1047. IEEE, 2016.

[BEJvR11] Anton Benz, Christian Ebert, Gerhard Jäger, and Robert van Rooij. Language, Games, and Evolution: An Introduction. In Anton Benz, Christian Ebert, Gerhard Jäger, and Robert van Rooij, editors, *Language, Games, and Evolution: Trends in current research on language and game theory*, volume 6207, pages 1–13. Springer-Verlag Berlin Heidelberg, 2011.

[BM99] Harry Bunt and Reinhard Muskens. Computational semantics. In *Computing meaning*, pages 1–32. Springer, 1999.

[Bol20] Gemma Boleda. Distributional semantics and linguistic theory. *Annual Review of Linguistics*, 2020.

[Bro10] Manfred Broy. A logical basis for component-oriented software and systems engineering. *Comput. J.*, 53(10):1758–1782, 2010.

[CS82] Vincent P Crawford and Joel Sobel. Strategic information transmission. *Econometrica*, pages 1431–1451, 1982.

[DAH01] Luca De Alfaro and Thomas A Henzinger. Interface theories for component-based design. In *International Workshop on Embedded Software*, pages 148–165. Springer, 2001.

[Far01] Berndt Farwer. Omega-Automata. In Erich Grädel, Wolfgang Thomas, and Thomas Wilke, editors, *Automata, Logics, and Infinite Games*, volume 2500 of *Lecture Notes in Computer Science*, pages 3–20. Springer, 2001.

[Gri89] H Paul Grice. *Studies in the Way of Words.* Harvard University Press, 1989.

[Grü10] Thomas Grüter. *Magisches Denken.* Scherz Verlag, Frankfurt, 2010.

[Hol91] Gerard J. Holzmann. *Design and validation of computer protocols.* Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1991.

[HP85] D. Harel and A. Pnueli. On the development of reactive systems. In Krzysztof R. Apt, editor, *Logics and Models of Concurrent Systems*, pages 477–498. Springer-Verlag, New York, 1985.

[HS97] Jaakko Hintikka and Gabriel Sandu. Game-theoretical semantics. In *Handbook of logic and language*, pages 361–410. Elsevier, 1997.

[IECff] IEC. IEC 60050 International Electrotechnical Vocabulary, 2001ff.

[Jas16] Kasia M Jaszczolt. *Meaning in linguistic interaction: semantics, metasemantics, philosophy of language*. Oxford University Press, 2016.

[Lew69] David Lewis. *Convention: A Philosophical Study*. Harvard University Press, 1 edition, 1969.

[LL78] Paul Lorenzen and Kuno Lorenz. *Dialogische Logik*. Wissenschaftliche Buchgesellschaft, Darmstadt, 1978.

[MPW92] R. Milner, J. Parrow, and D. Walker. A calculus of mobile processes (parts I and II). *Information and Computation*, 100(1):1–77, 1992.

[Mye91] Roger B. Myerson. *Game Theory: Analysis of Conflict*. Harvard University Press, Cambridge, Massachusetts, 1991. reprinted 1997 as paperback.

[Pos07] Stefan Poslad. Specifying protocols for multi-agent systems interaction. *ACM Trans. Auton. Adapt. Syst.*, 2(4), November 2007.

[Rei01] Johannes Reich. Über Struktur oder das Verhältnis der Teile zum Ganzen. *Philosophia Naturalis*, 38(1):37–69, 2001.

[Rei10] Johannes Reich. Finite system composition and interaction. In Klaus-Peter Fähnrich and Bogdan Franczyk, editors, *40. GI Jahrestagung (2)*, volume 176 of *LNI*, page 603. GI, 2010.

[Rei20] Johannes Reich. Composition, cooperation, and coordination of computational systems. *preprint arXiv:1602.07065*, 2016/2020.

[RS20] Johannes Reich and Tizian Schröder. A simple classification of discrete system interactions and some consequences for the solution of the interoperability puzzle. *ifac 2020*, 2020.

[Sak09] Jacques Sakarovitch. *Elements of Automata Theory*. Cambridge University Press, 2009.

[SD20] Tizian Schröder and Christian Diedrich. Formal Definition of the Term "Semantic" as a Foundation for Semantic Interoperability in the Industrial Internet of Things. *accepted at ifac2020*, 2020.

[Sha48] Claude E. Shannon. A Mathematical Theory of Information. *Bell System Technical Journal*, 27:379–423, 623–656, 1948.

[Sha49] Claude E. Shannon. Communication in the Presence of Noise. *Proceedings of the IRE*, 37(1):10–21, 1949.

[Sky10] Brian Skyrms. *Signals: Evolution, learning, and information.* Oxford University Press, 2010.

[SL18] Alexander J. Shackman and C. Lapate, Regina. How Do Emotion and Cognition Interact? In Andrew S. Fox, Regina C. Lapate, Alexander J. Shackman, and Richard J. Davidson, editors, *The nature of emotion. Fundamental questions*, pages 209–211. Oxford University Press, 2 edition, 2018.

[SP73] J. Maynard Smith and GEORGE R. Price. The logic of animal conflict. *Nature*, 246:15–18, 1973.

[Sza09] Zoltan Gendler Szabo. The Distinction between Semantics and Pragmatics. In Ernest Lepore and Barry C. Smith, editors, *The Oxford Handbook of Philosophy of Language*. Oxford University Press Oxford, 2009.

[Tal97] Carolyn Talcott. Interaction semantics for components of distributed systems. In E. Najm and J. B. Stefani, editors, *Formal Methods for Open Object-based Distributed Systems. IFIP Advances in Information and Communication Technology.*, pages 154–169. Springer, 1997.

[Tar35] Alfred Tarski. Der Wahrheitsbegriff in den formalisierten Sprachen. *Studia Philosophica. Commentarii Societatis Philosophicae Polonorum*, 1:261 – 405, 1935. Auch in: Tarski, Givant, McKenzie (1986), S. 53 – 197.

[TBF05] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics.* The Mit Press, 2005.

[vB+08] Johan van Benthem et al. Games that make sense: Logic, language and multi-agent interaction. *New perspectives on games and interaction*, 4:197–209, 2008.

[Wit53] Ludwig Wittgenstein. *Philosophical Investigations.* Oxford: Basil Blackwell, 1953. Transl. by G. E. M. Anscombe.

[WLM16] Sida I. Wang, Percy Liang, and Christopher D. Manning. Learning language games through interaction. *CoRR*, abs/1606.02447, 2016.

[Wä93] Karl Wärneryd. Cheap talk, coordination, and evolutionary stability. *Games and Economic Behavior*, 5(4):532–546, 1993.