

# Axiomatizing Complex Concepts from Fundamentals

Jerry R. Hobbs and Andrew Gordon  
University of Southern California  
Marina del Rey, California, USA

No Institute Given

## 1 Introduction

We have been engaged in the project of encoding commonsense theories of cognition, or how we think we think, in a logical representation. In this paper we use the concept of a “serious threat” as our prime example, and examine the infrastructure required for capturing the meaning of this complex concept. It is one of many examples we could have used, but it is particularly interesting because building up to this concept from fundamentals, such as causality and scalar notions, highlights a number of representational issues that have to be faced along the way, where the complexity of the target concepts strongly influences how we resolve those issues.

We first describe our approach to definition, defeasibility, and reification, where hard decisions have to be made to get the enterprise off the ground. We then sketch our approach to causality, scalar notions, goals, and importance. Finally we use all this to characterize what it is to be a serious threat. All of this is necessarily sketchy, but the key ideas essential to the target concept should be clear.

## 2 Characterization and Defeasibility

In order to get started in encoding commonsense knowledge, one must build up a great deal of conceptual and notational infrastructure, and make a large number of warranted but highly controversial decisions about representation.

Among the first of these involves how tightly we can hope to define or characterize commonsense concepts. Our view is that where we can *define* a concept by necessary and sufficient conditions, that is good, but it is the exception rather than the rule. In general, the most we can hope to do is *characterize* concepts with lots of necessary conditions and lots of sufficient conditions. For example, we can’t hope to define causality, but we can specify several key properties that follow from a causal relation between events, and we can list a great many pairs of causes and effects. By adding axioms, we constrain the set of possible interpretations of the predicates. It should be mentioned that this is not always done in efforts to encode commonsense knowledge. It is a common criticism that OpenCyc [14, 3] is axiomatically poor; a prose description is given for a predicate that is introduced, but the set of axioms involving the predicate in general do not begin to constrain its interpretation to what the description says it means. In an early version of another popular large-scale ontology, the predicate “near” had only the property of being symmetric, which does not distinguish it from “far”. In our effort, we have tried to focus on the axioms that delimit the meanings of predicates, rather than relying on the reader’s intuition about the meaning of a term.

A related property of formalizations of commonsense knowledge is the defeasibility of the rules. Inferences can be drawn that subsequently must be retracted because of further

information. The notation in which the axioms are expressed can be first-order logic, but there has to be a nonmonotonic proof procedure applied to them. The one we have assumed is weighted abduction [13], but our formalization could be adapted to any other approach to nonmonotonicity.

For a notation, we use a subset of Common Logic [2], essentially, textbook logic in a LISP-like format. In weighted abduction, it is possible to include “et cetera” predications in the antecedents of Horn clauses to indicate that other unspecified conditions may be relevant to the conclusion. Thus, an axiom saying that  $p$  defeasibly implies  $q$  might be written

```
(forall (x) (if (and (p x)(etc-i x)) (q x)))
```

where  $i$  is unique to this axiom. The “et cetera” predications can be thought of as the negations of McCarthy’s abnormality predications in circumscriptive logic [16]. In weighted abduction, they are never proved, but they can be assumed and thereby become part of the best abductive proof of the goal expression.

In this paper, we will abbreviate an axiom like the above to

```
(forall (x) (if (and (p x)(etc)) (q x)))
```

where  $(etc)$  is understood to stand for an “et cetera” predicate unique to this axiom applied to all the universally quantified variables whose scope it is within. More generally, the reader can view  $(etc)$  as simply an indication of the defeasibility of the rule, to be dealt with by the nonmonotonic inference procedure of choice.

### 3 Reification and Eventuality Types and Tokens

The domain of discourse for our logical theories is the class of possible individual entities, states, and events. They may or may not exist in the real world, and if they do, it is one of their properties, expressed as  $(Rexist\ x)$ . For example, in representing the sentence “John worships Zeus,” both John and Zeus are in the universe of possible individuals, but only John *really* exists.

In a narrowly focused inquiry it is often most perspicuous to utilize specialized notations for the concepts under consideration. But our view is that in a broad-based effort like ours, this is not possible, and that it can be avoided by sufficient judicious use of reification.

For example, we treat sets as first-class individuals. Moreover, sets are taken to have “type elements”, whose principal feature is that their properties are inherited by the real elements of the sets [15, 8, 11]. The expression  $(type\ elt\ x\ s)$  says that  $x$  is the type element of set  $s$ .

The term “eventuality” is used to cover both states and events [7, 9]. Eventualities like other individuals can be merely possible or can really exist in the real world. A notational convention we use is that whereas the expression  $(p\ x)$  says that predicate  $p$  is true of  $x$ , the expression  $(p'\ e\ x)$  says that  $e$  is the eventuality of  $p$  being true of  $x$ . The relation between the primed and unprimed predicates is given by the axiom schema

```
(forall (x) (iff (p x)(exist (e)(and (p' e x)(Rexist e))))))
```

Eventuality arguments allow us to specify properties of eventualities without introducing scoping. “Pat has the goal of Chris’s being happy” could be represented

```
(and (goal E P)(happy' E C))
```

That is,  $E$  is a goal of Pat’s, where  $E$  is the eventuality of Chris’s being happy.

Eventualities are very finely individuated. For example, Pat’s walking to work and Pat’s going to work are two different eventualities. The reason for this is that they may have different properties. The walking may be fast while the going isn’t.

Eventualities are therefore very nearly in one-one correspondence with predications in the logic, and we can be somewhat cavalier about the distinction. For example, we can speak of the “arguments” of eventualities as a way of referring to the participants in the states or events. The expression (`argn x 1 e`) says that `x` is the first direct argument of `e`, and the expression (`arg x e`) says that `x` is some direct argument of `e`. Since eventualities can be the arguments, it is useful to define a recursive equivalent of “argument”.

```
(forall (x e1)
  (iff (arg* x e1)
    (or (arg x e1)(exist (e2)(and (arg e2 e1)(arg* x e2))))))
```

Thus, in the above expression, `C` is an `arg*` of the eventuality of `P`'s having goal `E`. We can think of (`arg* x e`) as saying that `x` is somehow involved in eventuality `e`.

We have explicitly modelled substitution in axioms (cf. [8]). The expression (`subst x1 e1 x2 e2`) can be read as saying that `x1` plays the same role in eventuality `e1` that `x2` plays in eventuality `e2`, where `e1` and `e2` have the same predicate. Similarly, the expression (`subst2 x1 y1 e1 x2 y2 e2`) says that `x1` and `y1` play the same roles in eventuality `e1` that `x2` and `y2` play in eventuality `e2`, respectively.

Eventualities can have type elements of sets as their arguments, and when they do, they are eventuality types. An `instanceOf` relation relates eventuality types and tokens. If `e1` is an eventuality type whose only type element is `x`, the type element of set `s`, `y` is a member of `s`, and `e2` is an eventuality such that (`subst x e1 y e2`) holds, then `e2` is an eventuality token and an instance of `e1`.

Conjunctions, disjunctions, implications, and negations of eventualities are eventualities as well. The expression (`not' e1 e2`) says that `e1` is the eventuality of eventuality `e2`'s not really existing.

We have axiomatized a theory of time [12], and eventualities can have temporal properties. The expression (`atTime e t`) says that eventuality `e` occurs at time `t`. Thus, we use temporal properties rather than temporal arguments for eventualities.

The idea of reifying events is usually attributed to Davidson ([4]), although he was reluctant to reify states as well, and he did not individuate events as finely as we do. The linguist Emmon Bach ([1]) recognized the need for a concept that covered both states and events and introduced the term “eventuality”. A brief exposition of eventualities as used here can be found in [7] and a more extensive exposition in [9]. The latter contains a number of arguments for the need for eventualities, ways of looking at eventualities, and arguments for very fine individuation.

## 4 Causality

The account of causality we employ is that of [10]. This distinguishes between the monotonic, precise notion of “causal complex” and the nonmonotonic, defeasible notion of “cause”. The former gives us mathematical rigor; the latter is more useful for everyday reasoning and can be characterized in terms of the former. We begin with an abbreviated account of these concepts.

When we flip a switch to turn on a light, we say that flipping the switch caused the light to turn on. But many other factors had to be in place. The bulb had to be intact, the switch had to be connected to the bulb, the power had to be on in the city, and so on. We will use the predicate `cause` for flipping the switch, and introduce the predicate `causalComplex` to refer to the set of all the states and events that have to hold or happen for the effect to happen. The states of the bulb, the wiring, and the power supply would all be in the causal complex.

Causal complexes have two primary features. The first is that if all of the eventualities in the causal complex obtain or occur, then so does the effect. The second is that each of the members of the causal complex is relevant, in the sense that if it is removed from the set, the remainder is not a causal complex for the effect.

In practice, we can never specify all the eventualities in a causal complex for an event. So while the notion gives us a precise way of thinking about causality, it is not adequate for the kind of practical reasoning we do in planning, explaining, and predicting. For this, we need the defeasible notion of “cause”.

In a causal complex, for most events we can bring about, the majority of the eventualities are normally true. In the light bulb case, it is normally true that the bulb is not burnt out, that the wiring is intact, that the power is on in the city, and so on. What is not normally true is that someone is flipping the light switch. Those eventualities that are not normally true are identified as causes. They are useful in planning, because they are often the actions that the planner or some other agent must perform. They are useful in explanation and prediction because they frequently constitute the new information. They are less useful for diagnosis, because diagnosis is employed exactly when the normal cause fails to bring about its normal effect, and the rest of the causal complex has to be examined.

In [10] the interpretation of the predicate `cause` is constrained by axioms involving the largely unexplicated notion of “presumable”; most elements of a causal complex can be presumed to hold, and the others are identified as causes. We won’t repeat that development here, but we will place some looser constraints on causes.

First, a cause is an eventuality in a causal complex.

```
(forall (e1 e2)
  (if (cause e1 e2)
      (exist (s)(and (causalComplex s e2)(member e1 s))))))
```

This allows only single eventualities to be causes, and of course many events have multiple causes. But this is not a limitation because we can always bundle the multiple causes into a single conjunction of causes. So if `e1` is pouring starter fluid onto a pile of firewood and `e2` is lighting a match, then the cause of the fire starting is `e3` where `(and' e3 e1 e2)` holds.

The principal useful property of `cause` is a kind of causal modus ponens. When the cause happens or holds, then, defeasibly, so does the effect.

Causality is not strictly speaking transitive. Shoham ([17]) gives as an example that making a car lighter causes it to go faster, and taking the engine out causes the car to be lighter, but taking the engine out does not cause the car to go faster. In the second action, we have undone one of the presumable conditions in the causal complex for the first action. The two causal complexes are inconsistent. However, when they are consistent, `cause` is transitive, so it is defeasibly transitive.

```
(forall (e1 e2 e3)
  (if (and (cause e1 e2) (cause e2 e3) (etc))(cause e1 e3)))
```

Hobbs ([10]) is explicit about exactly what the content of the “et cetera” predicate is, in terms of presumable eventualities.

A causal complex consists of causes and other, presumable or nonproblematic, eventualities. The latter are frequently referred to as enabling conditions or preconditions. In the STRIPS model of Fikes and Nilsson [5] that has become the standard model for planning in artificial intelligence, the enabling conditions correspond to the preconditions and the body corresponds to the cause. The added and deleted states correspond to the effect.

## 5 Scales

A scale is a set of entities with a partial ordering among them.

```

(forall (s)
  (if (scale s)
    (exist (s1 e x y)
      (and (componentsOf s1 s)(partialOrdering e x y s))))))

```

The predicate `componentsOf` is explicated further in a theory of composite entities not discussed here [11]. The expression `(partialOrdering e x y s)` says that `e`, an eventuality type, is the relation of some `x` being less than some `y`, where `x` and `y` are components of the scale `s`.

```

(forall (e x y s)
  (if (partialOrdering e x y s)
    (and (scale s)(arg* x e)(arg* y e))))

```

In more conventional notation, we can think of `e` as a lambda expression and `x` and `y` as its two bound variables. However, since the `subst` predicate described above works equally well on types and tokens, we don't need to specify that `x` and `y` are variables, or types, or type elements, or anything else.

It is generally more convenient to speak directly of the partial ordering relation among elements. We can define a “less than” relation as follows, using the predicate name `lts` to indicate that it is relative to a particular scale `s`.

```

(forall (e1 x1 y1 s)
  (iff (lts' e1 x1 y1 s)
    (exist (e x y)
      (and (partialOrdering e x y s)
        (subst2 x y e x1 y1 e1))))))

```

Then the standard properties of partial orderings can be defined in terms of the predicate `lts`. The partial ordering is antireflexive, antisymmetric, and transitive. We also define the “less than or equal” relation.

We have frequent occasion to define particular scales. This is done by specifying the set of entities that are the components of the scale, and the relation that is the partial ordering of the scale.

```

(forall (s s1 e)
  (iff (scaleDefinedBy s s1 e)
    (and (scale s)(componentsOf s1 s)
      (exist (x y) (partialOrdering e x y s))))))

```

It is convenient to define a component of a scale, `(componentOf x s)`, as a member of its set of components. We can define subscales and the top and bottom of a scale in the obvious way.

Suppose we have two scales with the same set of components. Then we can define a composite scale that is consistent with the two original scales. For example, suppose the set is points in the United States, in the first scale the partial ordering (in this case total) is “northOf”, and in the second scale the partial ordering is “eastOf”. Then in the composite scale the partial ordering is at least consistent with the “northAndEastOf” relation. We may in addition impose further structure on the composite scale, for example, by saying that the “northOf” relation takes precedence.

The loose constraints on a composite scale are as follows:

```

(forall (s s1 s2)
  (if (compositeScale s s1 s2)
    (and (exist (s3)
      (and (componentsOf s3 s1)(componentsOf s3 s2)

```

```

      (componentsOf s3 s))
(forall (x y)
  (if (and (lts x y s1)(leqs x y s2))(lts x y s)))
(forall (x y)
  (if (and (leqs x y s1)(lts x y s2))(lts x y s))))))

```

The same set `s3` is the set of components of the two original scales and the composite scale. If an entity `x` is less than an entity `y` on one of the original scales and less than or equal to `y` on the other, then it is less than `y` on the composite scale.

There is a range of structures we can impose on scales. These map complex scales into simpler scales. For example, in much work in qualitative physics the actual measurement of some parameter may be any real number, but this is mapped into one of three values – positive, zero, and negative. Where the parameter is vertical velocity, this means we are only interested in whether something is going up, staying at the same elevation, or going down.

We have introduced another sort of structure on scales, one reflected in language. What we have defined so far is adequate for characterizing the comparative and superlative forms of adjectives – “taller” and “tallest” – but not for the absolute form of adjectives – “tall”. In natural language and in qualitative reasoning we often characterize something as being in the high or low region of a scale, with no more precise characterization of its location. We will call these regions the `Hi` and `Lo` regions of the scale. Each of these predicates is a relation between a scale `s` and one of its subscales `s1` – (`Hi s1 s`). The top of the scale, if there is one, is the top of the `Hi` region of the scale, and the bottom of a scale is the bottom of its `Lo` region. The bottom of the `Hi` region and the top of the `Lo` region will rarely be known exactly. There is no well-established height that is the minimum height that counts as tall. Nevertheless, we can say that if a point is in the `Lo` region, then it is less than all the points in the `Hi` region.

It is often useful to go from the absolute form of an adjective to its underlying scale, for example, from “tall” to the height scale. We use the predicate `scaleFor` for this relation.

```

(forall (s e)
  (iff (scaleFor s e)
    (exist (s1)
      (and (Hi s1 s)
        (forall (e1 x)
          (if (and (componentOf' e1 x s1)(argn x 1 e))
            (iff (Rexist e)(Rexist e1))))))))))

```

For example, suppose we have (`tall' e x`), that is, `e` is the property of `x`'s being tall. Then `s` is the height scale, `s1` is the `Hi` region of the height scale, and whenever we have a relation `e1` of `x` being in that `Hi` region, then `e1` holds exactly when `e` holds. That is, some entity `x` is tall exactly when `x` is in the `Hi` region of the height scale. The height scale is the `scaleFor` the property “tall”. In line 6 we specify that `x` must be the first argument of `e`, because if there are multiple arguments, we need to say which one is the relevant argument placed on the scale.

There are two primary external theories that a theory of the qualitative structure on scales should link to. The first is an as-yet-to-be-developed commonsense theory of distributions. The `Hi` and `Lo` regions usually correspond to the right and left tails of a distribution. As a first approximation, we can say that if something is in the `Hi` region of a scale, then defeasibly it is higher on the scale than *most* entities in some contextually determined comparison set.

The second is a theory of functionality or goals, as outlined in Section 6. Often when we say that an entity is tall, we mean that it is tall *enough* for something or *too* tall for

something. Discovering that something is recognizing the connection between qualitative scalar judgments and functionality.

More specifically, we can say that, defeasibly, if something is in the Hi region of a scale, then that property plays a causal or enabling role in some agent’s goal being achieved or not being achieved. We can state this as follows:

```
(forall (e x s1 s)
  (if (and (componentOf' e x s1)(Hi s1 s)(etc))
    (exist (c a g g1)
      (and (member e c)(goal g a)
        (or (causalComplex c g)
          (and (not' g1 g)(causalComplex c g1)))))))
```

That is, if *e* is the property of *x* being in the Hi region *s1* of some scale *s*, then defeasibly *e* is part of a causal complex *c* that will bring about some agent *a*’s goal *g* or its negation *g1*. This axiom does not tell us who the agent is or what the goal is. That has to be determined from context. But it does alert us to the possible relevance of such a goal.

We axiomatize the notion of a “likelihood scale”, as a qualitative, commonsense concept corresponding to standard probability and of which standard probability is one possible model. Space precludes presenting the details of this.

## 6 Agents and Goals

An agent is an entity that can, in the commonsense view of things, initiate a causal chain. People are agents. When someone decides to stand up and cross the room, there are neural events that are causing this, but we normally don’t carry our analysis of the event to this level. We view the person’s decision as the initial cause. Higher animals, organizations, and complex artifacts are also often viewed as agents.

Commonsense psychology is about people, but most of it applies more generally to agents. Agents have beliefs. We take the objects of belief to be eventualities. Because eventualities are very finely individuated, there is a straightforward translation between talking of belief in an eventuality and belief in a proposition. The expression (**believe a e**) can be read as saying that agent *a* believes the proposition that eventuality *e* really exists. We have developed but not included here our treatment of belief [6] because of space limitations and because it breaks no new ground in the abundant literature on logics of belief. Our use of the predicate here should be obvious and unproblematic.

Human beings are intentional agents. We have goals, we develop plans for achieving these goals, and we execute the plans. We monitor the executions to see if things are turning out the way we anticipated, and when they don’t, we modify our plans and execute the new plans. The concept of a goal is central to this formulation.

The key concept in modeling intentional behavior is that of an agent *a* having some eventuality type *e* as a goal. The expression (**goal e a**) says that eventuality *e* is a goal of agent *a*. Normally, *e* will be an eventuality type that can be satisfied by any number of specific eventuality tokens, but it is entirely possible in principle for an agent to have an eventuality token as a goal, where there is only one satisfactory way for things to work out. We won’t belabor the distinction here.

Agents know facts about what causes or enables what in the world, in most cases, facts of the form

```
(forall (e1 x)
  (if (p' e1 x) (exist (e2)(and (q' e2 x)(cause e1 e2))))))
```

and a similar axiom for **enable**. That is, if **e1** is the eventuality of **p** being true of some entities **x**, then there is an eventuality **e2** that is the eventuality of **q** being true of **x** and **e1** causes or enables **e2**. Or stated in a less roundabout way, **p** causes or enables **q**.

The agent uses these rules to plan to achieve goals and to infer the goals and plans of other agents. A plan is an agent's way of manipulating the causal properties of the world to achieve goals, and these axioms express causal properties.

We will work step by step toward a characterization of the planning process. The first version of the axiom we need says that if agent **a** has a goal **e2** and **e1** causes **e2**, then **a** will also have **e1** as a goal.

```
(forall (a e1 e2) (if (and (goal e2 a)(cause e1 e2))(goal e1 a)))
```

This is not a bad rule, and certainly is defeasibly true, but it is of course necessary for the agent to actually believe in the causality, and if the agent believes a causal relation that does not hold, **e1** may nevertheless be adopted as a goal. The causal relation needn't be true.

```
(forall (a e0 e1 e2)
  (if (and (goal e2 a)(cause' e0 e1 e2)(believe a e0))(goal e1 a)))
```

We can say furthermore that the very fact that **a** has goal **e2** causes **a** to have goal **e1**. We do this by reifying the eventuality **g2** that **e2** is a goal of **a**'s, and similarly **g1**. (The **e**'s in this axiom are the eventualities of having something; the **g**'s are the eventualities of wanting it.)

```
(forall (a e0 e1 e2 g2)
  (if (and (goal' g2 e2 a)(cause' e0 e1 e2)(believe a e0))
    (exist (g1)(and (goal' g1 e1 a)(cause g2 g1)))))
```

That is, if agent **a** wants **e2** and believes **e1** causes **e2**, that wanting will cause **a** to want **e1**. (The belief is also in **g1**'s causal complex, but that would not normally be thought of as the cause: Why do you want **e1**? Because I want **e2**.)

Note that while the antecedent and the consequent no longer assert the real existence of having the goal (i.e., **g2** and **g1**), if we know that **g2** really exists, then the real existence of **g1** follows from the properties of **cause**.

Note also that the predicate **goal** reverses causality. For example, because flipping a light switch causes a light to go on, having the goal of the light being on causes one to want to flip the switch.

The eventuality **e1** is a "subgoal" of **e2**, and we encode this in the axiom.

```
(forall (a e0 e1 e2 g2)
  (if (and (goal' g2 e2 a)(cause' e0 e1 e2)(believe a e0))
    (exist (g1)
      (and (goal' g1 e1 a)(cause g2 g1)(subgoal e1 e2 a)))))
```

Finally, this axiom is not always true. There may be many ways to cause the goal condition to come about, and the mystery of the agent's free choice intervenes. The axiom is only defeasible. We can represent this by means of an "et cetera" proposition in the antecedent.

```
(forall (a e0 e1 e2 g2)
  (if (and (goal' g2 e2 a)(cause' e0 e1 e2)(believe a e0)(etc))
    (exist (g1)
      (and (goal' g1 e1 a)(cause g2 g1)(subgoal e1 e2 a)))))
```

That is, if agent **a** has a goal **e2** (where **g2** is the eventuality of wanting **e2**) and **a** believes **e1** causes **e2**, then defeasibly this wanting **e2** will cause **a** to want **e1** as a subgoal of **e2** (where **g1** is the eventuality of wanting **e1**).

A similar succession of axioms can be written for enablement.

The "subgoal" relation is a relation between two goals, and implies the agent's belief that the subgoal is in a causal complex for the goal.



```

(forall (e1 e2 a)
  (if (subgoal e1 e2 a)
    (and (goal e2 a)(goal e1 a)
      (exist (e3 e4 e5 s)
        (and (causalComplex' e3 s e2)(member' e4 e1 s)
          (and' e5 e3 e4)(believe a e5))))))

```

In lines 5-6 of this axiom, *e3* is the proposition that *s* is a causal complex for *e2*, *e4* is the proposition that *e1* is a member of *s*, *e5* is the conjunction of these two propositions, and that's what agent *a* believes.

Goals do not have to be directly achievable by actions on the part of the agent, but successful plans have to bottom out in such actions or in states or events that will happen or hold at the appropriate time anyway.

It is formally convenient to assume that agents have one plan that they are always developing, executing, monitoring and revising, and that that plan is in the service of a single goal. We will call this goal "Thriving".

```

(forall (a) (if (agent a)(exist (e)(and (goal e a)(thrive' e a))))

```

More specific goals arise out of the planning process using the agents' beliefs about what will cause them to thrive.

The main reason for positing this top-level goal is that now instead of worrying about the mysterious process by which an agent comes to have goals, we can address the planning problems of what eventualities the agent believes cause other eventualities, including the eventuality of thriving, and of what alternative subgoals the agent should choose to achieve particular goals. We are still left with the problem of when one goal should be given priority over another, but this is now a plan construction issue.

We will not attempt to say what constitutes thriving in general, because there are huge differences among cultures and individuals. For most of us, thriving includes staying alive, breathing, and eating, as well as having pleasurable experiences. But many agents decide they thrive best when their social group thrives, and that may involve agents sacrificing themselves. This is a common view in all cultures, as seen in suicide bombers, soldiers going into battle, and people risking death to aid others. So thriving does not necessarily imply surviving.

A good theory of commonsense psychology should not attempt to define thriving, but it should provide the materials out of which the beliefs of various cultures and individuals can be stated in a formal manner.

## 7 Importance

Many scales, including the scale of importance, cannot be defined precisely, but constraints can be placed on their partial ordering. That is what we will do here.

A concept, entity or eventuality is more or less important to an agent depending its relation to the agent's goals. The "more important" relation is thus a partial ordering that depends on the agent. The expression (`moreImportant x1 x2 a`) says that something *x1* is more important than something else *x2* to agent *a*. We place no constraints on the things *x1* and *x2* whose importance is being measured. They can be anything.

A plan can be thought of as a tree-like structure representing the `subgoal` relation. The higher a goal is in an agent's plan to thrive, the more important it is to the agent, because of the greater amount of replanning that has to be done if the goal cannot be achieved. So the first constraint we can place on the importance scale is that it is consistent with the subgoal relation.

However, this is a bit tricky to specify because an eventuality can be a subgoal of a number of different higher-level goals in the same plan, and we do not want to say an eventuality is of little importance simply because one of its supergoals is of little importance. So we first need to define the notions of an “upper bound supergoal” and a “least upper bound supergoal”. An eventuality  $e1$  is an upper bound supergoal of  $e2$  if it is a supergoal of all of  $e2$ ’s immediate supergoals. More precisely, any supergoal of  $e2$ ’s must either be  $e1$ , be a subgoal of  $e1$ , or be a supergoal of  $e1$ .

```
(forall (e1 s a)
  (iff (ubSupergoal e1 s a)
    (and (agent a)(goal e1 a)
      (forall (e2) (if (member e2 s)(subgoal e2 e1 a)))
      (forall (e2 e) (if (and (member e2 s)(subgoal e2 e a))
        (or (subgoal e e1 a)(eq e e1)
          (subgoal e1 e a)))))))
```

The expression `(ubSupergoal e1 s a)` says that  $e1$  is an upper bound supergoal of all the goals of agent  $a$  in set  $s$ . The predicate holds if and only if any eventuality  $e$  which is a supergoal of a member  $e1$  of  $s$  is either a subgoal of  $e1$ ,  $e1$  itself, or a supergoal of  $e1$ .

A goal  $e1$  is a least upper bound supergoal if it is an upper bound supergoal and a subgoal of all other upper bound supergoals.

```
(forall (e1 s a)
  (iff (lubSupergoal e1 s a)
    (and (ubSupergoal e1 s a)
      (forall (e)
        (if (ubSupergoal e s a)
          (or (eq e e1)(subgoal e1 e)))))))
```

Because every goal is ultimately in the service of the top-level goal “To Thrive”, every goal has a least upper bound supergoal.

Now we can say that if eventuality  $e1$  dominates eventuality  $e2$  on every path in the agent’s plan that includes  $e2$ , then  $e1$  is more important than  $e2$ . Every reason for wanting  $e2$  is in the service of  $e1$ .

```
(forall (s e1 e2 a)
  (if (and (member e2 s)(lubSupergoal e1 s a))
    (moreImportant e1 e2 a)))
```

More generally, we can say that the least upper bound supergoal of a set of goals is more important than the whole set, since all the members of the set are in the service of the supergoal.

```
(forall (s e a)
  (if (lubSupergoal e s a)(moreImportant e s a)))
```

An agent’s goals are important. So are eventualities that affect the agent’s goals. Importance doesn’t care about polarity; if passing a course is important to you, so is not passing the course. Thus, we define an eventuality as “goal-relevant” to an agent, `(goalRelevant e a)`, if its existence implies the existence or nonexistence of one of the agent’s goals. The “goal consequences” of an eventuality, `(goalConsequences s e a)`, are those goals of the agent’s whose existence or nonexistence is implied by the eventuality. Then we can say the importance of an eventuality depends on the importance of its goal consequences. That is, if something  $x$  is more important than the goal consequences of eventuality  $e$ , then  $x$  is more important than  $e$ .

The importance of an entity depends on the importance of its properties and of the events it participates in. Thus, we define the set of “goal-relevant properties”.

```

(forall (s x a)
  (iff (grProps s x a)
    (forall (e)
      (iff (member e s)(and (arg* x e)(goalRelevant e a))))))

```

The expression `(grProps s x a)` says that `s` is the set of the set of properties of `x` that are relevant to a goal of `qa`'s.

The next axiom says that the importance of an entity depends on the importance of its goal-relevant properties.

```

(forall (s x a)
  (if (and (moreImportant s x1 a)(grProps s x2 a))
    (moreImportant x2 x1 a)))

```

To summarize, `x1` is more important than `x2` to `a` if `x2` is, or affects something that is, or has properties that affect something that is, in the service of `x1`. Note that there may be other properties constraining the `moreImportant` relation, but this one at least is among the most significant.

## 8 Threats and Seriousness

A threat situation is one in which the agent believes that things could turn out badly. To formalize this notion, we say that the agent `a` believes there is an eventuality `e` in a causal complex `s` whose effect `e0` would cause one of `a`'s goals not to be realized. The definition of a threat situation is as follows:

```

(forall (e s e0 a g)
  (iff (threatSituation e s e0 a g)
    (exist (e1 e2)
      (and (believe a e1)(member' e1 e s)(causalComplex s e0)
        (cause e0 e2)(not' e2 g)(goal g a))))))

```

The expression `(threatSituation e s e0 a g)` says that `e` is a threat by virtue of being a member of causal complex `s` with effect `e0` which threatens agent `a`'s goal `g`. In line 4 `e1` is the eventuality of `e`'s being in the causal complex; agent `a` believes this to be the case. In line 5 `e2` is the eventuality of goal `g` not obtaining.

Threats may be real, imaginary, or just imagined. These would be further properties of `e` and perhaps `s`.

The participating entities in a threat situation can be labelled by their role. In particular, the initiating event `e` is the threat.

```

(forall (e s e0 a g)
  (if (threatSituation e s e0 a g)(threat e a)))

```

That is, `e` is the threat to agent `a`.

Because the threat is only one element of the causal complex, the effect is only a possibility, not an inevitability. We generally refer to something as a threat when some evasive action is still possible.

Because of space considerations, this treatment of threats is actually somewhat simplified from that presented in [6]. There rather than mere belief, we have the agent in the dynamic process of envisioning a branching causal structure, one branch of which leads to the undesirable consequence.

A threat can be more or less serious to an agent. Seriousness is a composite scale that depends on importance and likelihood. Of two equally likely threats, the more important one is more serious. Of two equally important threats, the more likely is more serious. This composition of scales is captured in the predicate `compositeScale` introduced in Section 5. The likelihood scale was introduced in Section 5. The importance scale was introduced in Section 7. Hence, the definition of the "seriousness" scale is as follows:

```

(forall (s a)
  (iff (seriousnessScale s a)
    (exist (s1 s2)
      (and (likelihoodScale s1)(importanceScale s2 a)
        (compositeScale s s1 s2)
        (forall (e1)
          (if (componentOf e1 s)(threat e1 a)))))))

```

Lines 4-5 say that the seriousness scale is a composite of the likelihood scale and the importance scale. Lines 6-7 say that the elements of the scale are threats.

In terms of this we can define the partial ordering “more serious”.

```

(forall (e e1 e2 a)
  (iff (moreSerious' e e1 e2 a)
    (exist (s s1)
      (and (seriousnessScale s a)(scaleDefinedBy s s1 e))))

```

The expression `(moreSerious' e e1 e2 a)` says that `e` is the eventuality of one threat `e1` being more serious than threat `e2` to agent `a`. The predicate `scaleDefinedBy`, introduced in Section 5, relates a scale to its set of elements and its partial ordering.

We can define the qualitative predicate `serious` in terms of the Hi region of the scale, via the predicate `scaleFor`.

```

(forall (e e1 a)
  (iff (serious' e e1 a)
    (exist (s)
      (and (seriousnessScale s a)(scaleFor s e))))

```

The expression `(serious' e e1 a)` says that `e` is the eventuality of threat `e1` being serious to agent `a`. The predicate `scaleFor` relates a scale to the qualitative predicate of something being in the Hi region of the scale.

Thus, for a threat to an agent to be serious, it has to be high on a scale of seriousness, where “high” is related to the agent’s goals, most likely in this case the goals of the agent’s that are the focus of the threat. Seriousness in turn is characterized in terms of likelihood and importance of the threatened action. Importance is characterized by the action’s causal impact on a goal of the agent’s and that goal’s place in the agent’s subgoal hierarchy, or plan, for achieving the overarching goal of thriving.

## 9 Conclusion

Sophisticated natural language understanding will require a large knowledge base of commonsense knowledge. Much of that can be acquired automatically from large corpora and existing resources. But the very core of such a knowledge base will consist of rules that are too abstract and too complex for any automatic acquisition methods we can imagine today. It will require the manual encoding of theories such as we have described here for causality, scales, and goals. However, once these theories are explicated, some complex and deep accounts of meanings of difficult words can be constructed. We have demonstrated what this would look like for two particularly difficult words, “serious” and “threat”. These are just two representative examples of the knowledge we need and are in the position to begin to provide.

## References

1. Bach, E. (1981) "On Time, Tense, and Aspect: An Essay in English Metaphysics", in P. Cole, ed., *Radical Pragmatics*, pp. 63-81, Academic Press, New York.
2. Common Logic Working Group (2010) "Common Logic Standard". <http://common-logic.org/>.
3. Cycorp, 2008. <http://www.cyc.com/>.
4. Davidson, D. (1967) "The Logical Form of Action Sentences", in N. Rescher, ed., *The Logic of Decision and Action*, pp. 81-95, University of Pittsburgh Press, Pittsburgh, Pennsylvania.
5. Fikes, R., and Nilsson N. (1971) "STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving", *Artificial Intelligence*, 2: 189-208.
6. Gordon, A. and Hobbs, J. (2010) *How We Think We Think: Formalizing Commonsense Psychology*, in progress.
7. Hobbs, J. (1985) "Ontological Promiscuity", *Proceedings*, 23rd Annual Meeting of the Association for Computational Linguistics, Chicago, Illinois: 61-69.
8. Hobbs, J. (1995) "Monotone Decreasing Quantifiers in a Scope-Free Logical Form", in K. van Deemter and S. Peters (Eds.), *Semantic Ambiguity and Underspecification*, CSLI Lecture Notes No. 55, Stanford, California: 55-76.
9. Hobbs, J. (2003) "The Logical Notation: Ontological Promiscuity", <http://www.isi.edu/hobbs/disinf-tc.html>
10. Hobbs, J. (2005) "Toward a Useful Concept of Causality for Lexical Semantics", *Journal of Semantics*, 22: 181-209.
11. Hobbs, J. (2005) "Encoding Commonsense Knowledge", <http://www.isi.edu/hobbs/csk.html>.
12. Hobbs, J. and Pan F. (2004) "An Ontology of Time for the Semantic Web", *ACM Transactions on Asian Language Information Processing*, 3(1): 66-85.
13. Hobbs, J., Stickel, M., Appelt, D. and Martin P. (1993) "Interpretation as Abduction", *Artificial Intelligence*, Vol. 63, Nos. 1-2, pp. 69-142.
14. Lenat, D. and Guha, R. (1990). *Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project*. Addison-Wesley.
15. McCarthy, J. (1979) "First Order Theories of Individual Concepts and Propositions", in J. Hayes, D. Michie, and L. Mikulich, eds., *Machine Intelligence 9*, pp. 129-147, John Wiley and Sons, New York.
16. McCarthy, J. (1980) "Circumscription: A Form of Nonmonotonic Reasoning", *Artificial Intelligence*, 13: 27-39.
17. Shoham, Y. (1990) "Nonmonotonic Reasoning and Causation", *Cognitive Science*, Vol. 14, pp. 213-252.