From: AAAI Technical Report FS-98-02. Compilation copyright © 1998, AAAI (www.aaai.org). All rights reserved.

Interval-valued Epistemic (IVE) Fluents

John Funge
Microcomputer Research Lab
Intel Corporation
john.funge@intel.com

Abstract

We have developed a syntactic approach to representing knowledge within the situation calculus using interval arithmetic. Knowledge was first incorporated into the situation calculus using a possible-worlds approach. Unfortunately, this previous approach is not amenable to easy implementation. This is because it is not clear how to specify the initial situation as the number of possible worlds is potentially uncountable. We solve this problem by using interval-valued epistemic (IVE) fluents to represent the agent's knowledge of its world. With respect to the previous possible worlds approach, our approach is provably sound and (sometimes) complete.

Introduction

Autonomous agents are being used to help automate the process of making animations. The ultimate goal is to have virtual actors that can be directed in much the same manner as human actors. Such technology could also be used in computer games to provide more challenging adversaries and helpful partners. Unfortunately, such useful agents are still someway off, but the complexity of even some of today's virtual worlds means that these agents will have to be able to be able to reason, act and perceive in changing, incompletely known, unpredictable environments. It is natural, therefore, to turn to cognitive robotics for inspiration. In particular, we have adopted theories of action originally developed for robotics for use in animation and games. One of our key concerns in adopting these theories was that they should be practical and usable by programmers. To this end we have developed a syntactic or sentential approach to representing knowledge within the situation calculus using interval arithmetic. Knowledge was first incorporated into the situation calculus using a possible-worlds approach. Unfortunately, this previous approach is not amenable to easy implementation. This is because it is not clear how to specify the initial situation as the number of possible worlds is potentially uncountable. We solve this problem by using interval-valued epistemic (IVE) fluents to represent the agent's knowledge of its world. With respect to the previous possible worlds approach, our approach is provably sound and (sometimes) complete.

The situation calculus is a well known formalism for representing changing worlds in sorted first-order mathematical logic. The version we will be using is widely described in the literature (for example see (Reiter 1991; Scherl & Levesque 1993)). For the sake of completeness we briefly run through the main ideas. A *situation* is a "snapshot" of the state of the world. A domain-independent constant s_0 denotes the initial situation. Any property of the world that can change over time is known as a *fluent*. A fluent is a function, or relation, with a situation term as (by convention) its last argument. Actions are the fundamental instrument of change in our ontology. The situation s' resulting from doing action a in situation s is given by the distinguished function *do*, such that, $s' = do(a, s)$. The possibility of performing action a in situation s is denoted by a distinguished predicate $Poss(a.s)$. Sentences that specify what the state of the world must be before performing some action are known as *precondition axioms*. *Effect axioms* give necessary conditions for a fluent to take on a given value after performing an action. *Successor state* axioms address the well known frame problem by giving necessary and sufficient conditions.

The subject of this paper is how we can express, within the situation calculus, what an agent knows about its world. This is useful because it allows us to write axioms predicated not only on the state of the world, but also on an agent's knowledge of its world. For example, the following precondition axiom states that it is possible to call a person x if and only if we know their telephone number and we have a quarter, $Poss(call(x), s) \Leftrightarrow \exists y Knows(\mathsf{telNum}(x) = y, s) \wedge \mathsf{HaveQuarter}(s)$. Thus if an agents knowledge is deficient it can attempt to formulate a plan to aquire the necessary knowledge.

Actions that affect an agent's knowledge of its world are known as *knowledge producing actions*. This is because they do not affect the state of the world, but rather only the agent's knowledge of its world.

Previous Work

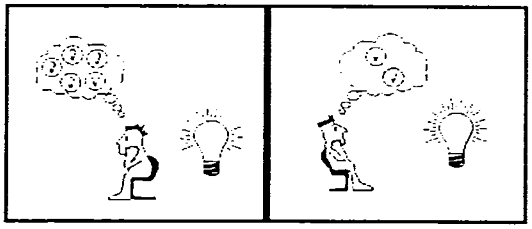

Figure 1: After sensing, only worlds where the light is on are possible.

Knowledge was first added to the situation calulus by Moore (Moore 1985) using the notion of possible worlds. In (Scherl & Levesque 1993), the technique was extended to incorporate Reiter's solution to the frame problem (Reiter 1991). The idea behind the possible worlds approach is to define an epistemic fluent to keep track of all the worlds an agent thinks it might possibly be in. The scenario is depicted graphically in figure 1. Initially the agent is unable to decide which world it is in. That is, whether in its world the light was on or off. The agent then turns around to see that the light is in fact turned on. The result of this sensing action is shown in the figure as the agent discarding some of the worlds it previously thought were possible. In particular, since it now knows that the light is on in its world, it must throw out all the worlds in which it thought the light was turned off.

An epistemic fluent

The way an agent keeps track of the possible worlds or, as the case may be, possible situations is to define an epistemic fluent K. The fluent keeps track of all the K-related worlds. These K-related worlds are precisely the ones in the bubbles above the agents head in the figure. They are the situations that the agent thinks might be its current situation. So we write $\mathsf{K}(s', s)$ to mean that in situation s, as far as the agent can tell, it might be in the alternative situation s'. That is, the agent's knowledge is such that s and s' are indistinguishable. It can only find out which situation it is actually in by sensing the value of certain terms, for example terms such as $\mathsf{light}(s)$.

When we say an agent *knows* the value of a term τ, in a situation s, is some constant c, we mean that τ has the value c in all the K-related worlds. For convenience, we introduce the following abbreviation:

$$\mathit{Knows}(\tau = c, s) \triangleq \forall s' \ \mathsf{K}(s', s) \Rightarrow \tau[s'] = c, \qquad (1)$$

where $\tau[s']$ is the term τ with the situation arguments appropriately inserted.

When an agent knows the value of a term, but we do not necessarily know the value of the term, we use the notation $\mathit{Kref}(\tau, s)$ to say that the agent *knows the referent* of τ:

$$\mathit{Kref}(\tau, s) \triangleq \exists z \ \mathit{Knows}(\tau = z, s). \qquad (2)$$

We now introduce some special notation for the case when τ takes on values in $\mathbb{B}$.[1] In particular, since there are only two possibilities for the referent, we say we *know whether* τ is true or not:

$$\mathit{Kwhether}(\tau, s) \triangleq \mathit{Knows}(\tau = \top, s) \vee \mathit{Knows}(\tau = \bot, s). \qquad (3)$$

Sensing

As in (Scherl & Levesque 1993), we shall make the simplifying assumption that for each term τ, whose value we are interested in sensing, we have a corresponding knowledge producing action sense_τ. In general, if there are n knowledge producing actions: sense_{τ_i}, $i = 0, \ldots, n-1$, then we shall assume there are n associated situation dependent terms: $\tau_0, \ldots, \tau_{n-1}$. The corresponding successor state axiom for K is then:

$$\begin{aligned}
\mathit{Poss}(a, s) \Rightarrow [&\mathsf{K}(s'', do(a, s)) \Leftrightarrow \\
&(\exists s')(\mathsf{K}(s', s) \wedge (s'' = do(a, s'))) \wedge \\
&((a \neq \mathit{sense}_{\tau_0} \wedge \cdots \wedge a \neq \mathit{sense}_{\tau_{n-1}}) \\
&\vee (a = \mathit{sense}_{\tau_0} \wedge \tau_0(s') = \tau_0(s)) \\
&\vdots \\
&\vee (a = \mathit{sense}_{\tau_{n-1}} \wedge \tau_{n-1}(s') = \tau_{n-1}(s)))]. \qquad (4)
\end{aligned}$$

The above successor state axiom captures the required notion of sensing and solves the frame problem for knowledge producing actions. We shall explain how it works through a simple example. In particular, let us consider the problem of sensing the current temperature. Firstly, we introduce a fluent $\mathsf{temp} : \text{SITUATION} \longrightarrow \mathbb{R}^+$, that corresponds to the temperature (in Kelvin) in the current situation. For now let us assume that the temperature remains constant: $\mathit{Poss}(a, s) \Rightarrow \mathsf{temp}(do(a, s)) = \mathsf{temp}(s)$. We will have a single knowledge producing action $\mathit{senseTemp}$. This gives us the following successor-state axiom for K: $\mathit{Poss}(a, s) \Rightarrow [\mathsf{K}(s'', do(a, s)) \Leftrightarrow (\exists s')(\mathsf{K}(s', s) \wedge (s'' = do(a, s'))) \wedge ((a \neq \mathit{senseTemp}) \vee (a = \mathit{senseTemp} \wedge \mathsf{temp}(s') = \mathsf{temp}(s)))]$.

The above axiom states that for any action other, than $\mathit{senseTemp}$, the set of K-related worlds is the set of images of the previous set of K-related worlds. That is, if s' was K-related to s, then the image $s'' = do(a, s')$, of s' after performing the action a is K-related to $do(a, s)$. Moreover, when the agent performs a $\mathit{senseTemp}$ action, in some situation s, the effect is to restrict the

[1] $\mathbb{B}$ just denotes the "Boolean numbers", consisting of the "numbers" $\bot, \top$ and all the usual connectives associated with Boolean algebras. To avoid introducing special notation to state all our results twice, we shall just view relational fluents as functional fluents that take on values in $\mathbb{B}$. We adopt the usual convention that $\mathsf{Foo}(s)$, and $\neg\mathsf{Foo}(s)$ are, respectively, just shorthand for $\mathsf{foo}(s) = \top$ and $\mathsf{foo}(s) = \bot$.

set of K-related worlds to those in which the temperature agrees with the temperature in the situation s. In other words, *senseTemp* is the only knowledge producing action, and its effect is to make the temperature denotation known: $\mathit{Kref}(\mathsf{temp}, \mathit{do}(\mathit{senseTemp}, s))$. The reader is referred to (Scherl & Levesque 1993) for any additional details, examples or theorems on any of the above.

Discussion

The formalization of knowledge within the situation calculus using the epistemic fluent K makes for an elegant mathematical specification language. It is also powerful. For example, suppose we have an effect axiom that states that if a gun is loaded then the agent is dead after shooting the gun: $\mathsf{Loaded}(s) \Rightarrow \mathsf{Dead}(\mathit{do}(\mathit{shoot}, s))$. Furthermore, suppose that we know the gun is initially loaded $\mathit{Knows}(\mathsf{Loaded}, \mathsf{s}_0)$, then we can infer that we know the agent is dead after shooting the gun $\mathit{Knows}(\mathsf{Dead}(\mathit{do}(\mathit{shoot}, \mathsf{s}_0))$.

There are a number of problems that make epistemic reasoning hard. Notably, knowledge by one agent of the knowledge of another; knowledge of an agent of her future knowledge; knowledge by an agent of other times; or knowledge of intensional terms. We will not be considering any of these problems. Instead we will address some of the problems that arise when we try to *implement* some of the ideas we have already discussed.

Implementation

The implementation problems revolve around how to specify the initial situation. For example, if we choose an implementation language like Prolog, specifying the initial situation may involve having to list out an exponential number of possible worlds. For example, if we do not initially know if the gun is loaded then we might consider explicitly listing the two possible worlds `s_a`, and `s_b`, such that:

```
k(s_a,s0).
k(s_b,s0).
loaded(s_a).
```

As we add more relational fluents, that we want to be able to refer to our knowledge of, the situation gets worse. In general, if we have n such fluents, there will be 2^n initial possible worlds that we have to list out. Once we start using functional fluents, however, things get even worse: we cannot, by definition, list out the uncountably many possible worlds associated with not knowing the value of a fluent that takes on values in $\mathbb{R}$.

Intuitively, we need to be able to specify rules that characterize, without having to list them all out, the set of initial possible worlds. It may be possible to somehow coerce Prolog into such an achievement. Perhaps, more reasonably, we could consider using a full first-order logic theorem prover. However, first-order logic theorem provers are often inefficient and experimental.

Ignoring all the above concerns let us assume that we can specify rules that characterize the set of initial possible worlds. For example, suppose that initially we know the temperature is between 10 and 50 Kelvin. We might express this using inequalities: $\forall s'\ \mathsf{K}(s', \mathsf{s}_0) \Rightarrow 10 \leqslant \mathsf{temp}(s') \leqslant 50$. This, however, brings us to our second set of problems related to reasoning about real numbers.

Real numbers

We just wrote down the formula that corresponds to: $\mathit{Knows}(10 \leqslant \mathsf{temp} \leqslant 50, \mathsf{s}_0)$. Suppose, we are now interested in what this tells us about what we know about the value of the temperature squared. In general, if we know a term τ lies in the range $[u, v]$ we would like to be able to answer questions about what we know about some arbitrary function f of τ. Such questions take us into a mathematical minefield of reasoning about inequalities. Fortunately, a path through this minefield has already been charted by the field of interval arithmetic.

Interval arithmetic

To address the issues we raised above we turn our attention to interval arithmetic (Moore 1966; Tupper January 1996). Some of the immediate advantages interval arithmetic affords us are listed below:

- Interval arithmetic enables us to move all the details of reasoning about inequalities into the rules for combining intervals under various mathematical operations.
- Interval arithmetic provides a finite (and succinct) way to represent uncertainty about a large, possibly uncountable, set of alternatives. Moreover, the representation remains finite after performing a series of operations of the intervals. In (Pesonen & Hyvonen 1995) interval arithmetic is compared to probability as a means of representing uncertainty.
- Writing a sound oracle for answering ground queries about interval arithmetic is a trivial task. Moreover, we can answer queries in time that is linear in the length of the query. Returning valid and optimal intervals is more challenging (see below). This should, however, be compared to the vastly unrealistic assumption people often make about the existence of oracles for answering queries about the real numbers.
- There is no discrepancy between the underlying theory of interval arithmetic, and the corresponding implementation.

We construct interval arithmetics from regular number systems (e.g. the real numbers $\mathbb{R}$, the integers $\mathbb{Z}$, etc.) as follows:

- For any number system $\mathbb{X}$, we add a new number system sort $\mathcal{I}_{\mathbb{X}}$. The constants of $\mathcal{I}_{\mathbb{X}}$ are the set of pairs $\langle u, v\rangle$ such that $u, v \in \mathbb{X}$ and $u \leqslant v$. There are functions and predicates corresponding to all the functions and predicates of $\mathbb{X}$.
- For an interval $\mathbf{x} = \langle u, v\rangle$, we use the notation $\underline{\mathbf{x}} = u$ for the lower bound, and $\overline{\mathbf{x}} = v$ for the upper bound.

- The function width, returns the width of an interval x, i.e. $\mathsf{width}(x) = \overline{x} - \underline{x}$.
- When we have a number x and an interval $x = \langle u, v \rangle$, such that $u \leqslant x \leqslant v$ we say that x contains x, we write $x \in x$. Similarly for two intervals x, y such that $\underline{y} \leqslant \underline{x}$ and $\overline{x} \leqslant \overline{y}$, we say that y contains x, we write $x \subseteq y$.
- For two intervals x_0, x_1 we say that $x_0 \leq x_1$ if and only if $\overline{x}_0 \leqslant \underline{x}_1$.
- We let $\bot$ and $\top$ represent, respectively, the minimum and maximum elements of the number system in question. For example, in $\mathbb{R}^*$, $\langle \bot, \top \rangle = \langle -\infty, \infty \rangle$.

As a simple example, consider the case of the number system $\mathcal{I}_\mathbf{B}$. There are three numbers in the number system: $\langle \bot, \bot \rangle$, $\langle \bot, \top \rangle$ and $\langle \top, \top \rangle$. Note that we have $\langle \bot, \bot \rangle \leqslant \langle \bot, \top \rangle \leqslant \langle \top, \top \rangle$, $\langle \bot, \bot \rangle \subset \langle \bot, \top \rangle$, and $\langle \top, \top \rangle \subset \langle \bot, \top \rangle$. In $\mathbb{B}$, $\top$ and $\bot$ can be used to represent, respectively, "true" and "false". Similarly, $\langle \top, \top \rangle$, $\langle \bot, \top \rangle$ and $\langle \bot, \bot \rangle$ in $\mathcal{I}_\mathbf{B}$ can be used to represent, respectively, "known to be true", "unknown", and "known to be false". We thus get what amounts to a *three-valued logic* which, by way of example, we develop further in section .

By way of analogy, complex numbers are also made up of a pair of (real) numbers, and operations on them are defined in terms of operations on the reals. However, it would lead to confusion, if when reading a text on complex analysis we could not comprehend complex numbers as a separate entity, distinct from pairs of real numbers. We therefore forewarn the reader against making the same mistake for intervals. That is, although numbers in $\mathcal{I}_\mathbf{X}$ are made up of a pair of numbers from $\mathbb{X}$ it is important to treat them as "first-class" numbers in their own right.

Interval-valued fluents

The epistemic K-fluent that we discussed previously allowed us to express an agent's uncertainty about the value of a fluent in its world. Unfortunately, as we explained above, we saw there were implementation problems associated with trying to represent an agent's knowledge of the initial situation. Fortunately, in the previous section we saw that intervals also allow us to express uncertainty about a quantity. Moreover, they allow us to do so syntactically, and in a way that circumvents the problem of how to represent infinite quantities with a finite number of bits. It is, therefore, natural to ask whether we can also use intervals to replace the troublesome epistemic K-fluent.

The answer, as we shall seek to demonstrate in the remainder of this paper, is a resounding "yes". In particular, we shall introduce new epistemic fluents that will be interval-valued. They will be used to represent an agent's uncertainty about the value of certain non-epistemic fluents.

We have previously used functional fluents that take on values in any of the number systems: $\mathbb{B}$, $\mathbb{R}$, etc. There is nothing noteworthy about now allowing fluents that take on values in any of the interval numbers systems: $\mathcal{I}_\mathbb{B}$, $\mathcal{I}_\mathbb{R}$. Firstly, let us distinguish those regular fluents whose value maybe learned through a knowledge-producing action. We term such fluents *sensory fluents.* Now, for each sensory fluent f, we introduce a new corresponding interval-valued epistemic (IVE) fluent $\mathcal{I}_f$.

For example, we can introduce an IVE fluent $\mathcal{I}_{\mathsf{temp}}$: SITUATION $\rightarrow \mathcal{I}_{\mathbb{R}^{*+}}$. We can now use the interval $\mathcal{I}_{\mathsf{temp}}(\mathrm{S}_0) = \langle 10, 50 \rangle$ to state that the temperature is initially between 10 and 50 Kelvin. Similarly, we can even specify that the temperature is initially completely unknown: $\mathcal{I}_{\mathsf{temp}}(\mathrm{S}_0) = \langle 0, \infty \rangle$.

Our ultimate aim is that in an implementation we can use IVE fluents to completely replace the troublesome K-fluent. Nevertheless, within our mathematical theory, there is nothing to prevent our IVE fluents co-existing with our previous sole epistemic K-fluent. Indeed, if we define everything correctly then there are many important relationships that should hold between the two. These relationships take the form of state constraints and, as we shall show, can be used to express the notion of validity and optimality of our IVE fluents. If these state constraints are maintained as actions are performed then the IVE fluents completely subsume the troublesome K-fluent. This will turn out to be true until we consider knowledge of general terms. In which case we can maintain validity but may have to sacrifice our original notion of optimality (see below).

Seeking to make IVE fluent ubiquitous necessitates an alternative definition for *Knows* that does not mention the K-fluent. To this end, we introduce a new abbreviation, $\mathcal{I}_{Knows}$ such that for any term τ, $\mathcal{I}_{Knows}(\tau, s) = \langle u, v \rangle$ means that τ's *interval value* is $\langle u, v \rangle$. By "interval value" we mean the value we get by evaluating the expression according the set of rules that we shall discuss below. For now, let us just consider the case when τ is some fluent f. When f is a sensory fluent then $\mathcal{I}_{Knows}$ is the value of the corresponding IVE fluent, otherwise it is completely unknown:

$$\mathcal{I}_{Knows}(f, s) = \begin{cases} \mathcal{I}_f(s) & \text{if } f \text{ is a sensory fluent,} \\ \langle \bot, \top \rangle & \text{otherwise.} \end{cases} \quad (5)$$

We now take the important step of redefining *Knows* to be the special case when $\mathcal{I}_{Knows}(\tau, s)$ has collapsed to a constant interval:

$$Knows'(\tau = c, s) \Leftrightarrow \mathcal{I}_{Knows}(\tau, s) = \langle c, c \rangle. \quad (6)$$

The definitions of *Kref*, and *Kwhether* are now in terms of the new definition for *Knows'*. As required, this new definition does not involve the problematic epistemic K-fluent.

We are now in a position to define what it means for an IVE fluent to be valid:

Definition 0.1 (Validity). *For every sensory fluent f, we say that the corresponding IVE fluent $\mathcal{I}_f$ is a* valid *interval if f's value in all of the* K*-related situations is contained within it:*

$$(\forall s, s')\ \mathsf{K}(s', s) \Rightarrow f(s') \in \mathcal{I}_f(s).$$

Note that since we have a logic of knowledge (as opposed to belief) we have that every situation is K-related to itself: $(\forall s)\ \mathsf{K}(s,s)$. Thus, as an immediate consequence of definition 0.1, we have that if an IVE fluent $\mathcal{I}_f$ is valid then it contains the value of f: $(\forall s)\ f(s) \in \mathcal{I}_f(s)$.

The validity criterion is a state constraint that ensures the interval value of the IVE fluents is wide enough to contain all the possible values of the sensory fluents. It does not however prevent intervals from being excessively wide. For example, the interval $\langle -\infty, \infty \rangle$ is a valid interval for any IVE fluent that takes on values in $\mathcal{I}_{\mathbb{R}^*}$. The notion of narrow intervals is captured in the definition of optimality:

Definition 0.2 (Optimality). *A valid IVE fluent $\mathcal{I}_f$ is also* optimal *if it is the smallest valid interval:*

$$(\forall\ y, s, s')\mathsf{K}(s', s) \Rightarrow (f(s') \in y \Rightarrow \mathcal{I}_f(s) \subseteq y).$$

Correctness

In this section we shall consider some of the consequences and applications of interval-valued fluents to formalizing sensing under various different assumptions. Our goal will be to show that we can maintain valid and optimal intervals as the agent performs actions. This leads to the soundness and completeness result given at the end of the section. Please note that in order to promote brevity, proofs of the theorems given below are omitted. However, the interested reader may find all the required proofs, given in full, in (Funge 1998).

The first step will be to define successor state axioms for IVE fluents. This is done in much the same way as it was for regular fluents. For example, suppose we have a perfect sensor, then the following successor-state axiom states that after sensing, we "know" the temperature in the resulting situation $\mathit{Poss}(a,s) \Rightarrow [\mathcal{I}_{\mathsf{temp}}(\mathit{do}(a,s)) = y \Leftrightarrow (a = \mathit{senseTemp} \wedge \overline{y} = \underline{y} = \mathsf{temp}(s)) \vee (a \neq \mathit{senseTemp} \wedge \mathcal{I}_{\mathsf{temp}}(s) = y)]$.

Now let us consider the case in general. Firstly, we note that there is always an initial valid IVE fluent.

Lemma 0.1. *For any initial situation s_0 and sensory fluent f we have that $\mathcal{I}_f = \langle \bot, \top \rangle$ is a valid interval.*

It is also the case that there will usually be an initial optimal interval.

Lemma 0.2. *If the initial set of* K*-related situations is either completely unspecified or specified with inequalities then we can find an initial optimal IVE fluent for each of the sensory fluents.*

Unless otherwise stated, we make the following assumptions about all sensory fluents f: Each fluent is independent (no changes to one affect either the value or the knowledge about the value of another); The value of $\mathcal{I}_f$, in the initial situation, is optimal and valid (justified by lemma 0.1); The successor-state axiom for f is such that f remains constant: $\mathit{Poss}(a,s) \Rightarrow [f(\mathit{do}(a,s)) = f(s)]$; The successor-state axioms for each of the corresponding IVE fluents $\mathcal{I}_f$ are of the form:

$$\mathit{Poss}(a,s) \Rightarrow [\mathcal{I}_f(\mathit{do}(a,s)) = y \Leftrightarrow$$
$$(a = \mathit{sense}_f \wedge \overline{y} = \underline{y} = \mathsf{f}(s)) \vee (a \neq \mathit{sense}_f \wedge \mathcal{I}_f(s) = y)].$$

We can now state our main correctness result.

Theorem 0.1. *With the above assumptions, for all situations s, and sensory fluents f, every IVE fluent $\mathcal{I}_f$ is valid and optimal.*

Sketch proof The proof is by induction on $s^* = \mathit{do}(a,s)$. The base case follows directly from the assumptions. For the inductive case we seek to prove that $(\forall s'')\mathsf{K}(s'', s^*) \Rightarrow f(s'') \in \mathcal{I}_f(s^*)$ and that $\mathcal{I}_f(s^*)$ is optimal. There are two cases to consider. The less interesting case is when $a \neq \mathit{sense}_f$ in which case the successor state axioms for K and $\mathcal{I}_f$ ensure the result follows by induction. When $a = \mathit{sense}_f$ the successor state axiom for $\mathcal{I}_f$ gives us that $\mathcal{I}_f(\mathit{do}(\mathit{sense}_f, s)) = \langle f(s), f(s) \rangle$. Meanwhile the successor state axiom for K ensures that $s'' = \mathit{do}(a, s') \wedge \mathsf{K}(s', s) \wedge f(s') = f(s)$. The result follows easily from these two facts and that $\langle f(s), f(s) \rangle$ is a thin interval. See (Funge 1998) for the details.

As a corollary we have that the definition of *Knows* given in equation 1 is equivalent to the one given in equation 6. Under the current set of assumptions this establishes the soundness and completeness of our approach with respect to the previous possible worlds approach (Moore 1985; Scherl & Levesque 1993).

Corollary 0.1. *For any sensory fluent f we have that:*

$$\mathit{Knows}(f = c, s) \Leftrightarrow \mathit{Knows}'(f = c, s).$$

The proof is straightforward from the definitions and theorem 0.1.

In (Scherl & Levesque 1993) a number of correctness results are proven for *Knows*. The above equivalence means that under the current set of assumptions the correctness results carry over for *Knows'*.

Operators for interval arithmetic

One of our original motivations, listed above, for introducing intervals was the promise of being able to conveniently calculate what we know about a term from our knowledge of its subcomponents. For example, suppose in a situation s we know the value of a fluent $f(s)$, what do we know about $(f(s))^2$?

The answer to this question leads us to the large and active research area of interval arithmetic. The fundamental principle used is that interval versions of a given function should be guaranteed to bound all possible values of the non-interval version. For example, let us consider a function $\phi : \mathbb{R} \rightarrow \mathbb{R}$. The interval version of this function is $\mathcal{I}_\phi : \mathcal{I}_{\mathbb{R}} \rightarrow \mathcal{I}_{\mathbb{R}}$. The result of applying $\mathcal{I}_\phi$ to some interval $\boldsymbol{x}$ is another interval $\boldsymbol{y} = \mathcal{I}_\phi(\boldsymbol{x})$. We say that the $\boldsymbol{y}$ is a *valid* interval if for every point $x \in \boldsymbol{x}$, we have that $\phi(x) \in \boldsymbol{y}$. Note also that for any valid interval $\boldsymbol{y}$, if $\boldsymbol{y} \subseteq \boldsymbol{y}'$ then, $\boldsymbol{y}'$ is also a valid interval. If, for every interval $\boldsymbol{x}$, $\mathcal{I}_\phi(\boldsymbol{x})$ gives a valid interval then we say that $\mathcal{I}_\phi$ is a *sound interval version* of ϕ.

As we might expect from our previous discussions defining a sound interval version of any function is trivial. In particular, we just let the interval version return the maximal interval of the relevant number system. For example, the function that, for any argument, returns $\langle -\infty, \infty \rangle$ is a sound interval version of any function $\phi : \mathbb{R} \to \mathbb{R}$.

Hence, we see that once again we also need to be concerned about returning intervals that are as narrow as possible. The *optimal interval version* of a function ϕ is thus defined to be the *sound interval version* that, for every argument, returns the smallest valid interval. Unfortunately, for most interesting functions, no such interval versions are known to exist. There are three basic approaches that have been found to address this shortcoming:

Special Forms Consider the expression $t + (50 - t)$. If we naïvely evaluate this expression for the interval $\langle 0, 50 \rangle$ we get back the interval $\langle 0, 100 \rangle$. It is clear, however, that the expression simplifies to 50 and the optimal interval is thus $\langle 50, 50 \rangle$. Therefore, researchers have looked at various standard forms for expressions in an attempt to give better results when evaluating the expression using intervals. In general, however, not only is there no known optimal form but there is also no known single form that is always guaranteed to give the best result. The closest researchers have been able to do so far is the so called "centered forms" (Alefeld & Herzberger 1983).

Subdivision The standard tool in the interval arithmetic arsenal is subdivision. Suppose we have an interval x and we evaluate $\mathcal{I}_\phi(x)$ to give us an interval that is too wide. Then we subdivide x into x_l and x_r such that $x = x_l \cup x_r$. We then evaluate each half separately in the hope that $\mathcal{I}_\phi(x_l) \cup \mathcal{I}_\phi(x_r) \subset \mathcal{I}_\phi(x)$. In practice this usually works well although in theory the functions can be noncomputable in which case any hopes of refining our intervals vanish.

Linear intervals The final approach we mention is a new approach that was recently invented by Jeffrey Tupper (Tupper January 1996). The idea is that instead of using constants to bound an interval we use linear functions. Thus for linear expressions, such as $t + (50 - t)$, we can define operators that are guaranteed to return optimal intervals. Of course, we can then recreate similar problems by considering quadratic expressions but Tupper also shows how we can generalize interval arithmetic all the way up to intervals that use general Turing machines as bounds!

Knowledge of terms

Previously, we introduced the abbreviation $\mathcal{I}_{Knows}$. In equation 5 we defined $\mathcal{I}_{Knows}$ for fluents and in what follows we shall show how to define $\mathcal{I}_{Knows}$ for general terms. We begin by stating what it means for our definitions to be valid.

Definition 0.3 (Validity for terms). *For every term τ, we say that the corresponding interval value of the term given by $\mathcal{I}_{Knows}(\tau, s)$ is a* valid *interval if τ's value in all of the* K*-related situations is contained within it:*

$$(\forall s, s')\ \mathsf{K}(s', s) \Rightarrow \tau[s'] \in \mathcal{I}_{Knows}(\tau, s).$$

Fortunately, the general notion of soundness for interval arithmetic carries over into our notion of validity for a $\mathcal{I}_{Knows}$.

Theorem 0.2. *Suppose $\mathcal{I}_\phi$ is a sound interval version of an n-ary function $\phi : \mathbb{X}^n \to \mathbb{X}$. Furthermore, let $x_0, \ldots, x_{n-1} \in \mathcal{I}_\mathbb{X}$ be, respectively, valid intervals for $\mathcal{I}_{Knows}(\tau_0, s), \ldots, \mathcal{I}_{Knows}(\tau_{n-1}, s)$. Then, $\mathcal{I}_\phi(x_0, \ldots, x_{n-1})$ is a valid interval for $\mathcal{I}_{Knows}(\phi(\tau_0, \ldots, \tau_{n-1}), s)$.*

The important consequence of this theorem is that our definition on $\mathcal{I}_{Knows}$ for terms can stand upon the shoulders of previous work in interval arithmetic. That is, we can define $\mathcal{I}_{Knows}$ recursively in terms of sound interval versions of functions. Assuming the same assumptions as in theorem 0.2 we have that

$$\mathcal{I}_{Knows}(\phi(\tau_0, \ldots, \tau_{n-1}), s) = \mathcal{I}_\phi(x_0, \ldots, x_{n-1}).$$

Note that some of our functions may be written using *infix* notation, in which case we may refer to them as *operators*. The important aspect of this definition is that we do not have to redesign a plethora of operators for interval arithmetic and prove each of them sound. In the previous section we noted the difficulties associated with defining optimal versions of operators. We also noted that there are a number of ways to deal with the problem. Each of the methods we outlines maintains validity and is thus appropriate for us to use. Which particular method we choose to narrow our intervals can be thought of as an implementation issue for our approach.

Usefulness

For a long list of useful operators for interval arithmetic the reader could do no better than to consult (Tupper January 1996). By way of example, however, we shall list some useful operators for $\mathcal{I}_\mathbf{B}$.

Interval versions of operators, and relations, are given in bold. Elsewhere, we rely on context to imply the intended meaning.

Definition 0.4 (Operators for $\mathcal{I}_{\mathbb{B}}$).

$$\begin{aligned}
\tau = \langle u, v\rangle &\Leftrightarrow \neg\tau = \langle \neg v, \neg u\rangle \\
\tau_0 = \langle u_0, v_0\rangle \wedge \tau_1 = \langle u_1, v_1\rangle &\Rightarrow \tau_0 \wedge \tau_1 \subseteq \\
& \quad \langle u_0 \wedge u_1, v_0 \wedge v_1\rangle \\
\tau_0 \wedge \tau_1 = \langle u, v\rangle &\Rightarrow \tau_0 \subseteq \langle u, \top\rangle \\
\tau_0 = \langle u_0, v_0\rangle \wedge \tau_1 = \langle u_1, v_1\rangle &\Rightarrow \tau_0 \vee \tau_1 \subseteq \\
& \quad \langle u_0 \vee u_1, v_0 \vee v_1\rangle \\
\tau_0 \vee \tau_1 = \langle u, v\rangle &\Rightarrow \tau_0 \subseteq \langle \bot, v\rangle \\
\exists x \tau(x) = \langle u, v\rangle &\Rightarrow \tau(c) \subseteq \langle \bot, v\rangle, \\
& \quad \textit{for any constant } c \\
\textit{For some constant } c, & \\
\tau(c) = \langle u, v\rangle &\Rightarrow \exists x \tau(x) \subseteq \langle u, \top\rangle \\
\forall x \tau(x) = \langle u, v\rangle &\Rightarrow \tau(c) \subseteq \langle u, \top\rangle, \\
& \quad \textit{for any constant } c \\
\textit{For some constant } c, & \\
\tau(c) = \langle u, v\rangle &\Rightarrow \forall x \tau(x) \subseteq \langle \bot, v\rangle
\end{aligned}$$

These definitions enable us to evaluate $\mathcal{I}_{\mathit{Knows}}$ for terms taking on values in $\mathbb{B}$. Notice however that most of the definitions are in terms of $\subseteq$. This is because we can, in general, only guarantee valid results, not optimal ones. For example, if we assume $\tau = \langle \bot, \top\rangle$ then we get that $\tau \vee \neg\tau \subseteq \langle \bot, \top\rangle$. While this is valid, it is clearly not optimal. Since there are only two numbers in $\mathbb{B}$ we can subdivide to perform an exhaustive search for the optimal value. That is, let $\tau = \tau_0 \cup \tau_1$, where $\tau_0 = \langle \bot, \bot\rangle$, and $\tau_1 = \langle 1, 1\rangle$. Now we get that $\tau_0 \vee \neg\tau_0 = \langle \top, \top\rangle$, and $\tau_1 \vee \neg\tau_1 = \langle \top, \top\rangle$. With more variables the exhaustive search approach has worst case exponential complexity. In general it may be observed that if each variable occurs only once in an expression then evaluating it will yield an optimal result. Also if we start with thin intervals then we will also get an optimal result. Finally, for a propositional formula in Blake canonical form (Blake 1938) evaluation with intervals *always* yields an optimal result (H. 1997). Moreover, all propositional formulas can be converted to this form. Thus we can evaluate propositional formulas in linear time and get optimal results. The catch is that converting propositional formulas to Blake canonical form is NP-hard.

Therefore, as we should expect, intervals do not provide us with a means to magically circumvent complexity problems. What they do provide, however, is the ability to track our progress in solving a problem. For the majority of real world problems, where exact knowledge is not imperative, this will often allow us to stop early once we have a "narrow enough" interval. At the very least we can give up early if convergence is too slow. This should be contrasted to other methods of evaluating expressions where we can never be sure whether the method is completely stuck, or is just about to return the solution.

Let us now consider some more examples in which our interval arithmetic approach can be shown to be useful and correct. Since *Knows'* is correct with respect to *Knows* we know that we can not derive any false relationships using *Knows'*. The important point is that many true and useful relationships hold. In particular, we have that the following can all be shown to hold:

$$\begin{aligned}
\mathit{Knows'}(P, s) \vee \mathit{Knows'}(Q, s) &\Rightarrow \mathit{Knows'}(P \vee Q, s) \\
\mathit{Knows'}(P \vee Q, s) &\not\Rightarrow \mathit{Knows'}(P, s) \vee \\
& \qquad \mathit{Knows'}(Q, s) \\
\mathit{Knows'}(\neg P, s) &\Rightarrow \neg\mathit{Knows'}(P, s) \\
\neg\mathit{Knows'}(P, s) &\not\Rightarrow \mathit{Knows'}(\neg P, s) \\
\exists x\ \mathit{Knows'}(P(x), s) &\Rightarrow \mathit{Knows'}(\exists x\ P(x), s) \\
\mathit{Knows'}(\exists x\ P(x), s) &\not\Rightarrow \exists x\ \mathit{Knows'}(P(x), s)
\end{aligned}$$

The proofs all follow by simple application of the rules given in definition 0.4 and can be found in (Funge 1998).

Finally, in our discussion of the possible worlds approach, we saw that we could make deductions based on *modus ponens.* Fortunately, we can perform similar reasoning with intervals.

Theorem 0.3. *Let τ_0 and τ_1 be terms for that take on values in $\mathbb{B}$, such that $\langle u, v\rangle$ is an optimal value for $\mathcal{I}_{\mathit{Knows}}(\tau_0, s)$, and $\tau_0[s] \Rightarrow \tau_1[s]$. Then, $\mathcal{I}_{\mathit{Knows}}(\tau_1, s) \subseteq \langle u, \top\rangle$.*

The proof follows from basic properties of intervals and the observation that $\tau_0[s] \leqslant \tau_1[s]$. Once again see (Funge 1998) details.

Inaccurate Sensors

In (Bacchus, Halpern, & Levesque 1995), the K-fluent approach is extended to handle noisy sensors. By redefining *Knows* we can also easily extend our approach to allow for inaccurate sensors. We may say that we know a fluent's value to within some Δ, if the width of the interval is less than twice Δ:

$$\mathit{Knows}(\Delta, f = z, s) \triangleq \mathcal{I}_f(s) \subseteq \langle z - \Delta, z + \Delta\rangle. \quad (7)$$

If we have a bound of $\pm\Delta$ on the greatest possible error for the sensor that recorded yesterday's temperature then we can state that the value sensed for the temperature is within $\pm\Delta$ of the actual value:

$$\begin{aligned}
& \mathit{Poss}(a, s) \Rightarrow [\mathcal{I}_{\mathsf{temp}}(\mathit{do}(a, s)) = \langle u, v\rangle \Leftrightarrow \\
& \quad (a = \mathit{senseTemp} \wedge u = \max(0, \mathsf{temp}(s) - \Delta) \wedge \\
& \qquad v = \min(\mathcal{I}_{\mathsf{temp}}(s) + \Delta, \overline{\mathsf{temp}(s)})) \\
& \qquad \vee (a \neq \mathit{senseTemp} \wedge \mathcal{I}_{\mathsf{temp}}(s) = \langle u, v\rangle)]. \quad (8)
\end{aligned}$$

Sensing Changing Values

Until now, we only considered sensing fluents whose value remains constant. In (Scherl & Levesque 1993) once a fluent becomes known then it stays known. That is, if the value of a known fluent changes then the agent will automatically know the fluents new value. This often counterintuitive. For example, if one has checked the temperature once then it is natural to assume that after a while the information may be out of date. That is, we would expect to have to sense the temperature periodically.

Using the epistemic K-fluent to model information becoming out of date corresponds to adding possible worlds back in. Unfortunately, the K-fluent keeps track of an agent's knowledge of all the sensory fluents all at once. It can therefore be hard to specify exactly which worlds the agent should be adding back into its consideration. In contrast, with intervals there is nothing noteworthy about allowing the particular relevant interval to expand. We must simply ensure that our axioms maintain the state constraint that the interval bounds the actual value of the fluent.

At the extreme we can extend our approach to handle fluents that are constantly changing in unpredictable ways. We can model this with exogenous actions (Giacomo, Lespérance, & Levesque 1997). We assume that the current temperature changes in a completely erratic and unpredictable way, according to some exogenous action *setTemp*. Then, we can write a successor-state axiom for temp that simply states that the temperature is whatever it was set to:

$$\begin{aligned} &Poss(a,s) \Rightarrow \mathsf{temp}(do(a,s)) = z \Leftrightarrow \\ &[(a = setTemp(z)) \vee (a \neq setTemp \wedge \mathsf{temp}(s) = z)]. \end{aligned}$$

We can, also, write a successor state axiom for $\mathcal{I}_{\mathsf{temp}}$. In particular, if we again assume accurate sensors, we can state that the temperature is known after sensing it, otherwise, it is completely unknown:

$$\begin{aligned} &Poss(a,s) \Rightarrow [\mathcal{I}_{\mathsf{temp}}(do(a,s)) = \langle u,v\rangle \Leftrightarrow \\ &\quad (a = senseTemp \wedge u = v = \mathsf{temp}(s)) \vee \\ &\quad\quad (a \neq senseTemp \wedge u = 0 \wedge v = \infty)]. \end{aligned} \quad (9)$$

Note that this definition works because, by definition, $(\forall s)\ \mathsf{temp}(s) \in \langle 0,\infty\rangle$. At first glance it may appear strange that we have, for example, $\mathcal{I}_{\mathsf{temp}}(do(setTemp(2),s)) = \langle 0,\infty\rangle$. Upon reflection, however, the reader will hopefully recall that our intention is to use the IVE fluents to model an agent's knowledge of its world. Therefore, until sensing, the agent rightly remains oblivious as to the effect of the exogenous action *setTemp*. For the fluent that keeps track of the temperature in the virtual world we of course get that $\mathsf{temp}(do(setTemp(2),s)) = 2$.

If we have a bound on the maximum rate of temperature change, per unit time, to be $\Delta\mathsf{temp}$, and we add the ability to track the time to our axiomatization, then we can do a lot better. Suppose we have an action *tick* that occurs once per unit of time. Moreover, we limit exogenous actions to only occurring directly before a tick action. Then we can have a successor-state axiom that states the temperature is known after sensing; or after a period of time it is known to have changed by less than some maximum amount; otherwise it is unchanged:

$$\begin{aligned} &Poss(a,s) \Rightarrow [\mathcal{I}_{\mathsf{temp}}(do(a,s)) = \langle u,v\rangle \Leftrightarrow \\ &\quad (a = senseTemp \wedge u = v = \mathsf{temp}(s)) \vee \\ &\quad (a = tick \wedge (\exists u_p, v_p)\ \mathcal{I}_{\mathsf{temp}}(s) = \langle u_p, v_p\rangle \wedge \\ &\quad u = \max(0, u_p - \Delta\mathsf{temp}) \wedge v = v_p + \Delta\mathsf{temp}) \vee \\ &\quad (a \neq senseTemp \wedge a \neq tick \wedge \mathcal{I}_{\mathsf{temp}}(s) = \langle u,v\rangle]. \end{aligned} \quad (10)$$

This type of axiom can be used to "plan to replan". That is, the degradation in our knowledge level is predicatable and can be used as the basis for a replanning action.

Conclusion

Our original aim was to develop theories of action and knowledge that are practical and usable by programmers. We envisaged using these theories to develop autonomous agents for use in computer animation and games. Thanks, in part, to the work described in this paper we have indeed successfully developed autonomous agents that can reason, act and perceive in changing, incompletely known, unpredictable environments. Some frames from some corresponding animations can be found at `www.cs.toronto.edu/~funge/images.html` . Additional documentation on the work can be found in (Funge 1998).

References

Alefeld, G., and Herzberger, J. 1983. *Introduction to Interval Computations.* Academic Press.

Bacchus, F.; Halpern, J.; and Levesque, H. 1995. Reasoning about noisy sensors in the situation calculus. In *IJCAI-95.*

Blake, A. 1938. *Canonical Expressions in Boolean Algebra.* University of Chicago: PhD thesis.

Funge, J. 1998. *Making Them Behave: Cognitive Models for Computer Animation.* CS Dept, University of Toronto: PhD thesis.

Giacomo, G. D.; Lespérance, Y.; and Levesque, H. 1997. Reasoning about concurrent execution, prioritized interrupts, and exogenous actions in the situation calculus. In *IJCAI-97.*

H. Levesque, University of Toronto. 1997. *Personal communication.*

Moore, R. E. 1966. *Interval Analysis.* Prentice-Hall.

Moore, R. C. 1985. A formal theory of knowledge and action. In *Formal Theories of the Commonsense World.*

Pesonen, J., and Hyvonen, E. 1995. Interval approach challanges monte carlo simulation. In *Scientific Computing, Computer Arithmetic and Validated Numerics.*

Reiter, R. 1991. The frame problem in the situation calculus. In *Artificial Intelligence and Mathematical Theory of Computation.*

Scherl, R., and Levesque, H. 1993. The frame problem and knowledge-producing actions. In *AAAI-93.*

Tupper, J. January 1996. *Graphing Equations with Generalized Interval Arithmetic.* CS Dept, University of Toronto: MSc thesis.