

# Simultaneous Events and the "Once-Only" Effect <sup>1</sup>

Haythem O. ISMAIL

*German University in Cairo  
Department of Computer Science  
haythem.ismail@guc.edu.eg*

**Abstract.** Some events recur, and some happen only once. Galton refers to the latter as "once-only" events [1]. In a first-order logic of events that makes a type-token distinction, the possibility of concurrent occurrences of the same event renders the characterization of the intuitive *once-onliness* not very intuitive. In particular, the paradigmatic case of the  $n^{\text{th}}$  occurrence of a recurring event is shown to be not necessarily once-only. Counter-examples give rise to a classification of events based on the temporal relations among their occurrences. The problematic cases turn out to be those events that involve an indefinite individual; we call these *indefinitely-specified* events. We consider two options. The first is to restrict our event ontology, as has been implicitly done in most logics of events, to events that are definitely-specified. The second is to admit all sorts of events into our ontology and distinguish those that are definitely-specified from those that are not by statements in the object language. We opt for a representation of events as functional terms in the logic, and those terms denoting indefinitely-specified events seem to inevitably contain variables. Such non-ground terms turn out to be semantically problematic. To smoothly resolve these problems, we adopt Shapiro's logic of arbitrary and indefinite objects in which indefinite individuals are denoted by special terms [2]. Thus, indefinitely-specified events are naturally represented by functional terms with at least one argument denoting an indefinite individual.

**Keywords.** Knowledge representation, events, indefinites.

## 1. Introduction

Most events possibly recur. For example, birthday parties, naps, and drives to work are all recurrent events. By their very nature, some events happen only once. As far as we know, Galton [1] was the first to explicitly point out this class of events. For example, someone's first birthday, the splitting of a particular cell, someone's turning 21, or, in general, someone's doing something for the  $n^{\text{th}}$  time, are all events that can happen only once. We shall follow Galton and call these events *once-only* events.

Once-only events are interesting to the logician, particularly a logician concerned with issues of tense and aspect. For example, an idealized conceptualization of the En-

---

<sup>1</sup>To appear as Haythem O. Ismail, Simultaneous Events and the "Once-Only" Effect. In Brandon Bennett and Christiane Fellbaum (eds.), *Formal Ontology in Information Systems: Proceedings of the Fourth International Conference (FOIS 2006)*, IOS Press, Amsterdam, 2006. All citations should be to the published version.

glish perfect is as an operator that yields a permanent state [1,3]. A permanent state is one that, once it starts to hold, will persist forever. The onset of a permanent state is therefore a once-only event, and a logic of aspect should support this inference. But once-only events are not only interesting to the logician, they are also interesting to state-of-the-art information systems. For example, any scheduling system would need to make a clear distinction between recurrent and once-only events.

For a knowledge representation system to reason about once-only events, we need to accomplish two tasks:

**Task 1.** Sharpening our intuitions regarding which events are once-only and which are recurrent.

**Task 2.** Developing a formal system with sufficient expressive and inferential powers to support representing and reasoning about once-onliness.

**Task 2** is a purely logical endeavor, but **Task 1** is an intellectual exercise that possibly falls outside the bounds of logic, but certainly fits within the realm of ontology. A system that accomplishes those tasks has been developed in [1], with a more thorough investigation of semantics, soundness, and completeness in [4].

However, a certain property of the two systems developed by Galton conceals some difficulties with accomplishing our two tasks. In particular, Galton's systems do not allow for simultaneous (or, more broadly, concurrent) occurrences of the same event. These systems are basically propositional tense logics extended with aspect operators and *event radicals*, which are symbols denoting event types. Thus, no event tokens are representable, and the notion of multiple simultaneous occurrences of the same event type does not even arise. In [1], and as far as event occurrence goes, only the fact that an event type occurs at some time (in the past or future) is representable. The system is silent about what that exactly means: does it mean that there is exactly one occurrence of the event at the time, or that there are possibly multiple simultaneous occurrences?<sup>2</sup> In [4], however, the model-theoretic semantics reveals how this is resolved. [4] identifies an event with its set of occurrences. Since Galton's ontology does not include event tokens (i.e., occurrences) as primitives, he identifies these (roughly) with the intervals over which they occur. Thus, the tacit assumption is that there can only be one occurrence of any event at any given time.

In the more recent work reported in [5] (which is not concerned with once-only events), event tokens are indeed first-class entities in the ontology denoted by terms in the logic. However, tokens are identified with pairs  $\langle \eta, e \rangle$ , where  $e$  is an event type and  $\eta$  is an *episode* of that type. (An episode is an interval carrying with it all the history in a branching time structure.) Thus, simultaneous tokens of the same event type are still not distinguished by this system.

In this paper, I investigate problems with accomplishing **Task 1** and **Task 2** in a common sense reasoning system that makes a type-token distinction and allows for simultaneous occurrences of the same event. One question is whether we need to do this, whether there are indeed events with simultaneous occurrences. This issue, together with

---

<sup>2</sup>For some, the preferred meaning may be the second. Asserting that an event occurs at a time could be thought of as an assertion about the existence of an occurrence of that event. The standard semantics of existence assertions would allow multiple occurrences of the event at the said time. But, in that case, [1] fails to correctly account for once-onliness; the best it can do is to account for events with multiple occurrences that are all simultaneous.

problems with **Task 1**, is taken on in Section 2. The rest of the paper is dedicated to different issues in accomplishing **Task 2**.

## 2. Patterns of Occurrence

### 2.1. Two Examples

Consider a T.V. channel correspondent reporting live from a war-torn city and delivering the following piece of troubling news.

- (1) The first explosion occurred at 3 p.m.

Is the correspondent reporting a once-only event? It would seem so, given that the  $n^{\text{th}}$  occurrence of a recurrent event is a typical example of once-onliness.<sup>3</sup> But this is not exactly true. For suppose that two explosions, not one, happen simultaneously, at 3 p.m. sharp. In this case, it is not clear whether the reporter is being sloppy in their choice of words, or whether the first-explosion event is, after all, not once-only. Consider the first alternative, where the reporter is assumed to have misused the phrase “first explosion”. In this case, the first-explosion event is certainly once-only, but it just never took place. In particular, since two explosions happened simultaneously, then neither of them could be considered the first. In fact, all explosions reported to have taken place are pre-ordered, and one might argue that a strict linear order is needed to be able to define the  $n^{\text{th}}$  occurrence. Although this is a reasonable analysis, it is not satisfying. In particular, the notion of a pre-order of events is not very intuitive. People would find it a little counter-intuitive to see how several explosions can occur, while a first-explosion never does. In addition, and if we take this line of thought to an extreme, since neither explosion qualifies as being *the* first-explosion, it would seem that the next explosion to occur would! The other alternative is to say that there are indeed two first-explosions and abandon the idea of the first-explosion being a once-only event. Note that the two alternatives correspond to two ways of interpreting “first explosion”. On one interpretation, a first-explosion is an explosion which is strictly followed by any other possible explosions. On another interpretation, a first-explosion is one which is not strictly preceded by any other explosions. We shall come back to explosions, but consider another example.

A professor asks a deep question in the classroom and announces that the first student to come up with the correct answer will get one bonus point. After this announcement has been made, everyone in the classroom is awaiting (or busily trying to be the agent of) some event. This is the event of someone coming up with the correct answer for the first time. Is this a once-only event? It is not clear. If a particular student pronounces the correct answer, then no later pronouncements of the same answer will earn other students bonus points. Thus, we may judge the event as being once-only. But suppose that *two* students come up with the correct answer simultaneously. Who will get the bonus point? Only one of them? Neither? Or both? If only one, then which one? And what criteria will the professor use to determine which of the two simultaneous events is the first? If neither, then, at the very best, it is just unfair. If both, which is probably the

---

<sup>3</sup>Needless to say, “first explosion” here means the first explosion in the city on the day the report is aired.

common practice in such cases, then, once again, we have to give up the assumption that the  $n^{\text{th}}$  occurrence of a recurrent event is *necessarily* once-only.<sup>4</sup>

## 2.2. *Serials, Races, and Witnesses*

The two examples discussed above reveal one thing: reasoning about once-onliness can be tricky. This is not confined to the case of the  $n^{\text{th}}$  occurrence event, but may also be shown to apply to other cases of once-onliness. This at least complicates **Task 1** as described in Section 1. But let's explore the issue a little further and see whether some structure could be imposed on the problematic patterns of occurrence. The unproblematic pattern is that where the occurrences of an event form a strict linear order. In this case, *no* occurrences of the event are simultaneous. I will call events exhibiting this pattern *serials*.

Another pattern, represented in the above two examples, is that where multiple  $n^{\text{th}}$  occurrences are possible. In this case, *some* occurrences of the event are simultaneous. I will call events that exhibit this pattern *races*. This choice of name is intended to remind us of the prototypical case of a race, where two runners arrive at the end line simultaneously—a situation very similar to that of the second example above.

A third pattern is that where *all* occurrences of the event are necessarily simultaneous. What is an example of an event exhibiting this pattern? Recall the T.V. report on explosions, and consider the reporter uttering (2) instead of (1).

(2) The first explosion was heard at 3 p.m.

The event reported here is the *hearing* of the first explosion. This event has many occurrences, as many as there are people who have heard the explosion. Yet, the event is not recurrent; all occurrences are simultaneous, they all took place at 3 p.m.<sup>5</sup> There are two events involved here: an explosion and a hearing of the explosion. In general, whenever there is a *perceived event* (generally a race) and a *perception event*, all occurrences of the latter are simultaneous. I will use the term *witnesses* to refer to such perception events and any other events exhibiting the same pattern. A particularly relevant example of a witness is the event of the  $n^{\text{th}}$  occurrence of a race.

But serials, races, and witnesses are not all there is. So far we have only considered examples of punctual events—those that could be conceived of as occurring at time points (or atomic intervals). For durative events, two more patterns are possible. Durative occurrences of the same event may overlap in time. Consider the first screening of a movie. If the movie is showing in two movie theaters, then we have two overlapping screenings. They can start (and, hence, end) at the same time, or one of them can start after the first starts but before it ends. In the first case, we have something similar to races, but for durative events. In the second case, things are a little vague. For we might prefer to pick as the first screening the one that starts first. Yet, we may also consider both screenings to be tokens of the first-screening event, adopting the view that a second screening is one that necessarily starts after the first ends. Overlapping can also

---

<sup>4</sup>Another alternative is to think of a single first-correct-answer event whose agent is a plural individual à la [6, for example]. But this analysis will be awkward in our situation, where the two students making up the purported plural individual competitively, rather than collectively, seek the correct answer.

<sup>5</sup>For a physicist, these events are probably not simultaneous on a fine-grained conception of time, due to the speed of sound and the spatial distribution of the different hearers.

take a third form if we consider events whose occurrences are not necessarily of the same duration. In that case, the period of one occurrence could be a proper sub-period of another. (Consider the event of running around a track, where the duration of an occurrence depends on the speed of the runner.) Durative events may also be interleaved. A multi-programmed operating system will concurrently run two processes by interleaving their executions. The  $n^{\text{th}}$  process-running is again not clearly once-only.

Without loss of generality, the rest of the paper will concentrate only on races and witnesses as problematic cases for once-onliness.

### 2.3. *Indefinitely-Specified Events*

Races and witnesses are problematic cases for the once-onliness of the  $n^{\text{th}}$  occurrence of an event. We have considered examples of each of these cases. What is an example of a serial—the unproblematic case? Here is a list:

1. The explosion of a particular bomb.
2. A particular student's coming up with the right answer.
3. A particular person's hearing the explosion of a particular bomb.
4. The screening of a particular movie in a particular movie theater.
5. The running of a particular process by an operating system.

The difference between these examples and the corresponding problematic examples, is that all individuals involved in the above serials are particular, definite individuals. A race or a witness (with respect to once-onliness) may be turned into a serial by fixing (or particularizing) all indefinite individuals it involves. This indefinite-specification of races and witnesses is the source of the difficulties they pose. Each of the simultaneous occurrences of an indefinitely-specified event (a race or a witness) is also an occurrence of a definitely-specified event (a serial). Consequently, it is also the unique occurrence of some once-only event, namely the event of the  $n^{\text{th}}$  occurrence of the serial. (In the case of an explosion, fixing the indefinite directly yields a once-only event—that of a particular bomb exploding.)

The above discussion suggests that the class of definitely-specified events is identical to that of serials. One direction of that identity is evident: a definitely-specified event is a serial. Once all individuals involved in an event are fixed, then, at any time, there can be only one occurrence of the event. The other direction deserves some discussion. Consider the act of driving a car. At any time, there can only be one driver of a particular car. This means that the indefinitely-specified event of someone's driving of a particular car is indeed a serial. Conversely, at any time, only one car may be driven by a particular driver. Thus, the indefinitely-specified event of a particular person's driving of some indefinite car is a serial. Other examples of such indefinitely-specified serials include the reelection of an American president, someone's eating of a particular apple (but not a particular person's eating of some apple) or someone's cutting of a particular rope. It could be argued that more than one person can eat the same apple or cut the same rope. One analysis may hold that these are not examples of serials. Another analysis may allude to plural individuals [6], asserting that multiple agents *collectively* eat the apple or cut the rope.

The above examples show that there are indeed indefinitely-specified serials. But wait. Even such cases have two special features. First, that the above examples of

indefinitely-specified events are serials seems to be only contingent. For one may construct possible worlds in which cars are designed such that the same car may be simultaneously driven by more than one driver, or such that a person may simultaneously drive more than one car, or such that more than one person can assume the American presidency at a given time. This is in contrast to the near-logical necessity of a definitely-specified event's being a serial. Second, one may always reduce the indefiniteness of the above serials to a very special type of indefiniteness. For example, the indefiniteness of the American president is only indirectly a person-indefiniteness, but is primarily temporal. The standard representation of "American president" is as a function of time; once you fix a time, you would definitely identify the reelected president. Fixing the indefinite time of a temporally-indefinite event yields a once-only event with the time of its unique occurrence as part of its specification. Similarly, we may think of the car driver, the apple eater, or the rope cutter as (partial) functions of time.

An important note on terminology is important at this point. A definitely-specified event is one involving only definite individuals. An indefinitely-specified event is one involving at least one non-temporal indefinite. A temporally-indefinite event is one involving only temporal indefinites.

I believe that a temporally-indefinite event is the only possible case of a non-definitely-specified serial. Let  $e(x)$  be an indefinitely-specified event, where  $x$  is the indefinite. (For simplicity, I am assuming a single indefinite.) There are three cases to consider.

1. At any given time, more than one individual satisfies the restrictions on  $x$ . If  $a$  and  $b$  are two such individuals, then possible simultaneous occurrences of  $e(a)$  and  $e(b)$  imply that  $e(x)$  is not a serial.
2. At all times, a single individual satisfies the restrictions on  $x$ . If this is a logical or a domain constraint, then the representation of the event is incorrect, since it should be definitely-specified (for example, using Russell's definite descriptor,  $\iota$ ). Otherwise, we are in a situation similar to case 1.
3. At any given time, a single individual satisfies the restrictions on  $x$ . This is a case of temporal indefiniteness.

We, thus, arrive at the following hypothesis.

- Serial = Definitely-specified  $\cup$  Temporally-indefinite.
- Race  $\cup$  Witness = Indefinitely-specified.

### 3. A Formal System

In this section, we investigate **Task 2**—developing a formal system for reasoning about once-onliness. Given the results of Section 2, we will need this system to distinguish serials, races, and witnesses; and to grant true once-onliness to serials. This system may easily be extended to account for overlapping occurrences, but some work is needed to include interleaved occurrences.

#### 3.1. Semantic and Ontological Commitments

The system is intended to be integrated into the logic of states and events introduced in [7] and described in detail in [3]. This logic has been proposed in the process of

developing an embodied cognitive agent based on the SNePS knowledge representation and reasoning and acting system [8,9,10].

The system to be developed here inherits two features of SNePS-based logics.

1. Terms represent anything that we can think or talk about. SNePS semantics is based on Alexius Meinong's theory of objects (see [8,11]), where terms denote objects of thought of a cognitive agent.
2. Given 1, the ontology includes objects, times, propositions, events and event occurrences, to mention only those entities that are particularly relevant to this paper. In addition, actual occurrences, possible occurrences, and incomplete occurrences are all members of the ontology.

It should be noted that SNePS-based logics are term-logics. In particular, there are no predicate symbols, only functions from tuples to proposition-denoting terms. Nevertheless, to simplify the exposition, I will assume a standard first-order logic. (But see [12] for details.)

As mentioned above, to reason about once-onliness, we need to distinguish serials. This distinction could be made at either the ontological level, or at the logical level. At the ontological level, we can opt for admitting only definitely-specified events into our ontology. This is indeed the implicit assumption of almost all logics of events in which event types are granted the status of first-class individuals.<sup>6</sup> In particular, this is the implicit (but possibly unintended) position taken in [1,4], where all examples of event radicals are definitely-specified. If we adopt this view, then we need not worry about once-onliness. In particular, the  $n^{\text{th}}$  occurrence event will be once-only for all event types.

This position, however, would limit the expressivity of our logic and deprives our ontology of many events that we can think or talk about. In particular, indefinitely-specified events allow us to make general statements about big classes of events. For example, we only need to state the effects of explosions once, for a single indefinitely-specified explosion, not once for *each* definitely-specified explosion of a particular bomb.

### 3.2. The Logic

*Sorts.* We use a sorted first-order logic with equality. The following is a list of the sorts and their intended meanings.

- $\mathcal{I}$ : A sort for individuals. We use the superscript  $i$  to designate variables and 0-ary functions of this sort. When not superscripted,  $x$ ,  $y$ , and  $z$  are variables of this sort. Self-explanatory small caps symbols are constants of sort  $\mathcal{I}$ . Examples include JOHN, MARY, BOMB12 (for a particular bomb), etc.
- $\mathcal{T}$ : A sort for times. We use the superscript  $t$  to designate variables and 0-ary functions of this sort. When not superscripted,  $t$ ,  $t_1$ ,  $t_2$ , ... are variables of this sort.
- $\mathcal{C}$ : A sort for event categories (or event types). We use the superscript  $c$  to designate variables and 0-ary functions of this sort.
- $\mathcal{E}$ : A sort for event tokens (or occurrences). We use the superscript  $e$  to designate variables and 0-ary functions of this sort. When not superscripted,  $e$ ,  $e_1$ ,  $e_2$ , ... are variables of this sort.

---

<sup>6</sup>There are notable exceptions—Process Specification Language [13], for example.

- $\mathcal{N}$ : A sort for natural numbers. Familiar numerals (1, 2, 3, ...) are constants of this sort. When not superscripted, variables  $i$  and  $j$  are of this sort.

*Function and Predicate Symbols.* These symbols denote typed  $n$ -ary functions or predicates ( $n > 0$ ). Here is a list of some domain-independent function and predicate symbols, with their informal semantics.

- $\text{Occ}: \mathcal{N} \times \mathcal{C} \longrightarrow \mathcal{C}$ , where  $\text{Occ}(i, c)$  is the event category of the  $[i]^{\text{th}}$  occurrence of event category  $[c]$ .<sup>7</sup>
- $\text{Cat}$ : a predicate symbol over  $\mathcal{E} \times \mathcal{C}$ , where  $[\text{Cat}(e, c)]$  means that event token  $[e]$  is of category  $[c]$ .
- $\text{Occurs}$ : a predicate symbol over  $\mathcal{E} \times \mathcal{T}$ , where  $[\text{Occurs}(e, t)]$  means that event token  $[e]$  occurs at time  $[t]$ .

In addition, domain-dependent function and predicate symbols will be used as we need to, with their semantics informally specified as we consider different examples.<sup>8</sup>

### 3.3. Representing Once-Onliness

With the formal machinery now at our disposal, consider how we can define serials, races, witnesses, and once-onliness.

In the following schemas,  $c$  denotes a term of sort  $\mathcal{C}$ .

- $\text{Serial}(c) =_{\text{def}} \forall t, e_1, e_2 [[\text{Cat}(e_1, c) \wedge \text{Cat}(e_2, c) \wedge \text{Occurs}(e_1, t) \wedge \text{Occurs}(e_2, t)] \Rightarrow e_1 = e_2]$
- $\text{Witness}(c) =_{\text{def}} \exists t_1 \forall e, t_2 [[\text{Cat}(e, c) \wedge \text{Occurs}(e, t_2)] \Rightarrow t_2 = t_1]$
- $\text{Race}(c) =_{\text{def}} \neg[\text{Serial}(c) \vee \text{Witness}(c)]$ .

Thus, an event category is a serial if no two distinct tokens occur simultaneously. An event category is a witness if there is a unique time ( $t_1$ ) such that, if a token of that event occurs, it will have to occur at that time. The uniqueness of  $t_1$  could be secured by proper axiomatization of "Occurs". A race is just the default, or unmarked, case of an event category.<sup>9</sup>

Let us now turn to once-onliness.

- $\text{OnceOnly}(c) =_{\text{def}} \text{Serial}(c) \wedge \text{Witness}(c)$ .

Why is the above a definition of once-onliness? To see why, note that if  $c$  is a serial, then there is a unique occurrence of  $c$  (if any) at a time  $t$ . In addition, if  $c$  is a witness, then  $t$  is the only possible time at which  $c$  may occur. Thus,  $c$ 's unique occurrence at  $t$  is its only occurrence.<sup>10</sup> In fact, we can derive the following result, given an appropriate set of

<sup>7</sup>Where  $\tau$  is a term,  $[\tau]$  is the denotation of  $\tau$ .

<sup>8</sup>To simplify the exposition, I have opted for a standard first-order logic with equality. In the SNePS tradition, an equivalence relation, denoted by the predicate "Equiv", typically replaces identity ("="). This relation holds between two intensions (Meinongian objecta) if they happen to pick out the same extension. See [14] for more details.

<sup>9</sup>A strengthening of the logic by a necessity operator is perhaps needed to dismiss cases where an event category is only contingently a serial or a witness.

<sup>10</sup>On a feature-analysis account, we can think of two binary features, given by the definitions of serials and witnesses above. Think of the first ( $\pm S$ ) as the uniqueness of occurrence at a time, and of the second ( $\pm W$ ) as the uniqueness of the time of occurrence. These two features define a  $2 \times 2$  matrix, where serials are  $\langle +S, -W \rangle$ , witnesses are  $\langle -S, +W \rangle$ , races are  $\langle -S, -W \rangle$ , and once-only events are  $\langle +S, +W \rangle$ .

inference rules and a proper axiomatization of Occurs. (The proof is straightforward, but is omitted for limitations of space.)

- $\text{OnceOnly}(c) \Leftrightarrow$   
 $\forall e_1[[\text{Cat}(e_1, c) \wedge \exists t_1[\text{Occurs}(e_1, t_1)]] \Rightarrow$   
 $\forall e_2[ [\text{Cat}(e_2, c) \wedge \exists t_2[\text{Occurs}(e_2, t_2)]] \Rightarrow e_2 = e_1]]]$

The complexity of the above formula stems from our admitting possible and incomplete tokens of an event category. In particular, stating something of the form  $\text{Cat}(e, c)$  does not mean that  $e$  actually occurs; we have to also state  $\text{Occurs}(e, t)$ , for some  $t$ . Once-only events can have multiple possible (imagined) and incomplete tokens, but only one occurrence. Thus, the above definition states that an event category is once only if an occurrence of any of its tokens is its unique occurrence.

Recalling our suggestion in Section 2.2 that the category of the  $n^{\text{th}}$  occurrence of an event is a witness, we may now easily derive (4), given (3).

$$(3) \forall c, i[\text{Witness}(\text{Occ}(i, c))]$$

$$(4) \forall c, i[\text{Serial}(c) \Rightarrow \text{OnceOnly}(\text{Occ}(i, c))]$$

## 4. A Problem with Indefinites

### 4.1. The Problem

Given the above system, we should be able to assert of particular domain-dependent event categories that they are serials, races, witnesses, or once-only. For example, we should be able to assert that the event of some (indefinite) bomb exploding is a race. But how would we represent this indefinitely-specified event? First of all, we cannot do it using a ground term; for a ground term would certainly pick out a definitely-specified event. The obvious option is to somehow introduce an existential quantifier. For example, to state that an explosion is a race, we can use (5).

$$(5) \exists x[\text{Bomb}(x) \wedge \text{Race}(\text{Explode}(x))]$$

But there are several problems with this representation. First, what does the term “Explode( $x$ )” denote? Does it denote the explosion of an indefinite bomb? Or the explosion of an indefinite entity, since the restriction of  $x$ ’s being a bomb is not part of the term? What if  $x$  is bound by a universal quantifier, would the meaning of the term change? More importantly, the above representation is obviously incorrect. It means that there is some bomb whose explosion is a race. But this is contradictory, since the explosion of that alleged bomb is certainly a serial (in fact, a once-only event).

What we seem to be facing here is yet another problem of quantifying-in. In this case, it is quantifying into an event category predication that is causing the trouble. What we need is to somehow bring the binding quantifier and the restriction on the variable inside the scope of “Race”. Unfortunately, this cannot be done in the current system.

We can, however, adopt Shapiro’s logic of arbitrary and indefinite objects [2], which solves the problem in an elegant and intuitive way.

## 4.2. Shapiro's Logic of Arbitrary and Indefinite Objects

Shapiro's logic,  $\mathcal{L}_A$ , is motivated by several issues in knowledge representation for commonsense reasoning and natural language understanding. For limitations of space, these motivations will not be discussed here. Nor will the complete logic be presented in detail; the interested reader may consult [2]. Only the relevant aspects of this logic will be minimally discussed.

Among its terms,  $\mathcal{L}_A$  includes what are called *quantified terms*. They come in two types:

1. If  $x$  is a variable and  $\phi(x)$  is a formula containing at least one free occurrence of  $x$ , then  $(any\ x)$  and  $(any\ x\ \phi(x))$  are *arbitrary terms*. If  $A$  is an arbitrary term, then  $x$  is the variable of  $A$ ,  $any$  is the determiner of  $A$ , and  $\phi(x)$  is the restriction of  $A$  (and  $x$ ).  $A$  denotes the arbitrary individual satisfying the restrictions in  $\phi(x)$ .
2. If  $x$  is a variable,  $q_1, \dots, q_n$  are variables or arbitrary terms, and  $\phi(x)$  is a formula containing at least one free occurrence of  $x$ , then  $(some\ x)$ ,  $(some\ x\ \phi(x))$ ,  $(some\ x\ (q_1, \dots, q_n))$ , and  $(some\ x\ (q_1, \dots, q_n)\ \phi(x))$  are *indefinite terms*. If  $I$  is an indefinite term, then  $x$ ,  $some$ , and  $\phi(x)$  are as for arbitrary terms (with  $some$  replacing  $any$ ).  $q_1, \dots, q_n$ , if included, are called the supporting variables of  $I$  and  $x$ .  $I$  denotes the indefinite individual satisfying the restrictions in  $\phi(x)$ , and dependent on the arbitrary individuals denoted by  $q_1, \dots, q_n$ .

One of the motivations behind  $\mathcal{L}_A$  is to provide a natural representation of donkey sentences. (6) is the famous donkey sentence due to [15], and (7) is its representation in  $\mathcal{L}_A$ .

(6) Every farmer who owns a donkey beats it.

(7) Beats( $(any\ x\ Farmer(x) \wedge Owns(x, (some\ y\ (x)\ Donkey(y))))$ ),  $y$ )

The arbitrary term  $(any\ x\ Farmer(x) \wedge Owns(x, (some\ y\ (x)\ Donkey(y))))$  denotes the arbitrary farmer who owns a donkey. The indefinite term  $(some\ y\ (x)\ Donkey(y))$  denotes the indefinite donkey that depends on the farmer  $x$ . (The inclusion of the supporting variable  $x$  maintains scoping information.) Here, due to *structure sharing*, the  $y$  that occurs as the second argument of *Beats* is the same one that occurs as the variable of the indefinite term.

## 4.3. Representing Indefinitely-Specified Events

We may now make use of Shapiro's indefinite terms to represent indefinitely-specified events. Going back to the example of explosions, here is a possible representation of the event category of some bomb exploding.

- Explode( $some\ x\ Bomb(x)$ )

Thus, to state that an explosion is a race, we may use (8).

(8) Race(Explode( $some\ x\ Bomb(x)$ ))

What is interesting here is that we have a rough syntactic characterization of definitely-specified events, indefinitely-specified events, and temporally-indefinite events.

- An event term is definitely-specified if it has no indefinite sub-terms.
- An event term is temporally-indefinite if the only indefinite sub-terms it has have variables of sort  $\mathcal{T}$ .
- An event term is indefinitely-specified if it has at least one indefinite sub-term with a variable not of sort  $\mathcal{T}$ .

## 5. Conclusions

We have examined the notion of once-only events and concluded that recognizing both event types and event tokens admits multiple simultaneous occurrences of typical cases of once-onliness. A classification of event types, distinguished by their patterns of occurrence, was shown to make the necessary distinctions between true cases of once-only events and problematic ones. The notion of indefinite specification was shown to underlie this classification. We have also shown how a logical system may account for this classification and how problems with quantifying into indefinitely-specified event terms may be resolved, using Shapiro's logic of arbitrary and indefinite objects.

## 6. Acknowledgements

The author thanks Stuart Shapiro and three anonymous reviewers for their comments on an earlier version of this paper.

## References

- [1] Antony Galton. *The Logic of Aspect*. Clarendon Press, Oxford, 1984.
- [2] Stuart C. Shapiro. A logic of arbitrary and indefinite objects. In Didier Dubois, Chris Welty, and MaryAnne Williams, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Ninth International Conference (KR2004)*, pages 565–575. AAAI Press, 2004.
- [3] Haythem O. Ismail. *Reasoning and Acting in Time*. PhD thesis, University at Buffalo, The State University of New York, 2001.
- [4] Antony Galton. The logic of occurrence. In Antony Galton, editor, *Temporal Logics and Their Applications*, pages 169–196. Academic Press, 1987.
- [5] Brandon Bennett and Antony Galton. A unifying semantics for time and events. *Artificial Intelligence*, 153(1–2):13–48, 2004.
- [6] Godehard Link. *Algebraic Semantics in Language and Philosophy*. CSLI Publications, Stanford, CA, 1998.
- [7] Haythem O. Ismail and Stuart C. Shapiro. Two problems with reasoning and acting in time. In Anthony Cohn, Fausto Giunchiglia, and Bart Selman, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Seventh International Conference (KR 2000)*, pages 355–365, San Francisco, CA, 2000. Morgan Kaufmann.
- [8] Stuart C. Shapiro and William J. Rapaport. SNePS considered as a fully intensional propositional semantic network. In N. Cercone and G. McCalla, editors, *The Knowledge Frontier*, pages 263–315. Springer-Verlag, New York, 1987.

- [9] Stuart C. Shapiro. SNePS: A logic for natural language understanding and commonsense reasoning. In Lucja M. Iwańska and Stuart C. Shapiro, editors, *Natural Language Processing and Knowledge Representation: Language for Knowledge and Knowledge for Language*, pages 175–195. AAAI Press/The MIT Press, Menlo Park, CA, 2000.
- [10] Stuart C. Shapiro and Haythem O. Ismail. Anchoring in a grounded layered architecture with integrated reasoning. *Robotics and Autonomous Systems*, 43(2–3):97–108, May 2003.
- [11] Stuart C. Shapiro and William J. Rapaport. Models and minds: Knowledge representation for natural-language competence. In Robert Cummins and John Pollock, editors, *Philosophy and AI: Essays at the Interface*, pages 215–259. MIT Press, Cambridge, MA, 1991.
- [12] Stuart C. Shapiro. Belief spaces as sets of propositions. *Journal of Experimental and Theoretical Artificial Intelligence*, 5:225–235, 1993.
- [13] Craig Schlenoff, Michael Gruninger, Florence Tissot, John Valois, Josh Lobell, and Jintae Lee. The process specification language (PSL): Overview and version 1.0 specification. Technical Report NISTIR 6459, National Institute of Standards and Technology, Gaithersburg, MD, 2000.
- [14] Anthony S. Maida and Stuart C. Shapiro. Intensional concepts in propositional semantic networks. *Cognitive Science*, 6:291–330, 1982.
- [15] P. T. Geach. *Reference and Generality*. Cornell University Press, Ithaca, NY, 1962.