The South African Institute of Computer Science
and
Information Technology

The 1997 National Research and Development Conference

Riverside Sun
Vanderbijlpark
13 & 14 November

Hosted by

Potchefstroomse Universiteit
vir Christelike Hoër Onderwys

The Department of Computer Science and Information Systems
Potchefstroom University for Christian Higher Education
Vaal Triangle Campus

SAICSIT 97

PROCEEDINGS

Edited by L.M. Venter & R.R. Lombard

The South African Institute of Computer Science
and
Information Technology

Proceedings

of the

The 1997 National Research and Development Conference

Towards 2000

Riverside Sun
Vanderbijlpark
13 & 14 November

Edited by
L.M. Venter
R.R. Lombard

ISBN 1-86822-300-0

Printed and Binded by Xerox Printers, Potchefstroom

The views expressed in this book are those of the individual authors

Foreword

This book contains a collection of papers presented at a Research and Development conference of the South African Institute of Computer Scientists and Information Technologists (SAICSIT). The conference was held on 13 & 14 November 1997 at the Riverside Sun, Vanderbijlpark. Most of the organization for the conference was done by the Department of Computer Science and Information Technology of the Vaal Triangle Campus, Potchefstroom University for Christian Higher Education.

The programming committee accepted a wide selection of papers for the conference. The papers range from detailed technical research work to reports of work in progress. The papers originate mainly from Academia, but also describe work done in and for Industry. It is hoped that the papers give a true reflection of the current research scene in Computer Science and Information Technology in South Africa. Since one of the aims of the conference is Research development, the papers were not subjected to a refereeing process.

A number of people spent numerous hours helping with the organization of this conference. In this regard, we wish to thank the members of the Organizing committee, and the Programming committee who had very little time to screen the abstracts and compile the program. A special thanks goes to the secretary of the department, Mrs Helei Jooste, whose very able work was interrupted by the birth of her first child.

Organizing Committee

Conference General Chairs
Prof. J.M. Hattingh (PU for CHE)

Organizing Chair
Prof. Lucas Venter (PU for CHE)

Organizing Committee
Mrs. S. Gilliland
Mr. J.P. Jooste
Mr. R.R. Lombard
Mrs. M. Huisman

Secretariat
Mrs. H. Jooste

Program Chair
Prof A de Waal (PU for CHE)

Program Committee
Prof D. Kourie (UP)
Prof. C. Bornman (UNISA)
Prof. L.M. Venter (PU for CHE)

Table of Contents

List of Contributors

S.A. Ajila
Department of Mathematics and Computer Science
National University of Lesotho
Roma, 180
Lesotho

L. Baart
Department of Mathematics
Vaal Triangle Campus of the PU for CHE
PO Box 1174
Vanderbijlpark, 1900

L. Barnard
Faculty of Computer Studies
Port Elizabeth Technikon
Private Bag X6011
Port Elizabeth, 6000

S. Berman
University of Cape Town
Rondebosch, 7701

L. Bester
Faculty of Computer Studies
Port Elizabeth Technikon
Private Bag X6011
Port Elizabeth 6000

J.M. Bishop
Computer Science Department
University of Pretoria
Pretoria, 0002

L. Botha
Computer Science Department
University of Pretoria
Pretoria, 0002

R.A. Botha
Faculty of Computer Studies
Port Elizabeth Technikon
Private Bag X6011
Port Elizabeth, 6000

B. Braude
Software Engineering Applications Laboratory,
Electrical Engineering
University of the Wiwatersrand
Private Bag 3
Wits, 2050

T. Breetzke
Faculty of Computer Studies
Port Elizabeth Technikon
Private Bag X6011
Port Elizabeth, 6000

C. Brink
University of Cape Town
Rondebosch, 7700

M. Bruynooghe
Departement Computerwetenschappen
Katholieke Universiteit Leuven
Celestjinenlaan 200A
B-3001 Heverlee
Belgium

S. Buffler
University of Capetown
Rondebosch, 7701

M.A. Coetzee
Department of Mathematics
PU for CHE
Private Bag X6001
Potchefstroom, 2520

R. Cools
Katholieke Universiteit Leuven
Celestjinenlaan 200A
B-3001 Heverlee
Belgium

E. de Preez
Faculty of Computer Studies
Port Elizabeth Technikon
Private Bag X6011
Port Elizabeth, 6000

D.A. De Waal
Department of Computer Science and Information Systems
PU for CHE
Private Bag X6001
Potchefstroom, 2531

B. Dekenah
The Board of Executors

M. Denecker
Departement Computerwetenschappen
Katholieke Universiteit Leuven
Celestjinenlaan 200A
B-3001 Heverlee
Belgium

M. Dunley-Owen
Department of Information Systems
University of Cape Town
Rondebosch, 7700

R. Fiqueira
University of Cape Town
Rondebosch, 7701

A. Foster
Department of Computer Science
University of Cape Town
Rondebosch, 7701

C. Gee
Software Engineering Applications Laboratory,
Electrical Engineering
University of the Witwatersrand
Private Bag 3
Wits 2050

M. Hajek
Department of Computer Science
University of Durban Westville
Private Bag X54001
Durban, 4000

M.L. Hart
Department of Information Systems
University of Cape Town
Rondebosch, 7700

J.M. Hattingh
Department of Computer Science and
Information Systems
PU for CHE
Private Bag X6001
Potchefstroom, 2520

S. Hazelhurst
Department of Computer Science
University of the Witwatersrand
Private Bag 3
Wits 2050

H.A. Kruger
Department of Computer Science and
Information Systems
PU for CHE
Private Bag X6001
Potchefstroom, 2520

J.W. Kruger
University of the Witwatersrand
Private Bag 3
Wits, 2050

M.F. Kruger
PU for CHE
Private Bag X6001
Potchefstroom, 2520

M.T. Lang
Eskom Information Technology Department

D. Laurie
Department of Mathematics
Vaal Triangle Campus of the PU for CHE
PO Box 1174
Vanderbijlpark, 1900

D. Lubinsky
Department of Computer Science
University of the Witwatersrand
Private Bag 3
Wits, 2050

R. McLeod
Saltire Software Inc.
Tigard
Oregon
U.S.A

H.J. Messerschidt
Department of Computer Science and
Informatics
University of the Orange Free State
PO Box 339
Bloemfontein, 9300

M. Mphahlele
Department of Computer Science
University of the North
Private Bag X1106
Sovenga, 0727

G.D. Oosthuizen
Department of Computer Science
University of Pretoria
Pretoria, 0002

J. Owen
University of Cape Town
Rondebosch, 7701

D. Petkov
Department of Computer Science
University of Natal
Private Bag X01
Scotsville, 3209

O. Petkova
Technikon Natal
PO Box 101112
Scotsville, 3209

N. Pillay
Department of Financial Studies
Technikon Natal, Pietermaritzburg
PO Box 101112
Scotsville, 3209

L. Pluym
Katholieke Universiteit Leuven
Celestijnenlaan 200A
B-3001 Heverlee
Belgium

K. Prag
Department of electrical Engineering
University of Durban-Westville
Private Bag X54001
Durban, 4000

P. Premjeeth
Department of electrical Engineering
University of Durban-Westville
Private Bag X54001
Durban, 4000

V. Ram
Department of Computer Science
University of Natal
Private Bag X01
Scotsville, 3209

J. Robertson
Department of Computer Science and
Informatics
University of the Orange Free State
PO Box 339
Bloemfontein, 9300

S. Rock
Department of Artificial Intelligence
Edinburgh University
United Kingdom

J. Roos
Department of Computer Science
University of Pretoria
Pretoria, 0002

I. Sanders
Department of Computer Science
University of the Witwatersrand
Private Bag 3
Wits, 2050

K. Sandrasegaran
Department of electrical Engineering
University of Durban-Westville
Private Bag X54001
Durban, 4000

C. Schoder
Faculty of Computer Studies
Port Elizabeth Technikon
Private Bag X6011
Port Elizabeth, 6000

M. Sears
Department of Mathematics
University of the Witwatersrand
Private Bag 3
Wits, 2050

E. Senior
International Center for Waste Technology
University of Natal, Pietermaritzburg
Private Bag X01
Scotsville, 3209

N.B. Serbedzija
GMD FIRST
Rudower Chausee 5
D-12489 Berlin
Germany

S.L. Serutla
Department of Computer Science
The University of Pretoria
Pretoria, 0002

T. Steyn
PU for CHE
Private Bag X6001
Potchefstroom, 2520

M. Thielscher
Fachgebiet Intellektik, Fachgebiet Informatik
Technische Hochschule Darmstadt
Alexanderstrasse 10
D-64283 Darmstadt
Germany

T. Thomas
Faculty of Computer Studies
Port Elizabeth Technikon
Private Bag X6011
Port Elizabeth, 6000

M. Thomas
Faculty of Computer Studies
Port Elizabeth Technikon
Private Bag X6011
Port Elizabeth, 6000

S. Tjasink
University of Cape Town
Rondebosch, 7700

E. Viljoen
Department of Computer Science and
Information Systems
University of South Africa
PO Box 392
Pretoria, 0001

E. Voges
University of Cape Town
Rondebosch, 7701

R. Von Solms
Faculty of Computer Studies
Port Elizabeth Technikon
Private Bag X6011
Port Elizabeth, 6000

A.J. Walker
Software Engineering Applications Laboratory,
Electrical Engineering
University of the Witwatersrand
Private Bag 3
Wits, 2050

P. Warren
Department of Computer Science
University of Natal
Private Bag X01
Scotsville, 3209

M. Watzenboeck
University of Botswana
Private Bag 0022
Gaberone
Botswana

K.L. Wortmann
Department of Computer Science
University of Natal, Pietermaritzburg
Private Bag X01
Scotsville, 3209

More readings than I thought: quantifier interaction in analysing the temporal structure of repeated eventualities

Sheila Rock, Department of Artificial Intelligence, Edinburgh University*
sheilar@dai.ed.ac.uk

October 20, 1997

Abstract

Focussing on the temporal structures of the events described by language, the sentence
'*Simmer the soup stirring it occasionally for fifteen minutes*'
without punctuation, has three syntactic structures, which give rise to five semantic interpretations.

All five readings are valid, but they can only be found if we take the view, proposed by Moltmann[8], that temporal (and spatial) measure adverbials play the role of part quantifiers.

Using this approach, and extending it to include the notion that it is appropriate to treat event quantifiers and object quantifiers in similar ways, we also are able to find the two interpretations of the sentence
'*Mary visited a friend every week*'
which many computational systems do not distinguish. The two interpretations can be paraphrased as
≪*There is a friend that Mary has, whom she visited every week*≫
and
≪*Every week, Mary participated in the activity of "visiting a friend"; the object of the activity need not be the same for each week's activity*≫

In this paper I describe first of all the justification for adopting Moltmann's view, and then demonstrate how this can be used in a computational system to highlight ambiguity. I argue that treating measure adverbials as quantifiers exposes some appropriate ambiguity that would otherwise remain hidden.

I have used this approach in the implementation of a computational system that identifies all the available meanings when temporal measure adverbials and quantifiers interact. The implementation makes use of a standard algorithm for quantifier scoping[4], that has been used for scoping quantified terms describing objects, and applies it to both objects and events.

1 Introduction

One of the exciting results of my thesis [10] came from analysing the sentence in (1).

(1) Simmer the soup stirring occasionally for fifteen minutes

When I analysed the sentence myself, I saw that it was ambiguous. That it is ambiguous between two readings is clear. I put it as input into the computational system that I have developed for analysing sentences that are about structured activities, and expected two analyses. When the system returned not two but five, I was concerned. However, it has turned out that in fact all five readings are valid, and the approach that I have taken in the computational analysis is essential if we are to find all five readings. The approach relies on the view proposed by Moltmann [8] that temporal measure adverbials play the role of part quantifiers, rather than that of conventional adverbial modifiers.

In this paper, I describe first of all Moltmann's view, and justify the reasons for addopting her approach. Then I show how, taking the position that eventualities and objects can be viewed analogously with respect

*Currently at Computer Science Department, University of the Witwatersrand, Johannesburg, sheila@cs.wits.ac.za

to time and space [11, 7, 5], it is possible to apply well accepted theory about the quantification of objects—in particular, the application of a standard algorithm for quantifier scoping [4]—to the quantification of eventualities.

2 Temporal modifiers as quantifiers

A common approach to analysing temporal adverbials has been as event predicates, much in the way that other adverbials are viewed. So, the modifier '*for ten minutes*' is seen as doing the same kind of thing as the modifier '*with a spoon*'. However, some authors have taken the view that temporal measure adverbials have properties that set them apart from adverbial event predicates, and this makes them closer to being part quantifiers. In particular, Moltmann[8] argues for this, and further does not include all temporal adverbials, only measure adverbials—so '*yesterday*' and '*at noon*' are not part of the discussion. This makes sense—after all, quantification is connected with measure rather than with location. So, her analysis is distinct from those that try to account for temporal adverbials more generally [9, 3]. This distinction—between measure adverbials and other kinds of adverbials—is in my view an extremely important one.

Moltmann argues for the view that measure adverbials in general are part quantifiers. This view begins with Dowty's[2] claim—that temporal measure adverbials (such as '*for two hours; until noon*') are part quantifiers ranging over the parts of some measuring entity; Moltmann extends the claim to other measure adverbials ('*worldwide; throughout the country*') strengthens the argument initiated by Dowty.

Measure adverbials have properties that set them apart from adverbial event predicates; these are used to support her thesis. Though the work she presents is a comprehensive analysis of the properties and distinctive aspects of measure adverbials—in particular, in relation to other adverbials—I only present, in summary form, those results from Moltmann's work that have direct impact on the work described in this paper. The reader is encouraged to consult Moltmann's paper[8] for convincing support of these results.

So, we accept Moltmann's argument that measure adverbials have the status of quantifiers; this is confirmed by the phenomena of scope interactions of measure adverbials with each other and with other quantifiers. Examples she gives are in (2) and (4); in (2b), there is no acceptable interpretation because of the syntax of the quantifiers and their scoping interactions.

(2) a. John listened to Mozart all the time for ten weeks.

b. #All the time John listened to Mozart for ten weeks.

I believe the syntax issue in (2) to which Moltmann alludes—though she does not present it in this way—can be seen if we translate (2a) and (2b) into logical forms (3a) and (3b) respectively, where $t \subseteq p$ represents $\ll t$ *is a subinterval of* $p \gg$. It is clear in (3b) that there is a syntactic scope problem—in the logical form—with p.

(3) a. $\text{ten}(p, \text{week}(p), \text{all}(t, t \subseteq p, \text{listen}(\mathit{john}, \mathit{mozart})))$

b. $\text{all}(t, t \subseteq p, \text{ten}(p, \text{week}(p), \text{listen}(\mathit{john}, \mathit{mozart})))$

So, treating measure adverbials as quantifiers then means that they are subject to the same treatment as quantifiers in terms of how they interact with other measure adverbials and quantifiers. To further show the need for this, Moltmann demonstrates the shortcoming of treating such adverbials as event predicates, using the sentences in (4) as examples. In an event predicate treatment, where '*for several years*' is treated as predicating the complaining event, both of these sentences would have a semantic representation as in (5). That is, only the meaning of (4b) would be accounted for. However, treating '*for several years*' as a part quantifier allows both meanings to be represented, as in (6).

(4) a. For several years a lot of students complained about the requirements.

b. A lot of students complained about the requirements for several years.

(5) $\exists t.\exists x.\exists e$ (several-years(t) $\wedge$ lot-of-students(x) $\wedge$ complain-about($e, x, [requirements]$) $\wedge at(e,t)$)

(6) a. several(t, year(t),$\exists x.\exists e$(lot-of-students(x) $\wedge$ complain-about($e, x, [requirements]$) $\wedge at(e,t)$))

b. $\exists x.\exists e$(lot-of-students(x)$\wedge$ several(t,year(t), complain-about($e, x, [requirements]$) $\wedge at(e,t)$))

I have given only enough detail of Moltmann's proposal[1] to demonstrate its basic concepts and to show that it is a useful approach to dealing with certain kinds of temporal adverbial. In particular, it also connects with the object-eventuality analogy that I have used extensively in analysing structure within events[10].

2.1 Event predicates, adverbials and measure adverbials

Distinguishing between adverbials that are about measure and those that are not (and are, say, about location) is important. This is particularly true at the semantic level, and when we incorporate the concept of quantification. In the case of simple quantification over objects, this distinction is easy to accept. There is a clear difference between things like ≪*many people*≫—denoting a quantity—and ≪*people at the desk*≫—denoting a point in space. There is also a need to distinguish between phrases, with respect to eventualities, that are about quantity and those that are not. I use adverbial phrases involving the preposition '*in*' to illustrate some distintions.

First, we look at the temporal arena. Prepositional phrases, such as '*in ten weeks*' have two senses in which they can be used—as a measure adverbial and as an adverbial denoting temporal location. This distinction is shown in (7); for me both sentences are ambiguous between the building taking ten weeks and the building commencing in ten weeks; however other authors [3] have argued that the ambiguity is not present in (7b). There are also different proposals about the source of the ambiguity[3, 10]; Hitzeman argues that the ambiguity does not come from the temporal adverbial, but from the way in which the adverbial combines with the rest of the sentence, while my view is that the ambiguity comes from the ambiguity of the preposition.

(7) a. They will build the bridge in ten weeks.

b. In ten weeks they will build the bridge.

If we take the view that measure adverbials are part quantifiers, we also need to accept that we have to distinguish semantically between the uses of *in*-adverbials as I describe above—as measure adverbials and as adverbials of location. I believe that it is the dual use of the adverbial that accounts for the ambiguity. In one case, it is being used as a measure adverbial, describing a period of ten weeks; in the other case it functions as a temporal location adverbial (much in the way '*tomorrow*' would function). The fact that both uses are delivered using an adverbial modifier is a phenomenon that is not unusual with prepositions—prepositions are well accepted as being ambiguous, and participating in ambiguous constructs. I would argue that recognising this ambiguity, **in conjunction with** a quantifier treatment of measure adverbials, is essential to developing a computational treatment of such adverbials. So, with Moltmann's analysis, in the first case it is a part quantifier, and in the second case it is not.

This analysis extends to the spatial domain, as exemplified in (8).

(8) a. Rosa rode a horse in Ben's picture.

b. Rosa found a scratch in Ben's picture.

[1] I note that Moltmann[8] provides a strong motivation for the quantifier approach, but does not define a formal mechanism for it.

Here, '*in*' is functioning as a spatial adverbial; however again it is functioning as a measure adverbial in (8a) and as an adverbial of location in (8b). To be clearer, ≪*within, throughout—in all of—Ben's picture*≫ Rosa rode a horse; ≪*at a particular place in Ben's picture*≫ Rosa found a scratch. Moltmann's analysis extends to quantification over space, as well as time, and hence this ambiguity is accounted for in the same way.

2.2 The interaction of multiple adverbials

The quantifier approach is also appropriate for dealing with multiple adverbials, as indicated indicated by the examples in (2) and (3). I note that again, alternate analyses, for dealing with the ill-formedness of some combinations of adverbials—that do not see these adverbials as quantifiers—have been proposed[3, 6] and limitations with these have been identified[10].

Taking the quantifier approach, both of the modifiers are measure adverbials—'*for 10 weeks*' and '*all the time*' both describe quantities of time. It is the quantifier scoping that shows us that (3b) is ill-formed.

3 Quantification

In the previous section I have argued that temporal measure adverbials—such as '*for half an hour*'—should be treated as event quantifiers, in much the same way that phrases like '*every week*' and '*three times*' are. In addition, the usual approach to treating nominal quantifiers applies.

The classic problem of meaning when we have quantified objects (nominal quantification) has received a lot of treatment in various fields—philosophy, logic, linguistics, computational linguistics, etc. The sentence in (9) is often used when discussing these issues.

(9) Every man loves a woman.

Clearly, there are two possible readings of this sentence. The one described in (10a) is found when the logic formulation is such that the existential quantifier takes wide scope, as in (10b). When it takes narrow scope, the meaning and logic formulation are as in (11).

(10) a. There is a (particular) woman of whom it is true that every man loves her.

b. $\exists w.\forall m.\text{loves}(m,w)$

(11) a. For every man, there is some (nonunique) woman that he loves.

b. $\forall m.\exists w.\text{loves}(m,w)$

Now, my work has involved taking work that has focused on objects, and applying in to eventualities. That is the main purpose of this section—to describe such application.

3.1 Feature structures

I have used feature structures to represent the meaning of sentences that describe eventualities. In this section I briefly describe this use.

I first demonstrate a feature structure representation of a simple quantification example—that in (12). This sentence is shown in unscoped form[2] in (13); in its two scoped forms, it can be represented in first-order logic as (14a) and (14b).

(12) Every chef bakes a cake.

[2]In this, an unscoped, 'in-place' quantified term is represented using notation (`qterm`) that indicates that it is a single expression containing the quantifier and its restriction, connected by a variable. So, '*many chefs*' would be represented by the quantified term `qterm(many, X, chef(X))`.

(13) $bake(qterm(\forall, y, chef(y)), qterm(\exists, x, cake(x)))$

(14) a. $\exists x(cake(x), \forall y(chef(y) \Rightarrow bakes(y, x)))$

b. $\forall y(chef(y), \exists x(cake(x) \Rightarrow bakes(y, x)))$

Taking the approach advocated by Pollard & Sag (with some minor modifications), feature structure representations for (14) are presented in (15); these are scoped representations. I note that I distinguish between the restriction and scope of a quantified expression, to embody a generalised quantifier notation like that proposed by Barwise & Cooper[1].

(15) a.

$$\begin{bmatrix} \text{quantifier} = \begin{bmatrix} \text{determiner} = \text{exists} \\ \text{var} = x \\ \text{restriction} = \text{cake} \end{bmatrix} \\ \text{scope} = \begin{bmatrix} \text{quantifier} = \begin{bmatrix} \text{determiner} = \text{for-all} \\ \text{var} = y \\ \text{restriction} = \text{chef} \end{bmatrix} \\ \text{scope} = \begin{bmatrix} \text{substance} = \text{bake} \\ \text{actor} = x \\ \text{patient} = y \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

b.

$$\begin{bmatrix} \text{quantifier} = \begin{bmatrix} \text{determiner} = \text{for-all} \\ \text{var} = y \\ \text{restriction} = \text{chef} \end{bmatrix} \\ \text{scope} = \begin{bmatrix} \text{quantifier} = \begin{bmatrix} \text{determiner} = \text{exists} \\ \text{var} = x \\ \text{restriction} = \text{cake} \end{bmatrix} \\ \text{scope} = \begin{bmatrix} \text{substance} = \text{bake} \\ \text{actor} = y \\ \text{patient} = x \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

3.2 Simple event quantifiers

A primary difference in representing quantification of events—and indeed simply the representing of events in general—is the need for the introduction of an event variable, which is then used for predication.

When we represent something like '*a chef bakes a cake*' from an object perspective, a representation such as those shown in (14) is sufficient. However, when we are trying to represent this from an event perspective, we find that we need to introduce something to represent the ≪*baking event*≫ in order to refer to it. So, (16) is an example of such representation.

(16) $\exists e, x, y(event(e) \wedge cake(x) \wedge chef(y) \wedge actor(e, y) \wedge patient(e, x) \wedge bake(e))$

This then carries through to the feature structure representation, so that structures of the form of (17) are appropriate. The innermost feature structure, representing '$actor(e, y) \wedge patient(e, x) \wedge bake(e)$', utilises the notion that it is structurally part of the feature structure for event e.

(17)
$$\begin{bmatrix} \text{quantifier} = \begin{bmatrix} \text{determiner} = \text{exists} \\ \text{var} = e \\ \text{restriction} = \text{event} \end{bmatrix} \\ \text{scope} = \begin{bmatrix} \text{quantifier} = \begin{bmatrix} \text{determiner} = \text{exists} \\ \text{var} = x \\ \text{restriction} = \text{cake} \end{bmatrix} \\ \text{scope} = \begin{bmatrix} \text{quantifier} = \begin{bmatrix} \text{determiner} = \text{exists} \\ \text{var} = y \\ \text{restriction} = \text{chef} \end{bmatrix} \\ \text{scope} = \begin{bmatrix} \text{substance} = \text{bake} \\ \text{actor} = x \\ \text{patient} = y \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Anything that is a basic event—a happening—is assumed to be quantified by a basic event quantifier that always takes narrow scope over the description of the happening.

3.3 Measure adverbials as quantifiers

We now look at an example of the use of feature structures to represent the semantics of measure adverbials as event quantifiers. I take this opportunity to move to the feature structure form that I use in the rest of this paper, which uses features [`content=`] and [`extent=`] instead of [`scope=`] and [`quantifier=`]. This equivalence is demonstrated using the sentence of (18).

(18) Stir the soup twice.

In this example, there is only one scoping possible, because there is only one measure adverbial, or quantifier. The feature structure (19) shows a direct representation of (18), using the usual quantification terminology.

(19)
$$\begin{bmatrix} \text{quantifier} = \begin{bmatrix} \text{determiner} = \text{exists} \\ \text{var} = e_1 \\ \text{restriction} = \text{event} \end{bmatrix} \\ \text{structure} = \text{ordered-set} \\ \text{scope} = \begin{bmatrix} \text{quantifier} = \begin{bmatrix} \text{determiner} = 2 \\ \text{var} = e_2 \\ \text{restriction} = \text{event} \end{bmatrix} \\ \text{structure} = \text{happening} \\ \text{scope} = \begin{bmatrix} \text{substance} = \begin{bmatrix} \text{action} = \text{stir} \\ \text{patient} = \text{soup} \end{bmatrix} \\ \text{type} = \text{discrete} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

However, this structured event is of the same form as that shown for (18), whose feature structure is shown in (20), repeated here. So, we already have available the machinery for representing it as quantification. The

feature [extent=] performs the quantification role, and the feature [content=] is analogous to the scope. The [index=] feature can be obtained from the event variable, and allows reference to sub-events when appropriate.

(20)
$$\begin{bmatrix} \text{structure} = \text{ordered-set} \\ \text{index} = e_1 \\ \text{extent} = \begin{bmatrix} \text{temporal-spacing} = \text{unfixed} \\ \text{cardinality} = \text{two} \end{bmatrix} \\ \text{content} = \begin{bmatrix} \text{structure} = \text{happening} \\ \text{index} = e_2 \\ \text{type} = \text{discrete} \\ \text{substance} = \begin{bmatrix} \text{action} = \text{stir} \\ \text{patient} = \text{soup} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

3.4 Representation prior to scoping

Until now, all the representations that have been shown are of feature structures that are already in scoped form. However, these representations are obtained by applying a scoping algorithm to an unscoped representation. An unscoped representation has 'in-place' quantified terms that are marked as such by the use of special features.

Event quantifiers in an unscoped form are collected into a list of [modifiers=], at the end of the basic event that they are modifying. Each modifier is simply represented as such; its quantification role comes from the fact that it is placed into the list of modifiers.

Object quantifiers in an unscoped form are those patients and recipients of the action of the basic event, that have a feature [term=][3], which has a [det=], a [var=] and a [restr=], corresponding to the determiner, variable and restriction of the generalised quantifier notation.

The sentence in (21) has two temporal measure modifiers and one object quantifier; the sentence in unscoped form would be represented as (22), where the list of modifiers represents the in-place event quantifiers.

(21) Stir some soup occasionally for half an hour.

[3]The feature is called term because it represents a quantified term.

(22)

$$\begin{bmatrix} \text{structure} = \text{happening} \\ \text{index} = \text{e2} \\ \text{type} = \text{iterable} \\ \text{substance} = \begin{bmatrix} \text{action} = \text{stir} \\ \text{patient} = \begin{bmatrix} \text{term} = \begin{bmatrix} \text{det} = \text{some} \\ \text{var} = v \\ \text{restr} = \text{soup} \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{modifiers} = \begin{bmatrix} \text{first} = \begin{bmatrix} \text{structure} = \text{delimitedmass} \\ \text{index} = \text{e5} \\ \text{extent} = \begin{bmatrix} \text{extent-type} = \text{time} \\ \text{term} = \begin{bmatrix} \text{det} = \text{exists} \\ \text{var} = w \\ \text{restr} = \text{hours} \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{rest} = \begin{bmatrix} \text{first} = \begin{bmatrix} \text{structure} = \text{ordered-set} \\ \text{index} = \text{e12} \\ \text{extent} = \begin{bmatrix} \text{det} = \text{occasional} \\ \text{var} = x \\ \text{restr} = \text{event} \end{bmatrix} \end{bmatrix} \\ \text{rest} = \text{end} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

3.5 Scoped representations

In the example of (3.5), which is the representation of the sentence in (23), there are two interacting event quantifiers, resulting in two possible scopings. The more natural one is that of ≪*stirring occasionally*≫ happening through a period of ≪*half an hour*≫. This is demonstrated in (25). However, another possible reading is that we have ≪*occasionally*≫ the ≪*stirring of soup for half an hour*≫, as shown in (26).

(23) Stir the soup occasionally for half an hour.

(24)

$$\begin{bmatrix} \text{structure} = \text{happening} \\ \text{index} = e_1 \\ \text{substance} = \begin{bmatrix} \text{action} = \text{stir} \\ \text{patient} = \text{soup} \end{bmatrix} \\ \text{type} = \text{iterable} \\ \text{modifiers} = \begin{bmatrix} \text{first} = \begin{bmatrix} \text{structure} = \text{delimitedmass} \\ \text{index} = e_2 \\ \text{extent} = \begin{bmatrix} \text{term} = \begin{bmatrix} \text{det} = \text{exists} \\ \text{var} = w \\ \text{restr} = \text{hours} \end{bmatrix} \\ \text{extent-type} = \text{time} \end{bmatrix} \end{bmatrix} \\ \text{rest} = \begin{bmatrix} \text{first} = \begin{bmatrix} \text{structure} = \text{ordered-set} \\ \text{index} = e_3 \\ \text{extent} = \begin{bmatrix} \text{det} = \text{occasional} \\ \text{var} = x \\ \text{restr} = \text{event} \end{bmatrix} \end{bmatrix} \\ \text{rest} = \text{end} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

(25)

$$\begin{bmatrix} \text{structure} = \text{delimited-mass} \\ \text{index} = e_2 \\ \text{extent} = \begin{bmatrix} \text{end} = \text{for-half-hour} \end{bmatrix} \\ \text{content} = \begin{bmatrix} \text{structure} = \text{ordered-set} \\ \text{index} = e_3 \\ \text{extent} = \begin{bmatrix} \text{temporal-spacing} = \text{unfixed} \\ \text{frequency} = \text{occasionally} \end{bmatrix} \\ \text{content} = \begin{bmatrix} \text{structure} = \text{happening} \\ \text{index} = e_1 \\ \text{type} = \text{discrete} \\ \text{substance} = \begin{bmatrix} \text{action} = \text{stir} \\ \text{patient} = \text{soup} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

(26)
$$\begin{bmatrix} \text{structure} = \text{ordered-set} \\ \text{index} = e_3 \\ \text{extent} = \begin{bmatrix} \text{temporal-spacing} = \text{unfixed} \\ \text{frequency} = \text{occasionally} \end{bmatrix} \\ \text{content} = \begin{bmatrix} \text{structure} = \text{delimited-mass} \\ \text{index} = e_2 \\ \text{extent} = \begin{bmatrix} \text{end} = \text{for-half-hour} \end{bmatrix} \\ \text{content} = \begin{bmatrix} \text{structure} = \text{happening} \\ \text{index} = e_1 \\ \text{type} = \text{discrete} \\ \text{substance} = \begin{bmatrix} \text{action} = \text{stir} \\ \text{patient} = \text{soup} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

I note that the feature structure representation brings along its own inherent structuring that contributes to the semantics of what it is representing. That is, I am using a structured representation to represent structured entities—activity events; the nesting that is provided in feature structures is used to show activity nesting. So, any activity that is nested in the feature structure representation of another activity, is actually a sub-activity of that other activity. In the example in (26), the activity indexed by e_1 is a sub-activity of e_2; similarly e_2 is a sub-activity of e_3. However, this information comes from the structure of the representation, and does not have to be explicitly represented.

3.6 Combining object and event quantification

The final example is one where a mix of object and event quantifiers is demonstrated; again the scope interactions allow for more than one reading. The example used is (27), which has two readings as glossed in (28a) and (28b). Feature structures for these readings are in (29a) and (29b)[4] respectively.

(27) Bake some cakes every week.

(28) a. *≪Every week, you are to bake some cakes.≫*

b. *≪There are some cakes that are to be baked every week.≫*

[4]The second reading is unusual, because we know that baking the same cake object week after week is an unlikely thing to do. However, it is a valid reading—especially if it is understood to mean *≪bake a cake of a particular sort≫*— and so we must allow for it.

(29) a.

$$\begin{bmatrix} \text{structure} = \text{ordered-set} \\ \text{index} = e_3 \\ \text{extent} = \begin{bmatrix} \text{frequency} = \text{every-week} \end{bmatrix} \\ \text{content} = \begin{bmatrix} \text{structure} = \text{ordinary-set} \\ \text{index} = e_2 \\ \text{extent} = \begin{bmatrix} \text{cardinality} = \text{some} \\ \text{var} = i \\ \text{restriction} = \text{cake} \end{bmatrix} \\ \text{content} = \begin{bmatrix} \text{structure} = \text{happening} \\ \text{index} = e_1 \\ \text{substance} = \begin{bmatrix} \text{action} = \text{bake} \\ \text{patient} = i \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

b.

$$\begin{bmatrix} \text{structure} = \text{ordinary-set} \\ \text{index} = e_2 \\ \text{extent} = \begin{bmatrix} \text{cardinality} = \text{some} \\ \text{var} = i \\ \text{restriction} = \text{cake} \end{bmatrix} \\ \text{content} = \begin{bmatrix} \text{structure} = \text{ordered-set} \\ \text{index} = e_3 \\ \text{extent} = \begin{bmatrix} \text{frequency} = \text{every-week} \end{bmatrix} \\ \text{content} = \begin{bmatrix} \text{structure} = \text{happening} \\ \text{index} = e_1 \\ \text{substance} = \begin{bmatrix} \text{action} = \text{bake} \\ \text{patient} = i \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

4 Scoping algorithm

The algorithm that extracts scopings has been based on the algorithm described by Hobbs & Shieber[4] for scoping unscoped logical forms. Their algorithm is a simple one that examines a logical form for quantified terms, and systematically 'pulls' these to outer levels of the logical form, ensuring syntactic well-formedness of the logical form.

The Hobbs & Shieber algorithm provided a very useful starting point; however there are three differences that are important in its use here:

1. I am concerned with feature structures, whereas their algorithm applied to logical forms.
2. I am concerned with events, and thus with temporal measure adverbials as well as event quantifiers and object quantifiers.
3. Feature structures already provide some 'structure' that unscoped logical forms do not.

The approach that is taken here is to use the notion of quantified terms, and look through the feature structure to find them. Once they are found, they are pulled into higher levels of the feature structure, in ways that are syntactically[5] appropriate; their position in the unscoped form is replaced by the variable of the three-part quantifier that has been 'pulled'. This is similar to the way the Hobbs & Shieber algorithm works.

However, rather than just pulling all of the quantifiers to the outermost position in appropriate orderings, we also need to ensure that they are pulled to appropriate levels in the feature structure. The level incorporates information about scoping range (the wide-narrow distinction in conventional scoping notions). So, it is a mix of quantifier level within the feature structure, and nesting of entities within the feature structure, that provides the scoping information for both quantified objects, and events that have been quantified by conventional quantifiers or measure adverbials.

4.1 Quantified terms

There are two kinds of quantified terms—those pertaining to objects, and those pertaining to events. Both of these contribute to extended event structure, but in slightly different ways. Quantified terms from objects suggest ordinary set structure[6], while quantified terms from events—such as measure adverbials, and prepositional phrases—produce delimited masses and ordered sets.

Quantified terms corresponding to objects are represented by feature structure of the form in (30a), that are patients or recipients[7] of actions, while those coming from events are of the form shown in (30b), and are in a 'first-rest' list of modifiers. Those feature values beginning with uppercase letters indicate I am demonstrating the general form, rather than a specific example; they play the role of variables, and may represent actual values or feature structures.

(30) a.
$$\begin{bmatrix} \text{patient} = \begin{bmatrix} \text{term} = \begin{bmatrix} \text{det} = \text{Det} \\ \text{var} = \text{Var} \\ \text{restr} = \text{Restr} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

b.
$$\begin{bmatrix} \text{first} = \begin{bmatrix} \text{structure} = \text{Structure} \\ \text{index} = \text{Index} \\ \text{extent} = \text{Extent} \end{bmatrix} \\ \text{rest} = \text{Rest} \end{bmatrix}$$

A feature structure that is of either of the above forms will be identified as a term, and is a candidate for being pulled. This means it is appropriate to 'pull' that term during the scoping process.

4.2 Pulling terms

For any term that is identified, it is 'pulled' by substituting it with an appropriate feature structure, and moving the term, in an appropriate form, to the outside of the feature structure currently being scoped.

Quantified terms from objects Terms that are of the form in (30a) are replaced with the feature structure [`patient=Var`]. The feature structure is pulled to the outside and a new feature structure of the form

[5]Here, syntactically refers to the syntax of the feature structure. I note that there are additional notions of syntax, particularly with regard to the relationship of sub-intervals, that should also affect well-formedness, which are not discussed here.

[6]The concepts of sets and masses, as used here, are discussed in Rock[10].

[7]Everything that applies to a [`patient=`] is applicable to a [`recipient=`]; for the rest of this discussion I do not explicitly refer to recipients.

shown in (31a) is created. The value of the [content=] feature is the old feature structure, with its term removed, and the [=Rest] feature or value left in its place.

Quantified terms from events Terms that are of the form in (30b) are replaced with the feature structure [dummyfirst=dummyrest][8]. The feature structure is pulled to the outside and a new feature structure of the form shown in (31b) is created. The value of the [content=] feature is the old feature structure (with its replaced term).

(31) a.
$$\begin{bmatrix} \text{structure} = \text{ordinary-set} \\ \text{index} = \textit{Nextevent} \\ \text{extent} = \begin{bmatrix} \text{Det} = \text{Var} \\ \text{restr} = \text{Restr} \end{bmatrix} \\ \text{content} = \_ \end{bmatrix}$$

b.
$$\begin{bmatrix} \text{structure} = \text{Structure} \\ \text{index} = \text{Index} \\ \text{extent} = \text{Extent} \\ \text{content} = \_ \end{bmatrix}$$

4.3 The overall scoping process

I begin with a clarification of the distinction between being scoped and being pulled:

- A feature structure is a candidate for being scoped if it contains at its topmost level, a feature of the form [structure=X]. This means that it is appropriate to look in that feature structure for quantified terms, and to pull any quantified terms found within it to a position that is immediately outside of the feature structure. In the feature structures in (32), which correspond to different syntactic structures for '*Simmer the soup stirring occasionally for five minutes*', there are in fact three feature structures that are candidates for being scoped. These are the ones beginning with [structure=concurrent], [structure=happening] and [structure=happening].
- A term (itself in the form of a feature structure) is a candidate for being pulled if it is an appropriate quantified term, as described in Section 4.1. In the feature structures in (32), there is only one of these—the feature [first=Firstmod]

The scoping of an entire feature structure is done by recursively scoping the feature structures within it. The quantified term to be pulled can only be pulled to a position that is immediately outside of the candidate for scoping in which it is found. Thus the scoping process is controlled. The unscoped feature structure in (32a), which is a simplified representation of '*Simmer the soup, stirring it, for five minutes*'[9], demonstrates why this is important. In this, it is the happening e9 that is being modified; when scoping is performed, it is important that this modifier ends up surrounding the happening, and not the entire concurrent event e6. In (32b) however, which represents an alternative syntactic structure of the same sentence[10], the modifier should be pulled to scope the entire activity of e6.

[8] I note that it might be preferable to replace these with empty feature structures; however this was an implementation restriction.
[9] This corresponds to the the syntactic analysis where '*for fifteen minutes*' is modifying only '*simmer the soup*'
[10] Here, '*for fifteen minutes*' modifies '*simmer the soup stirring occasionally*'

(32) a.

$$\begin{bmatrix} \text{structure} = \text{concurrent} \\ \text{index} = \text{e6} \\ \text{substance} = \begin{bmatrix} \text{major} = \begin{bmatrix} \text{structure} = \text{happening} \\ \text{index} = \text{e2} \\ \text{substance} = \text{Substance1} \\ \text{modifiers} = \text{end} \end{bmatrix} \\ \text{minor} = \begin{bmatrix} \text{structure} = \text{happening} \\ \text{index} = \text{e9} \\ \text{substance} = \text{Substance2} \\ \text{modifiers} = \begin{bmatrix} \text{first} = \text{Firstmod} \\ \text{rest} = \text{end} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

b.

$$\begin{bmatrix} \text{structure} = \text{concurrent} \\ \text{index} = \text{e6} \\ \text{substance} = \begin{bmatrix} \text{major} = \begin{bmatrix} \text{structure} = \text{happening} \\ \text{index} = \text{e2} \\ \text{substance} = \text{Substance1} \\ \text{modifiers} = \text{end} \end{bmatrix} \\ \text{minor} = \begin{bmatrix} \text{structure} = \text{happening} \\ \text{index} = \text{e9} \\ \text{substance} = \text{Substance2} \end{bmatrix} \end{bmatrix} \\ \text{modifiers} = \begin{bmatrix} \text{first} = \text{Firstmod} \\ \text{rest} = \text{end} \end{bmatrix} \end{bmatrix}$$

The algorithm that performs the overall scoping is thus a controlled version of the basic Hobbs & Shieber algorithm, which recursively traverses a feature structure, to find the innermost [`structure=`] feature, and then performs scoping—as it comes out of the recursion—to the level of the feature structure, which will be a [`=happening`], in which the term was found.

The algorithm can be summarised as follows:

```
To scope a feature structure FS --
  If FS directly includes a feature structure of the form
          'structure : FS2'
  then
     1. scope all the other parts of FS
     2. pull
          'structure : FS2'


To pull a feature-value pair  'structure : FS' --
  Non-deterministically select quantified terms within FS, and
    for each one, apply the quantified term to 'structure : FS'
    to construct a scoped feature structure
```

```
To apply a quantified term Q to a feature structure FS,
  and return NewFS --
    If Q is of the form
      'first : Fvalue'
    then
      1. substitute an empty feature structure for Q in FS
      2. make NewFS, which is of the form
              '[Fvalue,
                content : FS]'

    If Q is of the form
      'patient : [ term : [det : Det, var : Var, restr : Restr ]]'
    then
      1. substitute 'patient : Var' for Q in FS
      2. make NewFS, which is of the form
              '[structure : ordin-set,
                index     : Nextindex,
                extent : [Det : Var,
                          restr : Restr],
                content : FS ]
```

5 Summary

I end this paper with a return to where I began—and the demonstration of the five readings in (1), repeated here.

(1) Simmer the soup stirring occasionally for fifteen minutes

This sentence has three syntactic readings if we omit punctuation[11]. For each reading, the modifiers—the temporal measure adverbials that come from ≪*occasionally*≫ and ≪*for fifteen minutes*≫—attach to different parts of the syntactic structure.

They can be summarised as follows for the three structures

1. We have a core sentence,'*simmer the soup stirring it*'—which means that we have a concurrent activity that consists of some ≪*simmering of soup*≫ and some ≪*stirring it*≫—that is modified by both modifiers.

2. We have a concurrent activity, consisting of ≪*simmering the soup*≫ at the same time as ≪*stirring it occasionally*≫, and this concurrent activity is modified by the ≪*for fifteen minutes*≫ modifier.

3. We again have a concurrent activity, but in this case we have a ≪*simmering the soup*≫ at the same time as ≪*stirring it occasionally for fifteen minutes*≫. That is, both '*occasionally*' and '*for fifteen minutes*' modify only '*stirring it*'.

Now for the first and third of these syntactic structures, because in these both modifiers are at the same structural level (in the first case they modify the concurrent activity, while in the third they modify the ≪*stirring*≫ sub-activity), each has two potential scopings. The second syntactic structure only has one scoping; each modifier modifies one of the concurrent activities. Thus in total there are five possible readings for this sentence, once a complete semantic analysis has been done. Capturing all of these possibilities is

[11] I note that with punctuation, some of these readings could be excluded. However, this approach allows all possibilities, and punctuation can later be used to exclude where appropriate.

possible due to the 'temporal measure adverbials as quantifiers' approach argued for earlier. In addition, the use of feature structures has provided a degree of structuring that has allowed modifiers to be attached at different levels, thereby highlighting the significance of the scoping. Although it is possible to code up such structure in logic, the feature structure representation makes it clearly accessible.

The feature structures for the five scoped versions are shown below; these have been edited to demonstrate only structure.

```
``Simmering the soup and stirring it, for fifteen minutes,
  and doing this occasionally''
structure : ordered_set
extent :  occasional
content : structure : delimitedmass
          extent : 15 minutes
          content : structure : concurrent
                    substance : major : structure : happening
                                        substance : action : simmer
                                                    patient : soup
                                        type : continuous
                                minor : structure : happening
                                        substance : action : stir
                                                    patient : referent
                                        type : discrete


``Simmering the soup and stirring it, doing this occasionally,
  for fifteen minutes''
structure : delimitedmass
extent : 15 minutes
content : structure : ordered_set
          extent : occasional
          content: structure : concurrent
                    substance : major : structure : happening
                                        substance : action : simmer
                                                    patient : soup

                                        type : continuous
                                minor : structure : happening
                                        substance : action : stir
                                                    patient : referent
                                        type : discrete

``Simmering the soup and stirring it occasionally,
  and doing this for fifteen minutes''
structure : delimitedmass
extent : 15 minutes
content : structure : concurrent
          substance : major : structure : happening
                              substance : action : simmer
                                          patient : soup
                              type : continuous
                      minor : structure : ordered_set
                              extent : occasional
                              content : structure : happening
                                        substance : action : stir
                                                    patient : referent
```

```
''Simmering the soup, at the same time as stirring it
  for fifteen minutes occasionally''
structure : concurrent
substance : major : structure : happening
                    substance : action : simmer
                                patient : soup
                    type : continuous
            minor : structure : ordered_set
                    extent : occasional
                    content : structure : delimitedmass
                              extent : 15 minutes
                              content : structure : happening
                                        substance : action : stir
                                                    patient : referent

''Simmering the soup, at the same time as stirring
  it occasionally for fifteen minutes''
structure : concurrent
substance : major : structure : happening
                    substance : action : simmer
                                patient : soup
                    type : continuous
            minor : structure : delimitedmass
                    extent : 15 minutes
                    content : structure : ordered_set
                              extent : occasional
                              content : structure : happening
                                        substance : action : stir
                                                    patient : referent
```

The importance of this example is in demonstrating that adopting the quantifier approach allows us to successfully predict all the meaningful readings of a sentence; this includes valid readings that were not initially apparent, certainly not obvious, and would not be found using the approach that views temporal measure adverbials as event predicates.

References

[1] Barwise, J. and Cooper, R. Generalised quantifiers and natural language. *Linguistics and Philosophy*, 4(2):159–219, 1981.

[2] Dowty, D. *Word Meaning and Montague Grammar*. D.Reidel, Dordrecht, 1979.

[3] Hitzeman, J. *Temporal Adverbials and the Syntax-Semantics Interface*. PhD thesis, University of Rochester, 1993.

[4] Hobbs, J.R. and Shieber, S.M. An algorithm for generating quantifier scopings. *Computational Linguistics*, 13:47–63, 1987.

[5] Jackendoff, R. Parts and boundaries. *Cognition*, 41:9–45, 1991.

[6] Karlin, R. Defining the semantics of verbal modifiers in the domain of cooking tasks. In *Proc. 26th Annual Meeting of the Association for Computational Linguistics*, pages 61–67, Buffalo NY, USA, June 1988.

[7] Mayo, B. Objects, events and complementarity. *Philosophical Review*, LXX:340–361, 1961.

[8] Moltmann, F. Measure adverbials. *Linguistics and Philosophy*, 14:629–660, 1991.

[9] Pratt, I. & Brée, D. The expressive power of the english temporal prepositional system. Technical Report UMCS-93-1-7, University of Manchester, Department of Computer Science, 1993.

[10] Rock, S.T. *Understanding natural language about multiple eventualities and continuous eventualities.* PhD thesis, University of Edinburgh, 1996.

[11] Zemach, E. Four ontologies. In F. J. Pelletier, editor, *Mass Terms : some philosophical problems*, pages 63–80. D.Reidel, Dordrecht, 1979.