

Temporality and Quantification

Ron Artstein

ESSLLI 2004 – The 16th European Summer
School in Logic, Language and Information

Nancy, France
9–20 August, 2004

<http://esslli2004.loria.fr/>

Note This reader is an outline of my course plan. It is intended to serve as auxiliary notes, not as a comprehensive background reading. I plan to cover most of the material here, but there's a good chance I will skip some of it or introduce additional material. I hope to have an active and lively class, which would increase the likelihood of straying... At any rate, this reader is a plan which is subject to change. I will be happy if you send me comments, and point out any errors.

Ron Artstein, June 2004, artstein@cs.technion.ac.il

Contents

Class 1	Temporal logic	1
1.1	Propositional tense logic	1
1.2	An additional operator: “now”	2
1.3	Tensed predicate logic	4
1.4	Explicit representation of times	6
Class 2	Tense and time adverbs	8
2.1	Tenses as operators: problems	8
2.2	Tenses as predicates	10
2.3	Instants and intervals	10
2.4	<i>Before</i> and <i>after</i>	11
Class 3	Generalized quantifiers	13
3.1	Syntactic motivation	13
3.2	Conservativity	14
3.3	Conjunction and type shifting	15
Class 4	Quantificational temporal modifiers	18
4.1	Basic sentence meanings	19
4.2	Temporal generalized quantifiers	20
4.3	The scope of temporal generalized quantifiers	22
4.4	Temporal cascades	24
4.5	Temporal modification of nouns	25
4.6	Summary	26
Class 5	Temporal clauses	27
5.1	Basic clause meanings	27
5.2	Single-time readings	28
5.3	Dependent-time readings	29
5.4	Temporally modified temporal clauses	31
	Bibliography	35

Class 1

Temporal logic

1.1 Propositional tense logic

The foundations of propositional tense logic are associated with the work of Arthur N. Prior (Prior 1957, 1967, 1968). In this class we will develop a simplified system, in order to get a general idea of how an existing logic can be extended to interpret propositions at various times.

We start with a simple propositional logic. I will only give the definitions of the semantics, and let the syntax follow implicitly.

Definition 1.1 A model M for propositional logic consists of a valuation function $\llbracket \cdot \rrbracket_M$ which assigns to each proposition letter a truth value (0 or 1). Complex expressions are interpreted as follows: if ϕ and ψ are formulas, then

$$\llbracket (\neg\phi) \rrbracket_M = 1 \text{ iff } \llbracket \phi \rrbracket_M = 0.$$

$$\llbracket (\phi \wedge \psi) \rrbracket_M = 1 \text{ iff } \llbracket \phi \rrbracket_M = 1 \text{ and } \llbracket \psi \rrbracket_M = 1.$$

$$\llbracket (\phi \vee \psi) \rrbracket_M = 1 \text{ iff } \llbracket \phi \rrbracket_M = 1 \text{ or } \llbracket \psi \rrbracket_M = 1.$$

$$\llbracket (\phi \rightarrow \psi) \rrbracket_M = 1 \text{ iff } \llbracket \phi \rrbracket_M = 0 \text{ or } \llbracket \psi \rrbracket_M = 1.$$

$$\llbracket (\phi \leftrightarrow \psi) \rrbracket_M = 1 \text{ iff } \llbracket \phi \rrbracket_M = \llbracket \psi \rrbracket_M.$$

This simple logic is extended to allow propositions to have different truth values at different times (cf. Gamut 1991: 32–40).

Definition 1.2 A model M for propositional tense logic consists of:

- a. a non-empty set I of temporal instants, ordered by a precedence relation $<$ which is a strict total order (transitive, irreflexive, asymmetric).
- b. a valuation function $\llbracket \cdot \rrbracket_M$ which assigns to each proposition letter p and instant i in I a truth value $\llbracket p \rrbracket_{M,i}$ (0 or 1).

Complex expressions are interpreted as follows: if ϕ and ψ are formulas, then

$$\llbracket (\neg\phi) \rrbracket_{M,i} = 1 \text{ iff } \llbracket \phi \rrbracket_{M,i} = 0.$$

$$\llbracket (\phi \wedge \psi) \rrbracket_{M,i} = 1 \text{ iff } \llbracket \phi \rrbracket_{M,i} = 1 \text{ and } \llbracket \psi \rrbracket_{M,i} = 1.$$

$$\llbracket (\phi \vee \psi) \rrbracket_{M,i} = 1 \text{ iff } \llbracket \phi \rrbracket_{M,i} = 1 \text{ or } \llbracket \psi \rrbracket_{M,i} = 1.$$

$$\llbracket (\phi \rightarrow \psi) \rrbracket_{M,i} = 1 \text{ iff } \llbracket \phi \rrbracket_{M,i} = 0 \text{ or } \llbracket \psi \rrbracket_{M,i} = 1.$$

$$\llbracket (\phi \leftrightarrow \psi) \rrbracket_{M,i} = 1 \text{ iff } \llbracket \phi \rrbracket_{M,i} = \llbracket \psi \rrbracket_{M,i}.$$

$$\llbracket \mathbf{P}(\phi) \rrbracket_{M,i} = 1 \text{ iff for some instant } i' \text{ in } I \text{ such that } i' < i, \llbracket \phi \rrbracket_{M,i'} = 1.$$

$$\llbracket \mathbf{F}(\phi) \rrbracket_{M,i} = 1 \text{ iff for some instant } i' \text{ in } I \text{ such that } i < i', \llbracket \phi \rrbracket_{M,i'} = 1.$$

$$\llbracket \mathbf{H}(\phi) \rrbracket_{M,i} = 1 \text{ iff for each instant } i' \text{ in } I \text{ such that } i' < i, \llbracket \phi \rrbracket_{M,i'} = 1.$$

$$\llbracket \mathbf{G}(\phi) \rrbracket_{M,i} = 1 \text{ iff for each instant } i' \text{ in } I \text{ such that } i < i', \llbracket \phi \rrbracket_{M,i'} = 1.$$

Note: Parentheses will be omitted when no ambiguity may arise. Unary operators are interpreted prior to binary operators.

We say that a formula ϕ is *true in a model* M ($\llbracket \phi \rrbracket_M = 1$) if for all i in I , $\llbracket \phi \rrbracket_{M,i} = 1$; a formula is called a *theorem* if it is true in any model.

Exercise 1.1 *Prove the following theorems.*

- i. $\phi \rightarrow \mathbf{GP}\phi$
- ii. $\mathbf{P}\phi \rightarrow \mathbf{GP}\phi$
- iii. $\mathbf{PP}\phi \rightarrow \mathbf{P}\phi$
- iv. $\mathbf{P}\phi \leftrightarrow \neg \mathbf{H}\neg \phi$

Exercise 1.2 *What condition on $<$ would make the following into a theorem?*

- v. $\mathbf{P}\phi \rightarrow \mathbf{PP}\phi$

1.2 An additional operator: “now”

The operators $\mathbf{P}$, $\mathbf{F}$, $\mathbf{H}$ and $\mathbf{G}$ shift the interpretation to the past and to the future. How about an operator for the present? This may seem redundant; however, Kamp (1971: 229) observes that the term *now* in English can affect the meaning of a sentence.

- (1) I learned last week that there would be an earthquake.
- (2) I learned last week that there would now be an earthquake.

What *now* does is shift the time of interpretation back to the utterance time, regardless of any temporal operators that it's embedded under. The same is true of the English present tense (Kamp 1971: 231).

- (3) A child was born that would become ruler of the world.
- (4) A child was born that will become ruler of the world.

(This time-shifting property of the present tense does not hold for all languages: Sharvit 2003 observes a correlation between the obligatory reference of present tense to utterance time and the availability of a non-shifted reading of embedded past tense.)

We want to define a tense operator $\mathbf{N}$, which will act as the adverbial *now* or the present tense. However, its semantics cannot be defined in the above system, because formulas are interpreted with respect to a temporal index corresponding to the event time, and the valuation function cannot keep track of the utterance time. We therefore redefine the valuation function so that each formula gets a meaning relative to two indices – event time and utterance time.

Definition 1.3 A model M for propositional tense logic with *now* consists of:

- a. a non-empty strict total order $\langle I, < \rangle$ of temporal instants.
- b. a valuation function $\llbracket \cdot \rrbracket_M$ which assigns to each proposition letter p and pair of instants i, j in I a truth value $\llbracket p \rrbracket_{M,i,j}$ (0 or 1).

Complex expressions are interpreted as follows: if ϕ and ψ are formulas, then

$$\llbracket (\neg\phi) \rrbracket_{M,i,j} = 1 \text{ iff } \llbracket \phi \rrbracket_{M,i,j} = 0.$$

$$\llbracket (\phi \wedge \psi) \rrbracket_{M,i,j} = 1 \text{ iff } \llbracket \phi \rrbracket_{M,i,j} = 1 \text{ and } \llbracket \psi \rrbracket_{M,i,j} = 1.$$

$$\llbracket (\phi \vee \psi) \rrbracket_{M,i,j} = 1 \text{ iff } \llbracket \phi \rrbracket_{M,i,j} = 1 \text{ or } \llbracket \psi \rrbracket_{M,i,j} = 1.$$

$$\llbracket (\phi \rightarrow \psi) \rrbracket_{M,i,j} = 1 \text{ iff } \llbracket \phi \rrbracket_{M,i,j} = 0 \text{ or } \llbracket \psi \rrbracket_{M,i,j} = 1.$$

$$\llbracket (\phi \leftrightarrow \psi) \rrbracket_{M,i,j} = 1 \text{ iff } \llbracket \phi \rrbracket_{M,i,j} = \llbracket \psi \rrbracket_{M,i,j}.$$

$$\llbracket \mathbf{P}(\phi) \rrbracket_{M,i,j} = 1 \text{ iff for some instant } i' \text{ in } I \text{ such that } i' < i, \llbracket \phi \rrbracket_{M,i',j} = 1.$$

$$\llbracket \mathbf{F}(\phi) \rrbracket_{M,i,j} = 1 \text{ iff for some instant } i' \text{ in } I \text{ such that } i < i', \llbracket \phi \rrbracket_{M,i',j} = 1.$$

$$\llbracket \mathbf{H}(\phi) \rrbracket_{M,i,j} = 1 \text{ iff for each instant } i' \text{ in } I \text{ such that } i' < i, \llbracket \phi \rrbracket_{M,i',j} = 1.$$

$$\llbracket \mathbf{G}(\phi) \rrbracket_{M,i,j} = 1 \text{ iff for each instant } i' \text{ in } I \text{ such that } i < i', \llbracket \phi \rrbracket_{M,i',j} = 1.$$

$$\llbracket \mathbf{N}(\phi) \rrbracket_{M,i,j} = 1 \text{ iff } \llbracket \phi \rrbracket_{M,j,j} = 1.$$

In order to capture the intuition that the default interpretation of a formula (without any temporal operators) is in the present, we give the following definition of truth: a formula ϕ is *true in a model M at an instant i* if $\llbracket \phi \rrbracket_{M,i,i} = 1$; a formula ϕ is *true in a model M* ($\llbracket \phi \rrbracket_M = 1$) if for all i in I , $\llbracket \phi \rrbracket_{M,i,i} = 1$; a formula is called a *theorem* if it is true in any model.

Exercise 1.3 Show that the theorems from exercise 1.1 are still valid.

- vi. $\phi \rightarrow \mathbf{GP}\phi$
- vii. $\mathbf{P}\phi \rightarrow \mathbf{GP}\phi$
- viii. $\mathbf{PP}\phi \rightarrow \mathbf{P}\phi$
- ix. $\mathbf{P}\phi \leftrightarrow \neg\mathbf{H}\neg\phi$

Exercise 1.4 *Prove the following theorems.*

$$\text{x. } \phi \leftrightarrow \mathbf{N}\phi$$

$$\text{xi. } \mathbf{N}\phi \leftrightarrow \mathbf{F}\mathbf{N}\phi$$

Exercise 1.5 *Show that the following are not theorems (i.e. there are models in which they are false).*

$$\text{xii. } \mathbf{P}\phi \rightarrow \mathbf{P}\mathbf{N}\phi$$

$$\text{xiii. } \mathbf{P}\mathbf{N}\phi \rightarrow \mathbf{P}\phi$$

Using the above semantics, we can give a very rough expression of sentence (4) using the following formula.

$$\mathbf{P}(\text{a child is born} \wedge \mathbf{N}\mathbf{F}(\text{he is ruler of the world}))$$

But is the temporal operator **N** really necessary? The above formula is equivalent to the following one, without **N**.

$$\mathbf{P}(\text{a child is born}) \wedge \mathbf{F}(\text{he is ruler of the world})$$

Kamp (1971) shows that in general, any formula in a propositional logic which uses the operator **N** has an equivalent formula without **N**, making the operator in a sense redundant. However, this is not the case for predicate logic: a predicate logic with the operator **N** has formulas for which there exists no equivalent formula without **N**.

Exercise 1.6 *Kamp (1971: 232) gives the following predicate logic formula as the meaning representation for sentence (4).*

$$\mathbf{P}(\exists x)(x \text{ is born} \wedge \mathbf{N}\mathbf{F}(x \text{ is ruler of the world}))$$

*Is this an example of a formula that lacks an equivalent without **N**?*

*To answer this question, first extend the tensed predicate logic of definition 1.4 below by giving a two-index valuation function and adding the operator **N**. In the extended logic, compare Kamp's formula above to the following formula.*

$$(\exists x)(\mathbf{P}(x \text{ is born}) \wedge \mathbf{F}(x \text{ is ruler of the world}))$$

Are the two formulas equivalent?

1.3 Tensed predicate logic

We can take a predicate logic and add times to it the same way we did with a propositional logic.

Definition 1.4 A model M for tensed predicate logic consists of:

- a. a non-empty set E of individuals and a non-empty strict total order $\langle I, < \rangle$ of temporal instants.

The notation D_t will designate the set of truth values $\{0, 1\}$, D_e will designate the set of individuals E , and for any two types σ and τ , $D_{(\sigma, \tau)}$ will designate the set of functions from D_σ to D_τ .

- b. a valuation function $\llbracket \cdot \rrbracket_M$ which assigns to each constant c of any type σ and instant i in I a denotation $\llbracket c \rrbracket_{M,i}$ in D_σ .
- c. a set of assignment functions, such that each assignment function g assigns to each variable v of any type σ a denotation $\llbracket v \rrbracket_g$ in D_σ .

The notation g_u^x will be used to designate the assignment function which assigns the object u to the variable x , and is otherwise identical to g .

Complex expressions are interpreted as follows:

if c is a constant, then $\llbracket c \rrbracket_{M,g,i} = \llbracket c \rrbracket_{M,i}$

if v is a variable, then $\llbracket v \rrbracket_{M,g,i} = \llbracket v \rrbracket_g$

if a is an expression of type (σ, τ) and b is an expression of type σ , then

$$\llbracket a(b) \rrbracket_{M,g,i} = \llbracket a \rrbracket_{M,g,i}(\llbracket b \rrbracket_{M,g,i})$$

if x is a variable of type σ and ϕ is an expression of type t , then

$$\llbracket (\exists x)\phi \rrbracket_{M,g,i} = 1 \text{ iff for some object } u \text{ in } D_\sigma, \llbracket \phi \rrbracket_{M,g_u^x,i} = 1.$$

$$\llbracket (\forall x)\phi \rrbracket_{M,g,i} = 1 \text{ iff for each object } u \text{ in } D_\sigma, \llbracket \phi \rrbracket_{M,g_u^x,i} = 1.$$

if x is a variable of type σ and a is an expression of type τ , then

$$\llbracket (\lambda x)a \rrbracket_{M,g,i} \text{ is the function } h \text{ in } D_{(\sigma,\tau)} \text{ such that for each object } u \text{ in } D_\sigma, h(u) = \llbracket a \rrbracket_{M,g_u^x,i} = 1.$$

if ϕ and ψ are expressions of type t , then

$$\llbracket (\neg\phi) \rrbracket_{M,g,i} = 1 \text{ iff } \llbracket \phi \rrbracket_{M,g,i} = 0.$$

$$\llbracket (\phi \wedge \psi) \rrbracket_{M,g,i} = 1 \text{ iff } \llbracket \phi \rrbracket_{M,g,i} = 1 \text{ and } \llbracket \psi \rrbracket_{M,g,i} = 1.$$

$$\llbracket (\phi \vee \psi) \rrbracket_{M,g,i} = 1 \text{ iff } \llbracket \phi \rrbracket_{M,g,i} = 1 \text{ or } \llbracket \psi \rrbracket_{M,g,i} = 1.$$

$$\llbracket (\phi \rightarrow \psi) \rrbracket_{M,g,i} = 1 \text{ iff } \llbracket \phi \rrbracket_{M,g,i} = 0 \text{ or } \llbracket \psi \rrbracket_{M,g,i} = 1.$$

$$\llbracket (\phi \leftrightarrow \psi) \rrbracket_{M,g,i} = 1 \text{ iff } \llbracket \phi \rrbracket_{M,g,i} = \llbracket \psi \rrbracket_{M,g,i}.$$

$$\llbracket P(\phi) \rrbracket_{M,g,i} = 1 \text{ iff for some } i' \text{ in } I \text{ such that } i' < i, \llbracket \phi \rrbracket_{M,g,i'} = 1.$$

$$\llbracket F(\phi) \rrbracket_{M,g,i} = 1 \text{ iff for some } i' \text{ in } I \text{ such that } i < i', \llbracket \phi \rrbracket_{M,g,i'} = 1.$$

$$\llbracket H(\phi) \rrbracket_{M,g,i} = 1 \text{ iff for each } i' \text{ in } I \text{ such that } i' < i, \llbracket \phi \rrbracket_{M,g,i'} = 1.$$

$$\llbracket G(\phi) \rrbracket_{M,g,i} = 1 \text{ iff for each } i' \text{ in } I \text{ such that } i < i', \llbracket \phi \rrbracket_{M,g,i'} = 1.$$

A formula ϕ is *true in a model* M at an instant i ($\llbracket \phi \rrbracket_{M,i} = 1$) if for all assignment functions g , $\llbracket \phi \rrbracket_{M,g,i} = 1$; a formula ϕ is *true in a model* M ($\llbracket \phi \rrbracket_M = 1$) if for all i in I , $\llbracket \phi \rrbracket_{M,i} = 1$; a formula is called a *theorem* if it is true in any model.

Exercise 1.7 Which of the following formulas are theorems?

xiv. $P\exists x\phi \leftrightarrow \exists xP\phi$

xv. $G\exists x\phi \leftrightarrow \exists xG\phi$

xvi. $F\forall x\phi \leftrightarrow \forall xF\phi$

xvii. $H\forall x\phi \leftrightarrow \forall xH\phi$

Exercise 1.8 Of the above formulas that are not theorems: how can they be weakened to make a theorem?

Exercise 1.9 Are the theorems above intuitive? Do we want them to be theorems?

1.4 Explicit representation of times

The formulas from exercise 1.7 show us that that the operators **P** and **F** are similar to existential quantifiers, while **H** and **G** are similar to universal quantifiers. We can modify our language so that it expresses these operators explicitly, rather than evaluating expressions at various instants implicitly. To do this we need to add to our language constants and variables of temporal types.

Definition 1.5 A model M for tensed predicate logic with explicit representation of time consists of:

- a. a non-empty set E of individuals and a non-empty strict total order $\langle I, < \rangle$ of temporal instants.

The notation D_t will designate the set of truth values $\{0, 1\}$, D_e will designate the set of individuals E , D_i will designate the set of instants I , and for any two types σ and τ , $D_{(\sigma, \tau)}$ will designate the set of functions from D_σ to D_τ .

- b. a valuation function $\llbracket \cdot \rrbracket_M$ which assigns to each constant c of any type σ a denotation $\llbracket c \rrbracket_M$ in D_σ .
- c. a set of assignment functions, such that each assignment function g assigns to each variable v of any type σ a denotation $\llbracket v \rrbracket_g$ in D_σ .

The notation g_u^x will be used to designate the assignment function which assigns the object u to the variable x , and is otherwise identical to g .

Complex expressions are interpreted as follows:

if c is a constant, then $\llbracket c \rrbracket_{M,g} = \llbracket c \rrbracket_M$

if v is a variable, then $\llbracket v \rrbracket_{M,g} = \llbracket v \rrbracket_g$

if a is an expression of type (σ, τ) and b is an expression of type σ , then

$$\llbracket a(b) \rrbracket_{M,g} = \llbracket a \rrbracket_{M,g}(\llbracket b \rrbracket_{M,g})$$

if x is a variable of type σ and ϕ is an expression of type t , then

$$\llbracket (\exists x)\phi \rrbracket_{M,g} = 1 \text{ iff for some object } u \text{ in } D_\sigma, \llbracket \phi \rrbracket_{M,g_u^x} = 1.$$

$$\llbracket (\forall x)\phi \rrbracket_{M,g} = 1 \text{ iff for each object } u \text{ in } D_\sigma, \llbracket \phi \rrbracket_{M,g_u^x} = 1.$$

if x is a variable of type σ and a is an expression of type τ , then

$$\llbracket (\lambda x)a \rrbracket_{M,g} \text{ is the function } h \text{ in } D_{(\sigma, \tau)} \text{ such that for each object } u \text{ in } D_\sigma, \\ h(u) = \llbracket a \rrbracket_{M,g_u^x} = 1.$$

if ϕ and ψ are expressions of type t , then

$$\llbracket (\neg\phi) \rrbracket_{M,g} = 1 \text{ iff } \llbracket \phi \rrbracket_{M,g} = 0.$$

$$\llbracket (\phi \wedge \psi) \rrbracket_{M,g} = 1 \text{ iff } \llbracket \phi \rrbracket_{M,g} = 1 \text{ and } \llbracket \psi \rrbracket_{M,g} = 1.$$

$$\llbracket (\phi \vee \psi) \rrbracket_{M,g} = 1 \text{ iff } \llbracket \phi \rrbracket_{M,g} = 1 \text{ or } \llbracket \psi \rrbracket_{M,g} = 1.$$

$$\llbracket (\phi \rightarrow \psi) \rrbracket_{M,g} = 1 \text{ iff } \llbracket \phi \rrbracket_{M,g} = 0 \text{ or } \llbracket \psi \rrbracket_{M,g} = 1.$$

$$\llbracket (\phi \leftrightarrow \psi) \rrbracket_{M,g} = 1 \text{ iff } \llbracket \phi \rrbracket_{M,g} = \llbracket \psi \rrbracket_{M,g}.$$

A formula ϕ is *true in a model* M ($\llbracket \phi \rrbracket_M = 1$) if for all assignment functions g , $\llbracket \phi \rrbracket_{M,g} = 1$; a formula is called a *theorem* if it is true in any model.

Defining tense operators in the language. We first define the logical constant **prec**, which is a function of type $i \rightarrow i \rightarrow t$.

prec maps two intervals i and i' to 1 (true) just in case $i < i'$.

For clarity, we write **prec**(i)(i') as $i \text{ prec } i'$.

In our language, propositions are not expressions of type t , but rather of type it – functions from instants to truth values. We define our tense operators as modifiers of propositions, that is expressions of type $(it)it$.

$$\mathbf{P} = \lambda J \lambda i. \exists i' [i' \text{ prec } i \wedge J(i)]$$

$$\mathbf{F} = \lambda J \lambda i. \exists i' [i \text{ prec } i' \wedge J(i)]$$

$$\mathbf{H} = \lambda J \lambda i. \forall i' [i' \text{ prec } i \wedge J(i)]$$

$$\mathbf{G} = \lambda J \lambda i. \forall i' [i \text{ prec } i' \wedge J(i)]$$

A proposition $\lambda i. J(i)$ of type it can be considered *true in a model M at an instant j* if $\llbracket \lambda i. J(i) \rrbracket_M(j) = 1$

Class 2

Tense and time adverbs

2.1 Tenses as operators: problems

Tense operators shift the evaluation time of statements. This causes problems when two distinct expressions in a sentence seem to place a statement in the same time, for instance past tense and the adverb *yesterday* in the following sentence (Dowty 1982: 23).

- (1) John left yesterday.

In order to see the problem I will use a logical language much simpler than Dowty's – a propositional logic with explicit representation of times.

Definition 2.1 A model M for propositional tense logic with explicit representation of times consists of:

- a. a non-empty strict total order $\langle I, < \rangle$ of temporal instants.
- b. a valuation function $\llbracket \cdot \rrbracket_M$ which assigns to each constant c of any type σ a denotation $\llbracket c \rrbracket_M$ in D_σ .
- c. a set of assignment functions, such that each assignment function g assigns to each variable v of any type σ a denotation $\llbracket v \rrbracket_g$ in D_σ .

Complex expressions are interpreted as follows:

if c is a constant, then $\llbracket c \rrbracket_{M,g} = \llbracket c \rrbracket_M$

if v is a variable, then $\llbracket v \rrbracket_{M,g} = \llbracket v \rrbracket_g$

if a is an expression of type (σ, τ) and b is an expression of type σ , then

$$\llbracket a(b) \rrbracket_{M,g} = \llbracket a \rrbracket_{M,g}(\llbracket b \rrbracket_{M,g})$$

if x is a variable of type σ and ϕ is an expression of type t , then

$$\llbracket (\exists x)\phi \rrbracket_{M,g} = 1 \text{ iff for some object } u \text{ in } D_\sigma, \llbracket \phi \rrbracket_{M,g_u^x} = 1.$$

$$\llbracket (\forall x)\phi \rrbracket_{M,g} = 1 \text{ iff for each object } u \text{ in } D_\sigma, \llbracket \phi \rrbracket_{M,g_u^x} = 1.$$

if x is a variable of type σ and a is an expression of type τ , then

$\llbracket (\lambda x) a \rrbracket_{M,g}$ is the function h in $D_{(\sigma,\tau)}$ such that for each object u in D_σ ,
 $h(u) = \llbracket a \rrbracket_{M,g_u^x} = 1$.

if ϕ and ψ are expressions of type t , then

$$\begin{aligned}\llbracket (\neg\phi) \rrbracket_{M,g} &= 1 \text{ iff } \llbracket \phi \rrbracket_{M,g} = 0. \\ \llbracket (\phi \wedge \psi) \rrbracket_{M,g} &= 1 \text{ iff } \llbracket \phi \rrbracket_{M,g} = 1 \text{ and } \llbracket \psi \rrbracket_{M,g} = 1. \\ \llbracket (\phi \vee \psi) \rrbracket_{M,g} &= 1 \text{ iff } \llbracket \phi \rrbracket_{M,g} = 1 \text{ or } \llbracket \psi \rrbracket_{M,g} = 1. \\ \llbracket (\phi \rightarrow \psi) \rrbracket_{M,g} &= 1 \text{ iff } \llbracket \phi \rrbracket_{M,g} = 0 \text{ or } \llbracket \psi \rrbracket_{M,g} = 1. \\ \llbracket (\phi \leftrightarrow \psi) \rrbracket_{M,g} &= 1 \text{ iff } \llbracket \phi \rrbracket_{M,g} = \llbracket \psi \rrbracket_{M,g}.\end{aligned}$$

The following constants of type it will have the same denotation in every model (relative to the temporal domain in the model).

prec maps two intervals i and i' to 1 (true) just in case $i < i'$.

prev-day maps two intervals i and i' to 1 (true) just in case i is on the day that immediately precedes i' (this presupposes a division of the temporal domain into days).

For clarity, we write the expressions **prec**(i')(i) and **prev-day**(i')(i) as i **prec** i' and i **prev-day** i' respectively.

We now define temporal operators that use these constants; **P** is supposed to be the correlate of past tense, and **Y** – the correlate of the adverb *yesterday*.

$$P = \lambda J \lambda i. \exists i' [i' \text{ prec } i \wedge J(i')]$$

$$Y = \lambda J \lambda i. \exists i' [i' \text{ prev-day } i \wedge J(i')]$$

Exercise 2.1 Assume that **john-leave** is a constant of type it , true of all the instants in which John leaves. What do the following expressions denote?

- i. **PY(john-leave)**
- ii. **YP(john-leave)**

Is either expression a suitable representation of the sentence John left yesterday?

A related problem occurs with the following sentence (Partee 1973: 602).

- (2) I didn't turn off the stove.

Exercise 2.2 Assume that **I-turn-off-the-stove** is a constant of type it , true of all the instants in which I turn off the stove. Let $\mathfrak{N}$ be a high-type version of the negation operator: $\lambda J \lambda i. \neg J(i)$. What do the following expressions denote?

- iii. $\mathfrak{N}P(\mathbf{I-turn-off-the-stove})$
- iv. $P\mathfrak{N}(\mathbf{I-turn-off-the-stove})$

Is either expression a suitable representation of the sentence I didn't turn off the stove?

2.2 Tenses as predicates

Dowty (1982) offers a semantics in which tense and time adverbs are not operators that shift the time, but rather predicates that impose conditions on times. Tenses and certain adverbs make reference to the utterance time, so the evaluation function also needs to keep track of the utterance time. We revise our logical language in definition 2.1 by replacing clause b with the following:

- b. a valuation function $\llbracket \cdot \rrbracket_M$ which assigns to each constant c of any type σ and instant j in I a denotation $\llbracket c \rrbracket_{M,j}$ in D_σ .

We also replace each occurrence of $\llbracket \cdot \rrbracket_{M,\dots}$ in the definition with $\llbracket \cdot \rrbracket_{M,\dots,j}$.

We add a special indexical constant of type i , whose meaning is always the utterance time (this is *not* Dowty's constant t^* !).

$$\llbracket u \rrbracket_{M,j} = j$$

This constant is used in defining P and Y .

$$P = \lambda J \lambda i. i \text{ prec } u \wedge J(i)$$

$$Y = \lambda J \lambda i. i \text{ prev-day } u \wedge J(i)$$

Exercise 2.3 What do the expressions from the previous exercises denote now?

- v. $PY(\text{john-leave})$
- vi. $YP(\text{john-leave})$
- vii. $\mathfrak{N}P(\text{I-turn-off-the-stove})$
- viii. $P\mathfrak{N}(\text{I-turn-off-the-stove})$

Do they differ in meaning? Are they suitable representations of the sentences they're intended to stand for?

2.3 Instants and intervals

Partee (1973: 602–603) observes that sentence (2) *I didn't turn off the stove* makes reference to “a definite interval whose identity is generally clear from extralinguistic context”. A key term here is *interval*. Expression viii in exercise 2.3 $P\mathfrak{N}(\text{I-turn-off-the-stove})$ denotes the set of past times i such that I didn't turn off the stove at i . How can this set be connected to Partee's “definite interval”?

One possibility is to decide that the interval is the set of instants; this would amount to the following definition of truth: an expression J of type it is true relative to a contextually determined interval J' if for all instants i in J' , $J(i) = 1$.

Alternatively, we can decide that the basic units for evaluation of propositions are intervals. In this case the domain I is no longer a strictly ordered set of instants, but rather a set of temporal intervals with both a precedence relation $<$ (which may be only a partial order) and a subinterval relation $\subseteq$. Such a domain can be defined on a base of instants, or intervals may be considered primitive. Interval semantics raises many questions, for instance:

- What are the precedence relations between partly overlapping intervals? Fully overlapping intervals?
- What is the relation between truth at an interval and truth at a subinterval or superinterval?
- What does it mean to be false at an interval? How does this carry over to truth and falsity at a subinterval or superinterval?

These are all complicated questions, which we will not develop in this course. See Landman (1991: chapter 5) for references and the discussion of a particular proposal. For the discussion of temporal quantification in classes 4 and 5 it will be convenient to assume that some minimal interval structure is present, but we will not need to go into the full details (and complications!).

2.4 Before and after

We will now add constructs to the language in order to express the meanings of the prepositions *before* and *after*. We want to define the meanings of the symbols **B** and **A** to give translations like the following.

(3) John left before Sue arrived.

B(sue-arrive)(john-leave)

The temporal clause *before Sue arrived* is a constituent that functions as an adverbial, like *yesterday* or *on Tuesday*. On the way to determining the meaning of **B**, we look at the desired meaning of the expression **B**(sue-arrive), which functions as a one-place temporal predicate.

$\mathbf{B}(\text{sue-arrive}) = \lambda J \lambda i. \text{"before-sue-arrive"}(i) \wedge J(i)$

We now replace the pseudo-predicate "before-sue-arrive" by noting that it is true of an interval *i* if *i* precedes *some* interval at which the predicate **sue-arrive** holds (this is just an approximation and is open to debate, see discussion further down).

$\mathbf{B}(\text{sue-arrive}) = \lambda J \lambda i. \exists i' [\text{sue-arrive}(i') \wedge i \text{ prec } i'] \wedge J(i)$

Finally, we extract the meaning of **B** by abstracting over the temporal property **sue-arrive** from the previous definition. The result is a relational predicate, which maps two temporal properties to a third one. An analogous definition can be given for **A** ("after").

$\mathbf{B} = \lambda J' \lambda J \lambda i. \exists i' [J'(i') \wedge i \text{ prec } i'] \wedge J(i)$

$\mathbf{A} = \lambda J' \lambda J \lambda i. \exists i' [J'(i') \wedge i' \text{ prec } i] \wedge J(i)$

As with the relation **prec**, we will sometimes write $J \mathbf{B} J'$ instead of the more cumbersome $\mathbf{B}(J')(J)$.

Exercise 2.4 Presumably, the translation of *John left before Sue arrived* would also include tenses, as in the following examples.

ix. **P**(john-leave) **B** **P**(sue-arrive)

x. $P(\text{john-leave } B P(\text{sue-arrive}))$

Is there a difference between the above meanings? Is either of them the desired representation?

Exercise 2.5 Give a predicate meaning to F “future” along the lines of our meaning for P . Is there a difference between the two expressions below?

xi. $P(\text{john-leave}) B F(\text{sue-arrive})$

xii. $P(\text{john-leave } B F(\text{sue-arrive}))$

Is either of the above an adequate representation of a sentence in English? Which sentence?

Anscombe (1964) observed some curious differences between the behavior of *before* and *after*, which raise doubts about the correctness of the semantics we gave to B above. Here are some of the differences (not all of them are due to Anscombe).

Before is transitive, after is not. If Kennedy was assassinated before Johnson was president, and Johnson was president before King was assassinated, then Kennedy was assassinated before King was assassinated; but if Reagan was shot after he was president, and Reagan was president after the Challenger blew up, it does not follow that Reagan was shot after the Challenger blew up.

Before licenses Negative Polarity Items, after does not. It is possible to say in English *John left before Sue ever arrived*, but not **Sue arrived after John ever left*.

Before does not entail its complement, after does. The sentence *Beethoven died before he wrote ten symphonies* can be true while *Beethoven wrote ten symphonies* is false; similar examples can not be given with *after*.

One way of accounting for the differences (at least some of them) is to say that while “ ϕ after ψ ” should be true if ϕ takes place after *some* occurrence of ψ , the converse “ ϕ before ψ ” should be true if ϕ takes place before *all* occurrences of ψ .

Exercise 2.6 Modify the definition of B above to capture this new intuition.

For a detailed, guided exercise on the asymmetries between *before* and *after*, with particular emphasis on the Negative Polarity Item issue, see Landman (1991: 140–145). For an analysis of *before* and *after* as converses, see Beaver and Condoravdi (to appear).

Class 3

Generalized quantifiers

For this class we assume a logical language with lambda abstraction (at least second-order, that is with variables of type et), but with only one domain of individuals and consequently no temporal types, variables, constants etc.; we also ignore all tenses in the English examples.

3.1 Syntactic motivation

We want to find a systematic relation between natural language sentences with quantifiers and the logical formulas that represent their meanings, for example the following English sentences and their translations into first-order logic.

- (1) Every boy walks: $\forall x[\mathbf{boy}(x) \rightarrow \mathbf{walk}(x)]$
- (2) Some boy walks: $\exists x[\mathbf{boy}(x) \wedge \mathbf{walk}(x)]$

The relation should be *compositional*: it should respect the constituent structure of natural language, and assign a meaning representation to each constituent.

- (3) $[[\text{Every boy}]_{NP} \text{ walks}]_S$
- (4) $[[\text{Some boy}]_{NP} \text{ walks}]_S$

The words *boy* and *walks* translate as the constants **boy** and **walk** of type et , respectively. Since we know the meanings of these constituents as well as the desired meanings of sentences (1) and (2), we conclude that the meaning of the noun phrases *every boy* and *some boy* are functions that bring us from the former to the latter.

- (5) Every boy: $\lambda Q.\forall x[\mathbf{boy}(x) \rightarrow Q(x)]$
- (6) Some boy: $\lambda Q.\exists x[\mathbf{boy}(x) \wedge Q(x)]$

These NP denotations are functions of type $(et)t$, mapping properties of entities to truth values (they're properties of properties); they are called *generalized quantifiers*.

Going one step further, we can extract the meanings of *every* and *some* as functions that take us from properties to generalized quantifiers.

(7) Every: $\lambda P \lambda Q. \forall x [P(x) \rightarrow Q(x)]$

(8) Some: $\lambda P \lambda Q. \exists x [P(x) \wedge Q(x)]$

These are functions of type $(et)(et)t$ and are called *determiners*.

Many determiners and generalized quantifiers cannot be expressed as a lambda-abstract of a property variable over a first-order predicate logic formula. For example, there is no way to express the proposition “most boys walk” in first-order logic. The semantics can still be specified. For instance, the generalized quantifier *most boys* is a function which maps a property to true if it is a property that is true of most boys, and the determiner *most* maps any two properties P and Q to true if the property Q is true of most of the individuals of which P is true. A language with generalized quantifiers does not need to have property variables at all, nor predicate logic quantifiers like $\exists$ and $\forall$; see Barwise and Cooper (1981) for such a language.

Set notation. Instead of functions from objects to truth values we can talk about their characterizing sets. Thus, **boy** – the function which maps all the boys to 1 and all the non-boys to 0 – is the characteristic function of the set of boys. It is sometimes convenient to talk about denotations as sets, other times as functions. We can use the same symbols (like **boy**) in set notation; usually no confusion will arise as to whether we intend it as a function or a set.

Generalized quantifiers are the characteristic functions of sets of sets: the function denoted by *every boy* is true of a property Q if Q is true of all the boys; so it is true of a set Q if Q includes the set of boys. *Every boy* is thus the set of sets which are supersets of the set of boys.

(9) Every boy: $\{ Q \mid \mathbf{boy} \subseteq Q \}$

Exercise 3.1 Which set does the generalized quantifier *some boy* stand for? How about *no boys*, *few boys*, *most boys*, *at least five boys*?

Exercise 3.2 What would determiners represent as sets? Which sets do *every*, *some*, *no*, *few*, *most*, and *at least five* stand for?

3.2 Conservativity

We say that a generalized quantifier “lives on” a set A if it denotes the same function when the domain is restricted to the set A .

Definition 3.1 In a model with a domain E , a generalized quantifier T (a set of subsets of E) *lives on* a set $A \subseteq E$ if and only if for any set $X \subseteq E$,

$$X \in T \text{ if and only if } X \cap A \in T$$

A generalized quantifier can live on many sets. Trivially, every generalized quantifier lives on the entire domain. In a model where all men are people, the generalized quantifier denoted by *every man* will live on the set of men, and also on the set of people, but not necessarily on the set of women.

Exercise 3.3 Prove the following statements.

- i. If a generalized quantifier T lives on a set A , then it also lives on all sets B such that $A \subseteq B \subseteq E$.
- ii. If a generalized quantifier T lives on two sets A and B , then it also lives on $A \cap B$.

A determiner is called *conservative* if it maps each set A to a generalized quantifier that lives on A .

Definition 3.2 In a model with a domain E , a determiner D (a relation between subsets of E) is *conservative* if and only if for any sets $A, B \subseteq E$,

$$D(A)(B) \text{ if and only if } D(A)(A \cap B)$$

It appears that all natural language determiners are conservative. A classic example of something that looks like a determiner but is not conservative is *only*. If *only* is a determiner, then *only boys* should be a generalized quantifier – the set of all properties that are only true of boys, that is the set of subsets of **boy**: $\{X \mid X \subseteq \mathbf{boy}\}$. But this generalized quantifier doesn't live on the set **boy**: restricting our domain to the set of boys may change the denotation of the generalized quantifier. For instance, it may be that in a certain model both boys and girls talk; the set of talkers is not a subset of the set of boys: $\mathbf{talk} \not\subseteq \mathbf{boy}$, but the set of boy talkers is: $\mathbf{talk} \cap \mathbf{boy} \subseteq \mathbf{boy}$. Our conclusion is that *only* is not a determiner. This is supported by additional linguistic evidence, for instance *only* can modify various constituents that determiners can't, such as verb phrases (*John only eats cheese*) and preposition phrases (*Sue eats only with a fork*).

Keenan and Stavi (1986) show that the set of conservative determiners can be defined using a small generator set of basic determiners and the boolean operations of complement, meet and join.

3.3 Conjunction and type shifting

Generalized conjunction. The logical connector 'and' $\wedge$ is a truth function of type tt , mapping a pair of truth values to a third truth value. The natural language conjunction *and* can conjoin various constituents of different types, such as noun phrases (*every boy and some girl smiled*), transitive verbs (*John kissed and hugged Mike*), verb phrases (*John kissed Bill and hugged Mike*), sentences (*John laughs and Bill cries*), determiners (*at least three and at most five boys laughed*) and more. Gazdar (1980) offers a scheme for reducing such constituent coordination to propositional conjunction.

Definition 3.3 The generalized conjunction operation $\sqcap$ is defined as follows.

If a and b are expressions of type t , then $a \sqcap b = a \wedge b$.

If a and b are expressions of type $\sigma\tau$, with τ a conjoinable type (basically one "ending in t "), and x is a variable of type σ , then $a \sqcap b = \lambda x. a(x) \sqcap b(x)$.

Type shifting. Generalized quantifiers are functions of type $(et)t$, whose arguments are properties of individuals such as verb phrases. Transitive verbs like *kiss* are relations of type eet : $\lambda y \lambda x. \mathbf{kiss}(x, y)$. This type cannot combine with an object generalized quantifier; we therefore offer a rule of *type-shifting* which allows the transitive verb to accommodate an object of a higher type.

Definition 3.4 A relation R of type eet applies to a generalized quantifier of type $(et)t$ by raising to type $((et)t)et$:

$$R \mapsto \lambda T \lambda x. T(\lambda y. R(x, y))$$

Exercise 3.4 What is the translation of the verb phrase *kisses every boy* according to the above rule?

Exercise 3.5 Steedman (2003) gives a grammar in which transitive verbs remain in their original type, but the type of object generalized quantifiers shifts in order to accommodate transitive verb arguments. Write a type-shifting rule that has this effect.

Type ambiguity. (This is a simplified version of the argument in Partee and Rooth 1983; I will note major departures.) Transitive verbs are ambiguous between a low type and a type-raised version. This has consequences for conjunction, since conjunction at a low type leads to different results than conjunction at a high type. Assume the following translations: the transitive verbs *catch* and *eat* translate as the constants **catch**, **eat** of type eet , and the noun phrase *a fish* translates as the generalized quantifier $\lambda Q. \exists y [\mathbf{fish}(y) \wedge Q(y)]$ of type $(et)t$. We want to conjoin the verbs and apply the conjunct to the object, as in the sentence *John caught and ate a fish*.

Conjunction can be performed at a high or low type. Let's demonstrate first conjunction at the high type. We start with the type-shifted translations of the transitive verbs.

$$\text{catch: } \lambda T \lambda x. T(\lambda y. \mathbf{catch}(x, y)) \quad \text{eat: } \lambda T \lambda x. T(\lambda y. \mathbf{eat}(x, y))$$

These translations are conjoined according to the definition of generalized conjunction 3.3.

$$\begin{aligned} \text{catch and eat:} & \quad \lambda T \lambda x. T(\lambda y. \mathbf{catch}(x, y)) \sqcap \lambda T \lambda x. T(\lambda y. \mathbf{eat}(x, y)) \\ & = \lambda T. \lambda x. T(\lambda y. \mathbf{catch}(x, y)) \sqcap \lambda x. T(\lambda y. \mathbf{eat}(x, y)) \\ & = \lambda T \lambda x. T(\lambda y. \mathbf{catch}(x, y)) \wedge T(\lambda y. \mathbf{eat}(x, y)) \end{aligned}$$

And the conjoined verb phrase applies to the object.

$$\text{catch and eat a fish: } \lambda x. \exists y [\mathbf{fish}(y) \wedge \mathbf{catch}(x, y)] \wedge \exists y [\mathbf{fish}(y) \wedge \mathbf{eat}(x, y)]$$

The result: the sentence *John caught and ate a fish* will be true if there is a fish that he caught and a fish that he ate. This is not the natural reading of the sentence.

Next, we try conjunction at the low type. We conjoin the unshifted translations of the transitive verbs.

$$\text{catch and eat: } \mathbf{catch} \sqcap \mathbf{eat} = \lambda y. \mathbf{catch}(y) \sqcap \mathbf{eat}(y) = \lambda y \lambda x. \mathbf{catch}(x, y) \wedge \mathbf{eat}(x, y)$$

It is the conjoined meaning that is subject to the type shifting rule of definition 3.4 (Partee and Rooth only allow type shifting as a lexical rule, and use

quantifying-in in order to combine an unshifted transitive verb with a generalized quantifier object; I use type shifting here in order to avoid introducing a rule of quantifying-in).

catch and eat: $\lambda T \lambda x. T(\lambda y. \text{catch}(x, y) \wedge \text{eat}(x, y))$

The conjoined shifted verb applies to the object.

catch and eat a fish: $\lambda x. \exists y [\text{fish}(y) \wedge \text{catch}(x, y) \wedge \text{eat}(x, y)]$

The result: the sentence *John caught and ate a fish* will be true if there is one fish that he both caught and ate. This is the natural reading of the sentence. Our conclusion: conjunction of transitive verbs happens at the lower type.

However, Partee and Rooth observe that other conjoined transitive verbs show an exact opposite preference: the natural reading of *John wants and needs two secretaries* is that he wants two and needs two, but not necessarily the same two – not even any specific two secretaries. We can capture this by saying that the verbs *want* and *need* are not relations between individuals, but as their basic meanings are relations between an individual and a higher-order object, say a generalized quantifier. (In Partee and Rooth's treatment, such verbs take an even higher-order object, namely the *intension* of a generalized quantifier; this is because John may want or need a secretary and not want or need a lawyer, even if all the secretaries happen to be lawyers and vice versa. Here I want to avoid introducing intensionality, and having an extensional generalized quantifier object is enough to force conjunction to the higher type.)

want: $\lambda T \lambda x. \text{want}(x, T)$ need: $\lambda T \lambda x. \text{need}(x, T)$

want and need: $\lambda T \lambda x. \text{want}(x, T) \sqcap \lambda T \lambda x. \text{need}(x, T)$
 $= \lambda T. \lambda x. \text{want}(x, T) \sqcap \lambda x. \text{need}(x, T)$
 $= \lambda T \lambda x. \text{want}(x, T) \wedge \text{need}(x, T)$

Finally, coordination of a high-type transitive verb with a low-type verb also needs to be performed at the high type, since conjunction in definition 3.3 is only defined for expressions of the same type. This gives the correct prediction, that in a sentence like *John needed and bought a new coat*, it is not necessarily one coat that he both needed and bought. Conclusion: transitive verbs are coordinated at the lowest possible type, and are raised when needed.

Class 4

Quantificational temporal modifiers

The theory of quantificational temporal modifiers is based on the proposals in Pratt and Francez (1997, 2001), as developed further by von Stechow (2002), Francez and Steedman (to appear), Artstein (2004) and Artstein and Francez (2003). These works use somewhat different formalisms to account for similar facts, and here I will use the formalism of Artstein (2004). Actually, classes 4 and 5 follow Artstein (2004) rather closely – so closely, that most of the answers to the exercises can be found in the paper! Though you may be able to come up with better answers... ☺

We will use a two-sorted translation language with lambda abstraction, which represents times explicitly: e is the type of individuals and i of time intervals. Verbs and temporal nouns have temporal arguments: the basic meaning of an intransitive verb like *cry* is a relation between individuals and time intervals $\lambda x \lambda i. \text{cry}(x)(i)$ (type eit), and a noun like *meeting* in a temporal context is a property of time intervals $\lambda i. \text{meeting}(i)$ (type it) – those time intervals during which a meeting takes place.

The translation also explicitly represents temporal context. A distinguished free variable $\hat{t}$ stands for the overall temporal context of evaluation (its meaning is determined by an assignment function).

Temporal modification means subordinating the temporal context of a particular linguistic expression to that of another expression, through the binding of the free temporal context variable of the temporally subordinate expression. For a detailed discussion of why temporal modifiers should operate on the context of linguistic expressions see Pratt and Francez (2001: 200–206). Here's the crux of the argument. Modifying the event time, as in section 2.2 above (see also Dowty 1979; Stump 1985), works fine with non-quantificational temporal modifiers, as in the following example from Stump (1985: 103).

- (1) Yesterday John saw Mary in the morning, after bill arrived.

The temporal modifiers can apply in turn to a temporal property formed by abstracting over the event time of the verb; this results in a series of conjoined temporal predicates that do not stand in a scope relation to one another.

But when temporal modifiers are quantificational, they do stand in scope

relations: in the following example, the modifier *during most conferences* takes scope above *after each meeting*.

- (2) John saw Mary after each meeting during most conferences

In order to allow scoped relations, each modifier has to apply to a distinct temporal variable, but the verb (or any other temporal predicate) only has one event time. Temporal modifiers must therefore operate on something other than the event time of the expression they modify, and this is the context. Since each modifier introduces its own context, the number of modifiers is in principle unlimited and leads to the phenomenon of “cascading” (Pratt and Francez 2001, see also section 4.4 below).

4.1 Basic sentence meanings

Natural language predicates are indexed to time intervals. An intransitive verb like *cry* denotes a relation between individuals and time intervals (type eit); the denotation we get for a sentence like *Bill cried* is a property of time intervals – the set of times at which Bill cried (I ignore tense for the purpose of this class, but see exercise 4.1 below).

- (3) $\text{cry}: \lambda x \lambda i. \text{cry}(x)(i)$

- (4) *Bill cried*: $\lambda i. \text{cry}(\text{bill})(i)$

The sentence *Bill cried* is true if a time that satisfies (4) exists within the general time frame in which the sentence is evaluated. We express these truth conditions by altering the representation in (4), subjecting it to two operations that apply in succession: *contextualization* (5a) introduces the temporal context variable $\hat{t}$ which stands for the context of evaluation, and *existential closure* (6a) closes off the event time variable by turning the lambda operator into an existential quantifier (the latter is akin to the existential closure of an event argument in event semantics).

- (5) a. Contextualization operation (type $(\text{it})\text{it}$): $\lambda I \lambda i. i \subseteq \hat{t} \wedge I(i)$
 b. *Bill cried*: $\lambda i. i \subseteq \hat{t} \wedge \text{cry}(\text{bill})(i)$
- (6) a. Existential closure operation (type $(\text{it})t$): $\lambda I. \exists i [I(i)]$
 b. *Bill cried*: $\exists i [i \subseteq \hat{t} \wedge \text{cry}(\text{bill})(i)]$

The final representation (6b) is the translation of the sentence *Bill cried*; it is true with respect to a model M and an evaluation context $\hat{t}$ if there exists an interval i included in $\hat{t}$ such that Bill cried at i in M .

The inclusion relation between the event time and context is similar to the relation proposed by Kamp and Reyle (1993) between an eventuality and the discourse referent which plays the role of its “location time”. Temporal inclusion of an event time in its context is considered by von Stechow (2002: sections 10–11) to be the meaning of perfective aspect. In order to avoid the complications raised by aspectual class, I will always choose examples with predicates for which inclusion is the appropriate relation. Contextualization must precede existential closure because it has to apply at a point where the time variable of the predicate is accessible.

4.2 Temporal generalized quantifiers

Temporal modifiers like the PP *after each meeting* are temporal generalized quantifiers of type $(it)t$. I take common nouns like *meeting* to denote predicates of times rather than predicates of events when placed in a temporal context, e.g. as the complement of a temporal preposition; the temporal noun *meeting* is a predicate of type it .

(7) *meeting*: $\lambda i.\mathbf{meeting}(i)$

A predicate like *meeting* is itself evaluated with respect to the overall temporal context, so the temporal context variable $\hat{i}$ is introduced by the contextualization operation.

(8) *meeting*: $\lambda i.i \subseteq \hat{i} \wedge \mathbf{meeting}(i)$

The determiner *each* in a temporal NP like *each meeting* is similar to the familiar nominal determiner *each*, except that it denotes a relation between predicates of times rather than predicates of individuals – it is of type $(it)(it)t$ rather than type $(et)(et)t$. *Each* combines with the contextualized translation of the noun *meeting* in the obvious way.

(9) *each*: $\lambda I \lambda J.\forall i[I(i) \rightarrow J(i)]$

(10) *each meeting*: $\lambda J.\forall i[(i \subseteq \hat{i} \wedge \mathbf{meeting}(i)) \rightarrow J(i)]$

Since the common noun *meeting* was taken to be a predicate of times, it should come as no surprise that the expression *each meeting*, when placed in a temporal context, denotes a temporal generalized quantifier, that is a predicate of predicates of times.

The temporal preposition *after* translates as a function of type $((it)t)(it)t$, mapping temporal generalized quantifiers to temporal generalized quantifiers; this higher-order function is defined using the primitive temporal function $\lambda i \lambda j.\mathbf{after}_i(j)$ from interval pairs to intervals (type iii).

(11) $\mathbf{after}_i(j)$ is the interval spanning from the end of j to the end of i , if $j \subseteq i$; undefined otherwise.

The proposition that an interval j' follows j (within a context i) can be expressed by the formula $j' \subseteq \mathbf{after}_i(j)$.

As for the higher-order translation of the preposition *after*, we note that the temporal generalized quantifier *each meeting* (10) maps any temporal predicate J to the statement that J holds of each interval which is a meeting; the expression *after each meeting* should map a temporal predicate J to the statement that J holds after each interval which is a meeting. Therefore the preposition *after* should denote a function which takes a temporal generalized quantifier of the form $\lambda J.\phi$ and “injects” the temporal function **after** into the complement(s) of J in ϕ .

(12) *after*: $\lambda \mathcal{T}^{(it)t}.\lambda J.\mathcal{T}(\lambda i.J(\mathbf{after}_i(i)))$

The preposition *after* uses the overall temporal context of evaluation $\hat{i}$ as the context argument of the temporal primitive **after**; this will be used in the account of temporal cascades in section 4.4.

The temporal PP *after each meeting* gets its meaning by straightforward application of the preposition *after* (12) to the temporal generalized quantifier *each meeting* (10). The result is a temporal generalized quantifier.

$$(13) \text{ after each meeting: } \lambda J. \forall i[(i \subseteq \hat{t} \wedge \text{meeting}(i)) \rightarrow J(\text{after}_{\hat{t}}(i))]$$

The PP *after each meeting* modifies the sentence *Bill cried* (I assume for the moment that the PP attaches as an adjunct to the entire main clause, this will be revised in section 4.3). The temporal generalized quantifier (13) needs to apply to a temporal property; we form such a property by abstracting over the free temporal context variable $\hat{t}$ in the existentially closed meaning of the main clause (6b). This has the effect of subordinating the temporal context of the main clause to that of the temporal modifier.

$$\begin{aligned} (14) \text{ Bill cried after each meeting:} \\ \lambda J. \forall i[(i \subseteq \hat{t} \wedge \text{meeting}(i)) \rightarrow J(\text{after}_{\hat{t}}(i))](\lambda \hat{t}. \exists i' [i' \subseteq \hat{t} \wedge \text{cry}(\text{bill})(i')]) \\ = \forall i[(i \subseteq \hat{t} \wedge \text{meeting}(i)) \rightarrow \exists i' [i' \subseteq \text{after}_{\hat{t}}(i) \wedge \text{cry}(\text{bill})(i')]] \end{aligned}$$

We can view abstraction over a free temporal context variable as part of the rule of temporal modification, or as an independent operation that necessarily precedes it; an independent application of an abstraction operation will be needed for the computation of aggregate readings in section 5.4. In (23) below we will revise the abstraction rule to allow it to apply to types other than t .

Exercise 4.1 *How can tense be added to the system? Notice that sentence i can be uttered felicitously while the meeting is still going on, and that sentence ii can be uttered felicitously even though today is not in the past.*

- i. Bill cried before the meeting.
- ii. Bill cried today.

In light of this, should tense apply above or below the temporal modifier? Should it modify the context or the event time?

Exercise 4.2 *In sentence iii below, quoted by von Stechow (2002: 759) from Ogihara (1994), the domain of quantification is restricted to the past.*

- iii. John called every Monday.

How can tense enforce this kind of restriction?

Exercise 4.3 *Temporal primitives like **after** are functions of type ii ; this goes against a basic intuition, that **after** makes reference to a relation between times (recall that the primitive **prec** from section 2.1 was a relation of type ii). Can we replace our primitive **after** with **prec**? What changes would we have to make? What would be the benefits or disadvantages?*

Exercise 4.4 *Sentence (14) comes out true if Bill cries once, within the overall context after the last meeting. How can we get correct this undesirable behavior? Is there a way to refine the meaning of **after** so that it forces the interleaving of cryings between the meetings? (Kamp and Reyle 1993: 628 say that this is very difficult!)*

Exercise 4.5 Assume the obvious definitions for the temporal primitive **during** and the preposition *during*.

- iv. $\text{during}_i(j)$ is the interval j itself, if $j \subseteq i$; undefined otherwise.
- v. The preposition *during* translates as $\lambda T^{(it)t}.\lambda J.T(\lambda i.J(\text{during}_{\hat{c}}(i)))$.

It turns out, that the preposition *during* is almost redundant: applying the meaning in v. to a temporal generalized quantifier results in the original temporal generalized quantifier! Show how this follows from the assumptions that (a) temporal determiners only apply to contextualized NPs, and (b) natural language determiners are conservative.

4.3 The scope of temporal generalized quantifiers

Temporal generalized quantifiers can interact scopally with a quantificational argument of the verb. The following sentence displays a classical scope ambiguity.

- (15) $\left\{ \begin{array}{l} \text{Exactly three executives} \\ \text{Few executives} \\ \text{Some executive} \end{array} \right\}$ cried during each meeting.

The two scoped readings are independent of one another:

A wide scope reading of *exactly three executives* states that exactly three executives are such that they cried during each meeting (though at some or all of the meetings there may have been additional executives who cried).

A narrow scope reading of *exactly three executives* states that at each meeting, the number of executives who cried was exactly three (but not necessarily the same executives cried at all meetings).

In order to account for the scoped readings we need to resolve some type mismatches that arise in the course of semantic derivation, as well as generalize the temporal abstraction rule to apply to types other than t .

Scope ambiguities are usually dealt with using a scope mechanism, such as Quantifying-in (Montague 1973), Quantifier Storage (Cooper 1983) and Quantifier Raising (May 1985). The details depend on the theory of the syntax-semantics interface. In order to avoid the need to introduce a scope mechanism, I'll just read off the two readings from two imaginary surface structures.

- (16) a. $[[\text{Some executive cried}]]$ during each meeting].
 b. Some executive $[[\text{cried}]]$ during each meeting].

Wide scope temporal generalized quantifiers (16a). The meaning of the unmodified clause, *some executive cried*, is the result of applying the subject as well as contextualization and existential closure to the verb *cry*.

- (3) $\text{cry}: \lambda x \lambda i. \text{cry}(x)(i)$

In section 4.1 we saw how contextualization and existential closure apply to the entire sentence; contextualization and existential closure can also apply to the

verb directly (this can be motivated syntactically by using a mechanism that gives the subject scope above these operations, for instance Quantifying-in, Quantifier Storage or Quantifier Raising). In the sentence *some executive cried*, the relative scope of the subject and existential closure doesn't matter because they both contribute existential quantifiers. However, it is instructive to show how contextualization and existential closure apply to the verb, because this will be needed in for the narrow-scope temporal generalized quantifiers.

The contextualization operation (5a) is a function of type $(it)it$, and is thus of the wrong type to combine with the verb via functional application; it can however combine via functional composition.

$$(17) \text{ Functional composition: } f^{\sigma\tau} \circ g^{\rho\sigma} =_{df} \lambda r^{\rho}. f(g(r))$$

[ρ, σ, τ are variables over types]

$$(18) \text{ cry: } \lambda I \lambda i. i \subseteq \hat{I} \wedge I(i) \circ \lambda x \lambda i. \mathbf{cry}(x)(i) \\ = \lambda x. [\lambda I \lambda i. i \subseteq \hat{I} \wedge I(i)]([\lambda x \lambda i. \mathbf{cry}(x)(i)](x)) \\ = \lambda x \lambda i. i \subseteq \hat{I} \wedge \mathbf{cry}(x)(i)$$

The existentially closed meaning of the verb *cry* is likewise arrived at through functional composition of the existential closure operation (6a) with the contextualized meaning of the verb.

$$(19) \text{ cry: } \lambda I. \exists i [I(i)] \circ \lambda x \lambda i. i \subseteq \hat{I} \wedge \mathbf{cry}(x)(i) \\ = \lambda x. [\lambda I. \exists i [I(i)]]([\lambda x \lambda i. i \subseteq \hat{I} \wedge \mathbf{cry}(x)(i)](x)) \\ = \lambda x. \exists i [i \subseteq \hat{I} \wedge \mathbf{cry}(x)(i)]$$

The subject *some executive* is a familiar (nominal) generalized quantifier; it applies to the VP meaning to give the meaning of the sentence *some executive cried*, with a free temporal context variable.

$$(20) \text{ some executive: } \lambda P. \exists x [\mathbf{exec}(x) \wedge P(x)]$$

$$(21) \text{ some executive cried: } \\ \lambda P. \exists x [\mathbf{exec}(x) \wedge P(x)] (\lambda x. \exists i [i \subseteq \hat{I} \wedge \mathbf{cry}(x)(i)]) \\ = \exists x [\mathbf{exec}(x) \wedge \exists i [i \subseteq \hat{I} \wedge \mathbf{cry}(x)(i)]]$$

Finally, the temporal PP applies. Recall the result of exercise 4.5, which established that the meaning of *during each meeting* is identical to that of *each meeting* (10). This is a temporal generalized quantifier, and it applies to the $\lambda\hat{I}$ -abstract over the meaning of *some executive cried*.

$$(22) \text{ some executive cried during each meeting: } \\ \lambda J. \forall i [(i \subseteq \hat{I} \wedge \mathbf{meeting}(i)) \rightarrow J(i)] \\ (\lambda \hat{I}. \exists x [\mathbf{exec}(x) \wedge \exists i' [i' \subseteq \hat{I} \wedge \mathbf{cry}(x)(i')]]) \\ = \forall i [(i \subseteq \hat{I} \wedge \mathbf{meeting}(i)) \rightarrow \exists x [\mathbf{exec}(x) \wedge \exists i' [i' \subseteq i \wedge \mathbf{cry}(x)(i')]]]$$

The resulting meaning is that for each meeting there is an executive who cried during that meeting. Executives may vary with meetings, which is precisely the reading we want.

Narrow scope temporal generalized quantifiers (16b). The temporal PP has to apply to the verb before the subject does. As in the previous cases of temporal modification, the temporal generalized quantifier meaning of the modifier

has to combine with an abstract over the temporal context variable of the existentially closed meaning of the verb (19). Since the latter is of a type higher than t , simple prefixation of $\lambda\hat{t}$ to the verb meaning will not work. Rather, the lambda abstract over the temporal context variable is added to the verb meaning as the last lambda operator.

- (23) A temporal modifier denoting a temporal generalized quantifier $\mathcal{T}$ of type $(it)t$ combines with a constituent with a denotation $\lambda s_1 \dots \lambda s_n. \phi$ of type $\sigma_1 \dots \sigma_n t$ for some $n \geq 0$ via functional application or composition (whichever is appropriate) of $\mathcal{T}$ to the meaning $\lambda s_1 \dots \lambda s_n \lambda \hat{t}. \phi$ (the string of lambdas may be empty).

When the modified constituent is of type t , the rule results in prefixation of $\lambda\hat{t}$ as we have seen so far. When the rule is applied to a predicate of type et , the abstractor $\lambda\hat{t}$ skips over the initial λx ; this can be thought of as functional composition of the abstraction operation itself (thanks to Fred Landman for this observation).

The meaning of the VP *cried during each meeting* is derived through functional composition of the temporal generalized quantifier with the lambda abstract over the temporal context variable of the verb meaning (24); the VP then combines with the subject meaning (20) to give the meaning of the complete sentence (25).

- (24) *cried during each meeting*:

$$\begin{aligned} & \lambda J. \forall i [(i \subseteq \hat{t} \wedge \mathbf{meeting}(i)) \rightarrow J(i)] \circ \lambda x \lambda \hat{t}. \exists i' [i' \subseteq \hat{t} \wedge \mathbf{cry}(x)(i')] \\ &= \lambda x. [\lambda J. \forall i [(i \subseteq \hat{t} \wedge \mathbf{meeting}(i)) \rightarrow J(i)]] \\ & \quad ([\lambda x \lambda \hat{t}. \exists i' [i' \subseteq \hat{t} \wedge \mathbf{cry}(x)(i')]](x)) \\ &= \lambda x. \forall i [(i \subseteq \hat{t} \wedge \mathbf{meeting}(i)) \rightarrow \exists i' [i' \subseteq i \wedge \mathbf{cry}(x)(i')]] \end{aligned}$$
- (25) *some executive cried during each meeting*:

$$\begin{aligned} & \lambda P. \exists x [\mathbf{exec}(x) \wedge P(x)] \\ & \quad (\lambda x. \forall i [(i \subseteq \hat{t} \wedge \mathbf{meeting}(i)) \rightarrow \exists i' [i' \subseteq i \wedge \mathbf{cry}(x)(i')]]) \\ &= \exists x [\mathbf{exec}(x) \wedge \forall i [(i \subseteq \hat{t} \wedge \mathbf{meeting}(i)) \rightarrow \exists i' [i' \subseteq i \wedge \mathbf{cry}(x)(i')]]] \end{aligned}$$

This derivation entails the existence of one particular executive who cried at each meeting, which is the desired narrow-scope reading of the temporal generalized quantifier.

4.4 Temporal cascades

A temporally modified sentence can be modified again, resulting in what Pratt and Francez (2001) call temporal cascades. Take for example the sentence *Bill cried after each meeting on Tuesday*, where *Tuesday* is taken to refer to a particular day (e.g. Tuesday, August 17, 2004): the sentence is true if the sentence *Bill cried after each meeting* is true when evaluated on Tuesday. The word *Tuesday* will be translated as the constant **tues** of type it (a property of time intervals). Constituents like *every Tuesday* or *few Tuesdays* show that the function **tues** is true of all intervals which are Tuesdays, but when translating bare *Tuesday* the function is true of only one particular interval which is determined from context; how it gets chosen is a complicated matter which will not concern us here (for discussion see Kamp and Reyle 1993: 614ff).

Like other temporal predicates, *Tuesday* is interpreted in context (26); an implicit existential determiner meaning (27) turns the contextualized temporal property into a temporal generalized quantifier (28).

(26) Tuesday: $\lambda i. i \subseteq \hat{\tau} \wedge \mathbf{tues}(i)$

(27) Implicit existential determiner: $\lambda I \lambda J. \exists i [I(i) \wedge J(i)]$

(28) Tuesday: $\lambda J. \exists i [i \subseteq \hat{\tau} \wedge \mathbf{tues}(i) \wedge J(i)]$

The preposition *on*, like its counterpart *during*, does not change the meaning of its complement (exercise 4.5), and the resulting temporal generalized quantifier combines with the $\lambda\hat{\tau}$ -abstract over the meaning of the sentence *Bill cried after each meeting* (14).

(29) on Tuesday: $\lambda J. \exists i [i \subseteq \hat{\tau} \wedge \mathbf{tues}(i) \wedge J(i)]$

(30) Bill cried after each meeting on Tuesday:

$$\begin{aligned} & \lambda J. \exists i [i \subseteq \hat{\tau} \wedge \mathbf{tues}(i) \wedge J(i)] \\ & \quad (\lambda \hat{\tau} \forall i' [(i' \subseteq \hat{\tau} \wedge \mathbf{meeting}(i')) \rightarrow \exists i'' [i'' \subseteq \mathbf{after}_{\hat{\tau}}(i') \wedge \mathbf{cry}(\mathbf{bill})(i'')]]) \\ & = \exists i [i \subseteq \hat{\tau} \wedge \mathbf{tues}(i) \wedge \\ & \quad \forall i' [(i' \subseteq i \wedge \mathbf{meeting}(i')) \rightarrow \exists i'' [i'' \subseteq \mathbf{after}_i(i') \wedge \mathbf{cry}(\mathbf{bill})(i'')]]] \end{aligned}$$

This is the correct meaning of the sentence – there is some interval which is Tuesday, and each meeting during this interval was followed by an interval in which Bill cried.

4.5 Temporal modification of nouns

The sentence *Bill cried after each meeting on a Tuesday* has a reading which is true if there were several Tuesday meetings and Bill cried after each of them; the crying intervals themselves do not have to be on Tuesdays. This reading arises when the PP *on a Tuesday* modifies the noun *meeting* rather than the verb *cried* (Pratt and Francez 2001; Francez and Steedman to appear).

(31) Bill cried after [each [meeting on a Tuesday]]

We start with the modified common noun in (31). A $\lambda\hat{\tau}$ operator is added to the contextualized meaning of the noun *meeting* (8) according to the rule (23), and this composes with the temporal generalized quantifier *on a Tuesday*, yielding the representation in (32).

(32) meeting on a Tuesday:
 $\lambda J. \exists i [i \subseteq \hat{\tau} \wedge \mathbf{tues}(i) \wedge J(i)] \circ \lambda i' \lambda \hat{\tau}. i' \subseteq \hat{\tau} \wedge \mathbf{meeting}(i')$
 $= \lambda i'. \exists i [i \subseteq \hat{\tau} \wedge \mathbf{tues}(i) \wedge i' \subseteq i \wedge \mathbf{meeting}(i')]$

The temporal property (32) is true of an interval if it is both a meeting and on a Tuesday. The temporal determiner *each* applies to the above representation creating a temporal generalized quantifier (33); this is modified by the preposition *after*, and the resulting temporal generalized quantifier (34) applies to the sentence *Bill cried* (35).

(33) each meeting on a Tuesday:
 $\lambda J. \forall i' [\exists i [i \subseteq \hat{\tau} \wedge \mathbf{tues}(i) \wedge i' \subseteq i \wedge \mathbf{meeting}(i')] \rightarrow J(i')]$

- (34) after each meeting on a Tuesday:
 $\lambda J. \forall i' [\exists i [i \subseteq \hat{t} \wedge \mathbf{tues}(i) \wedge i' \subseteq i \wedge \mathbf{meeting}(i')] \rightarrow J(\mathbf{after}_{\hat{t}}(i'))]$
- (35) Bill cried after each meeting on a Tuesday:
 $\forall i' [\exists i [i \subseteq \hat{t} \wedge \mathbf{tues}(i) \wedge i' \subseteq i \wedge \mathbf{meeting}(i')] \rightarrow \exists i'' [i'' \subseteq \mathbf{after}_{\hat{t}}(i') \wedge \mathbf{cry}(\mathbf{bill})(i'')]]$

The final translation is indeed the desired reading: each interval which is a meeting on a Tuesday is followed by an interval of Bill crying. The contextual parameter of **after** is the temporal context variable $\hat{t}$ which stands for the overall context, so the crying intervals are not necessarily on any Tuesday.

Exercise 4.6 *What happens when the common noun meeting is modified by the PPs on each Tuesday and on no Tuesday? Do the results correspond to the intuitive meanings?*

Exercise 4.7 *What happens if temporal modifiers adjoin to full NPs instead of to common nouns?*

- vi. Bill cried after [[a meeting] on each Tuesday]

Does this correspond to an intuitive reading of the sentence? Can this replace another syntactic structure?

4.6 Summary

The basic theory of temporal modification involves the following operations.

Contextualization (5a) applies to all temporal nouns, and all verbs.

Existential closure (6a) applies at the top of the clause to all temporally modified clauses (as well as other clauses that are not temporal modifiers). It always applies to contextualized properties.

An implicit temporal determiner (27) applies to temporal common nouns that lack an explicit determiner. It always applies to contextualized properties.

Temporal modification (23) involves abstraction over the temporal context variable of the temporally modified constituent.

Quantificational arguments and temporal modifiers (nominal and temporal generalized quantifiers) may take scope over the existential closure operation; this has the effect of applying existential closure at a lower level, close to the verb. However, a universal or negative generalized quantifier may not intervene between a contextualized temporal property and its determiner (exercise 4.6).

Class 5

Temporal clauses

A temporal clause may have a quantificational subject.

- (1) A secretary cried after each executive resigned.

Under one reading of the sentence (possibly the most salient one), the sentence is true if each resignation is followed by the crying of a possibly different secretary. The subject of the temporal clause takes scope above the subject of the matrix clause, in what appears to be a violation of a locality constraint: quantifiers do not normally take scope outside of adjunct clauses (cf. Huang 1982). This is one of a family of readings which I will call *dependent-time* readings, where the times at which the matrix clause is evaluated depend on quantifiers within the temporal clause. I will show that dependent-time readings do not violate the adjunct island constraint; they arise from local scope relations within the adjunct clause, where the quantificational argument takes scope above the implicit temporal determiner.

5.1 Basic clause meanings

Temporal clauses like *after Sue arrived* denote temporal generalized quantifiers. Their meanings are derived in a similar way to temporal PPs, through application of the contextualization operation followed by a determiner meaning. The raw meaning of a clause like *Sue arrived* is a property of times (2), to which a temporal context variable is added by the contextualization operation (3).

- (2) Sue arrived: $\lambda i.\mathbf{arrive}(\mathbf{sue})(i)$

- (3) Sue arrived: $\lambda i.i \subseteq \hat{\tau} \wedge \mathbf{arrive}(\mathbf{sue})(i)$

However, unlike a matrix clause whose temporal argument is existentially closed at this stage, a temporal clause needs to have a determiner applied to it in order to turn it into a temporal generalized quantifier. There is no overt determiner, so an implicit existential determiner meaning (4) is applied, followed by the preposition *after*.

- (4) Implicit existential determiner: $\lambda I \lambda J.\exists i[I(i) \wedge J(i)]$

- (5) Sue arrived: $\lambda J.\exists i[i \subseteq \hat{\tau} \wedge \mathbf{arrive}(\mathbf{sue})(i) \wedge J(i)]$

(6) after Sue arrived: $\lambda J. \exists i [i \subseteq \hat{t} \wedge \mathbf{arrive}(\mathbf{sue})(i) \wedge J(\mathbf{after}_{\hat{t}}(i))]$

Exercise 5.1 Why not assume that the determiner meaning is part of the semantics of the preposition *after*?

The temporal generalized quantifier (6) combines with the meaning of the matrix clause and yields the correct results.

(7) Bill cried after Sue arrived:
 $\lambda J. \exists i [i \subseteq \hat{t} \wedge \mathbf{arrive}(\mathbf{sue})(i) \wedge J(\mathbf{after}_{\hat{t}}(i))](\lambda \hat{t} \exists i' [i' \subseteq \hat{t} \wedge \mathbf{cry}(\mathbf{bill})(i')])$
 $= \exists i [i \subseteq \hat{t} \wedge \mathbf{arrive}(\mathbf{sue})(i) \wedge \exists i' [i' \subseteq \mathbf{after}_{\hat{t}}(i) \wedge \mathbf{cry}(\mathbf{bill})(i')]]$

Sentence (7) is true if there exists an interval (within the overall frame) in which Sue arrived, followed by an interval at which Bill cried.

We note that the temporal clause in (7) takes semantic scope over the entire sentence; shortly we'll use this to explain how a nominal quantifier in a temporal clause can take scope above a quantifier in the matrix clause.

5.2 Single-time readings

The single-time reading of (1) entails that all the executives resigned at the same time. This reading obtains when contextualization and the implicit temporal determiner apply to the entire temporal clause, above its subject. Before contextualization, the quantificational subject *each executive* has to apply to the verb *resigned*. This results in a type mismatch, because the verb (8) is of type *e*t while the subject (9) expects an argument of type *e*t.

(8) resigned: $\lambda x \lambda i. \mathbf{resign}(x)(i)$
 (9) each executive: $\lambda P. \forall x [\mathbf{exec}(x) \rightarrow P(x)]$

The type mismatch can be resolved through type shifting, in an analogous way to the shifting of a transitive verb in order to accommodate a generalized quantifier object (definition 3.4).

Exercise 5.2 Formulate a type-shifting rule that has the desired effect.

After type-shifting, the combination of the subject and the verb results in the following temporal property as the raw sentence meaning.

(10) each executive resigned: $\lambda i. \forall x [\mathbf{exec}(x) \rightarrow \mathbf{resign}(x)(i)]$

Contextualization and a determiner meaning turn the above meaning into a temporal generalized quantifier (11), and are followed by the application of the preposition *after*.

(11) each executive resigned:
 $\lambda J. \exists i [i \subseteq \hat{t} \wedge \forall x [\mathbf{exec}(x) \rightarrow \mathbf{resign}(x)(i)] \wedge J(i)]$
 (12) after each executive resigned:
 $\lambda J. \exists i [i \subseteq \hat{t} \wedge \forall x [\mathbf{exec}(x) \rightarrow \mathbf{resign}(x)(i)] \wedge J(\mathbf{after}_{\hat{t}}(i))]$

Finally, the temporal clause modifies the matrix clause, whose meaning is derived by applying the generalized quantifier *a secretary* to the contextualized and existentially closed meaning of *cry*.

(13) a secretary cried: $\exists x[\text{secretary}(x) \wedge \exists i[i \subseteq \hat{t} \wedge \text{cry}(x)(i)]]$

(14) a secretary cried after each executive resigned:
 $\lambda J. \exists i[i \subseteq \hat{t} \wedge \forall x[\text{exec}(x) \rightarrow \text{resign}(x)(i)] \wedge J(\text{after}_{\hat{t}}(i))]$
 $(\lambda \hat{t}. \exists y[\text{secretary}(y) \wedge \exists i'[i' \subseteq \hat{t} \wedge \text{cry}(y)(i')]])$
 $= \exists i[i \subseteq \hat{t} \wedge \forall x[\text{exec}(x) \rightarrow \text{resign}(x)(i)] \wedge$
 $\exists y[\text{secretary}(y) \wedge \exists i'[i' \subseteq \text{after}_{\hat{t}}(i) \wedge \text{cry}(y)(i')]]]$

The result is indeed a single-time reading – there is a time interval at which all the executives resigned, and a secretary cried after that.

5.3 Dependent-time readings

If the implicit existential determiner applies to the verb of (1) before the subject does (for instance by using a scope mechanism on the subject, cf. section 4.3), we get a reading where resignation times vary with the resigning executives. Contextualization and a determiner meaning first apply to the the raw verb.

(15) resigned: $\lambda x \lambda J. \exists i[i \subseteq \hat{t} \wedge \text{resign}(x)(i) \wedge J(i)]$

The quantificational subject applies combines via type-shifting (exercise 5.2); the resulting temporal generalized quantifier is modified by the preposition *after*.

(16) each executive resigned:
 $\lambda J. \forall x[\text{exec}(x) \rightarrow \exists i[i \subseteq \hat{t} \wedge \text{resign}(x)(i) \wedge J(i)]]$

(17) after each executive resigned:
 $\lambda J. \forall x[\text{exec}(x) \rightarrow \exists i[i \subseteq \hat{t} \wedge \text{resign}(x)(i) \wedge J(\text{after}_{\hat{t}}(i))]]$

At this point we can already see how the quantificational relations will end up. The universal quantifier contributed by the subject NP outscopes the temporal existential determiner; since the temporal property variable J lies within the scope of the temporal existential determiner, it also lies within the scope of the subject. The variable J will get replaced with the meaning of the matrix clause, so the matrix clause itself will fall under the scope of the subject of the temporal clause.

The temporal clause can take scope over the entire main clause. In this case the temporal generalized quantifier (17) applies to the matrix clause meaning (13).

(18) a secretary cried after each executive resigned:
 $\lambda J. \forall x[\text{exec}(x) \rightarrow \exists i[i \subseteq \hat{t} \wedge \text{resign}(x)(i) \wedge J(\text{after}_{\hat{t}}(i))]]$
 $(\lambda \hat{t}. \exists y[\text{secretary}(y) \wedge \exists i'[i' \subseteq \hat{t} \wedge \text{cry}(y)(i')]])$
 $= \forall x[\text{exec}(x) \rightarrow \exists i[i \subseteq \hat{t} \wedge \text{resign}(x)(i) \wedge$
 $\exists y[\text{secretary}(y) \wedge \exists i'[i' \subseteq \text{after}_{\hat{t}}(i) \wedge \text{cry}(y)(i')]]]]]$

We get a representation of the desired meaning – each executive resigned, and following that resignation a secretary cried; the secretaries may vary with the executives. The end result is that the subject of the temporal adjunct clause has scope over the subject of the main clause, but this is achieved through local operations: within the temporal clause, the subject takes scope over the

temporal determiner; additionally, the entire temporal clause takes scope over the matrix subject. Thus, the representation (18) is not in violation of locality constraints.

The temporal clause may also modify the matrix predicate rather than the whole matrix clause. In this case the matrix subject takes scope above the temporal clause.

- (19) a secretary cried after each executive resigned:

$$= \exists y[\text{secretary}(y) \wedge \forall x[\text{exec}(x) \rightarrow \exists i[i \subseteq \hat{t} \wedge \text{resign}(x)(i) \wedge \exists i'[i' \subseteq \text{after}_t(i) \wedge \text{cry}(y)(i')]]]]]$$

This reading entails that there was one secretary who cried, but for each executive this secretary cried at a different time, namely after the executive's resignation. Readings (18) and (19) are both dependent-time readings, because the times at which the matrix predicate is evaluated depend on the subject of the temporal clause.

Exercise 5.3 *German speakers have told me that dependent-time readings are not possible in German. How can we parameterize our theory to rule out such readings in German?*

Exercise 5.4 *When the temporal clause contains a negative generalized quantifier, single-time readings are easy and dependent-time readings are difficult (Some speakers find the dependent-time reading easier when the negative determiner contrasts with another determiner).*

- i. Bill cried after no executive resigned.

- (a) Single time: $\exists i[i \subseteq \hat{t} \wedge \neg \exists x[\text{exec}(x) \wedge \text{resign}(x)(i)] \wedge \exists i'[i' \subseteq \text{after}_t(i) \wedge \text{cry}(\text{bill})(i')]]]$
 (b) Dependent time: $\neg \exists x[\text{exec}(x) \wedge \exists i[i \subseteq \hat{t} \wedge \text{resign}(x)(i) \wedge \exists i'[i' \subseteq \text{after}_t(i) \wedge \text{cry}(\text{bill})(i')]]]$

- ii. As opposed to Mary, who cried after *every* executive resigned, Bill cried after *no* executive resigned.

What can be the source of the difficulty? Consider the following facts. (a) There is no general problem with temporal dependency on negative quantifiers.

- iii. Bill cried after no resignation.

(b) When a negative adverbial is preposed, an auxiliary is added to the main clause ("negative inversion"). Without negative inversion, a preposed temporal clause with a negative quantifier only has a single-time reading; with negative inversion, the single-time reading disappears and the dependent-time reading is difficult.

- iv. After no resignation did Bill cry.

- v. After no executive resigned, Bill cried. (meaning: (ia))

- vi.?After no executive resigned did Bill cry. (meaning (ib) difficult)

Exercise 5.5 *What would happen if a negative generalized quantifiers intervened between the contextualization operation and the implicit temporal determiner? Compare this to your findings in exercise 4.6.*

5.4 Temporally modified temporal clauses

Our semantics is unable to handle temporal clauses which have temporal modifiers within them, such as the following sentences.

(20) Bill resigned after John disappeared every Friday.

(21) Bill did his homework before daddy came home every evening.

The semantics can interpret these sentences as cascades, where *every Friday* is a modifier of *resigned*, and *every evening* is a modifier of *did his homework*. While these are possible readings, it is more natural to interpret *every Friday* as a modifier of *disappeared* and *every evening* as a modifier of *came home*. But our semantics stipulates that temporally modified clauses are existentially closed (section 4.6), and this does not leave a way to subsequently turn the temporal clause into a temporal generalized quantifier. We therefore need to extend our grammar.

Sentences (20) and (21) show that two extensions of the grammar are needed. The most natural reading of (20) is that Bill resigned after a certain period, in which John disappeared every Friday; such a reading can be captured by subordinating the context of the main clause to the *context* of the temporal clause, which constitutes this period. This is what I call an aggregate reading.

The natural reading of (21) is not an aggregate reading; the sentence is true if Bill's dad came home every evening and Bill did his homework every day before his dad's arrival. The homework does not necessarily have to be done in the evening, so this is not a cascaded reading; rather, it is a special kind of dependent-time reading, where the dependency is not on a nominal quantifier but rather on a temporal quantifier inside the temporal clause. Such a reading can be captured by allowing the temporal modifier *every Friday* to apply to a temporal generalized quantifier rather than to an existentially closed sentence. I call this a temporally dependent reading.

Aggregate readings

Sentence (20), repeated below, is true if Bill resigned after a certain period, in which John disappeared every Friday.

(20) Bill resigned after John disappeared every Friday.

We want to subordinate the temporal context of the main clause to the context of the temporal clause. We start with the familiar clause-type meaning for the temporal clause, derived by applying the temporal generalized quantifier *every Friday* (22) to the contextualized and existentially closed meaning of the sentence *John disappeared* (23).

(22) every Friday: $\lambda J \forall i [(i \subseteq \hat{\tau} \wedge \mathbf{friday}(i)) \rightarrow J(i)]$

(23) John disappeared: $\exists i' [i' \subseteq \hat{\tau} \wedge \mathbf{disappear}(\mathbf{john})(i')]$

(24) John disappeared every Friday:
 $\forall i [(i \subseteq \hat{\tau} \wedge \mathbf{friday}(i)) \rightarrow \exists i' [i' \subseteq i \wedge \mathbf{disappear}(\mathbf{john})(i')]]$

The next step is an “aggregating” operation of abstraction over the temporal context variable of the temporal clause. This gives a temporal property – the set of intervals which can serve as a context for the temporal clause.

- (25) John disappeared every Friday:
 $\lambda \hat{i}. \forall i[(i \subseteq \hat{i} \wedge \text{friday}(i)) \rightarrow \exists i'[i' \subseteq i \wedge \text{disappear}(\text{john})(i')]]$

This property is contextualized and an existential determiner is applied (26), followed by the preposition *after* (27). The resulting temporal generalized quantifier modifies the main clause (28).

- (26) John disappeared every Friday:
 $\lambda J. \exists i[i \subseteq \hat{i} \wedge \forall i'[(i' \subseteq i \wedge \text{friday}(i')) \rightarrow \exists i''[i'' \subseteq i' \wedge \text{disappear}(\text{john})(i'')]] \wedge J(i)]$

- (27) after John disappeared every Friday:
 $\lambda J. \exists i[i \subseteq \hat{i} \wedge \forall i'[(i' \subseteq i \wedge \text{friday}(i')) \rightarrow \exists i''[i'' \subseteq i' \wedge \text{disappear}(\text{john})(i'')]] \wedge J(\text{after}_i(i))]$

- (28) Bill resigned after John disappeared every Friday:
 $\exists i[i \subseteq \hat{i} \wedge \forall i'[(i' \subseteq i \wedge \text{friday}(i')) \rightarrow \exists i''[i'' \subseteq i' \wedge \text{disappear}(\text{john})(i'')]] \wedge \exists i'''[i''' \subseteq \text{after}_i(i) \wedge \text{resign}(\text{bill})(i''')]]$

The representation of the sentence has the desired truth conditions: there exists an interval which serves as a context in which John disappeared every Friday, and after this interval, Bill resigned.

Temporally dependent readings

Sentence (21), repeated below, is true if Bill’s dad came home every evening and Bill did his homework every day before his dad’s arrival, though the homework does not necessarily have to be done in the evening.

- (21) Bill did his homework before daddy came home every evening.

Neither the aggregate reading nor the cascaded reading have the desired interpretation: the aggregate reading (29) says that Bill did his homework once, before a period in which daddy came home every evening; the cascaded reading (30) entails that Bill did his homework every evening (notice the parameter *i* on the temporal function **before**). For simplicity, I assume the VPs *did his homework* and *came home* translate as predicate constants (type eit), and that the word *daddy* is a proper name (type e).

- (29) Aggregate: $\exists i[i \subseteq \hat{i} \wedge \forall i'[(i' \subseteq i \wedge \text{evening}(i')) \rightarrow \exists i''[i'' \subseteq i' \wedge \text{come-home}(\text{dad})(i'')]] \wedge \exists i'''[i''' \subseteq \text{before}_i(i) \wedge \text{do-hw}(\text{bill})(i''')]]$

- (30) Cascaded: $\forall i[(i \subseteq \hat{i} \wedge \text{evening}(i)) \rightarrow \exists i'[i' \subseteq i \wedge \text{come-home}(\text{dad})(i')] \wedge \exists i''[i'' \subseteq \text{before}_i(i') \wedge \text{do-hw}(\text{bill})(i'')]]$

While (29) and (30) are both valid readings of the sentence, the most salient reading is the one described at the beginning of this passage, where the temporal generalized quantifier *every evening* is inside the temporal clause and the evaluation times of the matrix predicate depend on it. The obvious way to derive this temporally dependent reading is to allow temporal generalized quantifiers to modify other temporal generalized quantifiers.

The raw meaning of the sentence *daddy came home* (31) does not get existentially closed after contextualization, but rather a temporal determiner is applied, making the sentence into a temporal generalized quantifier (32).

(31) daddy came home: $\lambda i'. \text{come-home}(\text{dad})(i')$

(32) daddy came home: $\lambda J. \exists i' [i' \subseteq \hat{i} \wedge \text{come-home}(\text{dad})(i') \wedge J(i')]$

This temporal generalized quantifier is modified by the temporal generalized quantifier *every evening* through the rule (23) from class 4: the free temporal context variable in (32) is bound by the operator $\lambda \hat{i}$, and the temporal generalized quantifier *every evening* combines with it through functional composition.

(33) daddy came home every evening:
 $\lambda J. \forall i [(i \subseteq \hat{i} \wedge \text{evening}(i)) \rightarrow J(i)] \circ$
 $\lambda J \lambda \hat{i}. \exists i' [i' \subseteq \hat{i} \wedge \text{come-home}(\text{dad})(i') \wedge J(i')]$
 $= \lambda J. \forall i [(i \subseteq \hat{i} \wedge \text{evening}(i)) \rightarrow$
 $\exists i' [i' \subseteq i \wedge \text{come-home}(\text{dad})(i') \wedge J(i')]]$

The result is a temporal generalized quantifier, as expected, which can combine with *before* (34) and modify the matrix clause (35).

(34) before daddy came home every evening:
 $\lambda J. \forall i [(i \subseteq \hat{i} \wedge \text{evening}(i)) \rightarrow$
 $\exists i' [i' \subseteq i \wedge \text{come-home}(\text{dad})(i') \wedge J(\text{before}_i(i'))]]$

(35) Bill did his homework before daddy came home every evening:
 $\forall i [(i \subseteq \hat{i} \wedge \text{evening}(i)) \rightarrow$
 $\exists i' [i' \subseteq i \wedge \text{come-home}(\text{dad})(i') \wedge$
 $\exists i'' [i'' \subseteq \text{before}_i(i') \wedge \text{do-hw}(\text{bill})(i'')]]]$

This is indeed the desired temporally dependent reading of the sentence: every evening daddy came home, and before that – not necessarily in the evening – Bill did his homework.

Exercise 5.6 Temporally dependent readings are only available with a subclass of temporally modified temporal clauses. Here are some relevant data. Dependent readings are much more readily available with nominal generalized quantifiers than with temporal generalized quantifiers.

vii. John got angry after each executive cried.

viii. John got angry after Bill cried during each meeting.

Dependent readings are also more readily available with the preposition *at* than with *during*.

ix. (a) John showered before he testified during each hearing.

- (b) John showered before he testified at each hearing.
- x. (a) The lawyer prepared differently before he testified at each hearing.
(b) The lawyer prepared differently before he testified during each hearing.
- xi. (a) After John gave me a check at each meeting, he canceled it.
(b) After John gave me a check during each meeting, he canceled it.

How can these observations combine to a coherent restriction of temporally dependent readings?

Bibliography

- Anscombe, G. E. M. (1964). Before and after. *The Philosophical Review* 73(1): 3–24.
- Artstein, Ron (2004). Quantificational arguments in temporal adjunct clauses. Unpublished manuscript, June 1 (submitted to *Linguistics and Philosophy*); <http://www.cs.technion.ac.il/~artstein/>.
- Artstein, Ron and Nissim Francez (2003). Plural times and temporal modification. In Paul Dekker and Robert van Rooy (eds.), *Proceedings of the Fourteenth Amsterdam Colloquium*, pp. 63–68. ILLC, University of Amsterdam. <http://www.cs.technion.ac.il/~artstein/>.
- Barwise, Jon and Robin Cooper (1981). Generalized quantifiers and natural language. *Linguistics and Philosophy* 4(2): 159–219.
- Beaver, David and Cleo Condoravdi (to appear). A uniform analysis of *before* and *after*. In Rob Young and Yuping Zhou (eds.), *Proceedings from SALT XIII*. CLC Publications, Ithaca, NY. <http://montague.stanford.edu/~dib/Publications/Salt13.pdf>.
- Cooper, Robin (1983). *Quantification and Syntactic Theory*, vol. 21 of *Synthese Language Library*. D. Reidel, Dordrecht.
- Dowty, David R. (1979). *Word Meaning and Montague Grammar: The Semantics of Verbs and Times in Generative Semantics and in Montague’s PTQ*, vol. 7 of *Synthese Language Library*. D. Reidel, Dordrecht.
- Dowty, David R. (1982). Tenses, time adverbs, and compositional semantic theory. *Linguistics and Philosophy* 5(1): 23–55.
- Francez, Nissim and Mark Steedman (to appear). Categorical grammars and the semantics of indexical preposition phrases. *Linguistics and Philosophy*.
- Gamut, L. T. F. (1991). *Logic, Language and Meaning, Volume 2: Intensional Logic and Logical Grammar*. University of Chicago Press, Chicago.
- Gazdar, Gerald (1980). A cross-categorical semantics for coordination. *Linguistics and Philosophy* 3(3): 407–409.
- Huang, Cheng-Teh James (1982). *Logical Relations in Chinese and the Theory of Grammar*. Ph.D. thesis, MIT, Cambridge, MA.
- Kamp, Hans (1971). Formal properties of ‘now’. *Theoria* 37: 227–273.
- Kamp, Hans and Uwe Reyle (1993). *From Discourse to Logic: Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*, vol. 42 of *Studies in Linguistics and Philosophy*. Kluwer, Dordrecht.

- Keenan, Edward L. and Jonathan Stavi (1986). A semantic characterization of natural language determiners. *Linguistics and Philosophy* 9(3): 253–326.
- Landman, Fred (1991). *Structures for Semantics*, vol. 45 of *Studies in Linguistics and Philosophy*. Kluwer, Dordrecht.
- May, Robert (1985). *Logical Form: Its Structure and Derivation*, vol. 12 of *Linguistic Inquiry Monographs*. MIT Press, Cambridge, MA.
- Montague, Richard (1973). The proper treatment of quantification in ordinary English. In Jaakko Hintikka, Julius Moravcsik, and Patrick Suppes (eds.), *Approaches to Natural Language: Proceedings of the 1970 Stanford Workshop on Grammar and Semantics*, pp. 221–242. D. Reidel, Dordrecht. Reprinted in Montague (1974), ch. 8, pp. 247–270.
- Montague, Richard (1974). *Formal Philosophy: Selected Papers of Richard Montague*. Yale University Press, New Haven, CT. Edited and with an introduction by Richmond H. Thomason.
- Ogihara, Toshiyuki (1994). Adverbs of quantification and sequence-of-tense phenomena. In Mandy Harvey and Lynn Santelmann (eds.), *Proceedings from SALT IV*, pp. 251–267. CLC Publications, Ithaca, NY.
- Partee, Barbara (1973). Some structural analogies between tenses and pronouns in English. *The Journal of Philosophy* 70(18): 601–609.
- Partee, Barbara and Mats Rooth (1983). Generalized conjunction and type ambiguity. In Rainer Bäuerle, Christoph Schwarze, and Arnim von Stechow (eds.), *Meaning, Use and Interpretation of Language*, pp. 361–383. de Gruyter, Berlin.
- Pratt, Ian and Nissim Francez (1997). *On the Semantics of Temporal Prepositions and Preposition Phrases*. Tech. Rep. UMCS-97-4-2, Department of Computer Science, University of Manchester. <ftp://ftp.cs.man.ac.uk/pub/TR/UMCS-97-4-2.ps.Z>.
- Pratt, Ian and Nissim Francez (2001). Temporal prepositions and temporal generalized quantifiers. *Linguistics and Philosophy* 24(2): 187–222.
- Prior, A[rthur] N. (1957). *Time and Modality*. The Clarendon Press, Oxford.
- Prior, Arthur [N.] (1967). *Past, Present and Future*. The Clarendon Press, Oxford.
- Prior, Arthur N. (1968). *Papers on Time and Tense*. The Clarendon Press, Oxford.
- Sharvit, Yael (2003). Embedded tense and universal grammar. *Linguistic Inquiry* 34(4): 669–681.
- von Stechow, Arnim (2002). Temporal prepositional phrases with quantifiers: Some additions to Pratt and Francez (2001). *Linguistics and Philosophy* 25(5–6): 755–800.
- Steedman, Mark (2003). Scope alternation and the syntax-semantics interface. Manuscript, University of Edinburgh, draft 4.1, <http://www.informatics.ed.ac.uk/~steedman/papers.html>.
- Stump, Gregory T. (1985). *The Semantic Variability of Absolute Constructions*, vol. 25 of *Synthese Language Library*. D. Reidel, Dordrecht.