Grammar in TTR

Robin Cooper
University of Gothenburg
Jonathan Ginzburg
Université Paris-Diderot, Sorbonne Paris-Cité

An Introduction to Semantics using Type Theory with Records
ESSLLI 2012
Lecture 2

Speech acts as event (situations)

- The interactive stance, discussed yesterday, emphasizes that linguistic phenomena involve events that unfold in real time

Speech acts as event (situations)

- The interactive stance, discussed yesterday, emphasizes that linguistic phenomena involve events that unfold in real time
- In our discussion of TTR yesterday we introduced ptypes like hug(b,d) and record types like $\begin{bmatrix} \mathrm{x} & : & \mathit{Ind} \\ \mathrm{c_{boy}} & : & \mathrm{boy(x)} \\ \mathrm{y} & : & \mathit{Ind} \\ \mathrm{c_{dog}} & : & \mathrm{dog(y)} \\ \mathrm{e} & : & \mathrm{hug(x,y)} \end{bmatrix}$ as types of (records containing) situations (or events)

Speech acts as event (situations)

- The interactive stance, discussed yesterday, emphasizes that linguistic phenomena involve events that unfold in real time
- In our discussion of TTR yesterday we introduced ptypes like hug(b,d) and record types like $\left[\begin{array}{lll} \mathrm{x} & : & \mathit{Ind} \\ \mathrm{c}_{\mathrm{boy}} & : & \mathrm{boy(x)} \\ \mathrm{y} & : & \mathit{Ind} \\ \mathrm{c}_{\mathrm{dog}} & : & \mathrm{dog(y)} \\ \mathrm{e} & : & \mathrm{hug(x,y)} \end{array}\right]$ as types of (records containing) situations (or events)
- but is this enough?

Speech acts as event (situations)

- The interactive stance, discussed yesterday, emphasizes that linguistic phenomena involve events that unfold in real time
- In our discussion of TTR yesterday we introduced ptypes like hug(b,d) and record types like $\left[\begin{array}{lll} \mathrm{x} & : & \mathit{Ind} \\ \mathrm{c}_{\mathrm{boy}} & : & \mathrm{boy(x)} \\ \mathrm{y} & : & \mathit{Ind} \\ \mathrm{c}_{\mathrm{dog}} & : & \mathrm{dog(y)} \\ \mathrm{e} & : & \mathrm{hug(x,y)} \end{array}\right]$ as types of (records containing) situations (or events)
- but is this enough? – static snapshot, no analysis of change or progression in an event

Speech acts as event (situations)

- The interactive stance, discussed yesterday, emphasizes that linguistic phenomena involve events that unfold in real time
- In our discussion of TTR yesterday we introduced ptypes like hug(b,d) and record types like $\left[\begin{array}{lcl} \text{x} & : & \mathit{Ind} \\ \text{c}_{\text{boy}} & : & \text{boy(x)} \\ \text{y} & : & \mathit{Ind} \\ \text{c}_{\text{dog}} & : & \text{dog(y)} \\ \text{e} & : & \text{hug(x,y)} \end{array}\right]$ as types of (records containing) situations (or events)
- but is this enough? – static snapshot, no analysis of change or progression in an event
- a speech event which is an utterance of a sentence can consist of a noun-phrase event followed by a verb-phrase event

Speech acts as event (situations)

- The interactive stance, discussed yesterday, emphasizes that linguistic phenomena involve events that unfold in real time
- In our discussion of TTR yesterday we introduced ptypes like hug(b,d) and record types like $\begin{bmatrix} \mathrm{x} & : & \mathit{Ind} \\ \mathrm{c}_{\mathrm{boy}} & : & \mathrm{boy(x)} \\ \mathrm{y} & : & \mathit{Ind} \\ \mathrm{c}_{\mathrm{dog}} & : & \mathrm{dog(y)} \\ \mathrm{e} & : & \mathrm{hug(x,y)} \end{bmatrix}$ as types of (records containing) situations (or events)
- but is this enough? – static snapshot, no analysis of change or progression in an event
- a speech event which is an utterance of a sentence can consist of a noun-phrase event followed by a verb-phrase event
- this leads us to *string theory*

Outline

String theory of events

Inference from partial observation of events

Grammar rules as dependent event types (Cooper, 2012)

Frames in lexical semantics (Cooper, 2010, 2012)

Outline

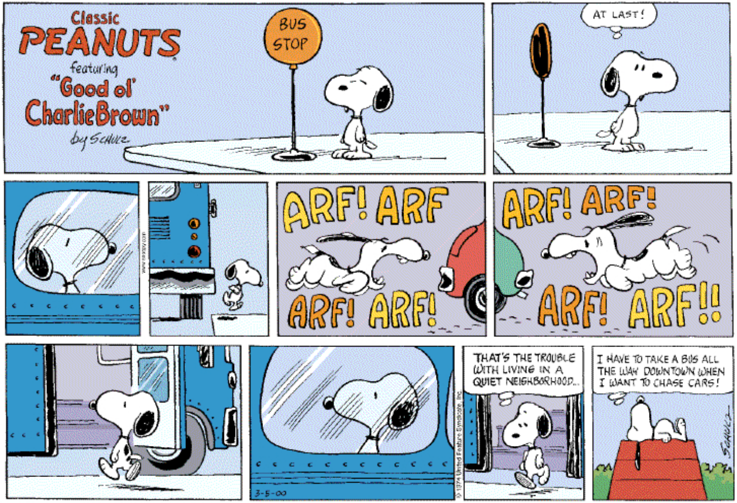
String theory of events

Inference from partial observation of events

Grammar rules as dependent event types (Cooper, 2012)

Frames in lexical semantics (Cooper, 2010, 2012)

Fernando's string theory



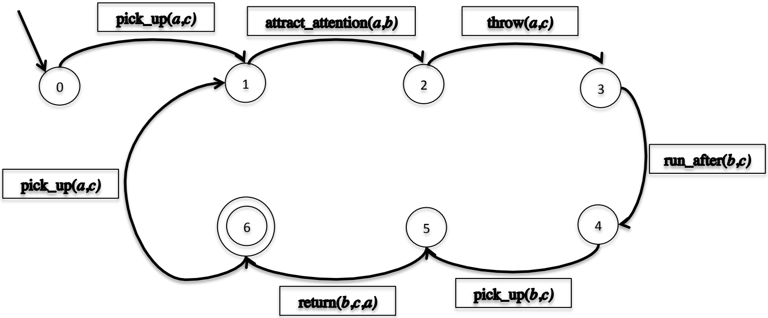
Some references to Fernando's work

Fernando (2004, 2006, 2008, 2009)

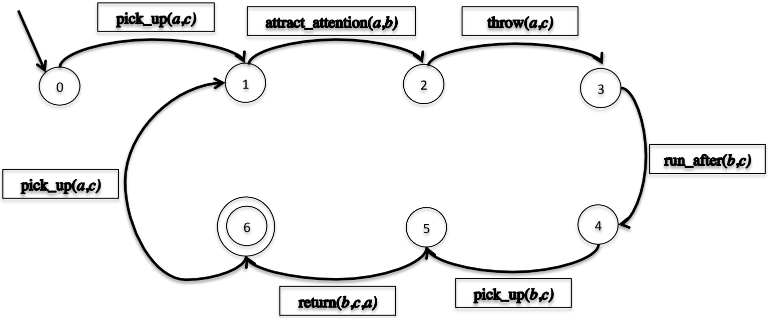
Regular types

1. if T_1, $T_2 \in$ **Type**, then $T_1 ⁀ T_2 \in$ **Type**
   $a : T_1 ⁀ T_2$ iff $a = x ⁀ y$, $x : T_1$ and $y : T_2$
2. if $T \in$ **Type** then $T^+ \in$ **Type**.
   $a : T^+$ iff $a = x_1 ⁀ \ldots ⁀ x_n$, $n > 0$ and for i, $1 \leq i \leq n$, $x_i : T$
   ...

A game of fetch



A game of fetch



$$(\text{pick\_up}(a,c)^\frown\text{attract\_attention}(a,b)^\frown\text{throw}(a,c)^\frown\text{run\_after}(b,c)^\frown \text{pick\_up}(b,c)^\frown\text{return}(b,c,a))^+$$

Outline

String theory of events

Inference from partial observation of events

Grammar rules as dependent event types (Cooper, 2012)

Frames in lexical semantics (Cooper, 2010, 2012)

Partiality of event perception

- Do not need to observe all the frames in an event
- Suffices to observe enough to uniquely identify event types agent has in its resources

Inferring an event type from a partial observation

$$\lambda r{:}\begin{bmatrix}\text{x}:\mathit{Ind}\\ \text{c}_{\text{human}}:\text{human(x)}\\ \text{y}:\mathit{Ind}\\ \text{c}_{\text{dog}}:\text{dog(y)}\\ \text{z}:\mathit{Ind}\\ \text{c}_{\text{stick}}:\text{stick(z)}\\ \text{e}:\text{pick\_up(x,z)}^\frown\text{attract\_attention(x,y)}\end{bmatrix}(\begin{bmatrix}\text{e}:\text{play\_fetch}(r.\text{x},r.\text{y})\end{bmatrix})$$

Three views of this inference

Three views of this inference

- function from objects (events) to a *type*

Three views of this inference

- function from objects (events) to a *type* – $\lambda a : T_1(T_2[a])$

Three views of this inference

- function from objects (events) to a *type* – $\lambda a : T_1(T_2[a])$
- a *dependent type*

Three views of this inference

- function from objects (events) to a *type* – $\lambda a : T_1(T_2[a])$
- a *dependent type*
- perceiving something and inferring the type of something not (yet) perceived from that perception

Three views of this inference

- function from objects (events) to a *type* – $\lambda a : T_1(T_2[a])$
- a *dependent type*
- perceiving something and inferring the type of something not (yet) perceived from that perception
- we will see a number of other uses of dependent types, for example as the interpretation of verb phrases

Outline

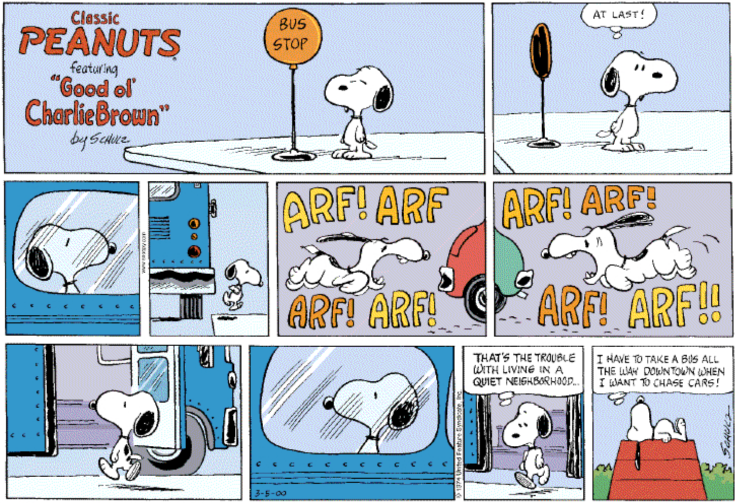
String theory of events

Inference from partial observation of events

Grammar rules as dependent event types (Cooper, 2012)

Frames in lexical semantics (Cooper, 2010, 2012)

Fernando's string theory



Two kinds of partiality

- not all frames are perceived – infer more

Two kinds of partiality

- not all frames are perceived – infer more
- individual frame type partially specified

Two kinds of partiality

- not all frames are perceived – infer more
- individual frame type partially specified
- $$\left[\begin{array}{lll} \mathrm{x} & : & \mathit{Ind} \\ \mathrm{c}_{\mathrm{dog}} & : & \mathrm{dog}(\mathrm{x}) \\ \mathrm{y} & : & \mathit{Ind} \\ \mathrm{c}_{\mathrm{bus-stop}} & : & \mathrm{bus\text{-}stop}(\mathrm{y}) \\ \mathrm{c}_{\mathrm{near}} & : & \mathrm{near}(\mathrm{x},\mathrm{y}) \\ \mathrm{e} & : & \mathrm{wait\text{-}for}(\mathrm{x}, \left[\begin{array}{lll} \mathrm{x} & : & \mathit{Ind} \\ \mathrm{c}_{\mathrm{bus}} & : & \mathrm{bus}(\mathrm{x}) \\ \mathrm{e} & : & \mathrm{arrive\text{-}at}(\mathrm{x},\Uparrow\mathrm{y}) \end{array}\right]) \end{array}\right]$$

$\Uparrow$ is part of the unofficial notation, used when you have to look outside a record type which is embedded as an argument to a predicate – points to the next higher record type.

Official notation for $\Uparrow$

$$\left[\begin{array}{l} \mathrm{x} \ \ldots \\ \mathrm{y} \ \ldots \\ \mathrm{e}{:}\mathrm{wait\text{-}for}(\mathrm{x}, \left[\begin{array}{l} \mathrm{x}{:}\mathit{Ind} \\ \mathrm{c_{bus}}{:}\mathrm{bus}(\mathrm{x}) \\ \mathrm{e}{:}\mathrm{arrive\text{-}at}(\mathrm{x},\Uparrow\mathrm{y}) \end{array}\right]) \end{array}\right]$$

$$\left[\begin{array}{l} \mathrm{x} \ \ldots \\ \mathrm{y} \ \ldots \\ \mathrm{e}{:}\langle \lambda v_1{:}\mathit{Ind}(\lambda v_2{:}\mathit{Ind}(\mathrm{wait\text{-}for}(v_1, \left[\begin{array}{l} \mathrm{x}{:}\mathit{Ind} \\ \mathrm{c_{bus}}{:}\langle \lambda v{:}\mathit{Ind}(\mathrm{bus}(v)),\ \langle \mathrm{x}\rangle\rangle \\ \mathrm{e}{:}\langle \lambda v{:}\mathit{Ind}(\mathrm{arrive\text{-}at}(v,v_2)),\ \langle \mathrm{x}\rangle\rangle \end{array}\right]))), \\ \qquad\qquad\qquad\qquad \langle \mathrm{x},\mathrm{y}\rangle\rangle \end{array}\right]$$

Specification with Snoopy

$$\left[\begin{array}{lcl} \mathrm{x}{=}\textcolor{red}{\mathrm{snoopy}} & : & \mathit{Ind} \\ \mathrm{c}_{\mathrm{dog}} & : & \mathrm{dog(x)} \\ \mathrm{y} & : & \mathit{Ind} \\ \mathrm{c}_{\mathrm{bus-stop}} & : & \mathrm{bus\text{-}stop(y)} \\ \mathrm{c}_{\mathrm{near}} & : & \mathrm{near(x,y)} \\ \mathrm{e} & : & \mathrm{wait\text{-}for}(\mathrm{x}, \left[\begin{array}{lcl} \mathrm{x} & : & \mathit{Ind} \\ \mathrm{c}_{\mathrm{bus}} & : & \mathrm{bus(x)} \\ \mathrm{e} & : & \mathrm{arrive\text{-}at(x,}\Uparrow\mathrm{y)} \end{array}\right]) \end{array}\right]$$

What are *manifest fields* $\left[\ \ell{=}a \quad : \quad T\ \right]$?

Manifest fields

$$\left[\ \ell{=}a \quad : \quad T\ \right]$$

is a convenient notation for

$$\left[\ \ell \quad : \quad T_a\ \right]$$

If $a : T$, then T_a is a type (the *singleton type of* a).
$b : T_a$ iff $b = a$

Partial perception of linguistic events

- perceive phonology (or phonetics)
- infer syntax/semantics

Grammar rules

Grammar rules are of the form

$\lambda s_1 : T_1 \ldots \lambda s_n : T_n(T)$

where T_i and T are *sign types*.

Sign types correspond to the notion of sign in HPSG.

The type *Sign*

$$\left[\begin{array}{lll} \text{s-event} & : & \mathit{SEvent} \\ \text{synsem} & : & \left[\begin{array}{lll} \text{cat} & : & \mathit{Cat} \\ \text{cnt} & : & \mathit{Cnt} \end{array}\right] \end{array}\right]$$

But what are *SEvent*, *Cat* and *Cnt*?

The type *SEvent*

$$\left[\begin{array}{lcl} \text{phon} & : & \mathit{Phon} \\ \text{s-time} & : & \mathit{TimeInt} \\ \text{e} & : & \text{uttered\_at(phon, s-time)} \end{array}\right]$$

A minimal solution, from Cooper (2012). We leave to one side the types *Phon*, "phonology", and *TimeInt*, "time interval".

The type *SEvent*, a more refined type

$$\begin{bmatrix} \text{s-time} & : & \textit{TimeInt} \\ \text{s-loc} & : & \textit{Loc} \\ \text{sp} & : & \textit{Ind} \\ \text{au} & : & \textit{Ind} \\ \text{phon} & : & \textit{Phon} \\ \text{e} & : & \text{utter(sp,phon,au,s-time,s-loc)} \end{bmatrix}$$

However, this may not always be correct: more than one person may be in the audience; more than one person may collaborate on a single speech event - *split utterances*.

The type *Cat*

s, np, vp, n_{prop}, v_{i} : *Cat*

There will be more in a more extensive fragment.

The type *Cnt*

$RecType \vee (Ppty \vee Quant)$

There will be more in a more extensive fragment

The type *Cnt*

$RecType \vee (Ppty \vee Quant)$

There will be more in a more extensive fragment

If T_1, T_2 are types, then $T_1 \vee T_2$ is a type (the *join* of T_1 and T_2).
$a : T_1 \vee T_2$ iff either $a : T_1$ or $a : T_2$

The type *Cnt*

$RecType \vee (Ppty \vee Quant)$

There will be more in a more extensive fragment

If T_1, T_2 are types, then $T_1 \vee T_2$ is a type (the *join* of T_1 and T_2).
$a : T_1 \vee T_2$ iff either $a : T_1$ or $a : T_2$

Ppty, "property" is to be $\left[\mathrm{x}{:}Ind\right] \rightarrow RecType$
(cf. $\langle e, t \rangle$)

The type *Cnt*

$\mathit{RecType} \vee (\mathit{Ppty} \vee \mathit{Quant})$

There will be more in a more extensive fragment

If T_1, T_2 are types, then $T_1 \vee T_2$ is a type (the *join* of T_1 and T_2).
$a : T_1 \vee T_2$ iff either $a : T_1$ or $a : T_2$

Ppty, "property" is to be $\left[\mathrm{x}{:}\mathit{Ind}\right] \rightarrow \mathit{RecType}$
(cf. $\langle e, t \rangle$)

Quant, "quantifier" is to be $\mathit{Ppty} \rightarrow \mathit{RecType}$
(cf. $\langle\langle e, t \rangle, t \rangle$)

The lexicon: Proper names

If W is a phonological type such as "Sam" or "John" and a:*Ind*, $\text{lex}_{\text{n}_{\text{Prop}}}(W, a)$ is

$$
\textit{Sign} \mathbin{\dot\wedge} \left[\begin{array}{lll} \text{s-event} & : & \left[\begin{array}{lll} \text{phon} & : & W \end{array}\right] \\ \text{synsem} & : & \left[\begin{array}{lll} \text{cat}{=}\text{n}_{\text{Prop}} & : & \textit{Cat} \\ \text{cnt}{=}\lambda v{:}\textit{Ppty}(v(\left[\text{x}{=}a\right])) & : & \textit{Quant} \end{array}\right] \end{array}\right]
$$

$T_1 \dot\wedge T_2$ is the *merge* of the two types T_1, T_2. If at least on of the two types is not a record type it is identical with the meet $T_1 \wedge T_2$. So what is the meet of two types and the merge of two record types?

Meets and merges

If T_1, T_2 are types, then $T_1 \wedge T_2$ is a type (the *meet* of T_1 and T_2).
$a : T_1 \wedge T_2$ iff $a : T_1$ and $a : T_2$

The basic idea of *merge* (full definition in Cooper, 2012):

$$\begin{bmatrix}\mathrm{f}\colon T_1\end{bmatrix} \underset{\cdot}{\wedge} \begin{bmatrix}\mathrm{g}\colon T_2\end{bmatrix} = \begin{bmatrix}\mathrm{f}\colon T_1 \\ \mathrm{g}\colon T_2\end{bmatrix}$$

$$\begin{bmatrix}\mathrm{f}\colon T_1\end{bmatrix} \underset{\cdot}{\wedge} \begin{bmatrix}\mathrm{f}\colon T_2\end{bmatrix} = \begin{bmatrix}\mathrm{f}\colon T_1 \underset{\cdot}{\wedge} T_2\end{bmatrix}$$

cf. unification

The lexicon: intransitive verbs (individual level)

If W is the phonological type "run" or "walk" and p is a predicate with arity $\langle Ind \rangle$, $\mathrm{lex}_{\mathrm{V_i}}(W, p)$ is

$$Sign \mathbin{\dot\wedge} \begin{bmatrix} \text{s-event}{:}\begin{bmatrix}\text{phon}{:}W\end{bmatrix} \\ \text{synsem}{:}\begin{bmatrix} \text{cat}{=}\text{v}_\text{i}{:}Cat \\ \text{cnt}{=}\lambda r{:}\begin{bmatrix}\text{x}{:}Ind\end{bmatrix}\ (\begin{bmatrix}\text{e}{:}p(r.\text{x})\end{bmatrix}){:}Ppty \end{bmatrix} \end{bmatrix}$$

The lexicon: intransitive verbs (frame level)

If W is the phonological type "rise" or "change" and p is a predicate with arity $\langle Ind \rangle$, $\mathrm{lex}_{\mathrm{V_i}}(W, p)$ is

$$Sign \mathbin{\dot\wedge} \begin{bmatrix} \text{s-event}: \begin{bmatrix} \text{phon}: W \end{bmatrix} \\ \text{synsem}: \begin{bmatrix} \text{cat}{=}\mathrm{v_i}: Cat \\ \text{cnt}{=}\lambda r{:}\begin{bmatrix} \text{x}: Ind \end{bmatrix}\ (\begin{bmatrix} \text{e}: p(r) \end{bmatrix}): Ppty \end{bmatrix} \end{bmatrix}$$

The temperature is rising (Cooper, 2010, 2012)

Composition rules as dependent types

unary rules $\lambda s : T_1(T_2)$, where $T_1, T_2 \sqsubseteq Sign$

binary rules $\lambda s : T_1 \frown T_2(T_3)$, where $T_1, T_2, T_3 \sqsubseteq Sign$

We need to address subtyping.

Subtyping

$T_1 \sqsubseteq T_2$ just in case $\{a \mid a : T_1\} \subseteq \{a \mid a : T_2\}$ *for all assignments to the basic types*.

Some examples:

- $\begin{bmatrix} \mathrm{f}: T_1 \\ \mathrm{g}: T_2 \end{bmatrix} \sqsubseteq \begin{bmatrix} \mathrm{f}: T_1 \end{bmatrix}$
- $\begin{bmatrix} \mathrm{f}{=}a: T \end{bmatrix} \sqsubseteq \begin{bmatrix} \mathrm{f}: T \end{bmatrix}$
- $T_1 \wedge T_2 \sqsubseteq T_1$
- $T_1 \sqsubseteq T_1 \vee T_2$

Rule components I

unary_sign $\lambda s{:}Sign(Sign)$

binary_sign $\lambda s{:}Sign^\frown Sign(Sign)$

phon_id $\lambda s{:}\left[\text{s-event}{:}\left[\text{phon}{:}Phon\right]\right]$
$\left(\left[\text{s-event}{:}\left[\text{phon}{=}s.\text{s-event.phon}{:}Phon\right]\right]\right)$

phon_concat $\lambda s{:}\left[\text{s-event}{:}\left[\text{phon}{:}Phon\right]\right]^\frown\left[\text{s-event}{:}\left[\text{phon}{:}Phon\right]\right]$
$\left(\left[\text{s-event}{:}\left[\text{phon}{=}s[1].\text{s-event.phon}^\frown s[2].\text{s-event.phon}{:}Phon\right]\right]\right)$

unary_cat $\lambda c_1{:}Cat(\lambda c_2{:}Cat(\lambda s{:}\left[\text{cat}{=}c_1{:}Cat\right]\left(\left[\text{cat}{=}c_2{:}Cat\right]\right)))$

binary_cat $\lambda c_1{:}Cat(\lambda c_2{:}Cat(\lambda c_3{:}Cat$
$(\lambda s{:}\left[\text{cat}{=}c_1{:}Cat\right]^\frown\left[\text{cat}{=}c_2{:}Cat\right]\left(\left[\text{cat}{=}c_3{:}Cat\right]\right))))$

cnt_id $\lambda s{:}\left[\text{synsem}{:}\left[\text{cnt}{:}Cnt\right]\right]$
$\left(\left[\text{synsem}{:}\left[\text{cnt}{=}s.\text{synsem.cnt}{:}Cnt\right]\right]\right)$

Rule components II

$$\text{cnt\_forw\_app} \;\; \lambda T_1{:}\,\mathit{Type}(\lambda T_2{:}\,\mathit{Type} \\ (\lambda s{:}\left[\text{synsem}{:}\left[\text{cnt}{:}\,T_1 \rightarrow T_2\right]\right]^\frown\left[\text{synsem}{:}\left[\text{cnt}{:}\,T_1\right]\right] \\ (\left[\text{synsem}{:}\left[\text{cnt}{=}s[1].\text{synsem.cnt}(s[2].\text{synsem.cnt}){:}\,T_2\right]\right])))$$

Merging functions

1. $\lambda v{:}T_1(T_2)\ \underset{:}{\wedge}\ \lambda v{:}T_3(T_4)$ is to be $\lambda v{:}T_1 \underset{.}{\wedge} T_3(T_2 \underset{.}{\wedge} T_4)$
2. $\lambda v{:}T_1{\frown}T_2(T_3)\ \underset{:}{\wedge}\ \lambda v{:}T_4{\frown}T_5(T_6)$ is to be $\lambda v{:}(T_1 \underset{.}{\wedge} T_4){\frown}(T_2 \underset{.}{\wedge} T_5)\ (T_3 \underset{.}{\wedge} T_6)$

Composition rules

$$
\begin{array}{rl}
\text{S} \rightarrow \text{NP VP} & \text{binary\_sign} \mathrel{\dot{\wedge}} \text{phon\_concat} \mathrel{\dot{\wedge}} \text{binary\_cat(np)(vp)(s)} \mathrel{\dot{\wedge}} \\
& \text{cnt\_forw\_app}(\mathit{Ppty})(\mathit{RecType}) \\
\text{NP} \rightarrow \text{N} & \text{unary\_sign} \mathrel{\dot{\wedge}} \text{phon\_id} \mathrel{\dot{\wedge}} \text{unary\_cat}(\text{n}_{\text{Prop}})\text{(np)} \mathrel{\dot{\wedge}} \\
& \text{cnt\_id} \\
\text{VP} \rightarrow \text{V}_i & \text{unary\_sign} \mathrel{\dot{\wedge}} \text{phon\_id} \mathrel{\dot{\wedge}} \text{unary\_cat}(\text{v}_i)\text{(vp)} \mathrel{\dot{\wedge}} \text{cnt\_id}
\end{array}
$$

Outline

String theory of events

Inference from partial observation of events

Grammar rules as dependent event types (Cooper, 2012)

Frames in lexical semantics (Cooper, 2010, 2012)

Getting serious about time: intervals

$$\textit{TimeInt} = \begin{bmatrix} \text{start} & : & \textit{Time} \\ \text{end} & : & \textit{Time} \\ \text{c}_{<} & : & \text{start}{<}\text{end} \end{bmatrix}$$

The content of *ran*

- $\lambda r{:}\left[\mathrm{x}{:}\mathit{Ind}\right]\left(\left[\begin{array}{lll} \text{e-time} & : & \mathit{TimeInt} \\ \mathrm{c}_{\mathrm{tns}} & : & \text{e-time.end} < \iota.\text{start} \\ \mathrm{c}_{\mathrm{run}} & : & \text{run}(r.\mathrm{x},\text{e-time}) \end{array}\right]\right)$

The content of *ran*

- $\lambda r{:}\begin{bmatrix}\text{x}:\mathit{Ind}\end{bmatrix}(\begin{bmatrix}\text{e-time} & : & \mathit{TimeInt} \\ \text{c}_{\text{tns}} & : & \text{e-time.end}< \iota\text{.start} \\ \text{c}_{\text{run}} & : & \text{run}(r\text{.x,e-time})\end{bmatrix})$
- Frames as arguments

The content of *ran*

- $\lambda r{:}\left[\text{x}{:}\textit{Ind}\right]\left(\begin{bmatrix} \text{e-time} & : & \textit{TimeInt} \\ \text{c}_{\text{tns}} & : & \text{e-time.end} < \iota\text{.start} \\ \text{c}_{\text{run}} & : & \text{run}(r\text{.x,e-time}) \end{bmatrix}\right)$
- Frames as arguments
- This is an individual level verb phrase interpretation, even though it takes a frame as argument, the predicate ‘run’ takes an individual as argument.

rises – a predicate of frames

- $\lambda r{:}\begin{bmatrix}\text{x}:\mathit{Ind}\end{bmatrix}\left(\begin{bmatrix}\text{e-time} & : & \mathit{TimeInt}\\ \text{c}_{\text{tns}} & : & \text{e-time}=\iota\\ \text{c}_{\text{run}} & : & \text{rise}(r,\text{e-time})\end{bmatrix}\right)$

rises – a predicate of frames

- $\lambda r{:}\left[\text{x}{:}\mathit{Ind}\right]\left(\begin{bmatrix} \text{e-time} & : & \mathit{TimeInt} \\ \text{c}_{\text{tns}} & : & \text{e-time}=\iota \\ \text{c}_{\text{run}} & : & \text{rise}(r,\text{e-time}) \end{bmatrix}\right)$
- This is a frame level verb phrase interpretation, the predicate ‘rise’ takes a frame as argument

rises – a predicate of frames

- $\lambda r{:}\begin{bmatrix}\text{x}{:}\mathit{Ind}\end{bmatrix}\left(\begin{bmatrix}\text{e-time} & : & \mathit{TimeInt} \\ \text{c}_{\text{tns}} & : & \text{e-time}= \iota \\ \text{c}_{\text{run}} & : & \text{rise}(r,\text{e-time})\end{bmatrix}\right)$
- This is a frame level verb phrase interpretation, the predicate 'rise' takes a frame as argument
- *The temperature is rising, the temperature is ninety* $\not\Rightarrow$ *Ninety is rising*

rises – a predicate of frames

- $\lambda r{:}\left[\text{x}{:}\textit{Ind}\right]\left(\begin{bmatrix} \text{e-time} & : & \textit{TimeInt} \\ \text{c}_{\text{tns}} & : & \text{e-time}{=}\ \iota \\ \text{c}_{\text{run}} & : & \text{rise}(r,\text{e-time}) \end{bmatrix}\right)$
- This is a frame level verb phrase interpretation, the predicate ‘rise’ takes a frame as argument
- *The temperature is rising, the temperature is ninety* $\not\Rightarrow$ *Ninety is rising*
- What can it mean for a frame to rise?

The record type *AmbTemp*

- $\left[\begin{array}{lll} \mathrm{x} & : & \mathit{Ind} \\ \text{e-time} & : & \mathit{Time} \\ \text{e-location} & : & \mathit{Loc} \\ \mathrm{c}_{\mathrm{temp\_at\_in}} & : & \text{temp\_at\_in(e-time, e-location, x)} \end{array}\right]$

The record type *AmbTemp*

- $\begin{bmatrix} \mathsf{x} & : & \mathit{Ind} \\ \mathsf{e\text{-}time} & : & \mathit{Time} \\ \mathsf{e\text{-}location} & : & \mathit{Loc} \\ \mathsf{c}_{\mathrm{temp\_at\_in}} & : & \mathsf{temp\_at\_in(e\text{-}time, e\text{-}location, x)} \end{bmatrix}$
- a record of this type will meet the following two conditions:
  - it will contain *at least* fields with the same labels as the type (it may contain more)

The record type *AmbTemp*

- $$\begin{bmatrix} \text{x} & : & \mathit{Ind} \\ \text{e-time} & : & \mathit{Time} \\ \text{e-location} & : & \mathit{Loc} \\ \text{c}_{\text{temp\_at\_in}} & : & \text{temp\_at\_in(e-time, e-location, x)} \end{bmatrix}$$
- a record of this type will meet the following two conditions:
  - it will contain *at least* fields with the same labels as the type (it may contain more)
  - each field in the record with the same label as a field in the record type will contain an object of the type in the corresponding field of the record type. (Any additional fields with different labels to those in the record type may contain objects of any type.)

The type *TempRise*

$$\left[\begin{array}{l}\text{e-time}:\textit{TimeInt}\\ \text{start}:\left[\begin{array}{l}\text{x}:\textit{Ind}\\ \text{e-time=e-time.start}:\textit{Time}\\ \text{e-location}:\textit{Loc}\\ \text{c}_{\text{temp\_at\_in}}:\text{temp\_at\_in(start.e-time, start.e-location, start.x)}\end{array}\right]\\ \text{end}:\left[\begin{array}{l}\text{x}:\textit{Ind}\\ \text{e-time=e-time.end}:\textit{Time}\\ \text{e-location=start.e-location}:\textit{Loc}\\ \text{c}_{\text{temp\_at\_in}}:\text{temp\_at\_in(end.e-time, end.e-location, end.x)}\end{array}\right]\\ \text{event=start}^\frown\text{end}:\textit{AmbTemp}^\frown\textit{AmbTemp}\\ \text{c}_{\text{incr}}:\text{start.x<end.x}\end{array}\right]$$

The type *TempRise*

$$\left[\begin{array}{ll}\text{e-time}: & \textit{TimeInt}\\ \text{start}: & \left[\begin{array}{l}\text{x}:\textit{Ind}\\ \text{e-time=e-time.start}:\textit{Time}\\ \text{e-location}:\textit{Loc}\\ \text{c}_{\text{temp\_at\_in}}:\text{temp\_at\_in(start.e-time, start.e-location, start.x)}\end{array}\right]\\ \text{end}: & \left[\begin{array}{l}\text{x}:\textit{Ind}\\ \text{e-time=e-time.end}:\textit{Time}\\ \text{e-location=start.e-location}:\textit{Loc}\\ \text{c}_{\text{temp\_at\_in}}:\text{temp\_at\_in(end.e-time, end.e-location, end.x)}\end{array}\right]\\ \multicolumn{2}{l}{\text{event=start}^\frown\text{end}:\textit{AmbTemp}^\frown\textit{AmbTemp}}\\ \multicolumn{2}{l}{\text{c}_{\text{incr}}:\text{start.x<end.x}}\end{array}\right]$$

If r:*AmbTemp* and i:*TimeInt* then e:rise(r,i) iff e:*TempRise*, e.start=r and e.e-time=i.

Does *rise* have a general meaning?

At this level of detail, the type of an event of a price rise is slightly different.

The type *PriceRise*

$$\left[\begin{array}{l}
\text{e-time}:\mathit{TimeInt}\\
\text{start}:\left[\begin{array}{l}
\text{x}:\mathit{Ind}\\
\text{e-time}{=}\text{e-time.start}:\mathit{Time}\\
\text{e-location}:\mathit{Loc}\\
\text{commodity}:\mathit{Ind}\\
\text{c}_{\text{price_of_at_in}}:\text{price_of_at_in(start.commodity,}\\
\qquad\qquad\qquad\text{start.e-time, start.e-location, start.x)}
\end{array}\right]\\
\text{end}:\left[\begin{array}{l}
\text{x}:\mathit{Ind}\\
\text{e-time}{=}\text{e-time.end}:\mathit{Time}\\
\text{e-location}{=}\text{start.e-location}:\mathit{Loc}\\
\text{commodity}{=}\text{start.commodity}:\mathit{Ind}\\
\text{c}_{\text{price_of_at_in}}:\text{price_of_at_in(end.commodity,}\\
\qquad\qquad\qquad\text{end.e-time, end.e-location, end.x)}
\end{array}\right]\\
\text{event}{=}\text{start}^\frown\text{end}:\mathit{Price}^\frown\mathit{Price}\\
\text{c}_{\text{incr}}:\text{start.x}<\text{end.x}
\end{array}\right]$$

The Titan rises

(description of a video game)

As they get to deck, they see the Inquisitor, calling out to a Titan in the seas. **The giant Titan rises through the waves**, shrieking at the Inquisitor.

The type *LocRise*

$$\left[\begin{array}{l}
\text{e-time}:\mathit{TimeInt}\\
\text{start}:\ \left[\begin{array}{l}\text{x}:\mathit{Ind}\\ \text{e-time=e-time.start}:\mathit{Time}\\ \text{e-location}:\mathit{Loc}\\ \text{c}_{\text{at}}:\text{at(start.x,start.e-location,start.e-time)}\end{array}\right]\\
\text{end}:\ \left[\begin{array}{l}\text{x=start.x}:\mathit{Ind}\\ \text{e-time=e-time.end}:\mathit{Time}\\ \text{e-location}:\mathit{Loc}\\ \text{c}_{\text{at}}:\text{at(end.x,end.e-location,end.e-time)}\end{array}\right]\\
\text{event=start}^\frown\text{end}:\mathit{Position}^\frown\mathit{Position}\\
\text{c}_{\text{incr}}:\text{height(start.e-location)}<\text{height(end.e-location)}
\end{array}\right]$$

Semantic coordination

(Work with Staffan Larsson: Larsson, 2007; Cooper and Larsson, 2009; Larsson and Cooper, 2009)

Instead of "What is the meaning of expression E?"

Semantic coordination

(Work with Staffan Larsson: Larsson, 2007; Cooper and Larsson, 2009; Larsson and Cooper, 2009)

Instead of "What is the meaning of expression E?"

we propose

the coordination question Given resources R, how can agent A construct a meaning for a particular utterance U of expression E?

Semantic coordination

(Work with Staffan Larsson: Larsson, 2007; Cooper and Larsson, 2009; Larsson and Cooper, 2009)

Instead of "What is the meaning of expression E?"

we propose

the coordination question Given resources R, how can agent A construct a meaning for a particular utterance U of expression E?

the resource update question What effect will this have on A's resources R?

Examples without context

How much do you understand?

- Mastercard rises
- China rises
- dog hairs rise

Mastercard rises

Visa Up on Q1 Beat, Forecast; **Mastercard Rises** in Sympathy
By Tiernan Ray
Shares of Visa (V) and Mastercard (MA) are both climbing in the aftermarket, reversing declines during the regular session, after Visa this afternoon reported fiscal Q1 sales and profit ahead of estimates and forecast 2010 sales growth ahead of estimates, raising enthusiasm for its cousin, Mastercard.

China rises

The rise of China will undoubtedly be one of the great dramas of the twenty-first century. China's extraordinary economic growth and active diplomacy are already transforming East Asia, and future decades will see even greater increases in Chinese power and influence. But exactly how this drama will play out is an open question. Will China overthrow the existing order or become a part of it? And what, if anything, can the United States do to maintain its position as **China rises**?

rise and clarification

Cherrilyn:	Yeah I mean ⟨pause⟩ **dog hairs rise** anyway so
Fiona:	What do you mean, rise?
Cherrilyn:	The hair ⟨pause⟩ it rises upstairs.

BNC file KBL, sentences 4201–4203

Cherrilyn 4192 Yeah , he likes, likes climbing on the bed.
4193 Makes him feel a bit comfy.
4194 Most dogs aren't <*pause*> allowed on beds.
Fiona 4195 Most dogs aren't, some dogs aren't even allowed upstairs!
Cherrilyn 4196 Most dogs aren't allowed up <*pause*> upstairs.
4197 He's allowed to go wherever he wants <*pause*> do whatever he likes.
Fiona 4198 Too right!
4199 So they should!
4200 Shouldn't they?
Cherrilyn 4201 Yeah I mean <*pause*> dog hairs **rise** anyway so
Fiona 4202 What do you mean, **rise**?
Cherrilyn 4203 The hair <*pause*> it **rises** upstairs.
4204 I mean I, you know friends said it was, oh God I wouldn't allow mine upstairs because of all the <*pause*> dog hairs!
4205 Oh well <*pause*> they go up there anyway.
Fiona 4206 So, but I don't know what it is, right, it's only a few bloody hairs!
4207 <*pause dur=8*> He's actually allowed to do whatever he likes.
4208 Gets his own way as well.
4209 <*pause dur=9*> Do you want a drink mum?
Cherrilyn 4210 No thanks.
Fiona 4211 Good!
4212 Ah he must be ill, he's in the bed!
Cherrilyn 4213 Eh? <*unclear*> <*pause*> not ill!

How easy is it to understand *rise*?

- 205 occurrences of *rise* as a verb in the BNC dialogue subcorpus

How easy is it to understand *rise*?

- 205 occurrences of *rise* as a verb in the BNC dialogue subcorpus
- One clarification

Summary

- detailed account of how syntactic rules can be related to the perception of events and reasoning about them
- strings of events rather than strings of symbols
- a speech act based view of grammar rather than a formal language view
- we have broken down syntactic rules into small components which can be combined
- we want ultimately to explain how agents construct new composition rules, change their language or learn new languages by combining rule components in different ways
- lexical semantics: frames, a system in flux

Bibliography I

Cooper, Robin (2010) Frames in formal semantics, in H. Loftsson, E. Rögnvaldsson and S. Helgadóttir (eds.), *IceTAL 2010*, Springer Verlag.

Cooper, Robin (2012) Type Theory and Semantics in Flux, in R. Kempson, N. Asher and T. Fernando (eds.), *Handbook of the Philosophy of Science*, Vol. 14: Philosophy of Linguistics, pp. 271–323, Elsevier BV. General editors: Dov M. Gabbay, Paul Thagard and John Woods.

Bibliography II

Cooper, Robin and Staffan Larsson (2009) Compositional and ontological semantics in learning from corrective feedback and explicit definition, in J. Edlund, J. Gustafson, A. Hjalmarsson and G. Skantze (eds.), *Proceedings of DiaHolmia: 2009 Workshop on the Semantics and Pragmatics of Dialogue*, pp. 59–66.

Fernando, Tim (2004) A finite-state approach to events in natural language semantics, *Journal of Logic and Computation*, Vol. 14, No. 1, pp. 79–92.

Fernando, Tim (2006) Situations as Strings, *Electronic Notes in Theoretical Computer Science*, Vol. 165, pp. 23–36.

Bibliography III

Fernando, Tim (2008) Finite-state descriptions for temporal semantics, in H. Bunt and R. Muskens (eds.), *Computing Meaning, Volume 3* (*Studies in Linguistics and Philosophy* 83), pp. 347–368, Springer.

Fernando, Tim (2009) Situations in LTL as strings, *Information and Computation*, Vol. 207, No. 10, pp. 980–999.

Larsson, Staffan (2007) A general framework for semantic plasticity and negotiation, in H. Bunt and E. C. G. Thijsse (eds.), *Proceedings of the 7th International Workshop on Computational Semantics (IWCS-7)*, pp. 101–117.

Larsson, Staffan and Robin Cooper (2009) Towards a formal view of corrective feedback, in A. Alishahi, T. Poibeau and A. Villavicencio (eds.), *Proceedings of the Workshop on Cognitive Aspects of Computational Language Acquisition*, pp. 1–9.