Austinian truth in Martin-Löf type theory

Robin Cooper
Göteborg University

Austinian truth

Barwise and Perry (*Situations and Attitudes*) paraphrase Austin:

> a statement is true when the actual situation to which it refers is of the type described by the statement.

Austin's original ('Truth', 1961)

> A statement is said to be true when the historic state of affairs to which it is correlated by the demonstrative conventions (the ones to which it 'refers') is of a type with which the sentence used in making it is correlated by the descriptive conventions.

- Statements are true of something (part of the world) not true *simpliciter*
- Truth *simpliciter* can be derived: there is some part of the world of which the statement is true
- Statements are correlated with types

Martin-Löf Type Theory

- Objects are of types
- Propositions are regarded as types of proofs (“propositions as types” principle)
- Proofs are objects, e.g. the proofs of *there is a prime number between 212 and 222* are the prime numbers between 212 and 222

Proofs of non-mathematical propositions

Ranta in *Type-Theoretical Grammar* draws a parallel with Davidson's event-based approach:

A proof of

> *Amundsen flew over the North Pole*

is

> a flight by Amundsen over the North Pole

In terms of situation theory:

A proof of

Amundsen flew over the North Pole

is

a **situation** in which Amundsen flies over the North Pole

Infons (states of affairs, soas) – types of situations

TT	**ST**
types of proofs (propositions)	infons

Sentences of situation theory

$s \models \sigma$ (“situation s supports infon (soa) σ” or “s is of type σ”)

Situation theoretic objects

$(s \models \sigma)$, an Austinian proposition, true just in case $s \models \sigma$

Judgements

Type theoretical judgements

$a : T$ (“object a is of type T”)

Type theoretical objects

an object a, of type T – $a : T$

No objects corresponding to judgements as such

Judgements concerning proofs

p : see(a, b)

p is a proof that a sees b

context: a:Ind, b:Ind

A note on individuals

What is the type Ind?

Schemes of individuation

Records

Contexts are not objects but records are.

We shall use record types which have fields corresponding both to the context and the propositional judgement.

$$T = \begin{bmatrix} \text{x} & : & \text{Ind} \\ \text{y} & : & \text{Ind} \\ \text{prf} & : & \text{see(x,y)} \end{bmatrix}$$

$$\begin{bmatrix} \text{x} & = & a \\ \text{y} & = & b \\ \text{prf} & = & p \\ \ldots & & \end{bmatrix} : T \text{ just in case } a : \text{Ind}, b : \text{Ind and } p : \text{see}(a, b)$$

Why records?

Three reasons independent of situation theory:

- allow us to reify contexts and quantify over them
- correspond to discourse representation structures – a record type is inhabited ("true") just in case there is something of the appropriate type corresponding to each of its labels.
- records can be values in fields as in typed feature structures – useful for compositional semantics, parallels with HPSG

Situations as records

Records and record types give us the possibility of grouping together collections of infon-like objects.

$$\left[\begin{array}{lll} \text{s} & : & \left[\begin{array}{lll} \text{x} & : & \text{Ind} \\ \text{y} & : & \text{Ind} \\ \text{s}_1 & : & \text{see(x,y)} \\ \text{s}_2 & : & \text{see(y,x)} \end{array}\right] \end{array}\right]$$

$$\left[\begin{array}{lll} \text{x} & : & \text{Ind} \\ \text{y} & : & \text{Ind} \\ \text{s} & : & \left[\begin{array}{lll} \text{s}_1 & : & \text{see(x,y)} \\ \text{s}_2 & : & \text{see(y,x)} \end{array}\right] \end{array}\right]$$

Constraints on universes of records

smoke means fire

Any universe of records which obeys this constraint is such that if for some labels l and s it contains a record of type

$$\begin{bmatrix} l & : & \text{Loc} \\ s & : & \text{smoke}(l) \end{bmatrix}$$

then it also, for some labels l' and s', contains a record of type

$$\begin{bmatrix} l' & : & \text{Loc} \\ s' & : & \text{fire}(l') \end{bmatrix}$$

kissing involves touching

Applies to single situations (you cannot perceive a kiss b without perceiving a touch b)

Any record in the universe which for some labels x,y and s is of type

$$\begin{bmatrix} x & : & \text{Ind} \\ y & : & \text{Ind} \\ s & : & \text{kiss}(x,y) \end{bmatrix}$$

is also, for some label s', of type

$$\begin{bmatrix} x & : & \text{Ind} \\ y & : & \text{Ind} \\ s' & : & \text{touch}(x,y) \end{bmatrix}$$

Note that x and y in the second record to depend on the first record.

Can constraints be internalized?

$$\left[\ \text{f} \ : \ (\left[\begin{array}{lll} l & : & \text{Loc} \\ s & : & \text{smoke}(l) \end{array}\right]) \left[\begin{array}{lll} l' & : & \text{Loc} \\ s' & : & \text{fire}(l') \end{array}\right]\ \right]$$

$$\left[\ \text{f} \ : \ (r\text{:}\left[\begin{array}{lll} x & : & \text{Ind} \\ y & : & \text{Ind} \\ s & : & \text{kiss}(x,y) \end{array}\right]) \left[\begin{array}{lll} r' & : & \left[\begin{array}{lll} s' & : & \text{touch}(r.x,r.y) \end{array}\right] \\ c & : & \text{eq}(\text{RecType}, r', r) \end{array}\right]\ \right]$$

Functions correspond to universal quantification.

What to do if the constraint is soft or *fallible*? Functions in TT are not normally partial.

Channel types

$$\mathcal{C} = \left[\begin{array}{lll} \text{signal} & : & T_1 \\ \text{target} & : & T_2 \end{array}\right]$$

A record $r : \mathcal{C}$ is a *link classified by* $\mathcal{C}$.

An agent attuned to $\mathcal{C}$ on observing an object of type T_1 will predict (defeasibly) the existence of an object of type T_2.

According to $\mathcal{C}$ an object $a : T_1$ *carries the information that* T_2 is inhabited.

Dependent target types

Becomes of some interest when we consider that the target type may depend on the signal type.

$$\left[\begin{array}{lll} \text{signal} & : & \left[\begin{array}{lll} \text{x} & : & \text{Ind} \\ \text{y} & : & \text{Ind} \\ \text{c}_1 & : & \text{r(x,y)} \end{array}\right] \\ \text{target} & : & \left[\begin{array}{lll} \text{c}_2 & : & \text{r}'\text{(signal.x, signal.y)} \end{array}\right] \end{array}\right]$$

Note that we do not know what the target type is until we have determined a particular signal.

Records and attitudes

A simple treatment of the attitudes

Record types as the object of the attitude (corresponding to a ST treatment where attitudes are relations between individuals and infons).

$$\begin{bmatrix} \mathrm{x} & : & \mathrm{Ind} \\ \mathrm{c}_1 & : & \mathrm{girl(x)} \\ \mathrm{c}_2 & : & \mathrm{believe(x,} \begin{bmatrix} \mathrm{y} & : & \mathrm{Ind} \\ \mathrm{c}_3 & : & \mathrm{man(y)} \\ \mathrm{z} & : & \mathrm{Ind} \\ \mathrm{c}_4 & : & \mathrm{donkey(z)} \\ \mathrm{c}_5 & : & \mathrm{own(y, z)} \end{bmatrix} \mathrm{)} \end{bmatrix}$$

- The object of belief is Austinian.
- Austinian propositions could be represented as pairs of records and record types (coded as a record).

Perception attitudes

a man sees that a donkey kicked a farmer

$$\left[\begin{array}{lll} \text{x} & : & \text{Ind} \\ \text{c}_1 & : & \text{man(x)} \\ \text{c}_5 & : & \text{see(x,}\left[\begin{array}{lll} \text{y} & : & \text{Ind} \\ \text{c}_2 & : & \text{donkey(y)} \\ \text{z} & : & \text{Ind} \\ \text{c}_3 & : & \text{farmer(z)} \\ \text{c}_4 & : & \text{kick(y, z)} \end{array}\right]\text{)} \end{array}\right]$$

Veridicality

$$\left[\begin{array}{lll} \text{f} & : & (r{:}\left[\begin{array}{lll} x & : & \text{Ind} \\ T & : & \text{RecType} \\ c & : & \text{see}(x,T) \end{array}\right])\left[\begin{array}{lll} a & : & r.T \end{array}\right] \end{array}\right]$$

Note that this does not require that the man saw a donkey-kicking-farmer situation – he may just have seen donkey hoof marks on the farmer's legs.

Naked infinitive perception complements

a man sees a donkey kick a farmer

$$\begin{bmatrix} \text{x} & : & \text{Ind} \\ \text{c}_1 & : & \text{man(x)} \\ \text{s} & : & \begin{bmatrix} \text{y} & : & \text{Ind} \\ \text{c}_2 & : & \text{donkey(y)} \\ \text{z} & : & \text{Ind} \\ \text{c}_3 & : & \text{farmer(z)} \\ \text{c}_4 & : & \text{kick(y, z)} \end{bmatrix} \\ \text{c}_5 & : & \text{see(x,s)} \end{bmatrix}$$

Veridicality is required.

Also perception of the donkey-kicking-farmer situation.

Types are not enough

- Proper names
- Presuppositions

Sandy kicks a farmer

$$\begin{bmatrix} \text{y} & : & \text{Ind} \\ \text{c}_2 & : & \text{named(y, “Sandy”)} \\ \text{z} & : & \text{Ind} \\ \text{c}_3 & : & \text{farmer(z)} \\ \text{c}_4 & : & \text{kick(y, z)} \end{bmatrix}$$

Frames of mind

A *family of types* – function from records of a given type to a record type.

$$\lambda r : \begin{bmatrix} \text{y} & : & \text{Ind} \\ \text{c}_2 & : & \text{named(y, “Sandy”)} \end{bmatrix} \begin{bmatrix} \text{z} & : & \text{Ind} \\ \text{c}_3 & : & \text{farmer(z)} \\ \text{c}_4 & : & \text{kick}(r\text{.y, z)} \end{bmatrix}$$

A *setting* for a frame of mind is a record in its domain.

The *presupposition* associated with a frame of mind is the type characterizing its domain.

A *mental state* is a pair consisting of a frame of mind (agent internal) and a setting (agent external).

The *content of a mental state* is the result of applying the frame of mind to the setting.

Advantage over Cooper and Ginzburg (1996) : no restricted objects

Pierre

Frame of mind

$$\lambda r : \begin{bmatrix} \text{x} & : & \text{Ind} \\ \text{c}_1 & : & \text{named(x, “Londres”)} \\ \text{y} & : & \text{Ind} \\ \text{c}_2 & : & \text{named(y, “London”)} \end{bmatrix} \begin{bmatrix} \text{c}_3 & : & \text{pretty}(r\text{.x}) \\ \text{c}_4 & : & \neg\text{pretty}(r\text{.y}) \end{bmatrix}$$

Setting

$$\begin{bmatrix} \text{x} & = & \text{london} \\ \text{c}_1 & = & \text{pf named(london “Londres”)} \\ \text{y} & = & \text{london} \\ \text{c}_2 & = & \text{pf named(london “London”)} \end{bmatrix}$$

Content of Pierre’s mental state

$$\begin{bmatrix} \text{c}_3 & : & \text{pretty(london)} \\ \text{c}_4 & : & \neg\text{pretty(london)} \end{bmatrix}$$

Compositionality

Sample derivation: *every man owns a donkey*

a donkey

$$\lambda R_1{:}(\left[\text{x:Ind}\right])\text{RecType}\ \lambda R_2{:}(\left[\text{x:Ind}\right])\text{RecType}\ \left[\begin{array}{lll} \text{par} & : & \left[\begin{array}{lll}\text{x} & : & \text{Ind}\end{array}\right] \\ \text{restr} & : & R_1\ @\ \text{par} \\ \text{scope} & : & R_2\ @\ \text{par} \end{array}\right]$$

@

$$\lambda r{:}\left[\text{x:Ind}\right](\left[\text{c:donkey}(r.\text{x})\right])$$

=

$$\lambda R_2{:}(\left[\text{x:Ind}\right])\text{RecType}\ \left[\begin{array}{lll} \text{par} & : & \left[\begin{array}{lll}\text{x} & : & \text{Ind}\end{array}\right] \\ \text{restr} & : & \left[\begin{array}{lll}\text{c} & : & \text{donkey(par.x)}\end{array}\right] \\ \text{scope} & : & R_2\ @\ \text{par} \end{array}\right]$$

own a donkey

$$\lambda\mathcal{N}\text{:}((\left[\text{x:Ind}\right])\text{RecType})\text{RecType}\ \lambda r_1\text{:}\left[\text{x:Ind}\right]\ (\mathcal{N}\ @\ \lambda r_2\text{:}\left[\text{x:Ind}\right](\left[\text{c:own}(r_1\text{.x},\ r\right.$$

@

$$\lambda R_2\text{:}(\left[\text{x:Ind}\right])\text{RecType}\ \begin{bmatrix} \text{par} & : & \begin{bmatrix} \text{x} & : & \text{Ind} \end{bmatrix} \\ \text{restr} & : & \begin{bmatrix} \text{c} & : & \text{donkey(par.x)} \end{bmatrix} \\ \text{scope} & : & R_2\ @\ \text{par} \end{bmatrix}$$

=

$$\lambda r_1\text{:}\left[\text{x:Ind}\right]\ (\begin{bmatrix} \text{par} & : & \begin{bmatrix} \text{x} & : & \text{Ind} \end{bmatrix} \\ \text{restr} & : & \begin{bmatrix} \text{c} & : & \text{donkey(par.x)} \end{bmatrix} \\ \text{scope} & : & \begin{bmatrix} \text{c} & : & \text{own}(r_1\text{.x,par.x}) \end{bmatrix} \end{bmatrix})$$

every man

λR_1:($\left[\text{x:Ind}\right]$)RecType λR_2:($\left[\text{x:Ind}\right]$)RecType

$$\left[\begin{array}{lll} \text{f} & : & (r : \left[\begin{array}{lll} \text{par} & : & \left[\begin{array}{lll} \text{x} & : & \text{Ind} \end{array}\right] \\ \text{restr} & : & R_1 \ @ \ \text{par} \end{array}\right]) \ R \end{array}\right.$$

@

λr:$\left[\text{x:Ind}\right]$($\left[\text{c:man}(r.\text{x})\right]$)

=

λR_2:($\left[\text{x:Ind}\right]$)RecType

$$\left[\begin{array}{lll} \text{f} & : & (r : \left[\begin{array}{lll} \text{par} & : & \left[\begin{array}{lll} \text{x} & : & \text{Ind} \end{array}\right] \\ \text{restr} & : & \left[\begin{array}{lll} \text{c} & : & \text{man(par.x)} \end{array}\right] \end{array}\right]) \ R_2 \ @ \ r.\text{par} \end{array}\right]$$

every man owns a donkey

$$\lambda R_2\text{:}(\begin{bmatrix}\text{x:Ind}\end{bmatrix})\text{RecType}$$

$$\begin{bmatrix} \text{f} & : & (r : \begin{bmatrix} \text{par} & : & \begin{bmatrix} \text{x} & : & \text{Ind} \end{bmatrix} \\ \text{restr} & : & \begin{bmatrix} \text{c} & : & \text{man(par.x)} \end{bmatrix} \end{bmatrix}) \; R_2 \mathbin{@} r\text{.par} \end{bmatrix}$$

@

$$\lambda r_1\text{:}\begin{bmatrix}\text{x:Ind}\end{bmatrix} \; (\begin{bmatrix} \text{par} & : & \begin{bmatrix} \text{x} & : & \text{Ind} \end{bmatrix} \\ \text{restr} & : & \begin{bmatrix} \text{c} & : & \text{donkey(par.x)} \end{bmatrix} \\ \text{scope} & : & \begin{bmatrix} \text{c} & : & \text{own}(r_1\text{.x,par.x}) \end{bmatrix} \end{bmatrix})$$

=

$$\begin{bmatrix} \text{f} & : & (r : \begin{bmatrix} \text{par} & : & \begin{bmatrix} \text{x} & : & \text{Ind} \end{bmatrix} \\ \text{restr} & : & \begin{bmatrix} \text{c} & : & \text{man(par.x)} \end{bmatrix} \end{bmatrix}) \begin{bmatrix} \text{par} & : & \begin{bmatrix} \text{x} & : & \text{Ind} \end{bmatrix} \\ \text{restr} & : & \begin{bmatrix} \text{c} & : & \text{donkey(par.x} \end{bmatrix} \\ \text{scope} & : & \begin{bmatrix} \text{c} & : & \text{own}(r\text{.par.x,p} \end{bmatrix} \end{bmatrix}$$

Flattening

$$\left[\begin{array}{lll} \text{f} & : & (r : \left[\begin{array}{lll} \text{par.x} & : & \text{Ind} \\ \text{restr.c} & : & \text{man(par.x)} \end{array}\right]) \left[\begin{array}{lll} \text{par.x} & : & \text{Ind} \\ \text{restr.c} & : & \text{donkey(par.x)} \\ \text{scope.c} & : & \text{own(}r\text{.par.x,par.x)} \end{array}\right] \end{array}\right]$$

Relabelling

$$\left[\begin{array}{lll} \text{f} & : & (r : \left[\begin{array}{lll} \text{x} & : & \text{Ind} \\ \text{c}_1 & : & \text{man(x)} \end{array}\right]) \left[\begin{array}{lll} \text{y} & : & \text{Ind} \\ \text{c}_2 & : & \text{donkey(y)} \\ \text{c}_3 & : & \text{own(}r\text{.x,y)} \end{array}\right] \end{array}\right]$$

r-suppression (syntactic sugar)

$$\left[\begin{array}{lll} \text{f} & : & (\left[\begin{array}{lll} \text{x} & : & \text{Ind} \\ \text{c}_1 & : & \text{man(x)} \end{array}\right]) \left[\begin{array}{lll} \text{y} & : & \text{Ind} \\ \text{c}_2 & : & \text{donkey(y)} \\ \text{c}_3 & : & \text{own(x,y)} \end{array}\right] \end{array}\right]$$

$[\![$ $[_{V_S}$ believes$]$ $]\!]$ = λT:RecType λr:$\left[\text{x:Ind}\right]$($\left[\text{c:believe}(r.\text{x},T)\right]$)

$[\![$ $[_{V_S}$ sees$]$ $]\!]$ = λT:RecType λr:$\left[\text{x:Ind}\right]$($\left[\text{c:see}_t(r.\text{x},T)\right]$)

$$[\![\ [_{V_{NI}}\ \text{sees}]\ ]\!] = \lambda\mathcal{N}:((\left[\text{x:Ind}\right])\text{RecType})\text{RecType}$$
$$\lambda R:(\left[\text{x:Ind}\right])\text{RecType}$$
$$\lambda r:\left[\text{x:Ind}\right](\begin{bmatrix} \text{s} & : & \mathcal{N}\ @\ R \\ \text{c} & : & \text{see}_s(r.\text{x},\text{s}) \end{bmatrix})$$

$[\![$ $[_{VP}$ V_S S$]$ $]\!]$ = $[\![$ V_S $]\!]$ @ $[\![$ S $]\!]$

$[\![$ $[_{VP}$ V_{NI} NP VP[inf]$]$ $]\!]$ = $[\![$ V_{NI} $]\!]$ @ $[\![$ NP $]\!]$ @ $[\![$ VP[inf] $]\!]$

Records as resources

Domain restrictions

every/the man owns a donkey

Resource situations in situation semantics

"everything which is a man in situation s"

"everything which is a man in record r"

We use

$$r \models T$$

to represent the type of objects in record r which are of type T

$a : r \models T$ just in case for some ℓ, $r : \left[\ell{:}T\right]$ and $r.\ell = a$

Anchored resources (specific)

$$\lambda r_1 : \text{Rec}$$
$$\lambda r_2 : \text{Rec}$$
$$\left[\ \text{f} \ : \ (r : \begin{bmatrix} \text{x} & : & r_1 \models \text{Ind} \\ \text{c}_1 & : & r_1 \models \text{man(x)} \end{bmatrix}) \begin{bmatrix} \text{y} & : & r_2 \models \text{Ind} \\ \text{c}_2 & : & r_2 \models \text{donkey(y)} \\ \text{c}_3 & : & \text{own}(r\text{.x,y}) \end{bmatrix}\ \right]$$
$$@\ \textit{res}_1\ @\ \textit{res}_2$$

"Every man in *res*$_1$ owns a donkey in *res*$_2$."

Non-anchored resources (non-specific, existentially quantified)

$$\begin{bmatrix} \text{res}_1 & : & \text{Rec} \\ \text{res}_2 & : & \text{Rec} \\ \text{f} & : & (r : \begin{bmatrix} \text{x} & : & \text{res}_1 \models \text{Ind} \\ \text{c}_1 & : & \text{res}_1 \models \text{man(x)} \end{bmatrix}) \begin{bmatrix} \text{y} & : & \text{res}_2 \models \text{Ind} \\ \text{c}_2 & : & \text{res}_2 \models \text{donkey(y)} \\ \text{c}_3 & : & \text{own}(r\text{.x,y}) \end{bmatrix} \end{bmatrix}$$

"There are resources *res*$_1$, *res*$_2$ such that every man in *res*$_1$ owns a donkey in *res*$_2$."

Quantification over resources (a kind of generic)

$$\left[\begin{array}{lll} \text{f} & : & (r : \left[\begin{array}{lll} \text{res}_1 & : & \text{Rec} \\ \text{x} & : & \text{res}_1 \models \text{Ind} \\ \text{c}_1 & : & \text{res}_1 \models \text{man(x)} \end{array}\right]) \left[\begin{array}{lll} \text{res}_2 & : & \text{Rec} \\ \text{y} & : & \text{res}_2 \models \text{Ind} \\ \text{c}_2 & : & \text{res}_2 \models \text{donkey(y)} \\ \text{c}_3 & : & \text{own}(r\text{.x,y}) \end{array}\right] \end{array}\right]$$

"For every resource *res*$_1$ and every man x in *res*$_1$ there is a resource *res*$_2$ such that x owns a donkey in *res*$_2$."

Records (situations) as modules

Types restricted to resources import information from the resource (cf importing from modules in programming languages).

If $r : \left[\begin{array}{lll} a & : & r_1 \models T \end{array}\right]$
then $r : \left[\begin{array}{lll} a & : & T \end{array}\right]$

r can be used as a resource for another record.

This kind of "information spreading" not present in situation theory.

Conclusions

Records in Martin-Löf type theory

- allow us to combine various natural language "technologies"
  - Montague lambda technology
  - DRT, dynamic semantics
  - typed feature structures (HPSG)
- give us a promising approach to situations and intensionality, including
  - an Austinian approach to perception and attitude complements
  - constraints/channels
- exploit key features of MLTT
  - dependent types
  - propositions as types