Dependent-Type Theory of Situated Information with Context Assessments

Roussanka Loukanova

Institute of Mathematics and Informatics (IMI), Bulgarian Academy of Sciences (BAS), Sofia

June 24, 2022

Origins: Model Theory of Situated Information with Applications to Semantics

- Barwise [1, 2] (1981–1983) is the most influential, early work
  - introducing a strategy on Situation Theory (SitT) and Situation Semantics (SitSem)
- Seligman and Moss [8] (2011): an introduction to mathematical model theory of SitT
  - Situation Theory is a mathematical model of situated, partial information
  - Situation Semantics is an application of Situation Theory to semantics of human languages, e.g., applications to computational semantics in:
    - large-scale grammars of human language, in particular: Head-Driven Phrase Structure Grammar (HPSG)
- Loukanova [4, 5] (2014–2019) initiated new prospects of Situation Theory based on
  - a new type-theory of the math notion of algorithm introduced by Moschovakis [7] (2006), currently in development

A Formal Language of Dependent Type-Theory of Situated Information

Here, I shall present some of the new development of a dependent-type theory of situated information, by introducing a formal language L_{ra}^{st}

- based on Moschovakis [7] type-theory of algorithms
- fundamentally close to Per Martin-Löf dependent-types [6]
- introducing possibilities for integration of situated propositions with quantitative information, e.g., from
  - approaches to data by mathematical statistics and probability
  - Machine Learning

- L_{ra}^{st} presents information in situations, which can depend on: other situations, space-time locations, agents
- primitive and complex terms, representing:
  - objects with partially available information
  - recursive restrictions, for satisfactions of typed conditions
  - objects in nature that are undeveloped or in developmental stage

I shall keep the presentation at an informal level, by simple examples from human language.

Primitive (basic) types of L_{ra}^{st}: a set of type constants

$$\text{BTypes} = \{ \text{IND}, \text{REL}, \text{FUN}, \text{ARGR}, \text{LOC}, \text{POL}, \text{EVAL}, \text{PAR}, \text{INFON}, \text{SIT}, \text{PROP}, \text{SET}, \text{TYPE}, \models \} \quad (1)$$

For example:

- IND: for primitive and complex individuals (entities)
- REL: for primitive and complex relations, without currying coding
- ARGR: for primitive and complex argument roles
- LOC: for space-time locations
- POL: for numerical polarities, e.g., between 0 and 1
  (these are for degree of having a property or being in a relation, not for truth values, even when limited to 0 and 1)
- EVAL: for value of numerical assessments of verification
- PAR: for primitive and complex parameters
- INFON: for basic or complex information units
- SIT: for situations
- PROP: for propositions, terms that may have truth values
- $\models$ is a designated type called "supports" / "holds"

- $\models$ is a constant for a primitive type called "supports" ("holds"), e.g., used in propositions that a situation s and an infon σ are of the type "supports", i.e., "s supports σ":

$$(s \models \sigma) \qquad \text{(a proposition)}$$
$$s \models \sigma \qquad \text{(a verified proposition)}$$

  The type $\models$ reminds for the semantic relation between models and predicate formulae of classic math logic.

- A class of primitive and complex types
  - Complex types are constructed at stages, e.g., as needed (not necessarily all of them)

$$\text{Types}_0, \text{Types}_1, \ldots, \text{Types}_n, \ldots \tag{3a}$$
$$\text{for } \text{Types}_i \subseteq \text{Types}_{i+1}, \text{ for } i \geq 0 \tag{3b}$$

Vocabulary and Syntax of L^{st}_{ra}

For all $\tau \in$ Types:

- Typed constants

$$K_\tau = \mathsf{Consts}_\tau = \{\, \mathsf{c}_0^\tau, \mathsf{c}_1^\tau, \ldots, \mathsf{c}_{k_\tau}^\tau, \ldots \} \tag{4}$$

- Typed pure and recursion (memory) variables
  - pure variables (for λ-abstractions)

$$\mathsf{PureV}_\tau = \mathsf{PureV}_\tau = \{\, \mathsf{v}_0^\tau, \mathsf{v}_1^\tau, \ldots \}$$

  - recursion variables (for memory "slots")

$$\mathsf{RecV}_\tau = \mathsf{RecV}_\tau = \{\, \mathsf{p}_0^\tau, \mathsf{p}_1^\tau, \ldots \}$$

- Notations for types of constants, variables, etc., terms

$$A : \tau \iff A^\tau \in \mathsf{Terms} \iff A \in \mathsf{Terms}_\tau \tag{5}$$

- Complex terms of situated information are defined by structural induction — mutual recursion

Relations, Functions, and Types have Restricted Argument Roles for Appropriateness

- Each γ that is (a term for) a relation, function, or type, has a set $\mathrm{ARGR}(\gamma)$ of argument roles
- The argument roles are restricted by types T for appropriateness
- For constants and variables — the typed argument roles are provided by the vocabulary
- For complex terms — by the recursive definitions

$$\begin{aligned}
&\mathrm{ARGR}(\gamma) = \{ \mathsf{arg}_1^{T_1}, \dots, \mathsf{arg}_n^{T_n} \} \\
&\text{for each } \gamma \in \mathsf{Terms}_{\mathrm{REL}} \cup \mathsf{Terms}_{\mathrm{FUN}} \cup \mathsf{Terms}_{\mathrm{TYPE}} \\
&\mathsf{arg}_i : \mathrm{ARGR}\text{: the argument roles of } \gamma, \\
&T_i : \text{Types: the type for appropriateness constraints of } \mathsf{arg}_i, \\
&i = 1, \dots, n
\end{aligned} \tag{6}$$

Relations, Functions, and Types with Argument Roles

- Every function constant and term γ, i.e., $\gamma \in \text{Terms}_{\text{FUN}}$, is associated with two sets of typed expressions for argument roles:

$$\text{ARGR}(\gamma) = \{ \text{arg}_1^{T_1}, \ldots, \text{arg}_n^{T_n} \} \tag{7a}$$

$$ValueArg(\gamma) = \{ \text{arg}_{n+1}^{T_{n+1}} \} \tag{7b}$$

- The graph term of $\gamma \in \text{Terms}_{\text{FUN}}$ is a term $G(\gamma) \in \text{Terms}_{\text{REL}}$, such that:

$$\text{ARGR}(G(\gamma)) = \{ \text{arg}_1^{T_1}, \ldots, \text{arg}_n^{T_n}, \text{arg}_{n+1}^{T_{n+1}} \} \tag{8a}$$

$$ValueArg(\gamma) = \{ \text{arg}_{n+1}^{T_{n+1}} \} \tag{8b}$$

Terms for *entities, infons, relations, propositions, and types: defined by recursion*

Typed terms are defined by recursion: here we exemplify some of them.
Infon Terms: The class of expressions of the form:

$$\ll \rho, \mathsf{arg}_1^{T_1} \mapsto \xi_1, \ldots, \\ \mathsf{arg}_n^{T_n} \mapsto \xi_n, \\ loc^{\mathrm{LOC}} \mapsto \tau,\ pol^{\mathrm{POL}} \mapsto t \gg : \mathrm{INFON}$$

for:

- $\rho \in \mathsf{Terms}_{\mathrm{REL}}$:

$$\mathrm{ARGR}(\rho) = \{\, \mathsf{arg}_1^{T_1}, \ldots, \mathsf{arg}_n^{T_n},\ loc^{\mathrm{LOC}},\ pol^{\mathrm{POL}} \,\} \tag{10}$$

- $\xi_1 \in \mathsf{Terms}_{T_1}, \ldots, \xi_n \in \mathsf{Terms}_{T_n}$
- $\tau \in \mathsf{Terms}_{\mathrm{LOC}}$
- $t \in \mathsf{Terms}_{\mathrm{POL}}$, where t is
  either a parametric term (formula), e.g., $t \in \mathsf{PureV}_{\mathrm{POL}} \cup \mathsf{RecV}_{\mathrm{POL}}$,
  or a term for a numerical value

Basic Infon: basic relation (constant) and names of its argument roles

$$\text{ARGR}(\textit{read-to}) = \{\,\mathsf{reader}^{T_{a_1}},\ \mathsf{read\text{-}ed}^{T_o},\ \mathsf{listener}^{T_{a_1}},\ loc^{\mathrm{LOC}},\ pol^{\mathrm{POL}}\,\} \tag{11a}$$

$$\ll \textit{read-to},\ \mathsf{reader}^{T_{a_1}} \mapsto c_a,\ \mathsf{read\text{-}ed}^{T_o} \mapsto c_b,\ \mathsf{listener}^{T_{a_1}} \mapsto c_c,\ loc^{\mathrm{LOC}} \mapsto l;\ pol^{\mathrm{POL}} \mapsto 0.60 \gg \tag{11b}$$

In (11a)–(11b), $\textit{read-to} \in \mathsf{Consts}_{\mathrm{REL}}$ is a constant denoting a 5-argument relation of reading, having three semantic argument roles for "participants"

- reader is a constant naming the argument role of *read-to* for the agent that does reading
- read-ed — for the object that is being read (this is not a verbal form)
- listener — for the participant that listens the reading

In predicate logic, the argument roles are conventionally ordered, e.g.:

$$\textit{read-to}(c_a, c_b, c_c) \tag{12}$$

General Practices for Names of Argument Roles of Relations

There are at least two approaches to naming semantic argument roles:

- Shared names of semantic arguments roles, e.g., in a version of L_{ra}^{st}:

$$\mathcal{BA}^{\tau}_{\mathrm{ARGR}} = \{\, arg^{\tau}_{1}, \ldots, arg^{\tau}_{n}, \ldots \},\ \tau \in \text{Types (by generation)} \quad (13)$$

- Individual names of semantic arguments roles
  Jon Barwise introduced naming via suffixes. In L_{ra}^{st}, e.g.:

$$append(relation\text{-}name, er) \in \text{Terms}_{\mathrm{ARGR}} \quad (14a)$$
$$append(relation\text{-}name, ed) \in \text{Terms}_{\mathrm{ARGR}} \quad (14b)$$
$$append(relation\text{-}name, ed) \equiv append(relation\text{-}name, \text{-}ed) \quad (14c)$$
$$\text{readed} \equiv \text{read-ed} \in (\text{Terms}_{\mathrm{ARGR}} - \text{Consts}_{\mathrm{REL}}) \quad (14d)$$

  Argument roles generated in this way, may look as if "misspelled" word forms, while, e.g.: readed $\notin$ $\text{Consts}_{\mathrm{REL}}$ is not a verb form. This can be avoided by adding dashes, (14c)–(14d).

- More complex roles are generated inductively, by the recursive definition of the terms

$$ArgR(read\text{-}to) = \{\mathsf{reader}^{T_{a_1}},\ \mathsf{read\text{-}ed}^{T_o},\ \mathsf{listener}^{T_{a_1}}\} \tag{15}$$

$$T_{a_1} \equiv \{\lambda(x)\big[\big(s_1 \models\ \ll human, \tag{16a}$$
$$\mathsf{arg}^{\mathrm{IND}} \mapsto x^{\mathrm{IND}}, \tag{16b}$$
$$loc^{\mathrm{LOC}} \mapsto l_d,\ pol^{\mathrm{POL}} \mapsto 1 \gg, \tag{16c}$$
$$eval^{\mathrm{EVAL}} \mapsto 40\%\big) \tag{16d}$$
$$\vee\ \big(s_1 \models\ \ll device, \tag{16e}$$
$$\mathsf{arg}^{\mathrm{IND}} \mapsto x^{\mathrm{IND}}, \tag{16f}$$
$$loc^{\mathrm{LOC}} \mapsto l_o,\ pol^{\mathrm{POL}} \mapsto 1 \gg, \tag{16g}$$
$$eval^{\mathrm{EVAL}} \mapsto 60\%\big)\big]\} \tag{16h}$$

$$T_o \equiv \{\lambda(x)\big(s_o \models\ \ll written, \tag{17a}$$
$$\mathsf{arg}^{\mathrm{IND}} \mapsto x^{\mathrm{IND}}, \tag{17b}$$
$$loc^{\mathrm{LOC}} \mapsto l_o,\ pol^{\mathrm{POL}} \mapsto 1 \gg, \tag{17c}$$
$$eval^{\mathrm{EVAL}} \mapsto 70\%\big)\} \tag{17d}$$

Given that $\gamma \in \text{Terms}_{\text{REL}}$, $\text{ARGR}(\gamma) = \{ \text{arg}_1^{T_1}, \ldots, \text{arg}_n^{T_n} \}$,
$\xi_i \in \text{Terms}_{T_i}$ $(i = 1, \ldots, n)$, infon terms are expressions of the form:

$$\ll \gamma, arg_1^{T_1} \mapsto \xi_1, \ldots, arg_n^{T_n} \mapsto \xi_n, loc^{\text{LOC}} \mapsto \tau; pol^{\text{POL}} \mapsto i \gg \quad \text{(18a)}$$

$$\ll \gamma, \xi_1, \ldots, \xi_n \gg \quad \text{(18b)}$$

Example (infons: specific or parametric)

- c_a reads c_b to c_c at the space-time location l

$$\ll read\text{-}to,\ \text{reader}^{T_{a_1}} \mapsto c_a,\ \text{read-ed}^{T_o} \mapsto c_b,\ \text{listener}^{T_{a_1}} \mapsto c_c,\ loc^{\text{LOC}} \mapsto l;\ pol^{\text{POL}} \mapsto 0.60 \gg \quad \text{(19)}$$

- c_a reads c_b to the unknown z at the unknown location $\dot{l}$

$$\ll read\text{-}to,\ \text{reader}^{T_{a_1}} \mapsto c_a,\ \text{read-ed}^{T_o} \mapsto c_b, \quad \text{(specific)} \quad \text{(20a)}$$

$$\text{listener}^{T_{a_1}} \mapsto z, \quad \text{(parametric)} \quad \text{(20b)}$$

$$loc^{\text{LOC}} \mapsto \dot{l};\ pol^{\text{POL}} \mapsto p \gg \quad \text{(20c)}$$

Example (Underspecified Complex Infons)

- $\mathsf{b}, z \in \mathsf{RecV}_{\text{IND}}$ are recursion (memory) variables
- $l \in \mathsf{RecV}_{\text{LOC}}$ is a recursion (memory) variable for space-time location
- $x \in \mathsf{PureV}_{\text{IND}}$ is a pure variable for an individual

Note: I in (21a)–(21b) is a term for a complex infon, not for a proposition!

$$I \equiv \ll book, \mathsf{arg} \mapsto \mathsf{b}, loc \mapsto l; pol \mapsto 1 \gg \land \tag{21a}$$

$$\ll read\text{-}to, \mathsf{reader}^{T_{a_1}} \mapsto x,\ \mathsf{read\text{-}ed}^{T_o} \mapsto \mathsf{b},\ \mathsf{listener}^{T_{a_1}} \mapsto z, \tag{21b}$$

$$loc^{\text{LOC}} \mapsto l; pol^{\text{POL}} \mapsto 1 \gg$$

R is a λ-term denoting a composite relation between objects x, z: conjuncts are terms for infons, not for propositions:

$$R \equiv \lambda(x, z)\Big[\ll book, \mathsf{arg} \mapsto \mathsf{b}, loc \mapsto l; pol \mapsto 1 \gg \land \tag{22a}$$

$$\ll read\text{-}to, \mathsf{reader}^{T_{a_1}} \mapsto x,\ \mathsf{read\text{-}ed}^{T_o} \mapsto \mathsf{b},\ \mathsf{listener}^{T_{a_1}} \mapsto z, \tag{22b}$$

$$loc^{\text{LOC}} \mapsto l; pol^{\text{POL}} \mapsto 1 \gg \Big]$$

Propositions and Situated Propositions

- For every type term (basic or complex) $\gamma \in \mathrm{Terms}_{\mathrm{TYPE}}$,
- associated with argument roles $(n \geq 0)$

$$ArgRof(\gamma) \equiv \{ T_1 : \mathrm{arg}_1, \ldots, T_n : \mathrm{arg}_n, \ \mathrm{EVAL} : \mathrm{arg}_{n+1} \} \tag{23}$$

- and for every sequence of terms: $\xi_1 \in \mathrm{Terms}_{T_1}, \ldots, \xi_n \in \mathrm{Terms}_{T_n}, t \in \mathrm{Terms}_{\mathrm{EVAL}} = \mathrm{Terms}_{\mathbb{R}}$

the following expressions are proposition terms:

$$(\gamma, T_1 : \mathrm{arg}_1 : \xi_1, \ldots, T_n : \mathrm{arg}_n : \xi_n) : \mathrm{PROP} \quad \text{(truth value 1)} \tag{24a}$$

$$(\gamma, T_1 : \mathrm{arg}_1 : \xi_1, \ldots, T_n : \mathrm{arg}_n : \xi_n, \ \mathrm{EVAL} : certainty : t) : \mathrm{PROP} \tag{24b}$$

Special case, for $s \in \mathrm{Terms}_{\mathrm{SIT}}$, $\sigma \in \mathrm{Terms}_{\mathrm{INFON}}$

$$(s \models \sigma) : \mathrm{PROP} \tag{25a}$$

$$(s \models \sigma, \mathrm{EVAL} : certainty : t) : \mathrm{PROP} \tag{25b}$$

λ-Abstraction Terms

Case 1: complex relations with complex argument roles

For every $\varphi : \mathrm{INFON}$ and $\xi_1, \ldots, \xi_n \in$ PureV,

$$\lambda\{\, \xi_1, \ldots, \xi_n \,\}\,(\varphi) : \mathrm{REL} \tag{26}$$

Case 2: complex types with complex argument roles

For every $\varphi : \mathrm{PROP}$ and $\xi_1, \ldots, \xi_n \in$ PureV,

$$\lambda\{\, \xi_1, \ldots, \xi_n \,\}\,(\varphi) : \mathrm{TYPE} \tag{27}$$

Case 3: complex function terms For $\varphi \in \mathrm{Terms}_\tau$ where $\tau \in$ Types, $\tau \not\equiv \mathrm{INFON}$, $\tau \not\equiv \mathrm{PROP}$, and for any $\xi_1, \ldots, \xi_n \in$ PureV,

$$\lambda\{\, \xi_1, \ldots, \xi_n \,\}\,(\varphi) : \mathrm{FUN} \tag{28}$$

The term $\lambda\{\, \xi_1, \ldots, \xi_n \,\}\,(\varphi)$ has an extra value role Val of type τ:

$$\mathit{Valof}\,(\lambda\{\, \xi_1, \ldots, \xi_n \,\}\,(\varphi)) = \{\, \tau : \mathsf{Val} \,\} \tag{29}$$

λ-Abstraction Terms

Complex Argument Roles and Appropriateness Constraints

$$ArgRof(\lambda\{\xi_1,\dots,\xi_n\}(\varphi)) = \{T_1 : [\xi_1],\dots,T_n : [\xi_n]\} \tag{30a}$$

$$ArgRof(\lambda\{\xi_1,\dots,\xi_n\}(\varphi)) = \{T_1 : [\xi_1],\dots,T_n : [\xi_n], \text{EVAL} : \mathsf{Val}\} \tag{30b}$$

$$ArgRof(\lambda\{\xi_1,\dots,\xi_n\}(\varphi)) = \{T_1 : [\xi_1],\dots,T_n : [\xi_n]\}$$
$$Valof(\lambda\{\xi_1,\dots,\xi_n\}(\varphi)) = \{\tau : \mathsf{Val}\} \quad \text{for Case 3: Terms}_{\text{FUN}} \tag{30c}$$

where, for $i \in \{1,\dots,n\}$, T_i is the set of all the types in the appropriateness constraints of all the argument roles filled by ξ_i, in all the occurrences of ξ_i in φ

Ongoing and Future Work

- Theoretical development of Dependent Type-Theory of Situated Information
  Immediate tasks:
  Reduction Calculi and canonical forms of the terms
- Choice and development of approach for linking the quantitive assessments and integration with situated information:
  Deep Machine Learning
- Reasoning based on semantic representations of formal and human languages
- Syntax-semantics interface in computational grammar of human language

Some References I

Barwise, J.: Scenes and other situations.
The Journal of Philosophy **78**, 369–397 (1981).
URL https://doi.org/10.2307/2026481

Barwise, J., Perry, J.: Situations and Attitudes.
Cambridge, MA:MIT press (1983).
Republished as [3]

Barwise, J., Perry, J.: Situations and Attitudes.
The Hume Series. CSLI Publications, Stanford, California (1999)

Loukanova, R.: Situation Theory, Situated Information, and Situated Agents.
In: N. et al. (ed.) Transactions on Computational Collective Intelligence XVII, *Lecture Notes in Computer Science*, vol. 8790, pp. 145–170. Springer, Berlin, Heidelberg (2014).
URL https://doi.org/10.1007/978-3-662-44994-3_8

Some References II

- Loukanova, R.: Formalisation of situated dependent-type theory with underspecified assessments.
  In: E. Bucciarelli, et al. (eds.) Decision Economics. Designs, Models, and Techniques for Boundedly Rational Decisions. DCAI 2018, *Advances in Intelligent Systems and Computing*, vol. 805, pp. 49–56. Springer International Publishing, Cham (2019).
  URL https://doi.org/10.1007/978-3-319-99698-1_6

- Martin-Löf, P.: Intuitionistic Type Theory.
  Bibliopolis, Napoli (1984)

- Moschovakis, Y.N.: A Logical Calculus of Meaning and Synonymy.
  Linguistics and Philosophy **29**(1), 27–89 (2006).
  URL https://doi.org/10.1007/s10988-005-6920-7

- Seligman, J., Moss, L.S.: Situation theory.
  In: J. van Benthem, A. ter Meulen (eds.) Handbook of Logic and Language, pp. 253–329. Elsevier, Amsterdam (2011)