Algorithmic Dependent-Type Theory of Situated Information and Context Assessments

Roussanka Loukanova

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

19th International Conference on Distributed Computing and Artificial Intelligence | L'Aquila (Italy), 13th–15th July, 2022

Thursday, July 14th, 11:00–11:20 am

Situation Theory (SitT) and Situation Semantics (SitSem): an application of SitT

- Barwise [1, 2] (1981–1983) is the influential, early work on the strategy of
  - Situation Theory (SitT):
    as a mathematical model of situated, partial information
  - Situation Semantics (SitSem): application of SitT

Mathematical Theories and Applications

- Loukanova, 1990–2001
  Math Model Theory of Situated Information by Dependent-Types
  & application to
  Computational Semantics of Human Language
- Loukanova [4, 5] since 2014
  - a new, dependent-type theory of Situated Information
  - a new math of algorithms
    extending Moschovakis Recursion [6] (2006)

A Formal Language L_{ra}^{st} of Dependent Type-Theory of Situated Information

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

- For integration of situated propositions and quantitative information
- Quantitative (numerical) information can be contributed by using
  - Approaches to data by mathematical statistics and probability
  - Machine Learning

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"

- the symbol $\models$ is a constant for a primitive type

$(s \models \sigma)$
a proposition that the infon σ holds in the situation s (2a)

$s \models \sigma$
a verified proposition, e.g., by data in a computer system (2b)

The type $\models$ reminds for the semantic relation between models s and predicate formulae σ 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)

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

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

For all $\tau \in$ Types:

- Typed constants

$$K_\tau = \mathsf{Consts}_\tau = \{\, \mathsf{c}^\tau_0, \mathsf{c}^\tau_1, \ldots, \mathsf{c}^\tau_{k_\tau}, \ldots \,\} \tag{4}$$

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

$$\mathsf{PureV}_\tau = \mathsf{PureV}_\tau = \{\, \mathsf{v}^\tau_0, \mathsf{v}^\tau_1, \ldots \,\}$$

  - recursion variables (for memory "slots")

$$\mathsf{RecV}_\tau = \mathsf{RecV}_\tau = \{\, \mathsf{p}^\tau_0, \mathsf{p}^\tau_1, \ldots \,\}$$

- 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 — i.e., by 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 $\textsc{argr}(\gamma)$ of argument roles
- The argument roles are restricted by types T for appropriateness

$$\begin{aligned}
&\textsc{argr}(\gamma) = \{\, \mathsf{arg}_1^{T_1}, \dots, \mathsf{arg}_n^{T_n} \,\} \\
&\text{for each } \gamma \in \mathsf{Terms}_{\textsc{rel}} \cup \mathsf{Terms}_{\textsc{fun}} \cup \mathsf{Terms}_{\textsc{type}} \\
&\mathsf{arg}_i : \textsc{argr}\ (i = 1, \dots, n) \text{ are the argument roles of } \gamma \\
&T_i : \textsc{type} \text{ the type for appropriateness constraints of } \mathsf{arg}_i \\
&i = 1, \dots, n
\end{aligned} \tag{6}$$

- For constants and variables — the typed argument roles are provided by the vocabulary
- For complex terms — by the recursive definitions

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:

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

for:

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

$$\mathrm{ARGR}(\rho) = \{ \mathsf{arg}_1^{T_1}, \ldots, \mathsf{arg}_n^{T_n},\ loc^{\mathrm{LOC}},\ pol^{\mathrm{POL}} \} \tag{8}$$

- $\xi_1 \in \mathsf{Terms}_{T_1}, \ldots, \xi_n \in \mathsf{Terms}_{T_n}$
- $\tau \in \mathsf{Terms}_{\mathrm{LOC}}$
- $t \in \mathsf{Terms}_{\mathrm{POL}}$, 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

$$\textsc{argr}(\mathit{read\text{-}to}) = \{\, \mathsf{reader}^{T_{a_1}},\ \mathsf{read\text{-}ed}^{T_o},\ \mathsf{listener}^{T_{a_1}},\ loc^{\mathrm{LOC}},\ pol^{\mathrm{POL}} \,\} \tag{9a}$$

$$\ll \mathit{read\text{-}to},\ \mathsf{reader}^{T_{a_1}} \mapsto c_a,\ \mathsf{read\text{-}ed}^{T_o} \mapsto c_b,\ \mathsf{listener}^{T_{a_1}} \mapsto c_c,\ loc^{\mathrm{LOC}} \mapsto l;\ pol^{\mathrm{POL}} \mapsto 0.60 \gg \tag{9b}$$

In (9a)–(9b), $\mathit{read\text{-}to} \in \mathsf{Consts}_{\mathrm{REL}}$ 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 $\mathit{read\text{-}to}$ 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.:

$$\mathit{read\text{-}to}(c_a, c_b, c_c) \tag{10}$$

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}:

$$\mathcal{BA}^{\tau}_{\text{ARGR}} = \{\, arg^{\tau}_{1}, \dots, arg^{\tau}_{n}, \dots \},\ \ \tau \in \text{Types}\ \ \text{(by generation)} \quad (11)$$

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

$$append(relation\text{-}name, er) \in \text{Terms}_{\text{ARGR}} \quad (12a)$$
$$append(relation\text{-}name, ed) \in \text{Terms}_{\text{ARGR}} \quad (12b)$$
$$append(relation\text{-}name, ed) \equiv append(relation\text{-}name, \text{-}ed) \quad (12c)$$
$$\text{readed} \equiv \text{read-ed} \in (\text{Terms}_{\text{ARGR}} - \text{Consts}_{\text{REL}}) \quad (12d)$$

  Argument roles generated in this way, may look as if “misspelled” word forms, while, e.g.: readed $\notin$ $\text{Consts}_{\text{REL}}$ is not a verb form. This can be avoided by adding dashes, (12c)–(12d).

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

$$ArgR(read\text{-}to) = \{\mathsf{reader}^{T_{a_1}},\ \mathsf{read\text{-}ed}^{T_o},\ \mathsf{listener}^{T_{a_1}}\} \tag{13}$$

$$T_{a_1} \equiv \{\lambda(x)\big[\big(s_1 \models\ \ll human, \tag{14a}$$
$$\mathsf{arg}^{\mathrm{IND}} \mapsto x^{\mathrm{IND}}, \tag{14b}$$
$$loc^{\mathrm{LOC}} \mapsto l_d,\ pol^{\mathrm{POL}} \mapsto 1 \gg, \tag{14c}$$
$$eval^{\mathrm{EVAL}} \mapsto 40\%\big) \tag{14d}$$
$$\vee\ \big(s_1 \models\ \ll device, \tag{14e}$$
$$\mathsf{arg}^{\mathrm{IND}} \mapsto x^{\mathrm{IND}}, \tag{14f}$$
$$loc^{\mathrm{LOC}} \mapsto l_o,\ pol^{\mathrm{POL}} \mapsto 1 \gg, \tag{14g}$$
$$eval^{\mathrm{EVAL}} \mapsto 60\%\big)\big]\} \tag{14h}$$

$$T_o \equiv \{\lambda(x)\big(s_o \models\ \ll written, \tag{15a}$$
$$\mathsf{arg}^{\mathrm{IND}} \mapsto x^{\mathrm{IND}}, \tag{15b}$$
$$loc^{\mathrm{LOC}} \mapsto l_o,\ pol^{\mathrm{POL}} \mapsto 1 \gg, \tag{15c}$$
$$eval^{\mathrm{EVAL}} \mapsto 70\%\big)\} \tag{15d}$$

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:

$$\ll \gamma, arg_1^{T_1} \mapsto \xi_1, \ldots, arg_n^{T_n} \mapsto \xi_n, loc^{\text{LOC}} \mapsto \tau; pol^{\text{POL}} \mapsto i \gg \qquad (16a)$$

$$\ll \gamma, \xi_1, \ldots, \xi_n \gg \qquad (16b)$$

Example (infons: specific or parametric)

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

$$\ll read\text{-}to,\ \text{reader}^{T_{a_1}} \mapsto c_a,\ \text{read-ed}^{T_o} \mapsto c_b,\ \text{listener}^{T_{a_1}} \mapsto c_c,\ loc^{\text{LOC}} \mapsto l;\ pol^{\text{POL}} \mapsto 0.60 \gg \qquad (17)$$

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

$$\ll read\text{-}to,\ \text{reader}^{T_{a_1}} \mapsto c_a,\ \text{read-ed}^{T_o} \mapsto c_b, \qquad \text{(specific)} \qquad (18a)$$

$$\text{listener}^{T_{a_1}} \mapsto z, \qquad \text{(parametric)} \qquad (18b)$$

$$loc^{\text{LOC}} \mapsto \dot{l};\ pol^{\text{POL}} \mapsto p \gg \qquad (18c)$$

Example (Underspecified Complex Infons)

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

Note: I in (19a)–(19b) is a term for a complex infon, not for a proposition!

$$I \equiv \ll book, \mathsf{arg} \mapsto \mathsf{b}, loc \mapsto l; pol \mapsto 1 \gg \land \tag{19a}$$

$$\ll read\text{-}to, \mathsf{reader}^{T_{a_1}} \mapsto x,\ \mathsf{read\text{-}ed}^{T_o} \mapsto \mathsf{b},\ \mathsf{listener}^{T_{a_1}} \mapsto z, \tag{19b}$$

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

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

$$R \equiv \lambda(x, z)\Big[\ll book, \mathsf{arg} \mapsto \mathsf{b}, loc \mapsto l; pol \mapsto 1 \gg \land \tag{20a}$$

$$\ll read\text{-}to, \mathsf{reader}^{T_{a_1}} \mapsto x,\ \mathsf{read\text{-}ed}^{T_o} \mapsto \mathsf{b},\ \mathsf{listener}^{T_{a_1}} \mapsto z, \tag{20b}$$

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

Propositions and Situated Propositions

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

$$ArgRof(\gamma) \equiv \{\, T_1 : \mathsf{arg}_1, \ldots, T_n : \mathsf{arg}_n,\ \text{EVAL} : \mathsf{arg}_{n+1} \,\} \tag{21}$$

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

the following expressions are proposition terms:

$$(\gamma, T_1 : \mathsf{arg}_1 : \xi_1, \ldots, T_n : \mathsf{arg}_n : \xi_n) : \text{PROP} \quad \text{(truth value 1)} \tag{22a}$$

$$(\gamma, T_1 : \mathsf{arg}_1 : \xi_1, \ldots, T_n : \mathsf{arg}_n : \xi_n, \text{EVAL} : \mathit{certainty} : t) : \text{PROP} \tag{22b}$$

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

$$(s \models \sigma) : \text{PROP} \tag{23a}$$

$$(s \models \sigma, \text{EVAL} : \mathit{certainty} : t) : \text{PROP} \tag{23b}$$

λ-Abstraction Terms

Case 1: complex relations with complex argument roles

For every φ : INFON and $\xi_1, \ldots, \xi_n \in$ PureV,

$$\lambda\{\xi_1, \ldots, \xi_n\}(\varphi) : \text{REL} \tag{24}$$

Case 2: complex types with complex argument roles

For every φ : PROP and $\xi_1, \ldots, \xi_n \in$ PureV,

$$\lambda\{\xi_1, \ldots, \xi_n\}(\varphi) : \text{TYPE} \tag{25}$$

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

$$\lambda\{\xi_1, \ldots, \xi_n\}(\varphi) : \text{FUN} \tag{26}$$

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

$$\mathit{Valof}(\lambda\{\xi_1, \ldots, \xi_n\}(\varphi)) = \{\tau : \mathsf{Val}\} \tag{27}$$

λ-Abstraction Terms

Complex Argument Roles and Appropriateness Constraints

$$ArgRof(\lambda\{\xi_1,\ldots,\xi_n\}(\varphi)) = \{T_1 : [\xi_1],\ldots,T_n : [\xi_n]\} \tag{28a}$$

$$ArgRof(\lambda\{\xi_1,\ldots,\xi_n\}(\varphi)) = \{T_1 : [\xi_1],\ldots,T_n : [\xi_n], \text{EVAL} : \mathsf{Val}\} \tag{28b}$$

$$\begin{aligned} ArgRof(\lambda\{\xi_1,\ldots,\xi_n\}(\varphi)) &= \{T_1 : [\xi_1],\ldots,T_n : [\xi_n]\} \\ Valof(\lambda\{\xi_1,\ldots,\xi_n\}(\varphi)) &= \{\tau : \mathsf{Val}\} \quad \text{for Case 3: Terms}_{\text{FUN}} \end{aligned} \tag{28c}$$

where, for $i \in \{1,\ldots,n\}$, 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
- Syntax-semantics interface in computational grammar of programming languages

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

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)