Modeling Dialogue in Dynamic Framework

Master's Thesis

defended on June, 21st, 2016

in order to get

Master of Science in
Cognitive Science & Applications (SCA)
specialized in
Natural Language Processing (NLP)

Presented by

Stéphane Tiv

Supervisor : Maxime Amblard
Co-supervisor : Jirka Maršík

Context of the work

This thesis details the work done during an internship in the LORIA in order to get the Master of Cognitive Sciences & Applications specialized in Natural Language Processing. The LORIA is a research laboratory focusing on computer science. The internship was done in one of the LORIA teams of the department 4, "*Knowledge and Language Management*", SEMAGRAMME, specialized on natural language processing and semantic analysis of discourse and utterances. This internship was supervised by SEMAGRAMME members Maxime Amblard and co-supervised by Jirka Maršík.

This work is done related to the SLAM project, "*Schizophrenic and Language : Analysis and Modeling*", that involves several disciplines in the project, including philosophy, psychology, linguistics and computer science in order to assist with the diagnosis of schizophrenia.

This work focuses on the formalization of dialogue at the semantic level. This work relies on the assumption of the rationality of schizophrenic people. The goal of the project is to help with diagnosis, focusing on cognitive dysfunction, in particular at the speech level. The basis of rationality brought some observed issues, like different ways to handle ambiguity. This work relates to this project by trying to bring a formalism that can represent and take into account these observations.

Abstract

This thesis presents an insight into dialogue modeling within a dynamic framework. This work aims to model propositions, questions and answers for setting the common ground between dialogue participants. Adapting type-theoretical continuation semantics with frame semantics, a dialogue structure is modeled with types and combinators for utterances and the content represented by the context in continuation semantics with frames. Questions and answers are modeled with the notion of focus, symbolizing a request on a previous proposition about a specific argument. As a result, a dialogue is seen as a sequence of typed utterances, and its meaning as frames.

Keywords : Dialogue, Type Theory, Dynamic Semantics, Frame, Continuation Semantics.

Contents

1 Context

1.1 Introduction

In the field of natural language processing, handling language involves a lot of disciplines such as linguistics, logic, philosophy and computer science. One of the most well-known way to handle this was developed by Richard Montague [Mon73] in the 1970s with a logical point of view and a close relation between semantics, logic and language. His work entails well-defined syntactic structures following compositionality and meaning given by lambda-expressions. For Montague, meaning is expressed by the principle that the meaning of the sentence is determined by its components and the way they are composed. His work brought a novel starting point to handle semantics for natural language processing, but lacks ways to handle specific issues in the discourse. As an example, his works can hardly express pronoun binding and anaphora, exemplified in the so-called donkey sentences.

As a result, recent researchers have developed theories based on his works to handle these topics to come up with a framework, namely dynamic semantics. In this framework, the earliest work was given by Hans Kamp with Discourse Representation Theory [Kam81] and Irene Heim with File Change Semantics [Hei82], developed independently with a similar idea. These works improve the process of discourse with such issues, by considering the meaning in a novel way. The meaning in discourse is given by a relation between contexts, what precedes an assertion and the assertion itself uttered in distinctive conditions. These works lead to the resolution of the issues in Montague's work to better handle discourse.

Dynamic semantics improved the way to process complex discourses but lead to other questions, the handling of dialogue and the concerns it raises. Dialogue entails interaction between participants, a peculiar structure and specialized ways to deal with. The meaning of a dialogue can vary from one context to another.

In this paper we investigate the points raised by dialogue and try to model it in a semantic and logical way. Dialogue also involves characteristic representations of the content by the participants. In the first part, we develop about these topics, such as dialogue interaction, structure and meaning, and what needs to be represented. In addition, we present the basics of dynamic semantics and the theories we will rely on to approach dialogue. In part 2, we propose a view to take care of specific dialogues, with questions and answers. And in part 3, we present an extension of the main idea to deal with more complex issues, namely quantifiers, their representations and scope issues.

1.2 Motivations

In natural language processing, dialogue raised particular questions that cannot be handled with within the dynamic semantics framework. In this part, we will investigate what needs to be done in order to properly model how dialogue is processed and what needs to be represented. The way discourse in context is treated can be extended to turns of speech in a dialogue. In dialogue, the interaction between participants involves a proper definition of the nature of the context, as showed by Ginzburg [Ginar], with the notion of conversational relevance and conversational meaning. We consider these notions in terms of dialogue structure and dialogue representation. The interpretation of a dialogue depends on the interpretation of each turn of speech and the way they are composed.

1.2.1 Dialogue issues

Dialogue, compared to discourse, involves more than one participant, the interaction between them brings news problem for the modeling, with turns of speech and content interpretation. The representation of the content may be different for each participant, and each person can bring or use their own knowledge and beliefs. We consider that the dialogue sets a context where participants wants to share information. The questions to ask for dialogue modeling follow from the need to properly define a structure for the dialogue and a representation for the content [Mul14].

In a semantic point of view, dialogue is characterized by precise issues. To define what need to be model in such a framework, we will follow Ginzburg for the definitions and description in the semantics of dialogue. According to Ginzburg, dialogue must take into account the context in which a dialogue occurs. Indeed, the context of a dialogue may lead what can be added to the current dialogue, what is considered possible to utter or not, the related content and its relevance. He describes this semantic in terms of conversational relevance and conversational meaning.

The notion of conversational relevance refers to the ability to consider, in a current state of a dialogue, what can be uttered in the context, what can be added to maintain a adequate conversation. This notion is related to the notion of coherence of a dialogue, considering if each utterance makes sense in a particular context where the dialogue occurs. Coherence for dialogue follows, according to Ginzburg, gricean maxims [Gri70]. By using Grice's maxims, the rationality of participants is assumed in a conversation, and allows to define if a dialogue is coherent or not, and to make inferences. The relevance depends on the way the dialogue is structured, and according to Ginzburg, depends on the interactions it conveys. They can be seen as adjacency pairs, such as queries/answers, repair or turn taking. These notions show

how dialogue can be represented, with questions and answers, how one can ask for information misunderstood or misinterpreted, and depending on the way each participant uttered in the conversation.

Conversational meaning relates to the content of the dialogue, for each utterance can have different meanings depending of the context. An important question to tackle is the necessity to represent this meaning. In dialogue, several issues concerning the meaning are developed : interaction representation, question/answer, structuring of dialogue, meaning, and understanding. This notion of conversational meaning focuses on how we can represent the content of a dialogue to have the meaning of a whole dialogue and how we can model its parts and their meanings. The structure is quite linked to the meaning in the same way that the principle of compositionality for sentences.

1.2.2 Dialogue modeling

These considerations represent an important basis for modeling dialogue. Indeed, we consider a model by explicitly define our stance on the semantics of dialogue. A major part consists of representing the structure of a dialogue, or at least, issues that are directly entailed in dialogue. To this regard, we will focus on assertions as propositions, questions and answers, for they are very specific to the topic of dialogue [AL98]. Actually, questions and answers in dialogue involve different syntax and semantics [Kar77], and specific ways to interpret them [LA+09]. Our model will follow Groenendijk and Stokhof [GS85] for the view that the meaning of questions is given by (adequate) answers. We will see the question-answer part of a dialogue as a whole for giving a meaning, as argued by Muller and Prévot [MP01] and exemplified in Prévot *et al* [PMDV02].

In the representational aspect, dialogue can be viewed as an information sharing where each participant tries to bring information with another in accordance to their own knowledge and beliefs. Since they can be different, the assumption made is that each participant in a dialogue tries to build the same representation and interprets it with their own knowledge. With this point of view, the representation can be built in the same way as a discourse, considering each speech turn in context and combining them. In addition, representing the way utterances from different participants can be linked is another issue. The goal is to have a level of representation that can be altered to take into account the effects of interaction. In consideration, one of the most important part for dialogue modeling is the common ground, the commonly shared information in an ongoing dialogue. It involves during a conversation presupposition [VF08], what is supposedly considered true when uttering a sentence.

Consider the following sentence.

(1) It was Margaret who broke the keyboard.

The sentence presupposes that someone broke the keyboard. Which means that in model point of view, it is assumed that this sentence sets that the context is represented by at least all the worlds that include that the sentence someone broke the keyboard is true. The speaker who uttered the sentence assumes that the proposition is true, and thus is included in the common ground. The idea of common ground is described by Stalnaker [Sta02] in terms of acceptance of proposition (considering a proposition true or at least not to consider it false).

“It is common that ϕ in a group if all members *accept* (for the purpose of the conversation) that ϕ, and all *believe* that all accept that ϕ, and all *believe* that all believe that all accept that ϕ, etc"

In our model, this notion will serve as a basis for computing interaction as advocated by Zeevat [$Z^{+}97$] and representing the information in context in discourse [Sta98]. We will focus on the way participants set the common ground with questions and answers.

1.3 Frameworks

Dynamic semantics contributed to a novel starting point for analyzing discourse. The development of dynamic frameworks enabled the handling of several issues from “classical" Montague grammar [Mon73], like anaphora resolution or presupposition. A natural issue follows from this framework, dialogue modeling. Indeed, as discourse can be seen as a relation between contexts, dialogue involves more complex issues, introduced by the interaction of two or more participants. To represent dialogue processing, we need to understand how dynamic semantics resolves specific issues, how to structure dialogues and the importance of the shared information by the speakers, the common ground. Dynamic semantics have extended the work of Richard Montague that proposed a view for the semantic analysis of sentences. In a dynamic point of view, the meaning is understood as a relation between contexts, or a Context Change Potential. This idea allows the proper treatment of issues raised from Montague’s works, namely, anaphora resolution, donkey sentences, and more generally quantifier scope.

Consider the following examples with an extra-sentential anaphora and the famous donkey sentence containing intra-sentential anaphora :

(2) A man walks in the street. He whistles.

(3) Every farmer who owns a donkey beats it.

In classical Montague semantics with first order logic, the meanings of the sentences are represented by the following expressions :

(4) $\exists x.(man(x) \wedge walk(x) \wedge in\_the\_street(x)) \wedge whistle(x)$

(5) $\forall x.(farmer(x) \implies \exists y.(donkey(y) \wedge own(x, y)) \implies beat(x, y))$

In both cases, the formula fails to represent the proper meaning of the sentence. The variable x in $whistle(x)$ in the first sentence and the variable y in $beat(x, y)$ in the second one are free variables. The meanings of these sentences are given by these formulas, where the former variables are bound to the quantifiers.

(6) $\exists x.(man(x) \wedge walk(x) \wedge in\_the\_street(x) \wedge whistle(x))$

(7) $\forall x.(farmer(x) \implies \exists y.(donkey(y) \wedge own(x, y) \implies beat(x, y))$

Notice in particular, the range of the parenthesis that bind the variables to the quantifiers in these expressions.

1.3.1 Discourse Representation Theory

One of the earliest works in dynamic semantics is Discourse Representation Theory (DRT) by Hans Kamp in 1981 [Kam81]. We will briefly explain the model and how it resolves those problems.

The principle of DRT is to model a representation of the information contained in the discourse by the listener. It consists of Discourse Representation Structures (DRS) that contain a set of “discourse referents" and conditions. The referents represent the objects in the discourse and the conditions, the information about these objects.

Discourse Conditions

If R is a n-ary relation symbol, $x_1, x_2, \ldots, x_n$ are reference markers, then $R(x_1, x_2, \ldots, x_n)$ is a condition;

If K_1 and K_2 are DRSs, then $K_1 \implies K_2$, $K_1 \vee K_2$ are conditions.

If K is a DRS, then $\neg K$ is a condition.
If x_1, x_2 are reference markers, $x_1 = x_2$ is a condition.
Nothing else is a condition.

Subordination

Let K_1 and K_2 be DRSs, K_1 subordinates K_2 if and only if one of the following conditions holds :
— K_1 contains a DRS condition of the form “$\neg K_2$";
— K_1 contains a DRS condition of the form “$K_2 \implies K_3$", where K_3 is another DRS;
— “$K_1 \implies K_2$" is a DRS condition of some other DRS K_3;
— K_1 contains a DRS condition of the form “$K_2 \vee K_3$", where K_3 is another DRS;
— Some DRS K_3 subordinates K_2, and K_1 subordinates K_3.

Accessibility

Discourse referents of a DRS K_1 are accessible for another DRS K_2 only when:
— K_1 subordinates K_2,
— K_1 equals K_2.

Representation

DRT's box representation of "*A man walks in the street.*" and "*whistles*" (without any context).

x
man(x)
walk(x)
in_the_street(x)

y
whistle(y)
y=?

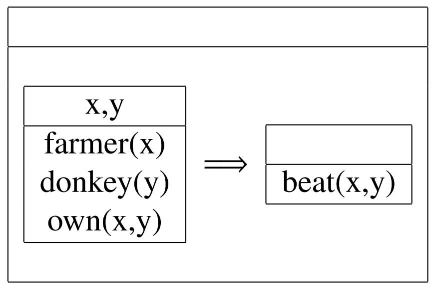
Consider again the example, but with the analysis brought by DRT.

(8) $[x, y : man(x), walk(x), in\_the\_street(x), whistle(y), y = x]$

x,y
man(x)
walk(x)
in_the_street(x)
whistle(y)
y = x

This box gives the representation of "*A man walks in the street. He whistles.*" The merging is done by filling the underspecified variable *y* to retrieve an accessible variable from the context, the first sentence. The only accessible variable is *x*, the merging resolves the anaphora and thus properly handles this discourse.

The analysis of the donkey sentence is done with DRT in the following way.



The equivalent meaning is given in first order logic in the following formula :

$$\forall x, y.((farmer(x) \land donkey(y) \land own(x, y)) \implies beat(x, y))$$

The variable *x* and *y* in $beat(x, y)$ are now bound to the universal quantifier, allowing the sentence to be properly computed.

1.3.2 Segmented Discourse Representation Theory

Segmented Discourse Representation Theory developed by Asher and Lascarides [AL03] is a semantic-driven theory of discourse, that is based on a dynamic theory of discourse, here DRT, and on rhetorical relations inspired from conversational analysis. The view they defend in their theory is that a discourse is logically structured with these so-called rhetorical relations that link utterances between them, and an internal structure for utterances and their meanings.

The basics taken from DRT allow to deal with the meaning in discourse, and the relations for its structure. Relations are given in terms of definite links, named after their utility, such as *Narration*, *Explanation*, *Elaboration*... The need for relations is explained to handle the way we use language, when one adds sentences in his discourse, it is used for a purpose, and according to the authors, this purpose is given in terms of rhetorical relations, to explain something, to narrate a story or to elaborate a sentence etc...

The principle of SDRT is that a discourse can be decomposed in smaller parts called segments that are related by one or several of those relations. The relations are used to take into account the fact that an utterance is linked to another and allow to retrieve an anaphora, for example, depending on the nature of the relation.

Consider the authors' famous example.

π_1 : John had a great evening last night.
π_2 : He had a great meal.
π_3 : He ate salmon.
π_4 : He devoured lots of cheese.
π_5 : He won a dancing competition.
π_6 : ??It was a beautiful pink.

Figure 1 – Discourse involving the right-frontier constraint

The last utterance is problematic, the "*It*" refers to the salmon, but this anaphora is infelicitous in natural language. The view given in SDRT can deal properly with it. The relation between utterances are given by the relations. There is first an *Elaboration* between π_1 and π_2, and an another *Elaboration* relation between π_2 and π_3, and between π_2 and π_4. Between π_2 and π_5, we have a *Narration* relation. Now the question is how can we relate the π_6 segment. The relations give a hierarchical structure, that allow or not links between utterances, this notion is called the Right-frontier constraint, by reference to the graphical structure the dialogue takes, and the fact that an utterance can only be linked to the outmost right frontier of the tree. According to SDRT, there is no such relation that allows a proper link with π_6 [1] since no segment is accessible respecting this constraint, allowing to express that the sentence π_6 is infelicitous.

1.3.3 Type-Theoretical Continuation Semantics

Later on, in a more traditional way following Montague's legacy, de Groote [dG06] proposed a compositional way to handle dynamics in discourse, using continuations, originating from computer science within theory of control, for representing contexts.

Based on computer science, dynamicity in this framework is treated with continuations. This proposal falls in the tradition of Montague, with the homomorphism

1. Intuitively we want to link this utterance to π_3 but the right-frontier constraint prevents it.

between syntax and semantics, and compositionality. This model uses the simply-typed lambda-calculus with the basic types from Church, as ι for individual and o for proposition. The idea consists in the introduction of a new type, γ, for representing contexts. The left context is what precedes the current sentence. The continuation consists of what is left to process, it returns a proposition given a left context. So the left context is of type γ, and the continuation of type $\gamma \rightarrow o$. The type for the usual proposition shifts for integrating the left context and the right context.

(9) $[\![s]\!] : \gamma \rightarrow (\gamma \rightarrow o) \rightarrow o$

Operators are defined to handle the content in context, and for resolving anaphora. The operator :: updates a current context with a new individual and the *sel* operator retrieves an individual from the context.

(10) $\_ :: \_ : \iota \rightarrow \gamma \rightarrow \gamma$

(11) $sel : \gamma \rightarrow \iota$

A binder is defined to interpret the combination of a discourse D and a sentence S.

(12) $[\![D.S]\!] = \lambda e\phi.[\![D]\!]e(\lambda e'.[\![S]\!]e'\phi)$

The content is interpreted with classical semantics, but with respect to contexts. The interpretation of a category is defined as λ-terms taking into account contexts. The sentence is combined using the binder with a previous discourse to obtain the meaning of the current discourse. We assume that pronoun binding is resolved with the *sel* operator that chooses a suitable variable extracted from the left context of the current utterance and introduced by the continuation of the previous one with the :: operator.

Consider an illustration in the following example.

(13) 1. John loves Mary.
2. He smiles at her.

For the computation of $[\![1]\!]$ and $[\![2]\!]$, we define as following the syntactic categories and the meaning.

Syntactic categories

(14) John, Mary, he, her : NP

(15) loves, smiles_at : $NP \rightarrow NP \rightarrow S$

Typing

(16) $$[\![s]\!] = \gamma \to (\gamma \to o) \to o$$

(17) $$[\![np]\!] = (\iota \to [\![s]\!]) \to [\![s]\!]$$

(18) $$[\![np \to np \to s]\!] = [\![np]\!] \to [\![np]\!] \to [\![s]\!]$$

Semantic representation

(19) $$[\![Mary]\!] = \lambda\psi e.\psi m(m :: e)$$

(20) $$[\![John]\!] = \lambda\psi e.\psi j(j :: e)$$

(21) $$[\![she]\!] = \lambda\psi e\phi.\psi(\mathtt{sel}_{she} e)e\phi$$

(22) $$[\![he]\!] = \lambda\psi e\phi.\psi(\mathtt{sel}_{he} e)e\phi$$

(23) $$[\![loves]\!] = \lambda ose\phi.s(\lambda xe.o(\lambda ye.love xy \wedge \phi e)e)e$$

(24) $$[\![smiles\_at]\!] = \lambda ose\phi.s(\lambda xe.o(\lambda ye.smile xy \wedge \phi e)e)e$$

Computation

(25) $$\begin{aligned}
&[\![loves]\!][\![Mary]\!][\![John]\!] = \\
&(\lambda ose\phi.s(\lambda xe.o(\lambda ye.love xy \wedge \phi e)e)e)[\![Mary]\!][\![John]\!] \\
&\to_\beta (\lambda se\phi.s(\lambda xe.[\![Mary]\!](\lambda ye.love xy \wedge \phi e)e)e)[\![John]\!] \\
&\to_\beta (\lambda e\phi.[\![John]\!](\lambda xe.[\![Mary]\!](\lambda ye.love xy \wedge \phi e)e)e) \\
&= (\lambda e\phi.(\lambda\psi e.\psi j(j :: e))(\lambda xe.[\![Mary]\!](\lambda ye.love xy \wedge \phi e)e)e) \\
&\to_\beta (\lambda e\phi.(\lambda e.(\lambda xe.[\![Mary]\!](\lambda ye.love xy \wedge \phi e)e) j(j :: e))e) \\
&\to_\beta (\lambda e\phi.(\lambda xe.[\![Mary]\!](\lambda ye.love xy \wedge \phi e)e) j(j :: e)) \\
&\to_\beta (\lambda e\phi.(\lambda e.[\![Mary]\!](\lambda ye.love jy \wedge \phi e)e)(j :: e)) \\
&\to_\beta (\lambda e\phi.[\![Mary]\!](\lambda ye.love jy \wedge \phi e)(j :: e)) \\
&= (\lambda e\phi.(\lambda\psi e.\psi m(m :: e))(\lambda ye.love jy \wedge \phi e)(j :: e)) \\
&\to_\beta (\lambda e\phi.(\lambda e.(\lambda ye.love jy \wedge \phi e)m(m :: e))(j :: e)) \\
&\to_\beta (\lambda e\phi.(\lambda ye.love jy \wedge \phi e)m(m :: j :: e)) \\
&\to_\beta (\lambda e\phi.(\lambda e.love jm \wedge \phi e)(m :: j :: e)) \\
&\to_\beta \lambda e\phi.love jm \wedge \phi(m :: j :: e)
\end{aligned}$$

In the same way, the meaning of $[\![2]\!]$ can be computed using the meaning of "*smiles_at*" in a similar way than "*loves*", and using the noun phrases representations of "*he*" and "*she*" instead of John and Mary. The computation follows a similar structure and gives the formula in $[\![2]\!]$. In the representation of $[\![1]\!]$, the continuation has two individuals, *John* and *Mary*, introduced with the :: operator, which will be available for future computations. In $[\![2]\!]$ the *sel* operator will choose from the context of the sentence.

(26) $[\![1]\!] = \lambda e\phi.love\, j\, m \wedge \phi(j :: m :: e)$

(27) $[\![2]\!] = \lambda e\phi.smile\ (sel_{he}e)\ (sel_{her}e) \wedge \phi e$

To combine the two sentences, the binder defined in (12) is used. It allows the *sel* operator of the second sentence to choose from the individuals introduced in the continuation of the first sentence.

(28) $[\![1.2]\!] = \lambda e\phi.[\![1]\!]e(\lambda e'.[\![2]\!]e'\phi)$

$= \lambda e\phi.[\lambda e\phi.love\, j\, m \wedge \phi(j :: m : e)]e\ (\lambda e'.[\lambda e\phi.smile\ (sel_{he}e)\ (sel_{her}e) \wedge \phi e]e'\phi)]$

$\rightarrow_\beta \lambda e\phi.[love\, j\, m \wedge (\lambda e'.\lambda e\phi.[smile\ (sel_{he}e)\ (sel_{her}e) \wedge \phi e]e'\phi)(j :: m : e)]$

$\rightarrow_\beta [\lambda e\phi.love\, j\, m \wedge (\lambda e\phi.[smile\ (sel_{he}e)\ (sel_{her}e) \wedge \phi e](j :: m : e)\phi)]$

$\rightarrow_\beta \lambda e\phi.[love\, j\, m \wedge [smile\ (sel_{he}(j :: m : e))\ (sel_{her}(j :: m : e))] \wedge \phi(j :: m : e)]$

With the right selection of the referents with *sel*, the final representation is obtained :

(29) $[\![1.2]\!] = \lambda e\phi.love\, j\, m \wedge smile\, j\, m \wedge \phi(j :: m : e)$

Seeing the context as a list of individuals allows the process of anaphora resolution and the handling of donkey sentences. In his work, de Groote [dG06] argues that the context may have more complex contents, as exemplified by Asher and Pogodalla [AP10] with SDRT. In accordance with this argument, we see context with a frame representation, allowing the process for setting the common ground, and the meaning of speech acts shared by the two participants.

1.3.4 Frame Semantics

Frame semantics is a theory in semantics developed by Fillmore [Fil76, Fil82]. The meaning of a sentence for Fillmore depends on the context in which it is uttered. To take into account the exterior knowledge needed to properly understand the meaning, Fillmore proposes to consider the whole context and the utterance as a scene, where actions take place with specific objects. Fillmore also argues that a specific representation is needed, called frame. A Frame is a relation between an action, usually a verb in a sentence, and objects, usually described as noun phrases. Frames describe actions with roles, and carry specific knowledge. These relations are represented with predicates, supported by thematic roles. Frames are used as a representation for the context with the thematic roles.

A role suits a scene by giving explicit names to the objects involved and related to the action. Consider an exchange context where someone buys an object from another person, we can see roles proposed by Fillmore as a buyer, a seller, goods for example. We will be using the usual general roles given below with their definitions [2].

Role	Meaning
AGENT	The participant that causes the action
THEME	The participant that is directly affected by the action
DIRECTION	The direction of the action
LOCATION	The place where the action occurs
INSTRUMENT	The instrument used to make the action

Figure 2 – Thematic roles

Consider the sentence (30) and its corresponding frame in (31) [3].

(30) Brutus stabbed Caesar with a knife in the agora.

(31)
$$\begin{bmatrix} STAB \\ Ag : brutus \\ Th : caesar \\ Ins : knife \\ Loc : agora \end{bmatrix}$$

Frame semantics is useful for the ability to represent knowledge, and is also well suited for questions, in particular wh-questions. In this kind of questions, the content of an adequate answer can be specified using those roles. Who made the action, or what is done or for whom is the action. These questions are the content of respectively, the agent, the verb or event, and the theme in the scene. In this representation, roles are bound with the verb and depend on it, and are restricted by it. Noun phrases restrain the use of particular roles. Actions and objects affect the analysis of sentences and determining the roles for each constituent elaborates the meaning of the sentences.

2. There exists several list of roles and different meanings. In the table, we will describe the ones we use in our examples

3. For clarity in the rest of the report, we will use abbreviations when using roles, Ag, Th, Dir, Loc, and Ins for respectively Agent, Theme, Direction, Location and Instrument.

2 Modeling dialogue with Type-Theoretical Semantics and Frames

To model dialogue, we need to define a structure and a way to represent its content. For the structure, we use type-theory for its structural aspect, that will allow a handling of dialogue turns of speech with a controlled composition of what can be uttered. The content is represented with frames, that is well-suited for knowledge representation and context representation. And the relation between the structure and the content will be computed with lambda-calculus. The computation is inspired by the representational aspect of DRT to take into account the context.

The purpose of our modeling is to propose a way to process short dialogues involving questions and answers, by computing the content of the common ground for the participants. We restricted ourselves in the view of questions and answers aiming only a specific argument in a previous utterance. To represent the structure of the dialogue (propositions, questions and answers), we use type theory for their syntactic aspect, and define a way to combine them. For the interpretation level, we use frames for the representation of the content and the context of the dialogue.

2.1 Modeling dialogue

2.1.1 A dialogue problem

We consider that in discourse, the meaning of a sentence is a Context Change Potential. In dialogue, the contribution of each participant is taken as well in context. Indeed, the meaning of every utterance cannot be computed in an ongoing dialogue. Each utterance does not always have a truth value on its own. Before going into further explanations, consider the example in Figure 3, from Prévot *et al* [PMDV02]. As we can observe, the meaning of the utterances B_2 or A_3 cannot be computed without considering their respective contexts. The meaning follows from the consideration of the whole dialogue. Each utterance brings new pieces of information, and together they grant the dialogue a meaning.

In discourse, each sentence contributes to the meaning of the discourse, considering it in its context. Similarly in a dialogue, each utterance contributes to the meaning of the dialogue. All the utterances expressed by participants can be seen as speech turns, that composed together form a speech act. We consider a speech act to have a meaning, and each speech turn builds this meaning by considering them in contexts. Frames are used as a representation for the context with the usual properties.

To simplify and for the sake of clarity, the representation is restricted in the frame.

A_1 : Tu tournes à gauche juste avant Monoprix.
: *You turn on the left before Monoprix.*
B_2 : Après Monoprix ?
: *After Monoprix ?*
A_3 : Non, avant Monoprix.
: *No, before Monoprix.*

A'_3 : Oui.
: *Yes.*

Figure 3 – Example of a simple dialogue

Embedded frames for this example will not be considered [4]. Instead the contents *before* and *after* are used to be respectively equivalent to the meaning of "before Monoprix" and "after Monoprix" as in (32), which means "You (B) turn on the left before Monoprix.".

(32) $$\begin{bmatrix} \quad TURN \\ Ag : B \\ Dir : left \\ Loc : before \end{bmatrix}$$

Intuitively the example from Figure 3 can be seen as following :

(33) $$[\![A_1]\!] = \begin{bmatrix} \quad TURN \\ Ag : B \\ Dir : left \\ Loc : before \end{bmatrix}$$

The content of A_1 is a simple proposition, represented in a frame.

(34) $$[\![B_2]\!] = \begin{bmatrix} \quad \dots \\ Loc :?\ (after) \end{bmatrix}$$

The utterance B_2 tries to modify an argument in a previous utterance and wait for an answer. The content of *Loc* is the focus of the question. The representation has the

4. Embedded frames involves some technical issues we will treat in future work especially with quantifiers in frames and scope issues.

form a *pending* frame, a frame with an argument currently under focus that needs a resolution.

(35) $$[\![A_3]\!] = \begin{bmatrix} \dots \\ Loc :?(after) \rightarrow before \end{bmatrix}$$

The utterance A_3 corrects the content of a *Loc* that has been previously questioned, the proposal by B is not taken into account and replaced with A's answer.

(36) $$[\![A'_3]\!] = \begin{bmatrix} \dots \\ Loc :?(after) \rightarrow after \end{bmatrix}$$

With the utterance A'_3, A accepts the proposal by B, the content is filled with *after*, making A's utterance an acknowledgment.

B_2 and A_3's utterances in this example do not have truth conditions. For them to have any kind of meaning, we need to combine them with their contexts. Using (33), (34) and (35), we may obtain an intuitive computation of the representation of the dialogue.

For $[\![A_1]\!]$, the representation is the same as in (33). For computing the two first utterances, the representation of A_1 serves as a context for the utterance of B_2 to have a full representation. B_2 is represented by a question Q uttered by B over a previous utterance from A in A_1.

(37) $$[\![Q\ B2\ A1]\!] = \begin{bmatrix} TURN \\ Ag : B \\ Dir : left \\ Loc :?(after) \end{bmatrix}$$

Notice in particular the question from B over the location and the necessity of introducing a way to retrieve the focus and the new proposal if any [5].

Similarly, for the answer, we consider the previous utterances, represented in (37), as context for the computation of the third utterance. This utterance is a correction (an answer) *CORR* from A in A_3 to the question Q from B in B_2 over the first utterance A_1 from A over the location.

(38) $$[\![CORR\ A3\ (Q\ B2\ A1)]\!] = \begin{bmatrix} TURN \\ Ag : B \\ Dir : left \\ Loc : before \end{bmatrix}$$

5. In the case of *B*'s uttering “where", only a pending frame is necessary.

The correction to the question uttered by B on the location allows A to complete the pending frame with its answer. The content of *Loc* is set to *before* thanks to the correction brought with its utterance in A_3 and replaces the questioning.

(39) $$[\![ACK\ A3'\ (Q\ B2\ A1)]\!] = \begin{bmatrix} TURN \\ Ag : B \\ Dir : left \\ Loc : after \end{bmatrix}$$

In the case of the acknowledgment, the content of *Loc* is set with the content of the proposal by B.

With B's question, we should have a way to represent and resolve the "conflict" between A and B. Our representation is based on frames and type theory, for the focus of the question, or currently questioned value.

The goal in this analysis is to set the common ground, what is understood commonly by the speakers. Each contribution after A_1 in the example serves only for this purpose.

2.1.2 Representing the focus

Our point of view on the dialogue uses the notion of focus, the current information highlighted in the conversation by a particular question and its answers. This analysis is limited to simple wh-questions that focus on a specific argument.

In the representation of the argument under focus, a pending frame is used to memorize the frame with the argument under focus, that need to be resolved. In the example in (40), the focused argument is the Location.

(40) $$\lambda l. \begin{bmatrix} TURN \\ Ag : B \\ Dir : left \\ Loc : \ l \end{bmatrix}$$

In the model, the operator $find_v$ is defined. It searches for a specific argument of type v in a frame of type γ, and returns the argument and its path in the frame, allowing it to be modified. The operator $find_v$ is used as a lens to find a specific argument in the context. The operator is used as an oracle and allows to propose a new value for the focus argument and to have the path in the frame to the argument.

The *path* is defined for the representation of the argument under focus in a frame. It consists of a couple, a specific argument, and a pending frame waiting for an argument of the same type.

(41) $path : P\ \gamma\ v = v \times (v \to \gamma)$

(42) $find_v : \gamma \rightarrow P\ \gamma\ v$

2.1.3 Representing utterances

With the definition of the way we represent an element under focus, types for utterances should be defined. Three kinds of utterances are defined, a proposition S, a question Q and an answer A. The type used is based on type theory and continuation semantics. In this view, contexts are seen as containing information for resolving some issues, like anaphora resolution or pronoun binding.

In our case, we will use contexts to represent the current state of the ongoing dialogue as utterances can not always have a truth value. So we consider a proposition in context that takes as argument a context and returns another updated context. For question, it uses the notion of path, defined in (41), for their type. A question takes as argument a context, and retrieves a *path*. An answer takes as parameter a path, and solves it, by returning a resolved context. S is a basic proposition, Q_v and A_v are respectively a question and an answer about a specific property of type v.

(43) $S \equiv \gamma \rightarrow \gamma$

(44) $Q_v \equiv \gamma \rightarrow P\ \gamma\ v$

(45) $A_v \equiv P\ \gamma\ v \rightarrow \gamma$

In order to compute the representation of utterances in a context, combinators for utterances are defined. In the model, three operators are defined : $.^s$, $.^q$ and $.^a$. Each operator takes two arguments, and is simply a composition of functions.

The operator $.^s$ combines two propositions of type S and returns another proposition of type S. $.^q$ combines a proposition of type S and a question of type Q and returns a question of type Q. And $.^a$ combines a question of type Q and an answer of type A and returns a proposition of type S.

(46) $.^s : S \rightarrow S \rightarrow S$

(47) $.^q : S \rightarrow Q_v \rightarrow Q_v$

(48) $.^a : Q_v \rightarrow A_v \rightarrow S$

For clarity, in the following, the type of the application is given as superscript and the type of the abstraction as subscript. In the exponent, the type of each element involved in the formula is displayed.

The operator $.^s$ can be decomposed in the following way.

(49) $$.^s = \underbrace{\lambda S_1^{\gamma\to\gamma} S_2^{\gamma\to\gamma} c^{\gamma}. S_2(\overbrace{\overbrace{S_1\ c}^{\gamma}}^{\gamma})}_{(\gamma\to\gamma)\to(\gamma\to\gamma)\to\gamma\to\gamma}$$

The type of $.^s$ is developed as follow :

(50) $.^s : (\gamma \to \gamma) \to (\gamma \to \gamma) \to \gamma \to \gamma$

Analogously, the types of the operators for $.^q$ and $.^a$ are :

(51) $.^q : (\gamma \to \gamma) \to (\gamma \to P\ \gamma\ v) \to \gamma \to P\ \gamma\ v$

(52) $.^a : (\gamma \to P\ \gamma\ v) \to (P\ \gamma\ v \to \gamma) \to \gamma \to \gamma$

2.2 Example

To give an insight of the model, the example described in Figure 3 is analyzed.

2.2.1 Types

Utterance typing represents the dialogue structure by allowing combinations of utterances with the operators. A_1 is a proposition in context, it is a context taking as argument another context. B_2 is a question relative to a Location, it takes as parameter a context and retrieves a path. A_3 and A'_3 are answers about a Location, they take as parameters a path to give back an updated context.

(53) $[\![A_1]\!] : S = \gamma \to \gamma$

(54) $[\![B_2]\!] : Q_{Loc} = \gamma \to P\ \gamma\ Loc$

(55) $[\![A_3]\!] : R_{Loc} = P\ \gamma\ Loc \to \gamma$

(56) $[\![A'_3]\!] : R_{Loc} = P\ \gamma\ Loc \to \gamma$

With these typings, the process can be controlled with a question to a proposition, an answer to a question, but not an answer to a proposition.

2.2.2 Representation

In the representation of the utterance, we show what information is contained and how they can be combined with following utterances. These representations differ depending on the purpose of the type. They set the way common ground can be accessed and updated during the conversation. The proposition is simply an information to take in context, here the turning event, whereas the question searches in a context for a path, the path of the Location questioned, and the answers apply a specific response to the question.

As A_1 takes a context to give a context, it has one parameter c, a frame, and gives an updated frame composed with the frame *TURN* and the context frame. We used the operator $\uplus$ for combining frames that takes two frames as parameters and returns one frame, this operator would be of type $\gamma \rightarrow \gamma \rightarrow \gamma$.

(57) $$[\![A_1]\!] = \lambda c. \begin{bmatrix} \quad TURN \\ Ag : B \\ Dir : left \\ Loc : before \end{bmatrix} \uplus c$$

B_2 is a question about a Location. So it takes a context as an argument, and retrieves a *path*, with the proposal *after*. The resulted path contains the new proposal for *Loc*, and the pending frame in which we can find that *Loc*.

(58) $$\begin{aligned} [\![B_2]\!] &= \lambda e.\texttt{let}\ (o, c)\ =\ find_{Loc}(e)\ \texttt{in}\ (after, c) \\ &\equiv \lambda e.((\lambda(o, c).(after, c))(find_{Loc}(e))) \\ &\equiv \lambda e.(after, \pi_2(find_{Loc}(e))) \end{aligned}$$

A_3 and A'_3 are respectively a corrective answer and an acknowledgment. They take as parameter a *path*, and apply the modification proposed by B or correct it with a new *Location* brought by the answer.

(59) $$\begin{aligned} [\![A_3]\!] &= \lambda(Loc, e).e(before) \\ [\![A'_3]\!] &= \lambda(Loc, e).e(Loc) \end{aligned}$$

These representations give a contextual meaning to each utterance that can be computed to give a full meaning to the speech act expressed in the example.

2.2.3 Computation

To compute the representation of the dialogue, each utterance is taken sequentially and combined with its context. The first two speech turns are combined with the $.^q$ operator as the first utterance is a proposition and the second one a question.

(60) $$[\![A_1.^q B_2]\!] = \lambda S\, Q\, c.Q(S\, c)\ [\![A_1]\!][\![B_2]\!] \\ \to_\beta \lambda c.[\![B_2]\!]([\![A_1]\!]\ c)$$

The type of $[\![A_1.^q B_2]\!]$ is indeed of type Q_{Loc} ($\gamma \to P\ \gamma\ Loc$), the type of questions. It awaits only for an answer and does not have a truth value on its own. It will be given by an adequate answer to the question.

(61) $$\underbrace{\lambda c^\gamma.\ \overbrace{[\![B_2]\!]^{\gamma \to P\gamma Loc}\ \overbrace{([\![A_1]\!]^{\gamma\to\gamma} c^\gamma)}^{\gamma}}^{P\ \gamma\ Loc}}_{\gamma \to P\ \gamma\ Loc}$$

The third utterance can be combined with the previous ones, with the $.^a$ operator, for it is an answer completing the question.

(62) $$[\![A_1.^q B_2.^a A_3]\!] \\ = (\lambda QA\ c.A(Qc))\ [\![A_1.^q B_2]\!][\![A_3]\!] \\ \to_\beta \lambda c.[\![A_3]\!]([\![A_1.^q B_2]\!]\ c) \\ \equiv \lambda c.[\![A_3]\!](\lambda c.[\![B_2]\!]([\![A_1]\!]c))c) \\ \to_\beta \lambda c.[\![A_3]\!]([\![B_2]\!]([\![A_1]\!]c))$$

The type of $[\![A_1.^q B_2]\!].^a A_3]\!]$ is of type S ($\gamma \to \gamma$), the type of propositions. It can be combined with a context to give a context, that can have a truth-conditional meaning, or be combined with another proposition or question.

(63) $$\underbrace{\lambda c^\gamma.\ \overbrace{[\![A_3]\!]^{P\gamma Loc\to\gamma}\ \overbrace{([\![B_2]\!]^{\gamma\to P\gamma Loc}([\![A_1]\!]^{\gamma\to\gamma} c^\gamma))}^{P\ \gamma\ Loc}}^{\gamma}}_{\gamma\to\gamma}$$

The context of the dialogue is empty, so for the computation of the dialogue representation, we consider an empty context, a frame $c_e = [\ ]$ of type γ, that will be passed as the context to the expression (62). As the operators we used are all compositions of functions, the representation can be computed sequentially, by applying the empty context to A_1, then the result applied to B_2, and finally to A_3.

(64) $$[\![A_1.^q B_2.^a A_3]\!]c_e \\ = (\lambda c.[\![A_3]\!]([\lambda c.[\![B_2]\!]([\![A_1]\!]c)]c)c_e \\ \to_\beta [\![A_3]\!]([\lambda c.[\![B_2]\!]([\![A_1]\!]c)]c_e) \\ \to_\beta [\![A_3]\!]([\![B_2]\!]([\![A_1]\!]c_e))$$

The representation with the utterance A'_3 is computed in a similar way.

(65) $[\![A_1.^q B_2.^a A'_3]\!] c_e = [\![A'_3]\!]([\![B_2]\!]([\![A_1]\!] c_e))$

First, the empty context is applied to $[\![A_1]\!]$. With the operator $\uplus$, we obtain the frame $[\![1]\!]$ that can be used for the next computation. At this step, the result is a representation of the proposition expressed by A. We suppose that the operator $\uplus$ applied with an empty context gives the same context[6].

(66)
$$
\begin{aligned}
[\![A_1]\!] c_e &= (\lambda c. \begin{bmatrix} \quad TURN \\ Ag : B \\ Dir : left \\ Loc : before \end{bmatrix} \uplus c)(c_e) \\
&\rightarrow_\beta \begin{bmatrix} \quad TURN \\ Ag : B \\ Dir : left \\ Loc : before \end{bmatrix} \uplus (c_e) \\
&= \begin{bmatrix} \quad TURN \\ Ag : B \\ Dir : left \\ Loc : before \end{bmatrix} \uplus [\,] \\
&\simeq \begin{bmatrix} \quad TURN \\ Ag : B \\ Dir : left \\ Loc : before \end{bmatrix} = [\![1]\!]
\end{aligned}
$$

(67) $[\![A_1.^q B_2.^a A_3]\!] c_e \equiv [\![A_3]\!]([\![B_2]\!][\![1]\!])$

Afterwards, the frame $[\![1]\!]$ is applied to $[\![B_2]\!]$. The operator $find_{Loc}$ takes the context frame $[\![1]\!]$ as argument and retrieves the pair of the form $(arg, Path)$ where arg is the *Location before* found in the context, and Path is the pending frame in (68).

(68)
$$
\lambda l. \begin{bmatrix} \quad TURN \\ Ag : B \\ Dir : left \\ Loc : l \end{bmatrix}
$$

6. The operator $\uplus$ should be defined to take care of the way frames are combined, a topic that will be the concern of future works.

The result obtained after computation of this part is the pair represented by $[\![2]\!]$, the argument of Location proposed by the question and the path in the context to be modified. This step shows how *B* with its utterance focuses a specific argument to question from *A*'s proposition. With the proposal in the first item in the pair, the content proposed by *B* is aiming for the argument *Loc* to be modified depending on an (adequate) answer.

(69) $$[\![B_2]\!][\![1]\!] = \lambda e.(after, \pi_2(find_{Loc}e))[\![1]\!]$$
$$= \lambda e.(after, \pi_2(find_{Loc}e)) \begin{bmatrix} TURN \\ Ag : B \\ Dir : left \\ Loc : before \end{bmatrix}$$
$$\rightarrow_\beta (after, \pi_2(find_{Loc} \begin{bmatrix} TURN \\ Ag : B \\ Dir : left \\ Loc : before \end{bmatrix}))$$
$$= (after, \lambda l. \begin{bmatrix} TURN \\ Ag : B \\ Dir : left \\ Loc : l \end{bmatrix}) = [\![2]\!]$$

(70) $$[\![A_1.^q B_2.^a A_3]\!]c_e \equiv [\![A_3]\!][\![2]\!]$$

The pending frame must be resolved with an answer about a *Location*. With the correction uttered by A in A_3, we apply the argument *before* to the pending frame in $[\![2]\!]$. So in the representation, the first item in the pair $after$, representing the proposal by B is corrected with the answer $before$.

(71) $$[\![A_3]\!][\![2]\!] = (\lambda(Loc, e).e\ before)([\![2]\!])$$
$$= (\lambda(Loc, e).e\ before)(after, \lambda l. \begin{bmatrix} TURN \\ Ag : B \\ Dir : left \\ Loc : l \end{bmatrix})$$
$$\rightarrow_\beta (\lambda l. \begin{bmatrix} TURN \\ Ag : B \\ Dir : left \\ Loc : l \end{bmatrix})(before)$$
$$= \begin{bmatrix} TURN \\ Ag : B \\ Dir : left \\ Loc : before \end{bmatrix}$$

In the case of the correction, the pending frame is resolved by correcting it with the proposal uttered by the answer. The result is shown in (72).

(72) $$[\![A_1.^q B_2.^a A_3]\!]c_e = \begin{bmatrix} TURN \\ Ag : B \\ Dir : left \\ Loc : before \end{bmatrix}$$

In the case of the acknowledgment, the pending frame is resolved by completing it with the proposed argument uttered by B in B_2, validated by A in A'_3 with its answer. In this case, the content of the first item in the pair $[\![2]\!]$, representing the proposal from B is used to replace the Loc in the pending frame, the argument currently under focus, introduced by the utterance B_2.

(73) $$[\![A3']\!][\![2]\!] = (\lambda(Loc, e).e\ Loc)([\![2]\!])$$
$$= (\lambda(Loc, e).e\ Loc)(after, \lambda l. \begin{bmatrix} TURN \\ Ag : B \\ Dir : left \\ Loc : l \end{bmatrix})$$
$$\rightarrow_\beta (\lambda l. \begin{bmatrix} TURN \\ Ag : B \\ Dir : left \\ Loc : l \end{bmatrix})(after)$$
$$\rightarrow_\beta \begin{bmatrix} TURN \\ Ag : B \\ Dir : left \\ Loc : after \end{bmatrix}$$

The resulting frame is updated within the conversation between A and B, in which B questions A's utterance with its own interpretation. A accepts B's modification by acknowledging the proposal by B, changing the content of the frame, setting another shared content between them. The result is showed in (74).

(74) $$[\![A_1.^q B_2.^a A_3']\!]c_e = \begin{bmatrix} TURN \\ Ag : B \\ Dir : left \\ Loc : after \end{bmatrix}$$

At this point, the result is of type γ, a context, but the corresponding meaning is straightforward. In the two cases, B's utterance questions a specific argument, the *Location*, and depending on the answer can have a different value for the Location. The computation of the example shows the structure of the dialogue, modeled by the types allowing or not the computation. The content is represented by frames in terms of contexts computed sequentially from each utterance. The common ground is set at the end of the computation, showing that in dialogue, the content of question and answer is seen as a whole and its contribution cannot be taken separately. If no questioning utterances follow, the resulted frame represents the common ground between the participants. It will serve, in addition to common world knowledge, as a basis for the conversation.

3 Extension

3.1 Hybrid Logic

3.1.1 Formal definition

We follow the notations of Kallmeyer *et al.*[KLO$^+$15] for the definitions of hybrid logic. The goal is to introduce quantification in frames. *Rel* is set of relational symbols, *Prop* a set of propositional variables, *Nom* a set of nominals and *Svar* a set of state variables.

The language of formulas is $Forms ::= \tau|p|s|\neg\phi|\phi_1 \wedge \phi_2|\langle R\rangle|\mathbf{\exists}\phi|@_s\phi|\downarrow x.\phi|\exists x.\phi$ where $p \in Prop$, $s \in Stat$, $R \in Rel$ and ϕ, ϕ_1 and $\phi_2 \in Forms$.

A model $\mathcal{M}$ is a triple $\langle M, (R^{\mathcal{M}})_{R\in Rel}, V\rangle$ such that M is a non-empty set, each $R^{\mathcal{M}}$ is a binary relation on M, and the valuation $V : Prop \cap Nom \rightarrow \Gamma(M)$ is such that if $i \in Nom$ then $V(i)$ is a singleton.

An assignment g is a mapping $g : Svar \rightarrow M$. For an assignment g, g^x_m is an assignment that differs from g at most on x and $g^m_x(x) = m$. For $s \in Stat$, we also define $[s]^{\mathcal{M},g} = g(s)$ if $s \in Svar$

Let $\mathcal{M}$ be a model, $w \in M$, and g an assignment for M. The satisfaction relation is defined as follows:

$\mathcal{M}, g, w \vDash \top$	
$\mathcal{M}, g, w \vDash s$	iff $w = [s]^{\mathcal{M},g}$for$s \in$ Stat
$\mathcal{M}, g, w \vDash \neg\phi$	iff $\mathcal{M}, g, w \nvDash \phi$
$\mathcal{M}, g, w \vDash \phi_1 \wedge \phi_2$	iff $\mathcal{M}, g, w \vdash \phi_1$and $\mathcal{M}, g, w \vdash \phi_2$
$\mathcal{M}, g, w \vDash \langle\mathcal{R}\rangle\phi$	iff there is a $w' \in \mathcal{M}$ such that $R^M(w, w')$ and $\mathcal{M}, g, w' \vdash \phi$
$\mathcal{M}, g, w \vDash p$	iff $w \in V(p)$for $p \in$ Prop
$\mathcal{M}, g, w \vDash @_s\phi$	iff $\mathcal{M}, g, [s]^{\mathcal{M},g} \nvDash \phi$ for $s \in$ Stat
$\mathcal{M}, g, w \vDash \downarrow x.\phi$	iff $\mathcal{M}, g^x_w, w \nvDash \phi$
$\mathcal{M}, g, w \vDash \mathbf{\exists} x.\phi$	iff there is a $w' \in M$ such that $\mathcal{M}, g^x_{w'}, w \nvDash \phi$
$\mathcal{M}, g, w \vDash \mathbf{\exists}\phi$	iff there is a $w' \in M$ such that $\mathcal{M}, g, w' \nvDash \phi$

Also, is defined $\mathbf{\forall}\phi \equiv \neg\mathbf{\exists}(\neg\phi) and \phi \implies \psi \equiv (\neg\phi) \vee \psi$.

A formula ϕ is :

— *satisfiable* if there is a model $\mathcal{M}$, and an assigment g on $\mathcal{M}$, and a state $w \in M$ such that $\mathcal{M}, g, w \models \phi$.

— *globally true* in a model $\mathcal{M}$ under an assignment g it is satisfiable at all states of the model.

3.1.2 Representation

The representation of frames using quantification can be seen as a graph, where each node is an event, or an object, and each arrow is a relation described by a thematic role [7].

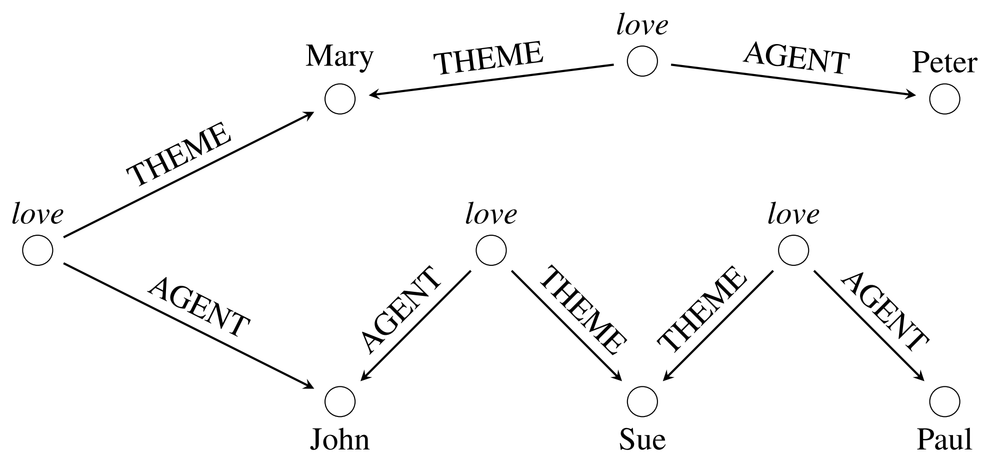


Figure 4 – Graph representation of a model using frames

Consider the following sentences.

(75) John loves Mary.

The formula in hybrid logic with frames is given by the following.

(76) $\exists(love \wedge \langle AGENT \rangle John \wedge \langle THEME \rangle Mary)$

This formula is satisfiable by a model where there exists a node "*love*" that have a relation "*AGENT*" with another node that is "*John*" and an relation "*THEME*" that is Mary.

Consider now the well-known ambiguous sentence.

(77) Every man loves a woman.

7. We assume that the property of being a man or woman is included in the person node

The representation of both reading is given in the following.

(78) $\forall(\downarrow i.man \implies \exists(\downarrow i'.woman \wedge \exists(love \wedge \langle AGENT\rangle i \wedge \langle THEME\rangle i')))$

In this expression, the content symbolizes that for all nodes that have the property "*man*" implies that there exists a node with the property "*woman*" and there exists a node love with the relation "*AGENT*" with the man, and a relation "*THEME*" with the woman.

(79) $\exists(\downarrow i.woman \wedge \forall(\downarrow i'.man \implies \exists(love \wedge \langle AGENT\rangle \wedge \langle THEME\rangle i)))$

This formula expresses the other reading of the sentence with the existence of a specific woman whom is loved by every man in the model. we can see, following the graphic representation how the model can fit the expression. The model shown in 4 properly fits the expression "*Every man loves a woman*" where several women are involved but not the other reading where only one woman is involved. The expression "*John loves Mary*" is also true in this model.

3.2 Modeling Quantifiers

For dealing with quantification, the use of hybrid logic allowed the representation of dialogue with frames. With this addition, our notion of focus can be applied for quantification. In a similar way as a specific argument in a frame, quantifier can be represented as searching for a specific noun phrase with a quantifier, and the path in the frame. The focus is in the same way, represented with a couple, with in the first argument, the quantified element focused by a specific question, and the second part the path. The representation of the noun phrase is given with an abstraction allowing the proper recovery of the meaning of the context. Indeed, the presence of the quantifier and its representation using hybrid logic forces the abstraction for a proper retrieval of the object.

(80) $\lambda P.\exists(\downarrow y.man \wedge P(y)))$

The operator $find_{NP}$ is also used for that representation, and allows to search for a specific element that can be quantified. The operator works as an oracle for it may have to search in multiple quantified expressions, but will always retrieves the "good" one.

In the same way a question can focus a location, the $find$ operator will search for either every existential quantifier or every universal ones. The path is given also with an abstraction for the same reason as the retrieval of the object, here the noun phrase.

(81) $\lambda Q.Q(\lambda x.\exists(action \wedge \langle Agent\rangle x \wedge \langle otherRole\rangle \ldots))$

3.3 Example

This short dialogue exemplifies the use of the representation of quantifier with hybrid logic and the model we use.

A_1 : Every man turn left before Monoprix.

B_2 : Every man ?

A_3 : Yes.

A'_3 : No, a man.

Figure 5 – Simple dialogue with quantified noun

3.3.1 Types

The structure of the dialogue is the same as the example in the main idea but focuses on a noun phrase that involves quantification. The sequence is the same as the previous example with first an assertion seen as a proposition, followed by a question, here over a noun phrase, and finally the answer over the same noun phrase.

A_1 : S

B_2 : Q_{NP}

A_3 : A_{NP}

A'_3 : A_{NP}

3.3.2 Representation

With the introduction of hybrid logic, the representation for frames shifts from the box notation to graph notation.

The equivalent representation of $\begin{bmatrix} TURN \\ Ag : you \\ Dir : left \\ Loc : before \end{bmatrix}$ is obtained with a graph representation with $\exists(turn \wedge \langle AG\rangle you \wedge \langle DIR\rangle left \wedge \langle LOC\rangle before_{MP})$

The representation of A_1 is given by the following expression, where for all nodes that have the property of being a man, there exists a node turn with the corresponding relation, namely, Agent, Direction and Location.

(82) $[\![A_1]\!] = \lambda c.\forall(\downarrow x.man \implies \exists(turn \wedge \langle AG\rangle x \wedge \langle DIR\rangle left \wedge \langle LOC\rangle before_{MP}) \uplus c$

The representation of B_2 uses the focus on quantification, with the definition given above. The proposal by B is stored in the first component of the couple, and the second component is the path that give the place that is questioned.

(83) $[\![B_2]\!] = \lambda c.(\lambda P.\forall(\downarrow y.man \implies P(y), \pi_2 find_{NP}(c))$

According to the answers, we apply, to the pending representation of the sentence, a different argument. In the first case, we apply the content stored and given by B, and in the second one, we apply the correction added by A in his answer.

(84) $[\![A_3]\!] = \lambda(NP, e).e(NP)$
$[\![A'_3]\!] = \lambda(NP, e).e(\lambda P.\forall(\downarrow y.man \implies P(y))$

3.3.3 Computation

The computation is done in the same way for the example in the main idea. It is done compositionally by adding each utterance to the context, that is what is previously uttered, beginning by an empty context.

(85)
$$\begin{aligned}[\![A_1.^q B_2]\!] &= \lambda S\,Qc.Q(S\,c)[\![A_1]\!][\![B_2]\!] \\ &= \lambda c.[\![B_2]\!]([\![A_1]\!]c)\end{aligned}$$

(86)
$$\begin{aligned}[\![A_1.^q B_2.^a A_3]\!] &= \lambda QAc.A(Qc)[\![A_1.^q B_2]\!][\![A_3]\!] \\ &= \lambda c.[\![A_3]\!]([\![A_1.^q B_2]\!]c) \\ &= \lambda c.[\![A_3]\!](\lambda c.[\![B_2]\!]([\![A_1]\!]c)c) \\ &= \lambda c.[\![A_3]\!]([\![B_2]\!]([\![A_1]\!]c)\end{aligned}$$

With an empty context c_{empty}, the computation is done as follow. As the context is empty, the representation of the first utterance is given by the following formula.

(87)

$$[\![A_1]\!]c_{empty} = \forall(\downarrow x.man \implies \exists(turn \wedge \langle AG \rangle x \wedge \langle DIR \rangle left \wedge \langle LOC \rangle before_{MP}) = [\![1]\!]$$

When B_2 is uttered, the question focuses on the quantified noun "*man*" that need to be represented. For that, the use of the representation described above is shown. The result obtained is a couple with the first argument being the proposed noun phrases by B, and the second argument, being its path. Its path is given in a λ-expression.

(88)

$$\begin{aligned}
&[\![B_2]\!][\![1]\!] = \lambda c.(\lambda P.\forall(\downarrow y.man \wedge P(y)), \pi_2 find_{NP}(c)) \\
&(\forall(\downarrow x.man \wedge \exists(turn \wedge \langle AG \rangle x \wedge \langle DIR \rangle left \wedge \langle LOC \rangle before_{MP})) \\
&= (\lambda P.\forall(\downarrow y.man \implies P(y)), \\
&\pi_2 find_{NP}(\forall(\downarrow x.man \wedge \exists(turn \wedge \langle AG \rangle x \wedge \langle DIR \rangle left \wedge \langle LOC \rangle before_{MP}))) \\
&= (\lambda P.\forall(\downarrow y.man \implies P(y), \\
&\lambda Q.Q(\lambda x.\exists(turn \wedge \langle AG \rangle x \wedge \langle DIR \rangle left \wedge \langle LOC \rangle before_{MP}) = [\![2]\!]
\end{aligned}$$

In the first argument of the couple, we can observe that the "*everyman*" part is indeed retrieved and that the structure remains with the $\implies$ symbols and the content abstracted with the λP. In the second argument, the path is kept with the frame representation, except for the "AGENT" that is abstracted and needs to be replaced by the content of the answer.

In the first case of the example, the acknowledgment is represented by applying the path to the proposed noun phrase.

(89)

$$\begin{aligned}
&[\![A_3]\!][\![2]\!] = \lambda(NP, e).e(NP)(\lambda P.\forall(\downarrow y.man \implies P(y), \\
&\lambda Q.Q(\lambda x.\exists(turn \wedge \langle AG \rangle x \wedge \langle DIR \rangle left \wedge \langle LOC \rangle before_{MP}) \\
&\rightarrow_\beta \lambda Q.Q(\lambda x.\exists(turn \wedge \langle AG \rangle x \wedge \langle DIR \rangle left \wedge \langle LOC \rangle before_{MP}))(\lambda P.\forall(\downarrow y.man \implies P(y)) \\
&\rightarrow_\beta \lambda P.\forall(\downarrow y.man \implies P(y)(\lambda x.\exists(turn \wedge \langle AG \rangle x \wedge \langle DIR \rangle left \wedge \langle LOC \rangle before_{MP}) \\
&\rightarrow_\beta \forall(\downarrow y.man \implies \lambda x.\exists(turn \wedge \langle AG \rangle x \wedge \langle DIR \rangle left \wedge \langle LOC \rangle before_{MP})(y) \\
&\rightarrow_\beta \forall(\downarrow y.man \implies \exists(turn \wedge \langle AG \rangle y \wedge \langle DIR \rangle left \wedge \langle LOC \rangle before_{MP})
\end{aligned}$$

Indeed in applying the content of the first couple to the man, the result is in the same representation as the first utterance. We can see how the repair can be done with this representation.

For the second answer however, the application is done with the correction given by A. In this case, the path is filled with the answer given by A.

(90) $[\![A'_3]\!][\![2]\!] = \lambda(NP, e).e(\lambda P.\exists(\downarrow y.man \wedge P(y))(\lambda P.\forall(\downarrow y.man \implies P(y),$
$\lambda Q.Q(\lambda x.\exists(turn \wedge \langle AG\rangle x \wedge \langle DIR\rangle left \wedge \langle LOC\rangle before_{MP})$
$\rightarrow_\beta \lambda Q.Q(\lambda x.\exists(turn \wedge \langle AG\rangle x \wedge \langle DIR\rangle left \wedge \langle LOC\rangle before_{MP}))(\lambda P.\exists(\downarrow y.man \wedge P(y))$
$\rightarrow_\beta \lambda P.\exists(\downarrow y.man \wedge P(y)(\lambda x.\exists(turn \wedge \langle AG\rangle x \wedge \langle DIR\rangle left \wedge \langle LOC\rangle before_{MP})$
$\rightarrow_\beta \exists(\downarrow y.man \wedge \lambda x.\exists(turn \wedge \langle AG\rangle x \wedge \langle DIR\rangle left \wedge \langle LOC\rangle before_{MP}(y))$
$\rightarrow_\beta \exists(\downarrow y.man \wedge \exists(turn \wedge \langle AG\rangle y \wedge \langle DIR\rangle left \wedge \langle LOC\rangle before_{MP}$

The result obtained represents the proper content of the dialogue after repair of the content that was misunderstood. The content is given in terms of model with frames, that can easily be converted in first order logic and thus have a meaning.

Another example is given for the proper treatment of quantified noun phrases and non quantified non phrases. As the noun phrases shift the types and the λ-expression. "*John*" in this case must be type-raised as $\lambda P.P(john)$. In this case the computation is properly handled with focus over quantification we propose.

A_1 : John turns left before Monoprix.
B_2 : John ?
A_3 : No, a man.

Figure 6 – Simple dialogue involving quantified noun and non-quantified noun phrase

The first utterance is the same expression as before represented with frames and now with hybrid logic. The expression means there exists a node "*turn*" with a relation "AGENT" with the node "*John*", a relation "DIRECTION" with "left" and a relation "LOCATION" with "$before_{MP}$".

(91) $[\![A_1]\!] = \lambda c.\exists(turn \wedge \langle AG\rangle john \wedge \langle DIR\rangle left \wedge \langle LOC\rangle before_{MP}) \uplus c$

The utterance B_2 represented the focus on the agent, and involves the noun phrase "*John*", that will be corrected with a quantified noun. The proper type for "*John*" needs a type-raising to fit the type NP.

(92) $[\![B_2]\!] = \lambda c.(\lambda P.P(john), \pi_2 find_{NP}(c))$

In the last utterance, A_3 corrects the question asked by B and adds the information that will build the proper common ground, assuming that B accepts the information. With no further information we suppose that B accepts A answer.

(93) $[\![A_3]\!] = \lambda(NP, e).e(\lambda P.\exists(\downarrow y.man \wedge P(y))$

The computation is done in the same way as the previous example, compositionally from the first utterance, with a question and an answer.

(94) $[\![A_1.^q B_2.^a A_3]\!] = \lambda c.[\![A_3]\!]([\![B_2]\!]([\![A_1]\!]c)$

With an empty context c_{empty}, we obtain the frame representation of the content of the utterance.

(95) $[\![A_1]\!]c_{empty} = \exists(turn \wedge \langle AG\rangle john \wedge \langle DIR\rangle left \wedge \langle LOC\rangle before_{MP}) = [\![1]\!]$

The proposal by B takes as a context the previous expression, that will be searched for the path of the NP. The first argument of the couple is the correct type-raised proposition for "*John*" and the second one is the path where we can find the NP, that will accept an NP and replaces the content of "AGENT" in our example.

(96) $[\![B_2]\!][\![1]\!] = \lambda c.(\lambda P.P(john), \pi_2 find_{NP}(c))(\exists(turn \wedge \langle AG\rangle john \wedge \langle DIR\rangle left \wedge \langle LOC\rangle before_{MP}))$
$\rightarrow_\beta (\lambda P.P(john), \pi_2 find_{NP}(\exists(turn \wedge \langle AG\rangle john \wedge \langle DIR\rangle left \wedge \langle LOC\rangle before_{MP}))$
$\rightarrow_\beta (\lambda P.P(john), \lambda Q.Q(\lambda x.\exists(turn \wedge \langle AG\rangle x \wedge \langle DIR\rangle left \wedge \langle LOC\rangle before_{MP})) = [\![2]\!]$

The last utterance complete the sequence with an answer. As it is a correction, the path with a currently pending "AGENT" is filled with the A utterance, with "a man". We apply the representation of the NP "a man" to the path, and complete pending content.

(97) $[\![A_3]\!][\![2]\!] = \lambda(NP, e).e($
$\lambda P.\exists(\downarrow y.man \wedge P(y))((\lambda P.P(john), \lambda Q.Q(\lambda x.\exists(turn \wedge \langle AG\rangle x \wedge \langle DIR\rangle left \wedge \langle LOC\rangle before_{MP})))$
$\rightarrow_\beta \lambda Q.Q(\lambda x.\exists(turn \wedge \langle AG\rangle x \wedge \langle DIR\rangle left \wedge \langle LOC\rangle before_{MP}))(\lambda P.\exists(\downarrow y.man \wedge P(y))$
$\rightarrow_\beta \lambda P.\exists(\downarrow y.man \wedge P(y))(\lambda x.\exists(turn \wedge \langle AG\rangle x \wedge \langle DIR\rangle left \wedge \langle LOC\rangle before_{MP}))$
$\rightarrow_\beta \exists(\downarrow y.man \wedge \lambda x.\exists(turn \wedge \langle AG\rangle x \wedge \langle DIR\rangle left \wedge \langle LOC\rangle before_{MP})(y))$
$\rightarrow_\beta \exists(\downarrow y.man \wedge \exists(turn \wedge \langle AG\rangle y \wedge \langle DIR\rangle left \wedge \langle LOC\rangle before_{MP}))$

In the final representation, the ”AGENT" is properly represented and erased the previous non-quantified noun phrase.

4 Conclusion and Perspective

Dynamic semantics extended the handling of sentences for discourse. With view that the meaning of a discourse is a relation between context. This framework raised new distinctive problems considering dialogue. In dialogue, the interactions between participants and its consequences have to be handled as well. Those issues have involved giving a structure for dialogue and novel ways to handle the meaning during its process. To this regard, we propose an insight into this resolution based on type-theory semantics using frames, by representing utterances with typing, and meaning as a relation between them. The meaning is represented as frames emphasizing on events and objects through thematic roles. The idea developed here takes into account common grounding between participants, by constructing the representation during the process of dialogue. The idea focuses mostly on questions and answers, by taking into account the request for information occurring in conversations. As result, a dialogue is seen as a relations between utterances, considered in contexts, where each utterances is typed with a definite types in accordance to its contribution, an assertion, a question or an answer. These different typings are linked with utterances connectors that structure the way we can compose them together. The meaning is handled using lambda-expressions, and the representation is given with frames. The computation is done compositionally by considering each utterances and the previous ones, with simply-typed lambda-calculus.

In further investigation, this idea needs to take into account issues already raised in dynamic framework but transposed to dialogue. In particular, the handling of quantifier range and accessibility of discourse referents. In this direction, the use of rhetorical relations described in SDRT for dialogue may be useful. Adding discourse connectors into the frames representation to take into account the right-frontier constraint. In addition, negation is an interesting question to deal with as it cancels the availability of some referents. Considering it may allow to improve the coverage of the idea presented in this thesis. In works related, we can rely on the works done by Qian[Qia14] regarding double negation and modality with type-theoretical continuation semantics to handle accessibility. And considering frames semantics, we may extend the representation by allowing embedded frames, with more described roles and contents. This will allow to extend the coverage and give more precise meaning to utterances. With the introduction of these notions, the coverage of the idea presented in this paper can be improved to analyze more complex dialogues.

References

[AL98] Nicholas Asher and Alex Lascarides. Questions in dialogue. *Linguistics and Philosophy*, 21(3):237–309, 1998.

[AL03] Nicholas Asher and Alex Lascarides. *Logics of conversation*. Cambridge University Press, 2003.

[AP10] Nicholas Asher and Sylvain Pogodalla. SDRT and continuation semantics. In *New Frontiers in Artificial Intelligence*, pages 3–15. Springer, 2010.

[dG06] Philippe de Groote. Towards a Montagovian account of dynamics. In *Semantics and Linguistic Theory*, volume 16, pages 1–16, 2006.

[Fil76] Charles J Fillmore. Frame semantics and the nature of language*. *Annals of the New York Academy of Sciences*, 280(1):20–32, 1976.

[Fil82] Charles Fillmore. Frame semantics. *Linguistics in the morning calm*, pages 111–137, 1982.

[Ginar] Jonathan Ginzburg. The semantics of dialogue. To appear in: Maria Aloni and Paul Dekker (eds) Cambridge Handbook of Formal Semantics, Cambridge University Press, To appear.

[Gri70] Herbert P Grice. *Logic and conversation*. na, 1970.

[GS85] Jeroen Groenendijk and Martin Stokhof. On the semantics of questions and the pragmatics of answers. *Semantics: Critical Concepts in Linguistics*, page 288, 1985.

[Hei82] Irene Heim. The semantics of definite and indefinite noun phrases. 1982.

[Kam81] Hans Kamp. A theory of truth and semantic representation. *Formal semantics-the essential readings*, pages 189–222, 1981.

[Kar77] Lauri Karttunen. Syntax and semantics of questions. *Linguistics and philosophy*, 1(1):3–44, 1977.

[KLO+15] Laura Kallmeyer, Timm Lichte, Rainer Osswald, Sylvain Pogodalla, and Christian Wurm. Quantification in frame semantics with hybrid logic. In *Type Theory and Lexical Semantics*, 2015.

[LA+09] Alex Lascarides, Nicholas Asher, et al. The interpretation of questions in dialogue. In *Proceedings of Sinn und Bedeutung*, volume 13, pages 17–30. Citeseer, 2009.

[Mon73] Richard Montague. *The proper treatment of quantification in ordinary English*. Springer, 1973.

[MP01] Philippe Muller and Laurent Prévot. The rhetorical attachment of questions and answers. In *First International Workshop on Semantics, Pragmatics, and Rhetoric, Donostia-San Sebastián, November 22*, volume 24, 2001.

[Mul14] Philippe Muller. *Computational pragmatics: time, space and discourse — Phénomènes contextuels en TAL: temps, espace et discours*. Habilitation à diriger des recherches, Université de Toulouse, Toulouse, France, mai 2014.

[PMDV02] Laurent Prévot, Philippe Muller, Pascal Denis, and Laure Vieu. Une approche sémantique et rhétorique du dialogue. un cas d'étude: l'explication d'un itinéraire. *TAL. Traitement automatique des langues*, 43(2):43–70, 2002.

[Qia14] Sai Qian. *Accessibility of Referents in Discourse Semantics*. PhD thesis, Université de Lorraine, 2014.

[Sta98] Robert Stalnaker. On the representation of context. *Journal of Logic, Language and Information*, 7(1):3–19, 1998.

[Sta02] Robert Stalnaker. Common ground. *Linguistics and philosophy*, 25(5):701–721, 2002.

[VF08] Kai Von Fintel. What is presupposition accommodation, again ?*. *Philosophical perspectives*, 22(1):137–170, 2008.

[Z+97] Henk Zeevat et al. The common ground as a dialogue parameter. In *Proceedings of MunDial*, volume 97, pages 97–106, 1997.

Appendices

A Type-Theoretical Continuation Semantics

Details for the computation of the meaning of the following sentences.

1. John loves Mary.
2. He smiles at her.

Syntactic categories

He, her : NP
smiles_at : $Np \to NP \to S$
Semantic meaning

$$
\begin{aligned}
[\![Mary]\!] &= \lambda\psi e.\psi m(m :: e)\\
[\![John]\!] &= \lambda\psi e.\psi j(j :: e)\\
[\![he]\!] &= \lambda\psi e.\psi(\mathtt{sel}_{he} e)e\\
[\![her]\!] &= \lambda\psi e.\psi(\mathtt{sel}_{she} e)e\\
[\![loves]\!] &= \lambda ose\phi.s(\lambda xe.o(\lambda ye.lovexy \wedge \phi e)e)e\\
[\![smiles\_at]\!] &= \lambda ose\phi.s(\lambda xe.o(\lambda ye.smilexy \wedge \phi e)e)e
\end{aligned}
$$

$$
\begin{aligned}
&([\![love]\!][\![Mary]\!])\\
&= \lambda ose\phi.s(\lambda xe.o(\lambda ye.lovexy \wedge \phi e)e)e[\lambda\psi e.\psi m(m :: e)]\\
&\to_\beta \lambda se\phi.s(\lambda xe.[\lambda\psi e.\psi m(m :: e)](\lambda ye.lovexy \wedge \phi e)e)e\\
&\to_\beta \lambda se\phi.s(\lambda xe.(\lambda ye.lovexy \wedge \phi e)m(m :: e))e\\
&\to_\beta \lambda se\phi.s(\lambda xe.lovexm \wedge \phi(m :: e))e
\end{aligned}
$$

$$(\llbracket love\rrbracket\llbracket Mary\rrbracket)\llbracket John\rrbracket =$$
$$(\lambda se\phi.s(\lambda xe.lovexm \wedge \phi(m :: e))e)[\lambda\psi e.\psi j(j :: e)]$$
$$\rightarrow_\beta \lambda e\phi.[\lambda\psi e.\psi j(j :: e)](\lambda xe.lovexm \wedge \phi(m :: e))e$$
$$\rightarrow_\beta \lambda e\phi.(\lambda xe.lovexm \wedge \phi(m :: e))j(j :: e)$$
$$\rightarrow_\beta \lambda e\phi.love\ j\ m \wedge \lambda e.\phi(j :: e)(m :: e)$$
$$\rightarrow_\beta \lambda e\phi.love\ j\ m \wedge \phi(j :: m :: e)$$

$$\llbracket smiles\_at\rrbracket\llbracket her\rrbracket =$$
$$(\lambda ose\phi.s(\lambda xe.o(\lambda ye.smilexy \wedge \phi e)e)e)[\lambda\psi e.\psi(\texttt{sel}_{she}e)e]$$
$$\rightarrow_\beta (\lambda se\phi.s(\lambda xe.[\lambda\psi e.\psi(\texttt{sel}_{she}e)e](\lambda ye.smilexy \wedge \phi e)e)e)$$
$$\rightarrow_\beta (\lambda se\phi.s(\lambda xe.(\lambda ye.smilexy \wedge \phi e)(\texttt{sel}_{she}e)e)e)$$
$$\rightarrow_\beta (\lambda se\phi.s(\lambda xe.smilex(\texttt{sel}_{she}e) \wedge \phi e)e)$$

$$(\llbracket smiles\_at\rrbracket\llbracket her\rrbracket)\llbracket he\rrbracket =$$
$$(\lambda se\phi.s(\lambda xe.smilex(\texttt{sel}_{she}e) \wedge \phi e)e)[\lambda\psi e.\psi(\texttt{sel}_{he}e)e]$$
$$\rightarrow_\beta \lambda e\phi.[\lambda\psi e.\psi(\texttt{sel}_{he}e)e](\lambda xe.smilex(\texttt{sel}_{she}e) \wedge \phi e)e$$
$$\rightarrow_\beta \lambda e\phi.(\lambda xe.smilex(\texttt{sel}_{she}e) \wedge \phi e)(\texttt{sel}_{he}e)e$$
$$\rightarrow_\beta \lambda e\phi.smile(\texttt{sel}_{he}e)(\texttt{sel}_{she}e) \wedge \phi e$$

$$\llbracket 1\rrbracket = \lambda e\phi.love\ j\ m \wedge \phi(j :: m :: e)$$
$$\llbracket 2\rrbracket = \lambda e\phi.smile(\texttt{sel}_{he}e)(\texttt{sel}_{she}e) \wedge \phi e$$

B Representing utterances

Details of type-checking for utterances types.

$$S \equiv \gamma \rightarrow \gamma$$
$$Q_v \equiv \gamma \rightarrow P\ \gamma\ v$$
$$A_v \equiv P\ \gamma\ v \rightarrow \gamma$$

$$.^s : S \to S \to S$$
$$.^q : S \to Q_v \to Q_v$$
$$.^a : Q_v \to A_v \to S$$

$$.^s = \underbrace{\lambda S_1^{\gamma\to\gamma} S_2^{\gamma\to\gamma} c^{\gamma} . \overbrace{S_2(\overbrace{S_1\ c}^{\gamma})}^{\gamma}}_{(\gamma\to\gamma)\to(\gamma\to\gamma)\to\gamma\to\gamma}$$

$$.^q = \underbrace{\lambda S^{\gamma\to\gamma} Q_v^{\gamma\to P\gamma v} c^{\gamma} . \overbrace{Q(\overbrace{S\ c}^{\gamma})}^{P\gamma v}}_{(\gamma\to\gamma)\to(\gamma\to P\ \gamma\ v)\to\gamma\to P\ \gamma\ v}$$

$$.^a = \underbrace{\lambda Q_v^{\gamma\to P\gamma v} A_v^{P\gamma v\to\gamma} c^{\gamma} . \overbrace{A_v(\overbrace{Q_v\ c}^{P\gamma v})}^{\gamma}}_{(\gamma\to P\ \gamma\ v)\to(P\ \gamma\ v\to\gamma)\to\gamma\to\gamma}$$

$$.^s : (\gamma \to \gamma) \to (\gamma \to \gamma) \to \gamma \to \gamma$$
$$.^q : (\gamma \to \gamma) \to (\gamma \to P\ \gamma\ v) \to \gamma \to P\ \gamma\ v$$
$$.^a : (\gamma \to P\ \gamma\ v) \to (P\ \gamma\ v \to \gamma) \to \gamma \to \gamma$$