CONSTRAINT-BASED COMPOSITIONAL SEMANTICS

VAN DEN BROECK, WOUTER J. M.

Sony Computer Science Laboratory, 6, rue Amyot,
Paris, 75005, France
wouter@csl.sony.fr

Abstract

The ability to interpret, conceive and learn composite meaning is a prerequisite for language use. Any computational model of the emergence and evolution of grammar needs to consider the emergence and evolution of such meaning. It needs to explain how interpretation deals with semantic ambiguity, how rich meaning is conceptualised, and how it is acquired and conventionalised. Various computational models have been proposed that deal with one or a few of these aspects. It is however hard to integrate them given the diverse and often hard to align underlying conceptual and computational metaphors and paradigms. To remedy this we propose a constraint-based model of compositional semantics which affords a uniform and grounded treatment of its interpretation, its conceptualisation and its acquisition.

1. Conceptual inventories

A model of the emergence and evolution of grounded meaning cannot simply assume the availability of a predicate-based world model. Even the vocabulary of meaning predicates – the conceptual repertoire – has to emerge and become conventionalised. Furthermore, in a grounded treatment of interpretation, the satisfiability of a meaning predicate is generally not a simple true-or-false affair.

Earlier work (Steels, 2000) showed how the formation and conventionalisation of sensorially grounded conceptual repertoires can be computationally modelled. These models generally involve mechanisms for partitioning some continuous sensory data space, where each partition corresponds to a concept. The actual partitioning is not pre-defined by e.g. a machine learning scheme, but rather built up in a constructivist manner. Each agent gradually constructs both a conceptual repertoire and a lexicon, which maps the concepts on arbitrary forms that can be used to represent them.

The successfulness of a communicative system relies on a sufficient alignment of both the conceptual repertoire and the lexicon among the population. Attaining such alignment is not a trivial problem since agents cannot directly communicate the nature of the concepts, i.e. the specifics of the underlying grounded representation. They have to make do with arbitrary symbols and gradually align the repertoire and lexicon through language use.

The conventionalisation of the repertoire and lexicon emerges from a number of local, situated interactions. In each of these interactions the participating agents play a language game and if necessary attempt to locally co-ordinate the relevant entries in their repertoire and lexicon. This coordination necessarily involves some hypothesising on what either the intended or interpreted meaning could be, and modification of the existing inventories accordingly.

2. Semantic primitives

The grounding of a conceptual repertoire can be realised using diverse techniques. Steels (1996) for instance uses *discrimination trees* in a generic model of perceptually grounded concept creation. Siskind (2001) uses *force dynamics and event logic* for grounding the lexical semantics of verbs. Belpaeme and Bleys (2005) use point representations of prototypes in their model of the formation of colour categories, while *radial basis function networks* were used in (Steels & Belpaeme, 2005). Countless other techniques are or could be considered.

The meaning of even the simplest of sentences involves various types of concepts. Rather than looking for a single technique that can deal with all types, we propose a system in which diverse techniques, each well suited for a particular type of concept, co-operate in a coherent framework. Each technique is therefore embedded in a semantic primitive. These encapsulate the procedural details and provide a uniform, abstract interface to the underlying, concept-specific mechanisms that deal with interpretation, conceptualisation and learning. Note that we consider these primitives to represent general cognitive capabilities that are recruited for dealing with linguistic needs. As such we assume them as given and do not consider their evolutionary origin.

Concepts such as categories do not refer to particular entities or phenomena in the world. They are not interpreted 'by themselves' but rather act as arguments for cognitive operations that fulfil certain functional needs. Most of our current experiments focus on discriminating one or more entities from the observed world. Consequently most of the primitives we currently consider somehow contribute to these goals. The bulk of these are what we refer to as *filtering* constraints. They take for example a category as arguments and filter from some source-set those entities for which the category does not apply. This operation thus takes two arguments, a category and a source-set, and produces a target-set.

During conceptualisation, the operation of the primitive is inverted. The goal is now to find a category that accounts for the filtering from a given source-set to a given target-set, which is a subset of the source-set. A suitable category is either found in the existing inventory, or invented and added to it. This operation thus takes a source-set and a target-set as arguments, and produces a category (or potentially a number of categories).

A semantic primitive thus establishes a relationship between a number of arguments. Unlike a function however, there exists more than one way in which some argument(s) can be inferred from the others. Such an omnidirectional relationship can be naturally modeled as a *constraint* as it is understood in artificial intelligence and operations research. Equally important, modelling the primitives as constraints also brings to the table a well understood framework for combining constraints in a constraint network and a rich range of techniques for solving so-called *constraint satisfaction problems* (Montanari, 1974; Freuder & Mackworth, 1994; Dechter, 2003).

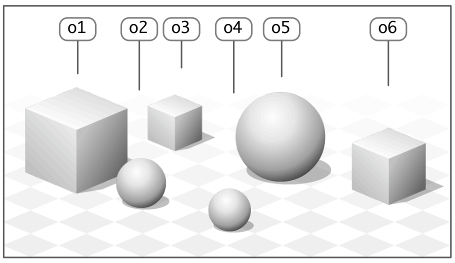
Concepts are traditionally considered to be the predicates. The semantic primitives we consider can however take concepts as arguments. We could thus say that our approach amounts to a second-order semantics.

3. Compositional semantics

Language users routinely combine diverse semantic primitives in rich meaning assemblies. By modelling the semantic primitives as constraints we can model these semantic compositions as constraint networks. Constraints are linked together in such networks by specifying them over a shared set of variables. The specification of constraint networks is intrinsically declarative. It specifies which primitives are used and how they are linked together. It does not, however, specify a particular control or data flow. The flow in fact adapts to the availability of bindings for the involved variables.

Let's consider an example. Figure 1 shows a labelled sensory context with a number of balls and boxes of varying size. The objective is to devise a semantic composition that discriminates one of the objects.

Figure 1. labelled sensory context example



Assume we have at our disposal a couple of simple semantic primitives. One of these primitives is a filtering constraint as described above and filters in terms of basic shape prototypes, i.e. balls or boxes. It is called *filter-set-prototype* and it has three slots: *filter-set-prototype(target-set, source-set, prototype)*.

A second filtering primitive – *filter-set-average(target-set, source-set, comparison)* – takes the average of the values of some numerical feature vector and filters those objects for which the respective feature value is for example smaller than the average. The comparison argument identifies a feature channel and a compare function. The comparison with channel 'size' and function $>$ would for instance correspond to the concept BIG.

3.1. *Example A*

For our first example we take object $o5$ in the above scene as the topic. This topic cannot be unambiguously identified by utterances such as "the ball" or "the big (thing)" since there are many balls and an equally big box. "The big ball" would do however, but requires the combined use of both aforementioned semantic primitives.

We actually use two more primitives. The first is *unique-element(object, set)* and asserts that the filler of the *set* slot is a set that contains one element; the filler of the *object* slot. It is used to cover the uniqueness of the topic. The other primitive is *equal-to-context(set)*, which simply asserts that the filler of the *set* slot equals the set of objects in the sensory context.

Combining these four constraints in a suitable composite meaning, yields the constraint network shown in figure 2. All slots are linked to variables. The four constraints are networked by linking slots of different constraints to the same variables, as is the case for the variables *context*, *set-1* and *set-2*.

Figure 2. Semantic program A

```
{ equal-to-context(context),
  filter-set-prototype(set-1, context, prototype),
  filter-set-average(set-2, set-1, comparison),
  unique-element(topic, set-2) }
```

Let's assume that the grammatical parsing of an utterance such as "the big ball" yields this semantic program plus the bindings: *prototype* $\leftarrow$ BALL and *comparison* $\leftarrow$ BIG, which are returned by the lexical look-up of "ball" and "big" respectively. No other variable can be bound for now.

The full interpretation of this composite meaning is obtained by resolving the constraint network, i.e. finding the set of bindings, one for each variable in the network, that does not violate any of the constraints in the network, or in other words, solving the constraint satisfaction problem. We do so by means of constraint propagation.

First the *equal-to-context* constraint binds the *context* variable to the complete set of objects in the scene, i.e. $\{o1, o2, o3, o4, o5, o6\}$. Given the bindings for both the *context* and *prototype* variables, the *filter-set-prototype* constraint can infer a binding for *set-1*, i.e. the set of ball-like objects: $\{o2, o4, o5\}$. Given this binding and the comparison, the *filter-set-average* constraint can now infer a binding for *set-2*, i.e. $\{o5\}$, since $o5$ is larger than the average size of the three balls. Finally, *unique-element* can correctly bind *topic* to $\{o5\}$, as such yielding the intended topic.

3.2. *Learning*

Say we hear "the froople ball" but do not know the meaning of "froople". If we signal our misunderstanding to the speaker, and the speaker manages to draw our attention to the intended topic through other means, such as pointing, an opportunity for learning presents itself. We take the same semantic composition and fill in the known bindings: *prototype* $\leftarrow$ BALL and *topic* $\leftarrow$ $o5$. This network can now be resolved.

Applying the *unique-element* constraint gives the binding *set-2* $\leftarrow \{o5\}$. Applying the *equal-to-context* and *filter-set-prototype* constraints gives *set-1* $\leftarrow \{o2, o4, o5\}$. Given these bindings the *filter-set-average* primitive can try to abduct a comparison that could account for the filtering from the *set-1* to *set-2*. The comparison with channel 'size' and function $>$ clearly applies. If this concept already exists in the inventory, a new hypothetical entry between this concept and the form "froople" can be added in the lexicon. If it was not conceptualised before, it can also be added in the conceptual inventory.

3.3. *Example B*

As a second example we will assume the same context, but take $o2$ as the topic. We cannot easily find a semantic program that discriminates this topic using the same constraints as before. Let's therefore introduce an additional semantic primitive: *filter-set-relation(target-set, source-set, relation, referent)*. This primitive filters all elements from the source-set for which the relation does not apply with respect to the referent. The relations we consider here are spatial relations, such as NEXT-TO, or IN-FRONT-OF. This enables us to construct the semantic composition that corresponds to "the ball next to the big box", which properly discriminates the intended topic. The resulting composition is shown in figure 3.

Figure 3. Semantic program B

```
{ equal-to-context(context),
  filter-set-prototype(set-1, context, proto-1),
  filter-set-average(set-2, set-1, comparison),
  unique-element(referent, set-2),
  filter-set-prototype(set-3, context, proto-2),
  filter-set-relation(set-4, set-3, relation, referent),
  unique-element(topic, set-4) }
```

For a regular interpretation the bindings are: *proto-1* $\leftarrow$ BOX, *comparison* $\leftarrow$ BIG, *proto-1* $\leftarrow$ BALL, and *relation* $\leftarrow$ NEXT-TO. Resolving the constraint network will first bind *referent* to $o1$ like in the previous example, and *set-3* to the set of balls, i.e. $\{o2, o4, o5\}$. Given these bindings the *filter-set-relation* primitive can now select from *set-3* those elements which are 'next-to' the referent and bind this set, i.e. $\{o2\}$, to *set-4*, giving us the correct topic.

4. Goal-directed construction of semantic compositions

The conceptualisation of the kind of composite meaning discussed above is realised as a process that constructs a constraint network. The input for this process is a communicative goal, e.g. 'discriminate topic X in the sensory context', and an inventory of primitive constraints. The resulting constraint network has to be coherent and fulfil the given goal when interpreted by the hearer. In order for the hearer to be able to properly interpret the decoded composition, all arguments that cannot be inferred should be expressed in the utterance. These *essential* arguments thus have to be representable in language, for instance as lexical forms.

Finding a suitable constraint network given some goal is a combinatorial problem. Blindly trying to link together various constraints in arbitrary configurations and checking if the results satisfy the requirements is not a viable strategy. We propose a structured, goal-directed strategy to manage the combinatorial explosion.

For a semantic composition to be useable, it must be resolvable given the essential arguments. All other bindings in the solution must be directly or indirectly inferable from this select set of bindings. In other words, there must exist a directed, non-cyclic dependency network among the bindings which reflects the inferential flow from the essential source bindings to the binding or bindings that represent or otherwise contribute to the communicative goal. The process of creating an appropriate semantic composition can be guided by this requirement.

Let's for example consider the construction of the semantic composition shown in figure 2. The initial goal is to discriminate object $o5$ from the sensory context shown in figure 1. We start the composition by introducing a variable and bind the topic to it. This binding is meant to be inferable during interpretation, so we need to add a constraint that can infer the binding. Most constraints however hold over more than one variable, which will need to be added. The bindings for these new variables also need to be either essential bindings or be inferable themselves. Introducing a new constraint to fulfil a goal might thus introduce new sub-goals, which need to be fulfilled recursively.

Let's say we add *unique-element(topic, set-2)* to infer the topic. This introduces a new sub-goal: find support for (the binding of) *set-2*. Adding *filter-set-average(set-2, set-1, comparison)* fulfils this sub-goal, but yields two new sub-goals: *set-1* and *comparison*. The comparison concept can be expressed in the utterance, but the set will have to be recursively dealt with.

A complete overview of the composition process is shown in figure 4. Each row represents a step in the process, starting with the initial step in the first row. The first column gives the goal for each step. The second column shows the 'action' taken to fulfil the goal, which is either a new constraint or an argument that has to be expressed in the utterance. The third column lists the sub-goals entailed by adding a constraint. Each of these sub-goals needs to be fulfilled in one of the subsequent rows.

Figure 4. Goal directed composition

goal	**constraint or argument**	**subgoals**
topic	unique-element(topic, set-2)	set-2
set-2	filter-set-average(set-2, set-1, comparison)	set-1, comparison
comparison	BIG	-
set-1	filter-set-prototype(set-1, context, prototype)	context, prototype
prototype	BALL	-
context	equal-to-context(context)	-

The composition process starts with the initial goal and ends when all the sub-goals that were introduced along the way, are fulfilled. For each goal there might be several constraints that could infer that goal. The composition shown in figure 4 thus represents but one particular path of potentially many. All these paths form a tree. Various strategies can be used to more efficiently explore this tree. We for instance apply an eager search strategy based on a heuristic that favours smaller compositions, with less unfulfilled goals and a smaller amount of essential arguments. We prune branches that involve a cyclic dependency and try to prune inconsistent branches as soon as possible by propagating the constraints where possible after each extension.

Finally we would like to note that this composition mechanism can also deal with situations in which the structure of the semantic composition was not fully understood. It can be used to hypothesise on a plausible completion of an incomplete network by adding constraints to account for bindings not yet accounted for in exactly the same way as outlined before.

5. Conclusions

In this paper we showed how modelling compositional semantics in terms of constraints and constraint networks, offers a uniform framework for dealing with its interpretation, acquisition and conceptualisation.

Grounded semantic primitives not only perform cognitive operations like categorising a set of visual stimuli in terms of categories, but also extend the conceptual inventory. This way the acquisition of a conceptual inventory is completely integrated in the process of conceptualising and interpreting language. It is therefore possible to have a strong interaction between the two.

Encapsulating the procedural details of these cognitive operations as primitives that expose a uniform interface, allows experimenters to combine diverse techniques for handling specific concept types, instead of being forced to apply a one-size-fits-all scheme. The representation of semantic compositions as constraint networks does not entail the specification of a particular control or data flow. This not only makes a closer fit to natural languages, but also allows for data flows to adapt to the differences in availability of information in conceptualisation, interpretation or learning.

Finally the proposed model does not favour any particular model or formalism concerning the emergence and evolution of language in general, or syntax in particular. It should thus be adoptable in a wide array of experimental and theoretical settings, while drawing upon a well-developed body of knowledge on constraint processing from the fields of artificial intelligence and operations research.

Acknowledgements

This research is supported by Sony Computer Science Laboratory in Paris and the ECAGENTS project funded by the Future and Emerging Technologies programme (IST-FET) of the European Community under EU R&D contract IST-2003-1940. It builds on the work first introduced in Steels (2000) and elaborated on in Steels and Bleys (2005).

References

Belpaeme, T., & Bleys, J. (2005). Explaining universal color categories through a constrained acquisition process. *Adaptive Behavior*, *13*(4), 293–310.

Dechter, R. (2003). *Constraint processing.* Morgan Kaufmann.

Freuder, E. C., & Mackworth, A. K. (1994). *Constraint-based reasoning.* MIT Press.

Montanari, U. (1974). Networks of constraints: Fundamental properties and application to picture processing. *Information Science*, *7*, 95–132.

Siskind, J. M. (2001). Grounding the lexical semantics of verbs in visual perception using force dynamics and event logic. *Journal of Artificial Intelligence Research (JAIR)*, *15*, 31-90.

Steels, L. (1996). Perceptually grounded meaning creation. In M. Tokoro (Ed.), *Icmas96.* AAAI Press.

Steels, L. (2000). The emergence of grammar in communicating autonomous robotic agents. In W. Horn (Ed.), *Ecai2000* (pp. 764–769). Amsterdam: IOS Press.

Steels, L., & Belpaeme, T. (2005). Coordinating perceptually grounded categories through language: A case study for colour. *Behavioral and Brain Sciences*, *28*(4), 469–89.

Steels, L., & Bleys, J. (2005). Planning what to say: Second order semantics for fluid construction grammars. In A. Bugarin Diz & J. S. Reyes (Eds.), *Proceedings of caepia '05. lecture notes in ai.* Berlin: Springer Verlag.

Steels, L., Kaplan, F., McIntyre, A., & Looveren, J. van. (2000). Crucial factors in the origins of word-meaning. In J.-L. Dessalles & L. Ghadakpour (Eds.), *Proceedings of the 3rd evolution of language conference* (pp. 214–217). Paris: ENST 2000 S 002.