Higher–Order Coloured Unification and Natural Language Semantics

Claire Gardent
Computational Linguistics
Universität des Saarlandes
D–Saarbrücken
claire@coli.uni-sb.de

Michael Kohlhase
Computer Science
Universität des Saarlandes
D–Saarbrücken
kohlhase@cs.uni-sb.de

Abstract

In this paper, we show that Higher–Order Coloured Unification – a form of unification developed for automated theorem proving – provides a general theory for modeling the interface between the interpretation process and other sources of linguistic, non semantic information. In particular, it provides the general theory for the Primary Occurrence Restriction which (Dalrymple et al., 1991)'s analysis called for.

1 Introduction

It is well known that Higher–Order Unification (HOU) can be used to construct the semantics of Natural Language: (Dalrymple et al., 1991) – henceforth, DSP – show that it allows a treatment of VP–Ellipsis which successfully captures the interaction of VPE with quantification and nominal anaphora; (Pulman, 1995; Gardent and Kohlhase, 1996) use HOU to model the interpretation of focus and its interaction with focus sensitive operators, adverbial quantifiers and second occurrence expressions; (Gardent et al., 1996) shows that HOU yields a simple but precise treatment of corrections; Finally, (Pinkal, 1995) uses linear HOU to reconstruct underspecified semantic representations.

However, it is also well known that the HOU approach to NL semantics systematically over–generates and that some general theory of the interface between the interpretation process and other sources of linguistic information is needed in order to avoid this.

In their treatment of VP–ellipsis, DSP introduce an informal restriction to avoid over–generation: the *Primary Occurrence Restriction* (POR). Although this restriction is intuitive and linguistically well–motivated, it does not provide a general theoretical framework for extra–semantic constraints.

In this paper, we argue that *Higher–Order Coloured Unification* (HOCU, (cf. sections 3,6), a restricted form of HOU developed independently for theorem proving, provides the needed general framework. We start out by showing that the HOCU approach allows for a precise and intuitive modeling of DSP's Primary Occurrence Restriction (cf. section 3.1). We then show that the POR can be extended to capture linguistic restrictions on other phenomena (focus, second occurrence expressions and adverbial quantification) provided that the notion of *primary occurrence* is suitably adjusted (cf. section 4). Obviously a treatment of the interplay of these phenomena and their related notion of primary occurrence is only feasible given a precise and well–understood theoretical framework. We illustrate this by an example in section 4.4. Finally, we illustrate the generality of the HOCU framework by using it to encode a completely different constraint, namely Kratzer's binding principle (cf. section 5).

2 Higher–Order Unification and NL semantics

The basic idea underlying the use of HOU for NL semantics is very simple: the typed λ–calculus is used as a semantic representation language while semantically under–specified elements (e.g. anaphors and ellipses) are represented by free variables whose value is determined by solving higher–order equations. For instance, the discourse (1a) has (1b) as a semantic representation where the value of R is given by equation (1c) with solutions (1d) and (1e).

(1) a. *Dan likes golf. Peter does too.*
b. $like(dan,golf) \wedge R(peter)$
c. $like(dan,golf) = R(dan)$
d. $R = \lambda x.\ like(x,golf)$
e. $R = \lambda x.\ like(dan,golf)$

The process of solving such equations is traditionally called unification and can be stated as follows:

given two terms M and N, find a substitution of terms for free variables that will make M and N equal. For first order logic, this problem is decidable and the set of solutions can be represented by a single most general unifier. For the typed λ–calculus, the problem is undecidable, but there is an algorithm which – given a solvable equation – will enumerate a complete set of solutions for this equation (Huet, 1975).

Note that in (1), unification yields a linguistically valid solution (1d) but also an invalid one: (1e). To remedy this shortcoming, DSP propose an informal restriction, the **Primary Occurrence Restriction**:

> Given a labeling of occurrences as either primary or secondary, the POR excludes of the set of linguistically valid solutions, any solution which contains a primary occurrence.

Here, a **primary occurrence** is an occurrence that is *directly associated* with a *source parallel element.* Neither the notion of direct association, nor that of parallelism is given a formal definition; but given an intuitive understanding of these notions, a **source parallel element** is an element of the source (i.e. antecedent) clause which has a parallel counterpart in the target (i.e. elliptic or anaphoric) clause.

To see how this works, consider example (1) again. In this case, *dan* is taken to be a primary occurrence because it represents a source parallel element which is neither anaphoric nor controlled i.e. it is directly associated with a source parallel element. Given this, equation (1c) becomes (2a) with solutions (2b) and (2c) (primary occurrences are underlined). Since (2c) contains a primary occurrence, it is ruled out by the POR and is thus excluded from the set of linguistically valid solutions.

(2) a. $like(\underline{dan}, golf) = R(dan)$
b. $R = \lambda x.like(x, golf)$
c. $R = \lambda x.like(\underline{dan}, golf)$

Although the intuitions underlying the POR are clear, two main objections can be raised. First, the restriction is informal and as such provides no good basis for a mathematical and computational evaluation. As DSP themselves note, a general theory for the POR is called for. Second, their method is a generate–and–test method: all logically valid solutions are generated before those solutions that violate the POR and are linguistically invalid are eliminated. While this is sufficient for a theoretical analysis, for actual computation it would be preferable never to produce these solutions in the first place. In what follows, we present a unification framework which solves both of these problems.

3 Higher–Order Coloured Unification (HOCU)

There is a restricted form of HOU which allows for a natural modeling of DSP's Primary Occurrence Restriction: Higher–Order Coloured Unification developed independently for theorem proving (Hutter and Kohlhase, 1995). This framework uses a variant of the simply typed λ-calculus where symbol occurrences can be annotated with so-called *colours* and substitutions must obey the following constraint:

> For any colour constant c and any c–coloured variable V_{c}, a well–formed coloured substitution must assign to V_{c} a c–monochrome term i.e., a term whose symbols are c–coloured.

3.1 Modeling the Primary Occurrence Restriction

Given this coloured framework, the POR is directly modelled as follows: Primary occurrences are pe-coloured whilst free variables are $\neg\mathsf{pe}$-coloured. For the moment we will just consider the colours pe (primary for ellipsis) and $\neg\mathsf{pe}$ (secondary for ellipsis) as distinct basic colours to keep the presentation simple. Only for the analysis of the interaction of e.g. ellipsis with focus phenomena (cf. section 4.4) do we need a more elaborate formalization, which we will discuss there.

Given the above restriction for well–formed coloured substitutions, such a colouring ensures that any solution containing a primary occurrence is ruled out: free variables are $\neg\mathsf{pe}$-coloured and must be assigned a $\neg\mathsf{pe}$-monochrome term. Hence no substitution will ever contain a primary occurrence (i.e. a pe-coloured symbol). For instance, discourse (1a) above is assigned the semantic representation (3a) and the equation (3b) with unique solution (3c). In contrast, (3d) is not a possible solution since it assigns to an $\neg\mathsf{pe}$-coloured variable, a term containing a pe-coloured symbol i.e. a term that is not $\neg\mathsf{pe}$-monochrome.

(3) a. $like(dan_{\mathsf{pe}}, golf) \wedge R_{\neg\mathsf{pe}}(peter)$
b. $like(dan_{\mathsf{pe}}, golf) = R_{\neg\mathsf{pe}}(dan_{\mathsf{pe}})$
c. $R_{\neg\mathsf{pe}} = \lambda x.like(x, golf)$
d. $R_{\neg\mathsf{pe}} = \lambda x.like(dan_{\mathsf{pe}}, golf)$

3.2 HOCU theory

To be more formal, we presuppose a finite set $\mathcal{C} = \{\mathsf{a}, \mathsf{b}, \mathsf{c}, \mathsf{pe}, \neg\mathsf{pe}, \ldots\}$ of **colour constants** and a countably infinite supply $\mathcal{CV} = \{\mathsf{A}, \mathsf{B}, \ldots\}$ of **colour variables**.

As usual in λ-calculus, the set *wff* of **well-formed formulae** consists of (coloured[1]) constants $c_{\mathsf{a}}, runs_{\mathsf{b}}, runs_{\mathsf{A}}, \ldots$, (possibly uncoloured) variables $x, x_{\mathsf{a}}, y_{\mathsf{b}}, \ldots$ (function) **applications** of the form MN and λ-abstractions of the form $\lambda x.M$. Note that only variables without colours can be abstracted over. We call a formula M **c-monochrome**, if all symbols in M are bound or tagged with c.

We will need the so-called **colour erasure** $|M|$ of M, i.e. the formula obtained from M by erasing all colour annotations in M. We will also use various elementary concepts of the λ-calculus, such as **free** and **bound** occurrences of variables or substitutions without defining them explicitly here. In particular we assume that free variables are coloured in all formulae occuring. We will denote the substitution of a term N for all free occurrences of x in M with $[N/x]M$.

It is crucial for our system that colours annotate symbol occurrences (i.e. colours are not sorts!), in particular, it is intended that different occurrences of symbols carry different colours (e.g. $f(x_{\mathsf{b}}, x_{\mathsf{a}})$) and that symbols that carry different colours are treated differently. This observation leads to the notion of coloured substitutions, that takes the colour information of formulae into account. In contrast to traditional (uncoloured) substitutions, a coloured substitution σ is a pair $\langle \sigma^t, \sigma^c \rangle$, where the **term substitution σ^t** maps coloured variables (i.e. the pair x_{c} of a variable x and the colour c) to formulae of the appropriate type and the **colour substitution σ^c** maps colour variables to colours. In order to be legal (a **$\mathcal{C}$-substitution**) such a mapping σ must obey the following constraints:

- If a and b are different colours, then $|\sigma(x_{\mathsf{a}})| = |\sigma(x_{\mathsf{b}})|$, i.e. the colour erasures have to be equal.
- If $\mathsf{c} \in \mathcal{C}$ is a colour constant, then $\sigma(x_{\mathsf{c}})$ is c-monochrome.

The first condition ensures that the colour erasure of a $\mathcal{C}$-substitution is a well-defined classical substitution of the simply typed λ-calculus. The second condition formalizes the fact that free variables with constant colours stand for monochrome subformulae, whereas colour variables do not constrain the substitutions. This is exactly the trait, that we will exploit in our analysis.

Note that $\beta\eta$-reduction in the coloured λ-calculus is just the classical notion, since the bound variables do not carry colour information. Thus we have all the known theoretical results, such as the fact that $\beta\eta$-reduction always terminates producing unique normal forms and that $\beta\eta$-equality can be tested by reducing to normal form and comparing for syntactic equality. This gives us a decidable test for *validity* of an equation.

In contrast to this, higher-order unification tests for *satisfiability* by finding a substitution σ that makes a given equation $M = N$ valid ($\sigma(M) =_{\beta\eta} \sigma(N)$), even if the original equation is not ($M \neq_{\beta\eta} N$). In the coloured λ-calculus the space of (semantic) solutions is further constrained by requiring the solutions to be $\mathcal{C}$-substitutions. Such a substitution is called a **$\mathcal{C}$-unifier** of M and N. In particular, $\mathcal{C}$-unification will only succeed if comparable formulae have unifiable colours. For instance, $intro_{\mathsf{a}}(p_{\mathsf{a}}, j_{\mathsf{b}}, x_{\mathsf{a}})$ unifies with $intro_{\mathsf{a}}(y_{\mathsf{a}}, j_{\mathsf{A}}, s_{\mathsf{a}})$ but not with $intro_{\mathsf{a}}(p_{\mathsf{a}}, j_{\mathsf{a}}, s_{\mathsf{a}})$ because of the colour clash on j.

It is well-known, that in first-order logic (and in certain related forms of feature structures) there is always a most general unifier for any equation that is solvable at all. This is not the case for higher-order (coloured) unification, where variables can range over functions, instead of only individuals. Fortunately, in our case we are not interested in general unification, but we can use the fact that our formulae belong to very restricted syntactic subclasses, for which much better results are known. In particular, the fact that free variables only occur on the left hand side of our equations reduces the problem of finding solutions to higher-order matching, of which decidability has been proven for the subclass of third-order formulae (Dowek, 1992) and is conjectured for the general case. This class, (intuitively allowing only nesting functions as arguments up to depth two) covers all of our examples in this paper. For a discussion of other subclasses of formulae, where higher-order unification is computationally feasible see (Prehofer, 1994).

Some of the equations in the examples have multiple most general solutions, and indeed this multiplicity corresponds to the possibility of multiple different interpretations of the focus constructions. The role of colours in this is to restrict the logically possible solutions to those that are linguistically sound.

---

[1] Colours are indicated by subscripts labeling term occurrences; whenever colours are irrelevant, we simply omit them.

4 Linguistic Applications of the POR

In section 3.1, we have seen that HOCU allowed for a simple theoretical rendering of DSP's Primary Occurrence Restriction. But isn't this restriction fairly idiosyncratic? In this section, we show that the restriction which was originally proposed by DSP to model VP–ellipsis, is in fact a very general constraint which far from being idiosyncratic, applies to many different phenomena. In particular, we show that it is necessary for an adequate analysis of focus, second occurrence expressions and adverbial quantification.

Furthermore, we will see that what counts as a primary occurrence differs from one phenomenon to the other (for instance, an occurrence directly associated with focus counts as primary w.r.t focus semantics but not w.r.t to VP–ellipsis interpretation). To account for these differences, some machinery is needed which turns DSP's intuitive idea into a fully–blown theory. Fortunately, the HOCU framework is just this: different colours can be used for different types of primary occurrences and likewise for different types of free variables. In what follows, we show how each phenomenon is dealt with. We then illustrate by an example how their interaction can be accounted for.

4.1 Focus

Since (Jackendoff, 1972), it is commonly agreed that focus affects the semantics and pragmatics of utterances. Under this perspective, **focus** is taken to be the semantic value of a prosodically prominent element. Furthermore, focus is assumed to trigger the formation of an additional semantic value (henceforth, the **Focus Semantic Value** or FSV) which is in essence the set of propositions obtained by making a substitution in the focus position (cf. e.g. (Kratzer, 1991)). For instance, the FSV of (4a)[2] is (4b), the set of formulae of the form *l(j,x)* where x is of type e, and the pragmatic effect of focus is to presuppose that the denotation of this set is under consideration.

(4) a. *Jon likes SARAH*
b. $\{l(j,x) \mid x \in \mathit{wff}_e\}$

In (Gardent and Kohlhase, 1996), we show that HOU can successfully be used to compute the FSV of an utterance. More specifically, given (part of) an utterance U with semantic representation Sem and foci $F^1 \dots F^n$, we require that the following equation, the **FSV equation**, be solved:

$$Sem = Gd(F^1)\dots(F^n)$$

On the basis of the Gd value, we then define the FSV, written $\overline{Gd}$, as follows:

Definition 4.1 *(Focus Semantic Value)*
Let Gd be of type $\alpha = \vec{\beta_k} \to t$ *and n be the number of foci* ($n \leq k$)*, then the Focus Semantic Value derivable from Gd, written* $\overline{Gd}$*, is* $\{Gd(t^1 \dots t^n) \mid t^i \in \mathit{wff}_{\beta_i}\}$.

This yields a focus semantic value which is in essence Kratzer's presupposition skeleton. For instance, given (4a) above, the required equation will be $l(j,s) = Gd(s)$ with two possible values for Gd: $\lambda x.l(j,x)$ and $\lambda x.l(j,s)$. Given definition (4.1), (4a) is then assigned two FSVs namely

(5) a. $\overline{Gd} = \{l(j,x) \mid x \in \mathit{wff}_e\}$
b. $\overline{Gd} = \{l(j,s) \mid x \in \mathit{wff}_e\}$

That is, the HOU treatment of focus over–generates: (5a) is an appropriate FSV, but not (5b). Clearly though, the POR can be used to rule out (5b) if we assume that occurrences that are directly associated with a focus are primary occurrences. To capture the fact that those primary occurrences are different from DSP's primary occurrences when dealing with ellipsis, we colour occurrences that are directly associated with focus (rather than a source parallel element in the case of ellipsis) pf. Consequently, we require that the variable representing the FSV be $\neg\mathsf{pf}$ coloured, that is, its value may not contain any pf term. Under these assumptions, the equation for (4a) will be (6a) which has for unique solution (6b).

(6) a. $l(j,s_{\mathsf{pf}}) = FSV_{\neg\mathsf{pf}}(s_{\mathsf{pf}})$
b. $FSV_{\neg\mathsf{pf}} = \lambda x.l(j,x)$

4.2 Second Occurrence Expressions

A second occurrence expression (SOE) is a partial or complete repetition of the preceding utterance and is characterised by a de-accenting of the repeating part (Bartels, 1995). For instance, (7b) is an SOE whose repeating part *only likes Mary* is deaccented.

(7) a. *Jon only likes MARY.*
b. *No, PETER only likes Mary.*

In (Gardent, 1996; Gardent et al., 1996) we show that SOEs are advantageously viewed as involving a deaccented anaphor whose semantic representation must unify with that of its antecedent. Formally, this is captured as follows. Let $SSem$ and $TSem$ be the semantic representation of the source and target clause respectively, and $TP^1 \dots TP^n, SP^1 \dots SP^n$

---

[2] Focus is indicated using upper–case.

be the target and source parallel elements[3], then the interpretation of an SOE must respect the following equations:

$$An(SP^1, \ldots, SP^n) = SSem$$
$$An(TP^1, \ldots, TP^n) = TSem$$

Given this proposal and some further assumptions about the semantics of *only*, the analysis of (7b) involves the following equations:

$$\begin{aligned} (8) \quad An(j) &= \forall P[P \in \{\lambda x.like(x,y) \mid y \in \mathit{wff}_e\} \\ &\qquad \wedge P(j) \rightarrow P = \lambda x.like(x,m)] \\ An(p) &= \forall P[P \in FSV \wedge P(p) \\ &\qquad \rightarrow P = \lambda x.like(x,m)] \end{aligned}$$

Resolution of the first equation then yields two solutions:

$$\begin{aligned} An &= \lambda z \forall P[P \in \{\lambda x.like(x,y) \mid y \in \mathit{wff}_e\} \\ &\qquad \wedge P(z) \rightarrow P = \lambda x.like(x,m)] \\ An &= \lambda z \forall P[P \in \{\lambda x.like(x,y) \mid y \in \mathit{wff}_e\} \\ &\qquad \wedge P(j) \rightarrow P = \lambda x.like(x,m)] \end{aligned}$$

Since An represents the semantic information shared by target and source clause, the second solution is clearly incorrect given that it contains information (j) that is specific to the source clause. Again, the POR will rule out the incorrect solutions, whereby contrary to the VP–ellipsis case, all occurrences that are directly associated with parallel elements (i.e. not just source parallel elements) are taken to be primary occurrences. The distinction is implemented by colouring all occurrences that are directly associated with parallel element ps, whereas the corresponding free variable (An) is coloured as ¬ps. Given these constraints, the first equation in (8) is reformulated as:

$$\begin{aligned} An_{\neg\mathsf{ps}}(j_{\mathsf{ps}}) &= \forall P[P \in \{\lambda x.like(x,y) \mid y \in \mathit{wff}_e\} \\ &\qquad \wedge P(j_{\mathsf{ps}}) \rightarrow P = \lambda x.like(x,m)] \end{aligned}$$

with the unique well–coloured solution

$$\begin{aligned} An_{\neg\mathsf{ps}} &= \lambda z.\forall P[P \in \{\lambda x.like(x,y) \mid y \in \mathit{wff}_e\} \\ &\qquad \wedge P(z) \rightarrow P = \lambda x.like(x,m)] \end{aligned}$$

4.3 Adverbial quantification

Finally, let us briefly examine some cases of adverbial quantification. Consider the following example from (von Fintel, 1995):

Tom always takes SUE to Al's mother.
Yes, and he always takes Sue to JO's mother.

In (Gardent and Kohlhase, 1996), we suggest that such cases are SOEs, and thus can be treated as involving a deaccented anaphor (in this case, the anaphor *he always takes Sue to _'s mother*). Given some standard assumptions about the semantics of *always*, the equations constraining the interpretation An of this anaphor are:

$$\begin{aligned} An(al) &= always \quad (\textit{Tom take } x \textit{ to al's mother}) \\ &\qquad\qquad\ (\textit{Tom take Sue to al's mother}) \\ An(jo) &= always \quad FSV \\ &\qquad\qquad\ (\textit{Tom take Sue to Jo's mother}) \end{aligned}$$

Consider the first equation. If An is the semantics shared by target and source clause, then the only possible value for An is

$$\begin{aligned} \lambda z.always \quad &(\textit{Tom take } x \textit{ to } z\textit{'s mother}) \\ &(\textit{Tom take Sue to } z\textit{'s mother}) \end{aligned}$$

where both occurrences of the parallel element m have been abstracted over. In contrast, the following solutions for An are incorrect.

$$\begin{aligned} \lambda z.always \quad &(\textit{Tom take } x \textit{ to al's mother}) \\ &(\textit{Tom take Sue to } z\textit{'s mother}) \\ \lambda z.always \quad &(\textit{Tom take } x \textit{ to al's mother}) \\ &(\textit{Tom take Sue to al's mother}) \\ \lambda z.always \quad &(\textit{Tom take } x \textit{ to } z\textit{'s mother}) \\ &(\textit{Tom take Sue to al's mother}) \end{aligned}$$

Once again, we see that the POR is a necessary restriction: by labeling as primary, all occurrences representing a parallel element, it can be ensured that only the first solution is generated.

4.4 Interaction of constraints

Perhaps the most convincing way of showing the need for a theory of colours (rather than just an informal constraint) is by looking at the interaction of constraints between various phenomena. Consider the following discourse

(9) a. *Jon likes SARAH*
b. *Peter does too*

Such a discourse presents us with a case of interaction between ellipsis and focus thereby raising the question of how DSP' POR for ellipsis should interact with our POR for focus.

As remarked in section 3.1, we have to interpret the colour ¬pe as the concept of being not primary for ellipsis, which includes pf (primary for focus). In order to make this approach work formally, we have to extend the supply of colours by allowing boolean combinations of colour constants. The semantics of these ground colour formula is that of propositional logic, where ¬d is taken to be equivalent to the disjunction of all other colour constants.

Consequently we have to generalize the second condition on $\mathcal{C}$-substitutions

- For all colour annotations d of symbols in $\sigma(x_{\mathsf{c}})$ $d \models c$ in propositional logic.

Thus $X_{\neg\mathsf{d}}$ can be instantiated with any coloured formula that does not contain the colour d. The

---

[3] As in DSP, the identification of parallel elements is taken as given.

HOCU algorithm is augmented with suitable rules for boolean constraint satisfaction for colour equations.

The equations resulting from the interpretation of (9b) are:

$$l(j_{\mathsf{pe}}, s_{\mathsf{pf}}) = R_{\neg\mathsf{pe}}(j_{\mathsf{pe}})$$
$$R_{\neg\mathsf{pe}}(p) = FSV_{\neg\mathsf{pf}}(F)$$

where the first equation determines the interpretation of the ellipsis whereas the second fixes the value of the FSV. Resolution of the first equation yields the value $\lambda x.l(x, s_{\mathsf{pf}})$ for $R_{\neg\mathsf{pe}}$. As required, no other solution is possible given the colour constraints; in particular $\lambda x.l(j_{\mathsf{pe}}, s_{\mathsf{pf}})$ is not a valid solution. The value of $R_{\neg\mathsf{pe}}(j_{\mathsf{pe}})$ is now $l(p_{\mathsf{pe}}, s_{\mathsf{pf}})$ so that the second equation is[4]:

$$l(p, s_{\mathsf{pf}}) = FSV_{\neg\mathsf{pf}}(F)$$

Under the indicated colour constraints, three solutions are possible:

$$FSV_{\neg\mathsf{pf}} = \lambda x.l(p, x), F = s_{\mathsf{pf}}$$
$$FSV_{\neg\mathsf{pf}} = \lambda O.O(p), F = \lambda x.l(x, s_{\mathsf{pf}})$$
$$FSV_{\neg\mathsf{pf}} = \lambda X.X, F = l(p, s_{\mathsf{pf}})$$

The first solution yields a *narrow focus* reading (only *SARAH* is in focus) whereas the second and the third yield *wide focus* interpretations corresponding to a VP and an S focus respectively. That is, not only do colours allow us to correctly capture the interaction of the two PORs restricting the interpretation of ellipsis of focus, they also permit a natural modeling of focus projection (cf. (Jackendoff, 1972)).

5 Another constraint

An additional argument in favour of a general theory of colours lies in the fact that constraints that are distinct from the POR need to be encoded to prevent HOU analyses from over–generating. In this section, we present one such constraint (the so-called *weak–crossover constraint*) and show how it can be implemented within the HOCU framework.

In essence, the main function of the POR is to ensure that some occurrence occuring in an equation appears as a bound variable in the term assigned by substitution to the free variable occurring in this equation. However, there are cases where the *dual* constraint must be enforced: a term occurrence appearing in an equation must appear unchanged in the term assigned by substitution to the free variable occurring in this equation. The following example illustrates this.

(Chomsky, 1976) observes that focused NPs pattern with quantified and wh–NPs with respect to pronominal anaphora: when the quantified/wh/focused NP precedes and c–commands the pronoun, this pronoun yields an ambiguity between a co-referential and a bound–variable reading. This is illustrated in example

(10) *We only expected HIM$_i$ to claim*
that he$_i$ was brilliant

where the presence of the pronoun *he$_i$* gives rise to two possible FSVs[5]

$$FSV = \{\lambda x.ex(x, y, i) \mid y \in \mathit{wff}_e\}$$
$$FSV = \{\lambda x.ex(x, y, y) \mid y \in \mathit{wff}_e\}$$

thus allowing two different readings: the **corefential** or **strict** reading

$$\forall P[P \in \{\lambda x.ex(x, y, i) \mid y \in \mathit{wff}_e\}$$
$$\wedge\, P(we) \rightarrow P = \lambda x.ex(x, i, i)]$$

and the **bound-variable** or **sloppy** reading.

$$\forall P[P \in \{\lambda x.ex(x, y, y)) \mid y \in \mathit{wff}_e\}$$
$$\wedge\, P(we) \rightarrow P = \lambda x.ex(x, i, i))]$$

In contrast, if the quantified/wh/focused NP does not precede and c–command the pronoun, as in

(11) *We only expected him$_i$ to claim*
that HE$_i$ was brilliant

there is no ambiguity and the pronoun can only give rise to a co-referential interpretation. For instance, given (11) only one reading arises

$$\forall P[P \in \{\lambda x.ex(x, i, y) \mid y \in \mathit{wff}_e\}$$
$$\wedge\, P(we) \rightarrow P = \lambda x.ex(x, i, i)]$$

where the FSV is $\{\lambda x.ex(x, i, y) \mid y \in \mathit{wff}_e\}$.

To capture this data, Government and Binding analyses postulate first, that the antecedent is raised by quantifier raising and second, that pronouns that are c–commanded and preceded by their antecedent are represented either as a λ–bound variable or as a constant whereas other pronouns can only be represented by a constant (cf. e.g. (Kratzer, 1991)'s *binding principle*). Using HOCU, we can model this restriction directly. As before, the focus term is pf- and the FSV variable $\neg\mathsf{pf}$-coloured. Furthermore, we assume that pronouns that are preceded and c–commanded by a quantified/wh/focused antecedent are variable coloured whereas other pronouns are $\neg\mathsf{pf}$-coloured. Finally, all other terms are taken to

---

[4] Note that this equation falls out of our formal system in that it is untyped and thus cannot be solved by the algorithm described in section 6 (as the solutions will show, we have to allow for FSV and F to have different types). However, it seems to be a routine exercise to augment HOU algorithms that can cope with type variables like (Hustadt, 1991; Dougherty, 1993) with the colour methods from (Hutter and Kohlhase, 1995).

[5] We abbreviate $exp(x, cl(y, blt(i)))$ to $ex(x, y, i)$ to increase legibility.

be $\neg\mathsf{pf}$-coloured. Given these assumptions, the representation for (10) is $ex_{\neg\mathsf{pf}}(we_{\neg\mathsf{pf}}, i_{\mathsf{pf}}, i_{\mathsf{A}})$ and the corresponding FSV equation

$$R_{\neg\mathsf{pf}}(i_{\mathsf{pf}}) = \lambda x.ex_{\neg\mathsf{pf}}(x, i_{\mathsf{pf}}, i_{\mathsf{A}})$$

has two possible solutions

$$R_{\neg\mathsf{pf}} = \lambda y.\lambda x.ex_{\neg\mathsf{pf}}(x, y, i_{\neg\mathsf{pf}})$$
$$R_{\neg\mathsf{pf}} = \lambda y.\lambda x.ex_{\neg\mathsf{pf}}(x, y, x)$$

In contrast, the representation for (11) is $ex_{\neg\mathsf{pf}}(we_{\neg\mathsf{pf}}, i_{\neg\mathsf{pf}}, i_{\mathsf{pf}})$ and the equation is

$$R_{\neg\mathsf{pf}}(i_{\mathsf{pf}}) = \lambda x.ex_{\neg\mathsf{pf}}(x, i_{\neg\mathsf{pf}}, i_{\mathsf{pf}})$$

with only one well–coloured solution

$$R_{\neg\mathsf{pf}} = \lambda y.\lambda x.ex_{\neg\mathsf{pf}}(x, i_{\neg\mathsf{pf}}, y)$$

Importantly, given the indicated colour constraints, no other solutions are admissible. Intuitively, there are two reasons for this. First, the definition of coloured substitutions ensures that the term assigned to $R_{\neg\mathsf{pf}}$ is $\neg\mathsf{pf}$-monochrome. In particular, this forces any occurrences of i_{pf} to appear as a bound variable in the value assigned to $R_{\neg\mathsf{pf}}$ whereas i_{A} can appear either as $i_{\neg\mathsf{pf}}$ (a colour variable unifies with any colour constant) or as a bound variable – this in effect models the sloppy/strict ambiguity. Second, a colour constant only unifies with itself. This in effect rules out the bound variable reading in (11): if the $i_{\neg\mathsf{pf}}$ occurrence were to become a bound variable, the value of $R_{\neg\mathsf{pf}}$ would then $\lambda y.\lambda x.ex_{\neg\mathsf{pf}}(x, y, y)$. But then by β–reduction, $R_{\neg\mathsf{pf}}(i_{\mathsf{pf}})$ would be $\lambda x.ex_{\neg\mathsf{pf}}(x, i_{\mathsf{pf}}, i_{\mathsf{pf}})$ which does not unify with the right hand side of the original equation i.e $\lambda x.ex_{\neg\mathsf{pf}}(x, i_{\neg\mathsf{pf}}, i_{\mathsf{pf}})$.

For a more formal account of how the unifiers are calculated see section 6.1.

6 Calculating Coloured Unifiers

Since the HOCU is the principal computational device of the analysis in this paper, we will now try to give an intuition for the functioning of the algorithm. For a formal account including all details and proofs see (Hutter and Kohlhase, 1995).

Just as in the case of unification for first-order terms, the algorithm is a process of recursive decomposition and variable elimination that transform sets of equations into solved forms. Since $\mathcal{C}$-substitutions have two parts, a term– and a colour part, we need two kinds ($M =^t N$ for term equations and $c =^c d$ for colour equations). Sets $\mathcal{E}$ of equations in solved form (i.e. where all equations are of the form $x = M$ such that the variable x does not occur anywhere else in M or $\mathcal{E}$) have a unique most general $\mathcal{C}$-unifier $\sigma_{\mathcal{E}}$ that also $\mathcal{C}$-unifies the initial equation.

There are several rules that decompose the syntactic structure of formulae, we will only present two of them. The rule for abstractions transforms equations of the form $\lambda x.A =^t \lambda y.B$ to $[c/x]A =^t [c/y]B$, and $\lambda x.A =^t B$ to $[c/x]A =^t Bc$ where c is a new constant, which may not appear in any solution. The rule for applications decomposes $h_{\mathsf{a}}(s^1, \ldots, s^n) =^t h_{\mathsf{b}}(t^1, \ldots, t^n)$ to the set $\{a =^c b, s^1 =^t t^1, \ldots, s^n =^t t^n\}$, provided that h is a constant. Furthermore equations are kept in $\beta\eta$-normal form.

The variable elimination process for colour variables is very simple, it allows to transform a set $\mathcal{E} \cup \{\mathsf{A} =^c \mathsf{d}\}$ of equations to $[\mathsf{d}/\mathsf{A}]\mathcal{E} \cup \{\mathsf{A} =^c \mathsf{d}\}$, making the equation $\{\mathsf{A} =^c \mathsf{d}\}$ solved in the result. For the formula case, elimination is not that simple, since we have to ensure that $|\sigma(x_{\mathsf{A}})| = |\sigma(x_{\mathsf{B}})|$ to obtain a $\mathcal{C}$-substitution σ. Thus we cannot simply transform a set $\mathcal{E} \cup \{x_{\mathsf{d}} =^t M\}$ into $[M/x_{\mathsf{d}}]\mathcal{E} \cup \{x_{\mathsf{d}} =^t M\}$, since this would (incorrectly) solve the equations $\{x_{\mathsf{c}} = f_{\mathsf{c}}, x_{\mathsf{d}} = g_{\mathsf{d}}\}$. The correct variable elimination rule transforms $\mathcal{E} \cup \{x_{\mathsf{d}} =^t M\}$ into $\sigma(\mathcal{E}) \cup \{x_{\mathsf{d}} =^t M, x_{\mathsf{c}_1} = M^1, \ldots, x_{\mathsf{c}_n} =^t M^n\}$, where c_i are all colours of the variable x occurring in M and $\mathcal{E}$, the M^i are appropriately coloured variants (same colour erasure) of M, and σ is the $\mathcal{C}$-substitution that eliminates all occurrences of x from $\mathcal{E}$.

Due to the presence of function variables, systematic application of these rules can terminate with equations of the form $x_{\mathsf{c}}(s^1, \ldots, s^n) =^t h_{\mathsf{d}}(t^1, \ldots, t^m)$. Such equations can neither be further decomposed, since this would loose unifiers (if G and F are variables, then $Ga = Fb$ as a solution $\lambda x.c$ for F and G, but $\{F = G, a = b\}$ is unsolvable), nor can the right hand side be substituted for x as in a variable elimination rule, since the types would clash. Let us consider the uncoloured equation $x(a) =^t a$ which has the solutions $(\lambda z.a)$ and $(\lambda z.z)$ for x.

The standard solution for finding a complete set of solutions in this so-called **flex/rigid** situation is to substitute a term for x that will enable decomposition to be applicable afterwards. It turns out that for finding all $\mathcal{C}$-unifiers it is sufficient to bind x to terms of the same type as x (otherwise the unifier would be ill-typed) and compatible colour (otherwise the unifier would not be a $\mathcal{C}$-substitution) that either

- have the same head as the right hand side; the so-called **imitation** solution ($\lambda z.a$ in our example) or
- where the head is a bound variable that enables the head of one of the arguments of x to become head; the so-called **projection** binding ($\lambda z.z$).

In order to get a better understanding of the situation let us reconsider our example using colours.

$x(a_{\mathsf{c}}) = a_{\mathsf{d}}$. For the imitation solution $(\lambda z.a_{\mathsf{d}})$ we "imitate" the right hand side, so the colour on a must be d. For the projection solution we instantiate $(\lambda z.z)$ for x and obtain $(\lambda z.z)a_{\mathsf{c}}$, which β-reduces to a_{c}. We see that this "lifts" the constant a_{c} from the argument position to the top. Incidentally, the projection is only a $\mathcal{C}$-unifier of our coloured example, if c and d are identical.

Fortunately, the choice of instantiations can be further restricted to the most general terms in the categories above. If x_{c} has type $\overline{\beta_n} \to \alpha$ and h_{d} has type $\overline{\gamma_m} \to \alpha$, then these so-called **general bindings** have the following form:

$$\mathcal{G}^h_{\mathsf{d}} = \lambda z^{\alpha_1} \ldots z^{\alpha_n}.h_{\mathsf{d}}(H^1_{\mathsf{e}_1}(\overline{z}), \ldots, H^m_{\mathsf{e}_m}(\overline{z}))$$

where the H^i are new variables of type $\overline{\beta_n} \to \gamma_i$ and the e_i are either distinct colour variables (if $\mathsf{c} \in \mathcal{CV}$) or $\mathsf{e}_i = \mathsf{d} = \mathsf{c}$ (if $\mathsf{c} \in \mathcal{C}$). If h is one of the bound variables z^{α_i}, then $\mathcal{G}^h_{\mathsf{d}}$ is called an **imitation binding**, and else, (h is a constant or a free variable), a **projection binding**.

The general rule for flex/rigid equations transforms $\{x_{\mathsf{c}}(s^1, \ldots, s^n) =^t h_{\mathsf{d}}(t^1, \ldots, t^m)\}$ into $\{x_{\mathsf{c}}(s^1, \ldots, s^n) =^t h_{\mathsf{d}}(t^1, \ldots, t^m), x_{\mathsf{c}} =^t \mathcal{G}^h_{\mathsf{c}}\}$, which in essence only fixes a particular binding for the head variable x_{c}. It turns out (for details and proofs see (Hutter and Kohlhase, 1995)) that these general bindings suffice to solve all flex/rigid situations, possibly at the cost of creating new flex/rigid situations after elimination of the variable x_{c} and decomposition of the changed equations (the elimination of x changes $x_{\mathsf{c}}(s^1, \ldots, s^n)$ to $\mathcal{G}^h_{\mathsf{c}}(s^1, \ldots, s^n)$ which has head h).

6.1 Example

To fortify our intuition on calculating higher-order coloured unifiers let us reconsider examples (10) and (11) with the equations

$R_{\neg\mathsf{pf}}(i_{\mathsf{pf}}) =^t \lambda x.ex_{\neg\mathsf{pf}}(x, i_{\mathsf{pf}}, i_{\mathsf{A}})$
$R_{\neg\mathsf{pf}}(i_{\mathsf{pf}}) =^t \lambda x.ex_{\neg\mathsf{pf}}(x, i_{\neg\mathsf{pf}}, i_{\mathsf{pf}})$

We will develop the derivation of the solutions for the first equations (10) and point out the differences for the second (11). As a first step, the first equation is decomposed to

$R_{\neg\mathsf{pf}}(i_{\mathsf{pf}}, c) =^t ex_{\neg\mathsf{pf}}(c, i_{\mathsf{pf}}, i_{\mathsf{A}})$

where c is a new constant. Since $R_{\neg\mathsf{pf}}$ is a variable, we are in a flex/rigid situation and have the possibilities of projection and imitation. The projection bindings $\lambda xy.x$ and $\lambda xy.y$ for $R_{\neg\mathsf{pf}}$ would lead us to the equations $i_{\mathsf{pf}} =^t ex_{\neg\mathsf{pf}}(c, i_{\mathsf{pf}}, i_{\mathsf{A}})$ and $c =^t ex_{\neg\mathsf{pf}}(c, i_{\mathsf{pf}}, i_{\mathsf{A}})$, which are obviously unsolvable, since the head constants i_{pf} (and c resp.) and $ex_{\neg\mathsf{pf}}$ clash[6]. So we can only bind $R_{\neg\mathsf{pf}}$ to the imitation binding $\lambda yx.ex_{\neg\mathsf{pf}}(H^1_{\neg\mathsf{pf}}(y, x), H^2_{\neg\mathsf{pf}}(y, x), H^3(y, x))$. Now, we can directly eliminate the variable $R_{\neg\mathsf{pf}}$, since there are no other variants. The resulting equation

$$\begin{aligned} & ex_{\neg\mathsf{pf}}(H^1_{\neg\mathsf{pf}}(i_{\mathsf{pf}}, c), H^2_{\neg\mathsf{pf}}(i_{\mathsf{pf}}, c), H^3(i_{\mathsf{pf}}, c)) \\ =^t \; & ex_{\neg\mathsf{pf}}(c, i_{\mathsf{pf}}, i_{\mathsf{A}}) \end{aligned}$$

can be decomposed to the equations

$$\begin{aligned} (17) \quad & H^1_{\neg\mathsf{pf}}(i_{\mathsf{pf}}, c) =^t c \\ & H^2_{\neg\mathsf{pf}}(i_{\mathsf{pf}}, c) =^t i_{\mathsf{pf}} \\ & H^3_{\neg\mathsf{pf}}(i_{\mathsf{pf}}, c) =^t i_{\mathsf{A}} \end{aligned}$$

Let us first look at the first equation; in this flex/rigid situation, only the projection binding $\lambda zw.w$ can be applied, since the imitation binding $\lambda zw.c$ contains the forbidden constant c and the other projection leads to a clash. This solves the equation, since $(\lambda zw.w)(i_{\mathsf{pf}}, c)$ β-reduces to c, giving the trivial equation $c =^t c$ which can be deleted by the decomposition rules.

Similarly, in the second equation, the projection binding $\lambda zw.z$ for H^2 solves the equation, while the second projection clashes and the imitation binding $\lambda zw.i_{\mathsf{pf}}$ is not $\neg$pf-monochrome. Thus we are left with the third equation, where both imitation and projection bindings yield legal solutions:

- The imitation binding for $H^3_{\neg\mathsf{pf}}$ is $\lambda zw.i_{\neg\mathsf{pf}}$, and not $\lambda zw.i_{\mathsf{A}}$, as one is tempted to believe, since it has to be $\neg$pf-monochrome. Thus we are left with $i_{\neg\mathsf{pf}} =^t i_{\mathsf{A}}$, which can (uniquely) be solved by the colour substitution $[\neg\mathsf{pf}/\mathsf{A}]$.

- If we bind $H^3_{\neg\mathsf{pf}}$ to $\lambda zw.z$, then we are left with $i_{\mathsf{pf}} =^t i_{\mathsf{A}}$, which can (uniquely) be solved by the colour substitution $[\mathsf{pf}/\mathsf{A}]$.

If we collect all instantiations, we arrive at exactly the two possible solutions for $R_{\neg\mathsf{pf}}$ in the original equations, which we had claimed in section 5:

$R_{\neg\mathsf{pf}} = \lambda yx.ex_{\neg\mathsf{pf}}(x, y, i_{\neg\mathsf{pf}})$
$R_{\neg\mathsf{pf}} = \lambda yx.ex_{\neg\mathsf{pf}}(x, y, x)$

Obviously both of them solve the equation and furthermore, none is more general than the other, since $i_{\neg\mathsf{pf}}$ cannot be inserted for the variable x in the second unifier (which would make it more general than the first), since x is bound.

In the case of (11) the equations corresponding to (17) are $H^1_{\neg\mathsf{pf}}(c, i_{\mathsf{pf}}) =^t c$, $H^2_{\neg\mathsf{pf}}(c, i_{\mathsf{pf}}) =^t i_{\neg\mathsf{pf}}$ and $H^3_{\neg\mathsf{pf}}(i_{\mathsf{pf}}) =^t i_{\mathsf{pf}}$. Given the discussion above, it is immediate to see that $H^1_{\neg\mathsf{pf}}$ has to be instantiated with the projection binding $\lambda zw.w$, H^2 with the imitation

---

[6] For (11) we have the same situation. Here the corresponding equation is $i_{\mathsf{pf}} =^t ex_{\neg\mathsf{pf}}(c, i_{\neg\mathsf{pf}}, i_{\mathsf{pf}})$.

binding $\lambda zw.i_{\neg\mathsf{pf}}$, since the projection binding leads to a colour clash ($i_{\neg\mathsf{pf}} =^t i_{\mathsf{pf}}$) and finally $H^3_{\neg\mathsf{pf}}$ has to be bound to the projection binding $\lambda zw.z$, since the imitation binding $\lambda zw.i_{\mathsf{pf}}$ is not $\neg$pf-monochrome. Collecting the bindings, we arrive at the unique solution $R_{\neg\mathsf{pf}} = \lambda yx.ex_{\neg\mathsf{pf}}(x, i_{\neg\mathsf{pf}}, x)$.

7 Conclusion

Higher–Order Unification has been shown to be a powerful tool for constructing the interpretation of NL. In this paper, we have argued that Higher–Order Coloured Unification allows a precise specification of the interface between semantic interpretation and other sources of linguistic information, thus preventing over–generation. We have substantiated this claim by specifying the linguistic, extra–semantic constraints regulating the interpretation of VP–ellipsis, focus, SOEs, adverbial quantification and pronouns whose antecedent is a focused NP.

Other phenomena for which the HOCU approach seems particularly promising are phenomena in which the semantic interpretation process is obviously constrained by the other sources of linguistic information. In particular, it would be interesting to see whether coloured unification can appropriately model the complex interaction of constraints governing the interpretation and acceptability of gapping on the one hand, and sloppy/strict ambiguity on the other.

Another interesting research direction would be the development and implementation of a monostratal grammar for anaphors whose interpretation are determined by coloured unification. Colours are tags which decorate a semantic representation thereby constraining the unification process; on the other hand, there are also the reflex of linguistic, non-semantic (e.g. syntactic or prosodic) information. A full grammar implementation would make this connection more precise.

8 Acknowledgements

The work reported in this paper was funded by the Deutsche Forschungsgemeinschaft (DFG) in Sonderforschungsbereich SFB–378, Project C2 (LISA).

References

[Bartels1995] Christine Bartels. 1995. Second occurrence test. Ms.

[Chomsky1976] Noam Chomsky. 1976. Conditions on rules in grammar. *Linguistic Analysis*, 2(4):303–351.

[Dalrymple et al.1991] Mary Dalrymple, Stuart Shieber, and Fernando Pereira. 1991. Ellipsis and higher-order-unification. *Linguistics and Philosophy*, 14:399–452.

[Dougherty1993] Daniel Dougherty. 1993. Higher-order unification using combinators. *Theoretical Computer Science B*, 114(2):273–298.

[Dowek1992] Gilles Dowek. 1992. Third order matching is decidable. In *Proc. LICS92*, pages 2–10. IEEE Computer Society Press.

[Gardent and Kohlhase1996] Claire Gardent and Michael Kohlhase. 1996. Focus and higher–order unification. In *Proc. COLING96* forthcoming.

[Gardent et al.1996] Claire Gardent, Michael Kohlhase, and Noor van Leusen. 1996. Corrections and higher–order unification. CLAUS report 77, University of Saarland.

[Gardent1996] Claire Gardent. 1996. Anaphores parallèles et techniques de résolution. *Langages.*

[Huet1975] Gérard Huet. 1975. A unification algorithm for typed λ–calculus. *Theoretical Computer Science 1*, pages 27–57.

[Hustadt1991] Ulrich Hustadt. 1991. A complete transformation system for polymorphic higher-order unification. Technical Report MPI-I-91-228, MPI Informatik, Saarbrücken, Germany.

[Hutter and Kohlhase1995] Dieter Hutter and Michael Kohlhase. 1995. A coloured version of the λ-calculus. SEKI-Report SR-95-05, Universität des Saarlandes.

[Jackendoff1972] Ray S. Jackendoff. 1972. *Semantic Interpretation in Generative Grammar.* The MIT Press.

[Kratzer1991] Angelika Kratzer. 1991. The representation of focus. In Arnim van Stechow and Dieter Wunderlich, editors, *Semantik: Ein internationales Handbuch der zeitgenoessischen Forschung.* Berlin: Walter de Gruyter.

[Pinkal1995] Manfred Pinkal. 1995. Radical underspecification. In *The 10th Amsterdam Colloquium.*

[Prehofer1994] Christian Prehofer. 1994. Decidable higher-order unification problems. In Alan Bundy, editor, *Proc. CADE94*, LNAI, pages 635–649, Nancy, France.

[Pulman1995] Steve G. Pulman. 1995. Higher-order unification and the interpretation of focus. Paper submitted for publication.

[von Fintel1995] Kai von Fintel. 1995. A minimal theory of adverbial quantification. Unpublished draft Ms. MIT, Cambridge, March.