The combinatorics of interpreted languages

Marcus Kracht
Department of Linguistics
UCLA
3125 Campbell Hall
405 Hilgard Avenue
Los Angeles, CA 90095–1543
kracht@humnet.ucla.edu

§1. Standard Formal Language Theory

① An **alphabet** A is a finite set of letters.

② A (finite) set C of **categories**, with a distinguished member (typically S).

③ A **language** is a subset $L \subseteq A^*$.

④ A **grammar** is a finite set of functions on $A^* \times C$.

Abstractly : F a finite set, $\Omega : F \to \mathbb{N}$ a **signature**. A (Ω-)**grammar** is a pair $G = \langle \Omega, \mathfrak{I} \rangle$, where for every $f \in F$:

$$\mathfrak{I}(f) : (A^* \times C)^{\Omega(f)} \hookrightarrow A^* \times C$$

($f : A \hookrightarrow B$ means f is partial.)

§2. Generated Language

We define $S(G)$ to be the minimal set closed under under all functions of G. (Alternatively, it the set of all denotations of zeroary terms of the clone.) Then

$$L(G) := \{\vec{x} : \langle \vec{x}, S \rangle \in S(G)\}$$

§3. Example: Boolean Formulae

$A := \{(,), \mathtt{p}, \mathtt{0}, \mathtt{1}, \neg, \vee, \wedge\}$. $\frown$ is concatenation. In CFG format:

$$\begin{aligned}
\langle \text{Ind} \rangle &\to \varepsilon \mid \langle \text{Ind} \rangle \frown \mathtt{0} \mid \langle \text{Ind} \rangle \frown \mathtt{1} \\
\langle \text{Var} \rangle &\to \mathtt{p} \frown \langle \text{Ind} \rangle \\
\langle \text{Form} \rangle &\to \langle \text{Var} \rangle \mid (\frown \neg \frown \langle \text{Form} \rangle \frown) \mid (\frown \langle \text{Form} \rangle \frown \wedge \frown \langle \text{Form} \rangle \frown) \\
&\quad \mid (\frown \langle \text{Form} \rangle \frown \vee \frown \langle \text{Form} \rangle \frown)
\end{aligned}$$

Start with the symbol $\langle \text{Form} \rangle$ and do step by step string replacement. Rephrased in a bottom up fashion.

> "If $\vec{x}$ and $\vec{y}$ are formulae then so is $(\frown \vec{x} \frown \wedge \frown \vec{y} \frown)$."

§4. Bottom Up Version (Informal)

$C := \{I, V, F\}$. Designated category F. S is minimal s.t.

① $\langle \varepsilon, I \rangle \in S$.

② If $\langle \vec{x}, I \rangle \in S$ then $\langle \vec{x}^\frown \mathtt{0}, I \rangle \in S$ and $\langle \vec{x}^\frown 1, I \rangle \in S$.

③ If $\langle \vec{x}, I \rangle \in S$ then $\langle \mathtt{p}^\frown \vec{x}, V \rangle \in S$.

④ If $\langle \vec{x}, V \rangle \in S$ then $\langle \vec{x}, F \rangle \in S$.

⑤ If $\langle \vec{x}, F \rangle \in S$ then $\langle (^\frown \neg ^\frown \vec{x}^\frown), F \rangle \in S$.

⑥ If $\langle \vec{x}, F \rangle, \langle \vec{y}, F \rangle \in S$ then $\langle (^\frown \vec{x}^\frown \wedge ^\frown \vec{y}^\frown), F \rangle \in S$ as well as $\langle (^\frown \vec{x}^\frown \vee ^\frown \vec{y}^\frown), F \rangle \in S$.

$L = \{\vec{x} : \langle \vec{x}, F \rangle \in S\}$.

§5. Bottom Up Version (Formal)

Let $F := \{f_0, f_1, f_2, f_3, f_4, f_5, f_6, f_7\}$,

$\Omega : f_0 \mapsto 0; f_1, f_2, f_3, f_4 \mapsto 1; f_5, f_6 \mapsto 2.$

$$\mathcal{I}(f_0)() := \langle \varepsilon, I \rangle$$

$$\mathcal{I}(f_1)(\langle \vec{x}, I \rangle) := \langle \vec{x}^\frown \mathtt{0}, I \rangle$$

$$\mathcal{I}(f_2)(\langle \vec{x}, I \rangle) := \langle \vec{x}^\frown \mathtt{1}, I \rangle$$

$$\mathcal{I}(f_3)(\langle \vec{x}, I \rangle) := \langle \mathtt{p}^\frown \vec{x}, V \rangle$$

$$\mathcal{I}(f_4)(\langle \vec{x}, V \rangle) := \langle \vec{x}, F \rangle$$

$$\mathcal{I}(f_5)(\langle \vec{x}, F \rangle) := \langle (^\frown \neg ^\frown \vec{x}^\frown), F \rangle$$

$$\mathcal{I}(f_6)(\langle \vec{x}, F \rangle, \langle \vec{y}, F \rangle) := \langle (^\frown \vec{x}^\frown \wedge ^\frown \vec{y}^\frown), F \rangle$$

$$\mathcal{I}(f_7)(\langle \vec{x}, F \rangle, \langle \vec{y}, F \rangle) := \langle (^\frown \vec{x}^\frown \vee ^\frown \vec{y}^\frown), F \rangle$$

§6. Bottom Up Version (Categoriless)

We can drop the categories. Define language in the wide sense:

$$L^+ := \{\vec{x} : \text{there is } c \in C : \langle \vec{x}, c\rangle \in S\}$$

☆ ⟨Ind⟩ consists of all $\vec{x} \in L^+$ such that $\vec{x} \in \{\mathtt{0}, \mathtt{1}\}^*$.

☆ ⟨Var⟩ consists of all $\vec{x} \in L^+$ of the form $\mathtt{p}^\frown\vec{x}$, $\vec{x}$ an index.

☆ ⟨Form⟩ contains all the other strings of L^+.

So: L^+ is the least set containing /p/ and ε such that

① If $\vec{x} \in \langle\text{Ind}\rangle$ then $\vec{x}^\frown\mathtt{0} \in L^+$ and $\vec{x}^\frown\mathtt{1} \in L^+$.

② If $\vec{x} \in \langle\text{Ind}\rangle$ then $\mathtt{p}^\frown\vec{x} \in L^+$.

③ If $\vec{x} \in \langle\text{Form}\rangle$ then $(^\frown\neg^\frown\vec{x}^\frown) \in L^+$.

④ If $\vec{x}, \vec{y} \in \langle\text{Form}\rangle$ then $(^\frown\vec{x}^\frown\wedge^\frown\vec{y}^\frown) \in L^+$ and $(^\frown\vec{x}^\frown\vee^\frown\vec{y}^\frown) \in L^+$.

§7. Categoriless Version

$\mathcal{I}(f_4)$ is now empty. $\langle \mathrm{Var} \rangle$ is not needed.

$$\mathcal{I}(f_0)() := \varepsilon$$

$$\mathcal{I}(f_1)(\vec{x}) := \begin{cases} \vec{x}^\frown \mathtt{0} & \text{if } \vec{x} \in \langle \mathrm{Ind} \rangle \\ \text{undefined} & \text{else} \end{cases}$$

$$\mathcal{I}(f_3)(\vec{x}) := \begin{cases} \mathtt{p}^\frown \vec{x} & \text{if } \vec{x} \in \langle \mathrm{Ind} \rangle \\ \text{undefined} & \text{else} \end{cases}$$

$$\mathcal{I}(f_5)(\vec{x}) := \begin{cases} (^\frown \neg ^\frown \vec{x}^\frown) & \text{if } \vec{x} \in \langle \mathrm{Form} \rangle \\ \text{undefined} & \text{else} \end{cases}$$

$$\mathcal{I}(f_6)(\vec{x}, \vec{y}) := \begin{cases} (^\frown \vec{x}^\frown \wedge ^\frown \vec{y}^\frown) & \text{if } \vec{x}, \vec{y} \in \langle \mathrm{Form} \rangle \\ \text{undefined} & \text{else} \end{cases}$$

§8. Categories Partially Encode Derivations

Consider

$$S \to L \mid R$$

$$L \to a \mid aL$$

$$R \to a \mid Ra$$

The categoriless grammar has more derivations! (It conflates the categories L and R.)

§9. Part 2: Interpreted Languages

Let M be a set ("meanings"). An **interpreted language** is a subset of $A^* \times M$ (a many-to-many relation between strings and meanings).

An **interpreted grammar** is a pair $G = \langle \Omega, \mathfrak{I} \rangle$, where $\Omega : F \to \mathbb{N}$ is a signature and for every $f \in F$:

$$\mathfrak{I}(f) : (A^* \times M)^{\Omega(f)} \hookrightarrow A^* \times M$$

§10. Boolean Logic

Strings as before. Valuation: a function $\beta : \langle \text{Ind} \rangle \to \{0, 1\}$. Val the set of all valuations.

$$\{\langle \vec{x}, \vec{x} \rangle : \vec{x} \in \langle \text{Ind} \rangle\}$$
$$\cup \{\langle \vec{x}, \{\beta : \beta(\vec{x}) = 1\} \rangle : \vec{x} \in \langle \text{Form} \rangle\}$$

(Need to have *some* meaning for indices!) This language is called Bool.

§11. A Grammar

$$\mathcal{I}(f_0)() := \langle \varepsilon, \varepsilon \rangle$$

$$\mathcal{I}(f_1)(\langle \vec{x}, m \rangle) := \begin{cases} \langle \vec{x}^\frown \mathtt{0}, m^\frown \mathtt{0} \rangle & \text{if } \vec{x} \in \langle \text{Ind} \rangle \\ \text{undefined} & \text{else} \end{cases}$$

$$\mathcal{I}(f_3)(\langle \vec{x}, m \rangle) := \begin{cases} \langle \mathtt{p}^\frown \vec{x}, \{\beta : \beta(\vec{x}) = 1\} \rangle & \text{if } \vec{x} \in \langle \text{Ind} \rangle \\ \text{undefined} & \text{else} \end{cases}$$

$$\mathcal{I}(f_5)(\langle \vec{x}, m \rangle) := \begin{cases} \langle (^\frown \neg ^\frown \vec{x}^\frown), \text{Val} - m \rangle & \text{if } \vec{x} \in \langle \text{Form} \rangle \\ \text{undefined} & \text{else} \end{cases}$$

$$\mathcal{I}(f_6)(\langle \vec{x}, m \rangle, \langle \vec{y}, n \rangle) := \begin{cases} \langle (^\frown \vec{x}^\frown \wedge ^\frown \vec{y}^\frown), m \cap n \rangle & \text{if } \vec{x}, \vec{y} \in \langle \text{Form} \rangle \\ \text{undefined} & \text{else} \end{cases}$$

§12. Two Grammars

Let $\vec{x} \in \{\mathsf{L}, \mathsf{0}\}^*$ be a binary string, and $n(\vec{x})$ its associated number (eg $n(\mathsf{L0LLL}) = 23$). How to generate this language? Left to right:

$$\mathcal{I}(f_0)() := \langle \mathsf{0}, 0 \rangle$$

$$\mathcal{I}(f_1)() := \langle \mathsf{L}, 1 \rangle$$

$$\mathcal{I}(a_0)(\langle \vec{x}, n \rangle) := \langle \vec{x}^\frown \mathsf{0}, 2n \rangle$$

$$\mathcal{I}(a_1)(\langle \vec{x}, n \rangle) := \langle \vec{x}^\frown \mathsf{L}, 2n + 1 \rangle$$

§13. Right to Left

$$\mathcal{I}(f_0)() := \langle \mathsf{0}, 0 \rangle$$
$$\mathcal{I}(f_1)() := \langle \mathsf{L}, 1 \rangle$$
$$\mathcal{I}(p_0)(\langle \vec{x}, n \rangle) := \langle \mathsf{0}^\frown \vec{x}, n \rangle$$
$$\mathcal{I}(p_1)(\langle \vec{x}, n \rangle) := \langle \mathsf{L}^\frown \vec{x}, n + 2^{|\vec{x}|} \rangle$$

(Function on the meanings is dependent on the length of the string!)

§14. Bigrammars

A **bigrammar** is a triple $B = \langle \Omega, \mathfrak{I}^{\varepsilon}, \mathfrak{I}^{\mu} \rangle$, where $\Omega : F \to \mathbb{N}$ is a signature and for every $f \in F$:

$$\mathfrak{I}^{\varepsilon}(f) : (A^* \times M)^{\Omega(f)} \hookrightarrow A^*$$
$$\mathfrak{I}^{\mu}(f) : (A^* \times M)^{\Omega(f)} \hookrightarrow M$$

B is **autonomous** if for every f there is $f^{\varepsilon}_* : (A^*)^{\Omega(f)} \hookrightarrow A^*$ such that for all m_i:

$$\mathfrak{I}^{\varepsilon}(f)(\langle e_0, m_0 \rangle, \cdots, \langle e_{\Omega(f)-1}, m_{\Omega(f)-1} \rangle) = f^{\varepsilon}_*(e_0, \cdots, e_{\Omega(f)-1})$$

B is **compositional** if for every f there is $f^{\mu}_* : M^{\Omega(f)} \hookrightarrow M$ such that for all e_i:

$$\mathfrak{I}^{\mu}(f)(\langle e_0, m_0 \rangle, \cdots, \langle e_{\Omega(f)-1}, m_{\Omega(f)-1} \rangle) = g^{\mu}_*(m_0, \cdots, m_{\Omega(f)-1})$$

§15. Remarks

① The notions of autonomy and compositionality can be formulated also for grammars.

② The equations can be read in the weak sense (if both sides exist then they are equal) or in the strong sense (if one side exists so does the other and they are equal).

§16. Unavoidable Ambiguity

L is **ambiguous** if there are e, m and m' such that $m \neq m'$ and $\langle e, m\rangle, \langle e, m'\rangle \in L$. Put

$$e^{\circ} := \{m : \langle e, m\rangle \in L\}$$

The **functional transform** $L^{\S}$:

$$L^{\S} := \{\langle e, e^{\circ}\rangle : e \in \varepsilon[L]\}$$

Given a grammar for L, can we construct a grammar for $L^{\S}$?

§17. Unbracketed Boolean Expressions I

The language UBool. Let $F := \{f_0, f_1, f_2, f_3, f_4, f_5, f_6, f_7\}$, $\Omega : f_0 \mapsto 0, f_1, f_2, f_3, f_4 \mapsto 1, f_5, f_6 \mapsto 2$.

$$\mathfrak{I}(f_0)() := \langle \varepsilon, I \rangle$$
$$\mathfrak{I}(f_1)(\langle \vec{x}, I \rangle) := \langle \vec{x}\frown \mathtt{0}, I \rangle$$
$$\mathfrak{I}(f_2)(\langle \vec{x}, I \rangle) := \langle \vec{x}\frown \mathtt{1}, I \rangle$$
$$\mathfrak{I}(f_3)(\langle \vec{x}, I \rangle) := \langle \mathtt{p}\frown \vec{x}, V \rangle$$
$$\mathfrak{I}(f_4)(\langle \vec{x}, V \rangle) := \langle \vec{x}, F \rangle$$
$$\mathfrak{I}(f_5)(\langle \vec{x}, F \rangle) := \langle \neg\frown \vec{x}, F \rangle$$
$$\mathfrak{I}(f_6)(\langle \vec{x}, F \rangle, \langle \vec{y}, F \rangle) := \langle \vec{x}\frown \wedge\frown \vec{y}, F \rangle$$
$$\mathfrak{I}(f_7)(\langle \vec{x}, F \rangle, \langle \vec{y}, F \rangle) := \langle \vec{x}\frown \vee\frown \vec{y}, F \rangle$$

§18. Unbracketed Boolean Expressions II

Theorem 1 *The language* UBool *has a compositional context free grammar.*

Theorem 2 *The language* $\text{UBool}^{\S}$ *has* <u>no</u> *compositional context free grammar.*

For a proof look at expressions of the form

$$p_0 \vee p_1(\vee p_1) \vee p_2(\vee p_2) \cdots \vee p_{n+2}(\vee p_{n+2}) \vee p_{n+3}$$

§19. Adjunction

A **2-context** is a triple $\langle \vec{u}, \vec{v}, \vec{w} \rangle$. A **locale** is a set of 2-contexts. An **adjunction rule** is a pair $\rho = \langle \langle \vec{x}, \vec{y} \rangle, \Lambda \rangle$, where Λ is a locale.

We write $\vec{p} \rightarrow_\rho \vec{q}$ if there is a $\langle \vec{u}, \vec{v}, \vec{w} \rangle \in \Lambda$ such that $\vec{p} = \vec{u}\vec{v}\vec{w}$ and $\vec{q} = \vec{u}\vec{x}\vec{v}\vec{y}\vec{w}$. An **adjunction grammar** is a pair $\langle C, R \rangle$ where C is a set of strings, and R a set of adjunction rules.

A **compositional interpreted adjunction bigrammar** is a bigrammar in which all syntactic functions are adjunction rules and the semantic functions are independent of the strings.

Tree adjunction grammars are similar, except that adjunction may only be to constituents and is only determined by the syntactic label.

§20. Adjunction Grammars

There is a compositional CF bigrammar for Bool in which all semantic functions are total.

Theorem 3 *There is no compositional interpreted tree adjunction bigrammar for* Bool *in which all semantic functions are total.*

I have not been able to determine what happens when the semantic functions may be partial.

Problem 4 *Is there a compositional interpreted adjunction bigrammar for* Bool*?*

§21. Predicate Logic

Let Rel be a finite set of relation symbols with signature τ.

(1) $\quad A := \{\mathtt{x}, \mathtt{0}, \mathtt{1}, \wedge, \neg, \vee, \exists\} \cup \mathrm{Rel}$

Let L be the language of predicate logic over Rel, L_n the n variable fragment of L. A **model structure** is a pair $\mathscr{M} = \langle M, I \rangle$ such that $I(R) \subseteq M^{\tau(R)}$. For a formula φ, $[\varphi]_{\mathscr{M}}$ denotes the set of satisfying assignments.

(2) $\quad [\varphi]_{\mathscr{M}} := \{\beta : \langle \mathscr{M}, \beta \rangle \models \varphi\}$

If the $\varphi \in L_n$, then we may think of $[\varphi]_{\mathscr{M}}$ as an n-ary relation on M.

§22. A Grammar

Add a type of expression, **variable**, and let $\beta^y : \beta \mapsto \beta(y)$.

$$
\begin{aligned}
L(\mathscr{M}) := \quad & \{\langle x, \beta^x \rangle : x \in \mathrm{Var}\} \\
& \cup \{\langle \varphi, [\varphi]_{\mathscr{M}} \rangle : \varphi \in L\}
\end{aligned} \tag{3}
$$

Proposition 5 *$L(\mathscr{M})$ has a compositional context free grammar.*

§23. Alphabetical Innocence

Let $\ll\varphi\gg$ be the meaning of φ. The semantics $\ll\cdot\gg$ is **alphabetically innocent** if for all injective $s : \mathrm{Var} \to \mathrm{Var}$ and all $x_i \in \mathrm{Var}$:

① $\ll\varphi^s\gg = \ll\varphi\gg$

② $\ll\varphi \wedge x_i = x_i\gg = \ll\varphi\gg$

③ $\ll\varphi \wedge x_i = x_j\gg = \ll[x_i/x_j]\varphi\gg$

Standard semantics for predicate logic is not alphabetically innocent. (Sets of assignments are sensitive to the names of variables.)

§24. Construction

Let $R, R' \subseteq M^n$ be relations. Then $R' \sim R$ if

① $R' = \pi[R]$, π a permutation of n;

② $R' = R \times M$; or

③ $R' = \{\langle a_0, \cdots, a_{n-1}, a_{n-1}\rangle : \langle a_0, \cdots, a_{n-1}\rangle \in R\}$

$\approx$ is the reflexive transitive closure of $\sim$.

Definition 6 *A **concept** is a set of relations of the form* $[\![R]\!] := \{R' : R' \approx R\}$.

Let $\varphi(x_0, \cdots, x_{n-1})$ denote an n-ary relation $[\varphi(x_0, \cdots, x_{n-1})]_{\mathscr{M}}$ on the set M. Then

$$\text{«}\varphi(x_0, \cdots, x_{n-1})\text{»} := [\![[\varphi(x_0, \cdots, x_{n-1})]]\!]$$

§25. Concept based predicate logic

Fix a model structure $\mathscr{M}$ over M. The language $\mathrm{LC}_n(\mathscr{M})$ ($\mathrm{LC}(\mathscr{M})$) is defined by

$$\{\langle \varphi, \ll\varphi\gg_{\mathscr{M}} \rangle : \varphi \in L_n\} \qquad (\{\langle \varphi, \ll\varphi\gg_{\mathscr{M}} \rangle : \varphi \in L_n\})$$

Theorem 7 *For every n and every model structure* $\mathscr{M}$ *the language* $\mathrm{LC}_n(\mathscr{M})$ *has a compositional grammar.*

Problem 8 *Does* $\mathrm{LC}(\mathscr{M})$ *have a compositional grammar?*

§26. Proof of Theorem ??.

Let f map concepts to formulae such that $\mathfrak{c} = \ll f(\mathfrak{c}) \gg_{\mathscr{M}}$. f delivers for every concept a formulae defining it.

$$\mathfrak{I}(f_{\neg})(\langle e, m\rangle) := \langle (\frown \neg \frown e \frown), \ll M \gg^{n} - [f(m)]_{\mathscr{M}}]\!]$$

Given φ, let the type of φ be

$$t(\varphi) := \{\pi : \mathscr{M} \vDash \varphi \leftrightarrow \pi(f(\ll \varphi \gg_{\mathscr{M}}))\}$$

Let π be a permutation.

$$\mathfrak{I}(f_{\wedge}^{\pi})(\langle e, m\rangle, \langle e', m'\rangle)$$

$$:= \begin{cases} \langle (\frown e \frown \wedge \frown e' \frown), [\![[f(m)]_{\mathscr{M}} \cap \pi[[f(m')]_{\mathscr{M}}]\!]\rangle \\ \qquad \text{if } \pi \circ \pi_1 = \pi_2, \pi_1 \in t(e), \pi_2 \in t(e') \\ \text{undefined} \qquad \text{else} \end{cases}$$

§27. Conclusion

❶ The combination of expression and meaning introduces more structure. Some grammars become more natural than others.

❷ Negative results are difficult to obtain, however. We need new combinatorial ideas.

❸ Benefit: Plenty of open problems!

§28. Thank You!