Computational Linguistics and Computer Languages Vol. XV. 1982.

INDUCTIVE DOMAINS AND ALGEBRAIC SEMANTICS OF CF LANGUAGES*

*Stephen D. Comer**
Mathematical Institute, Oxford
Great – Britain

1. Introduction

This note contains two simple observations on the effect of allowing a CF language to admit inductive definitions. Such languages can be generated by an unambiguous grammar and they allow the construction of an "adequate" algebraic semantics in the sense of Andreka – Nemeti – – Sain [2] (henceforth referred to as ANS).

The development of an algebraic semantics for a well presented language $L = \langle S,M,k \rangle$ in [2] depends on using a grammar G that satisfies an "adequateness criterion". Section 3 of [2] presents examples of grammars that are adequate and those that are not. A careful analysis of these examples reveals a common thread. In all cases the "meaning function" k for L was defined by induction on the complexity of the words in S. This, of course, is the most common way of defining a function on S. In most cases where G was not adequate for L it happens that G is an ambiguous grammar and it is precisely the ambiguity that leads to the inadequacy.In a nutshell, the rewrite rules of G conflict with the inductive clauses used to define k. The point is that the ability to make inductive definitions on the syntax S of a language implicitly gives a "parse" of S. We formalize this below (Proposition 1) and show that, in the situation where the meaning function is defined by induction, the induced unambiguous grammar is always adequate (Proposition 2).

2. Preliminaries

We briefly review terminology used in ANS [2].

A *well presented language* is a triple $L = \langle S,M,k \rangle$ where S is a nonempty set (the *syntax* of L) defined by a generative grammar G, M is a nonempty set-theoretically defined class (the *models* of L), and k is a function on $S \times M$ (the *meaning function* of L) that is also assumed to be set-theoretically defined.

We assume throughout that the grammar G generating S is context-free (CF). Formally, $G = \langle N,X,\langle R_i : i \in I \rangle \rangle$ with nonterminals N, terminal symbols

* This work was supported by National Science Foundation Grant MCS-8003896.
Author's address after July 1981: The Citadel, Charleston, S.C. 29409, USA..

$X(N \cap X = \emptyset)$, and rewrite rules (or productions) R_i, $i\epsilon I$. Each R_i has the form $A \vdash u$ for some $A\epsilon N$ and $u\epsilon(N \cup X)^*$. For each $A\epsilon N$ the *syntactic category of* A is the set

$$S_A = \{u\epsilon X^* : A \vdash^* u\}.$$

The syntax $L(G)$ generated by G is defined as $L(G) = \cup\{S_A \mid A\epsilon N\}$, which we assume equals S.

The first task in developing an algebraic semantics for L is to turn S into a "syntactic algebra", actually an N-sorted algebra or operator domain $\underset{\sim}{S}$ (see ADJ[1] or ANS[2]). The universe assigned to the sort $A\epsilon N$ is S_A. The operations on $\underset{\sim}{S}$ are derived from the production rules in the following way. If R_i is the rule

$$A_0 \vdash u_0 A_1 u_1 \ldots A_n u_n$$

where $u_j\epsilon X^*$ and $A_j\epsilon N$ for $j < n + 1$, the associated operation F_i has type $S_{A_1} \times \ldots \times S_{A_n} \to S_{A_0}$ and is defined as

$$F_i(a_1, \ldots, a_n) = u_0 a_1 u_1 \ldots a_n u_n$$

for each n-tuple $(a_1, \ldots, a_n)\epsilon S_{A_1} \times \ldots \times S_{A_n}$.

For each $\mathcal{U}\epsilon M$, $k(-, \mathcal{U}) = \lambda u.k(u, \mathcal{U})$ is a function on S, hence a function on the algebra $\underset{\sim}{S}$. A grammar G is called *adequate* for L (see ANS[2]) if $k(-, \mathcal{U})$ is a homomorphism on $\underset{\sim}{S}$ for each $\mathcal{U}\epsilon M$.

We also use the following terminology about sequences. Suppose $\sigma,\tau\epsilon S \subseteq X^*$. We say σ is a *part* of τ, in symbols $\sigma \sqsubseteq \tau$, if $\tau = \mu\,\sigma\,\nu$ for some $\mu, \nu \in X^*$. σ is a *proper part* of τ if $\sigma \sqsubseteq \tau$ and $\sigma \neq \tau$. σ is a *maximal part* of τ if τ covers σ in the poset $\langle S, \sqsubseteq \rangle$. $\tau\epsilon S$ is called a *basic word* (or *atom*) of S if it is a minimal element in the poset $\langle S, \sqsubseteq \rangle$.

3. Inductive domains

What is going on when a function k is defined on a syntax $S = L(G)$ by induction on the "complexity" of words? To begin with, the value of k is specified on "atoms" of L(G). Then, for every $\sigma\epsilon L(G)$, not an "atom", the value $k(\sigma)$ is given by a function applied to values $k(\sigma_1), \ldots, k(\sigma_n)$ where $\sigma_1, \ldots, \sigma_n$ are "less complex" parts of σ. If k is really a function, that is, well-defined, the parts $\sigma_1, \ldots, \sigma_n$ must be uniquely determined from σ. Thus, for inductive definitions to be possible, a "structuring" of S must be present. This motivates the following notion of an inductive domain.

Definition 1. An *inductive domain* is a triple $\langle S, \langle C_j : j\epsilon J\rangle, j_0\rangle$ where S is a CF syntax, $\{C_j : j\epsilon J\}$ is a partition of S, and $j_0\epsilon J$ such that the following hold:

(i) $\sigma \in C_{j_0}$ iff σ is a basic word of S;

(ii) for every $j \in J - \{j_0\}$ there is a sort $A_0 \in N$ with $C_j \subseteq S_{A_0}$ and there exist a unique sequence $\langle A_1, \ldots, A_k \rangle$ of sorts and unique $\mu_i \in X^*$ (for $i \leqslant k$) such that every $\sigma \in C_j$ has the form

(†) $\qquad \sigma = \mu_0 \tau_1 \mu_1 \ldots \tau_k \mu_k$

where $\tau_i \in S_{A_i}$ for $i = 1, \ldots, k$ and $\tau_1, \ldots, \tau_k$ are maximal parts of σ. Conversely, for every choice $\tau_i \in S_{A_i}$ for $i = 1, \ldots, k$, the σ with description (†) is an element of C_j.

The subsets C_j are called *clauses*. Each clause has a type. C_{j_0} has type 0. A clause C_j, $j \neq j_0$ has type k where k is the lenght of the sequence $\langle A_1, \ldots, A_k \rangle$ associated with it. We say that a syntax S *admits inductive definitions* if there exist $\langle C_j : j \in J \rangle$ and $j_0 \in J$ such that $\langle S, \langle C_j : j \in J \rangle, j_0 \rangle$ is an inductive domain. □

Definition 2. Suppose $\langle S, \langle C_j : j \in J \rangle, j_0 \rangle$ is an inductive domain.

(i) An *inductive definition* over S is a system of functions $\langle \mathbf{O}_j : j \in J \rangle$ such that domain $\mathbf{O}_{j_0} = C_{j_0}$ and $\mathbf{O}_j$ has type k whenever C_j has type k for all $j \in J - \{j_0\}$.

(ii) A function h on S is *defined by induction* by the inductive definition $\langle \mathbf{O}_j : j \in J \rangle$ if

(a) $h(\sigma) = \mathbf{O}_{j_0}(\sigma)$ for all $\sigma \in C_{j_0}$;

(b) for all $\sigma \in C_j$, $j \neq j_0$, $h(\sigma) = \mathbf{O}_j(h(\tau_1), \ldots, h(\tau_k))$ where

σ has the form (†) in Definition 1. □

Our first goal is to characterize when a CF syntax admits inductive definitions.

Proposition 1. A context-free syntax S admits inductive definitions iff $S = L(G')$ for some unambiguous grammar G' (that is, no word possesses two distinct G'-derivation trees, cf., Clark – Cowell [3]).

Proof. Suppose $\langle S, \langle C_j : j \in J \rangle, j_0 \rangle$ is an inductive domain where $S = L(G)$ for some CF grammar G. Let $K = C_{j_0} \cup (J - \{j_0\})$. We define $G' = \langle N, X, \langle P_j : j \in K \rangle \rangle$ where N and X are the nonterminals and terminals, respectively, of G. The rewrite rules P_j, $j \in K$, of G' are as follows:

for $\sigma \in C_{j_0} \cap S_A$, let P_σ be $A \underset{G'}{\vdash} \sigma$;

for each clause C_j, $j \neq j_0$ let P_j be $A_0 \underset{G'}{\vdash} \mu_0\, A_1\, \mu_1 \ldots A_k\, \mu_k$

where $A_0, A_1, \ldots, A_k$ are the unique sorts and $\mu_0, \ldots, \mu_k$ the unique sequences in (ii) of Definition 1.

An easy induction shows that G' generates S. We show that G' is unambiguous by induction on word lenght. Clearly each basic word of S has exactly one derivation tree. Now consider a word σ in C_j where σ has the form (†) in (ii) of Definition 1 and each maximal part τ_i of σ has exactly one derivation tree. Any derivation tree for σ has root labled A_0 (recall $C_j \subseteq S_{A_0}$) and the lables, left-to-right, of the sons of A_0 are exactly the symbols in the sequence $\mu_0\, A_1\, \mu_1 \ldots A_k\, \mu_k$ and the subtree rooted at each A_i is exactly the derivation tree for τ_i. By the uniqueness in (ii) of Definition 1 and the induction hypothesis, σ has exactly one derivation tree.

For the converse, suppose, $S = L(G)$ where G is an unambiguous CF grammar $\langle N, X, \langle R_i : i \in I \rangle \rangle$. Let $I' = \{ i \in I : R_i$ has the form $A \underset{G}{\vdash} \sigma$ for some $A \in N$ and $\sigma \in X^* \}$. $I' \neq \emptyset$ (we disallow $A \vdash A$ as a production) so fix $i_0 \in I'$ and define clause $C_{i_0} = \{\sigma \in X^* : A \underset{G}{\vdash} \sigma$ for some $A \in N\}$. For each $i \in I - I'$, the rewrite rule R_i contains nonterminals on the right-hand side. Suppose R_i has the form

$$A_0 \vdash_G \mu_0 A_1 \ldots . A_k \mu_k$$

for some $\mu_i \in X^*$ and $A_i \in N$ $(i \leqslant k)$. Define the clause C_i as

$$C_i = \{\sigma \in X^* : \sigma = \mu_0 \tau_1 \mu_1 \ldots \tau_k \mu_k \text{ for some } \tau_1 \in S_{A_1}, \ldots, \tau_k \in S_{A_k}\}.$$

Using the fact G is unambiguous, it is easily checked that $\langle S, \langle C_j : j \in (I - I') \cup \{i_0\}, i_0 \rangle$ is an inductive domain. □

For a CF syntax S which admits inductive definitions we call the grammar G' for S constructed in Proposition 1 the *grammar induced by the inductive structure.* It is easy to give examples of CF languages that can be generated by different unambiguous grammars; see ANS, Example 3(i).

We now consider the relationship between having an inductively defined meaning function and an adequate grammar.

Proposition 2. Suppose $L = \langle S, M, k \rangle$ is a well presented language. Then the existence of an inductive definition for $k(-, \mathcal{M})$, for each $\mathcal{M} \in M$, is equivalent to the existence of an adequate grammar for L.

Proof. First, suppose S admits inductive definitions and, for each $\mathcal{M} \in M$, $k(-, \mathcal{M})$ is defined by induction. We claim the grammar for S induced by the inductive structure is adequate for L. It needs to be shown, for $\mathcal{M} \in M$, that the equivalence relation $\equiv_{\mathcal{M}}$ on $\underset{\sim}{S}$, defined by $\sigma \equiv_{\mathcal{M}} \tau$ iff $k(\sigma, \mathcal{M}) = k(\tau, \mathcal{M})$, is a congruence relation on $\underset{\sim}{S}$. Towards this end, suppose F_j is a k-ary operation of the type $S_{A_1} \times \ldots \times S_{A_k} \to S_{A_0}$ associated with a clause C_j, $j \neq j_0$, and suppose $\sigma_i \equiv_{\mathcal{M}} \tau_i$ for all $i = 1, \ldots, k$, $\sigma = F_j(\sigma_1, \ldots, \sigma_k)$ and $\tau = F_j(\tau_1, \ldots, \tau_k)$. Then, by Definition 2(ii),

$$\begin{aligned} k(\sigma, \mathcal{M}) &= \mathbf{O}_j(k(\sigma_1, \mathcal{M}), \ldots, k(\sigma_k, \mathcal{M})) \\ &= \mathbf{O}_j(k(\tau_1, \mathcal{M}), \ldots, k(\tau_k, \mathcal{M})) \\ &= k(\tau, \mathcal{M}), \end{aligned}$$

hence $\sigma \equiv_{\mathcal{M}} \tau$ as desired.

Conversely, if G is adequate for L, the homomorphism property for $k(-, \mathcal{M})$ on $\underset{\sim}{S}$ gives an inductive definition of the function $k(-, \mathcal{M})$ on S.□

4. Example

We illustrate the remarks in the preceeding section for the implication language $L = \langle S, M, k \rangle$ treated in ANS[2], Example 3(iii), Section 3. S is generated by the grammar $G_6 = \langle N, X, \langle R_i : i \in I_6 \rangle \rangle$ where $N = \{F\}$, $X = \{p_i : i \in \omega\} \cup \{\to\}$, $I_6 = \omega \cup \{\to\}$ and the productions are:

$$R_{\to} : F \vdash F \to F$$

$$R_i : F \vdash p_i \qquad (\text{for } i \in \omega).$$

The class of models is $M = {}^{\omega}2$. The meaning function k is defined, for $\mathcal{M} \in {}^{\omega}2$, by induction as

$$k(p_i, \mathcal{M}) = \mathcal{M}(i)$$

$$k(p_i \to \phi, \mathcal{M}) = \begin{cases} 0 & \text{if } \mathcal{M}(i) = 1 \text{ and } k(\phi, \mathcal{M}) = 0 \\ 1 & \text{otherwise} \end{cases}$$

for each $i \in \omega$. The grammar G_6 is not adequate for L. An adequate grammar can be obtained from G_6 in either of two ways, depending upon what we desire to preserve.

Method 1. Keep the meaning function.

We must change the grammar. The inductive definition of k implicitly gives S the structure of an inductive domain. The induced grammar has rewrite rules, for $i \in \omega$,

$$F \vdash p_i \to F$$

$$F \vdash p_i.$$

This is exactly the grammar G_7 considered in ANS[2] and is adequate by the proof of Proposition 2.

Method 2. Keep the grammar (but make unambiguous by parsing). In the case of G_6, extend X to $X' = X \cup \{(,)\}$ and let

$$R'_{\to} : F \vdash (F \to F)$$

$$R'_i : F \vdash p_i \quad (\text{for } i \in \omega).$$

Then $G' = \langle N, X', \langle R'_i : i \in I_6 \rangle \rangle$ is an unambiguous grammar which will admit inductive definitions. In particular, we can define a new meaning function $k'(-, \mathfrak{M})$ for ω_2, by

$$k'(p_i, \mathfrak{M}) = \mathfrak{M}(i), \text{ for each } i \in \omega, \text{ and}$$

$$k'((\phi \to \psi), \mathfrak{M}) = \begin{cases} 0 \text{ if } k'(\phi, \mathfrak{M}) = 1 \text{ and } k'(\psi, \mathfrak{M}) = 0 \\ 1 \text{ otherwise.} \end{cases}$$

G' is the grammar induced by the inductive structure, so by Proposition 2, G' is adequate for the language $\langle S', M, k' \rangle$. Of course, $S' = L(G')$ is slightly different from S due to the added punctuation.

The other examples of inadequate grammars in ANS[2] can be modified in the same way.

5. Concluding remarks

Semantics (or meaning functions) used in the vast majority of languages are defined by induction. Thus, for all practical purposes, in dealing with context-free languages, we may as well assume we have such a language. The content of Proposition 2 shows that, under this assumption, an adequate algebraic semantics can always be constructed.

The content of Proposition 1 may be viewed as a justification for assuming from the beginning the syntax is "parsed" (as in, for example, ADJ[1]). The inductive structure induces a natural unambiguous grammar that generates the syntax anyway. On this point we quote Clark and Cowell [3], page 159.

"When a context-free grammar is used to specify the syntax of a programming language it is clearly important that the grammar be unambiguous. For, since the productions used in the generation of the program indicate the way the program should be 'parsed' and its meaning derived, the existence of two distinct parses for the same program might lead to an interpretation of the program by the compiler different from the interpretation intended by the programmer."

The author is grateful to H. Andréka and I. Németi for their encouragement and correspondence concerning the ideas presented in this paper.

References

[1] ADJ. Gougen, J.A., Thatcher, J.W., Wagner, E.G., and Wright, J.B., Initial Algebra Semantics and Continuous Algebras. JACM 24(1977), 68-95.

[2] ANS. Andréka, H., Németi, I., and Sain, I., Connections between Algebraic Logic and Initial Algebra Semantics of CF languages Part I and Part II. Preprint, Math. Inst. Hung. Acad. Sci. October 1978. Appeared in: Mathematical Logic in Computer Science (Proc. Coll. held in Salgótarján 1978), Dömölki, B., Gergely, T. (Eds), Colloq. Math. Soc. J. Bolyai Vol. 26, North–Holland, 1981. Part I: pp. 25-83, Part II: pp. 561-606.

[3] Clark, K.L. and Cowell, D.F., Programs, Machines, and Computation, McGraw-Hill, 1976.