On the Semantic Approaches to Boolean Grammars*

Vassilis Kountouriotis[1], Christos Nomikos[2], and Panos Rondogiannis[1]

[1] Department of Informatics & Telecommunications
University of Athens, Athens, Greece
{bk,prondo}@di.uoa.gr
[2] Department of Computer Science, University of Ioannina,
P.O. Box 1186, 45 110 Ioannina, Greece
cnomikos@cs.uoi.gr

Abstract. Boolean grammars extend context-free grammars by allowing conjunction and negation in rule bodies. This new formalism appears to be quite expressive and still efficient from a parsing point of view. Therefore, it seems reasonable to hope that boolean grammars can lead to more expressive tools that can facilitate the compilation process of modern programming languages. One important aspect concerning the theory of boolean grammars is their *semantics*. More specifically, the existence of negation makes it difficult to define a simple derivation-style semantics (such as for example in the case of context-free grammars). There have already been proposed a number of different semantic approaches in the literature. The purpose of this paper is to present the basic ideas behind each method and identify certain interesting problems that can be the object of further study in this area.

1 Introduction

Boolean grammars [Okh04] is a recent extension of context-free grammars which allows conjunction and negation in the right hand sides of rules. It has been demonstrated [Okh04] that boolean grammars can be parsed efficiently and that they can express interesting languages that are not context-free. These facts render these new grammars a promising alternative to the traditional formalisms that are currently used for the syntax analysis phase of compiler construction.

Despite their syntactic simplicity, boolean grammars proved to be non-trivial from a semantic point of view. In particular, the use of negation makes it difficult to define a simple derivation-style semantics (such as for example in the case of context-free languages). For example, it is not immediately obvious whether a grammar of the form $S \to \neg S$ has any meaning at all. There have already been proposed a number of different approaches for coping with the semantics of boolean grammars. It is the purpose of this paper to present these different techniques in a unified way and to identify certain interesting problems that can be the object of further study in this area.

* The first author's work is being supported by a doctoral research grant by the General Secretariat for Research and Technology under the program $\Pi ENE\Delta$ (grant number $03E\Delta$ 330).

Dagstuhl Seminar Proceedings 08271
Topological and Game-Theoretic Aspects of Infinite Computations
http://drops.dagstuhl.de/opus/volltexte/2008/1652

2 Boolean Grammars

The class of boolean grammars was introduced by A. Okhotin in [Okh04]. Intuitively, boolean grammars extend context-free grammars with conjunction and negation. Formally:

Definition 1 ([Okh04]). *A Boolean grammar is a quadruple $G = (\Sigma, N, P, S)$, where Σ and N are disjoint finite nonempty sets of terminal and nonterminal symbols respectively, P is a finite set of rules, each of the form*

$$A \to \alpha_1 \& \cdots \& \alpha_m \& \neg\beta_1 \& \cdots \& \neg\beta_n \qquad (m + n \geq 1, \alpha_i, \beta_i \in (\Sigma \cup N)^*),$$

and $S \in N$ is the start symbol of the grammar. We will call the α_i's positive literals *and the $\neg\beta_i$'s* negative*.*

To illustrate the use of Boolean grammars, consider the following example (which is a slightly modified version of an example taken from [Okh04]):

Example 1. Let $\Sigma = \{a, b\}$. We define:

$$\begin{array}{l} S \to \neg(AB) \;\&\; \neg(BA) \;\&\; \neg A \;\&\; \neg B \\ A \to a \\ A \to CAC \\ B \to b \\ B \to CBC \\ C \to a \\ C \to b \end{array}$$

It can be shown that the above grammar defines the language $\{ww \mid w \in \{a, b\}^*\}$ (see [Okh04] for details). It is well-known that this language is not context-free.

3 Unique Solution and Naturally Feasible Semantics

In this section we describe the two initial approaches that were proposed [Okh04] for the semantics of boolean grammars. Both approaches are based on the notion of *solution of language equations*, which is developed below. Our presentation is based on the ones given in [Okh04] and in [Wro05]. We start by defining the notion of language formulas. The definition is slightly more general than the one that will actually be needed in this section. More specifically, we assume that a language formula may also contain language symbols taken from a set Φ, which stand for arbitrary fixed languages. This extension will be needed in the next section. For the purposes of this section we assume that Φ is empty.

Definition 2. *Let Σ be a finite non-empty alphabet, N a finite non-empty set of non-terminal symbols and Φ a finite set of language symbols each corresponding to a fixed language. A* language formula *over Σ, Φ and N is defined inductively as follows:*

– *The empty string ϵ is a formula*

- *Any symbol from* $\Sigma \cup \Phi \cup N$ *is a formula*
- *If* ϕ *and* ψ *are formulas, then* $(\phi \cdot \psi)$, $(\phi \& \psi)$, $(\phi \vee \psi)$ *and* $(\neg \phi)$ *are formulae.*

Definition 3. *Let* Σ *be a finite non-empty alphabet and* N *a finite non-empty set of non-terminal symbols. An* interpretation I *of* N *is a function* $I : N \to 2^{\Sigma^*}$. *We denote by* $\perp$ *the interpretation which assigns to every non-terminal symbol in* N *the empty set.*

An interpretation I can be recursively extended to apply to any language formula, as follows:

Definition 4. *Let* Σ *be a finite non-empty alphabet,* N *a finite non-empty set of non-terminal symbols and* $\Phi = \{\mathcal{L}_1, \ldots, \mathcal{L}_m\}$ *a finite set of language symbols, such that* $\mathcal{L}_i$ *corresponds to a fixed language* F_i. *Then an interpretation* I *of* N *can be extended to* $\hat{I}$ *which assigns a value to every language formula over* Σ, Φ *and* N *as follows:*

- $\hat{I}(\epsilon) = \{\epsilon\}$.
- $\hat{I}(a) = \{a\}$, *for every* $a \in \Sigma$.
- $\hat{I}(A) = I(A)$, *for every* $A \in N$.
- $\hat{I}(\mathcal{L}_i) = F_i$, *for every* $\mathcal{L}_i \in \Phi$.
- $\hat{I}(\psi \cdot \phi) = \hat{I}(\psi) \circ \hat{I}(\phi)$.
- $\hat{I}(\psi \vee \phi) = \hat{I}(\psi) \cup \hat{I}(\phi)$.
- $\hat{I}(\psi \& \phi) = \hat{I}(\psi) \cap \hat{I}(\phi)$.
- $\hat{I}(\neg \psi) = \Sigma^* - \hat{I}(\psi)$.

We can now define the notion of *system of language equations*:

Definition 5. *Let* Σ *be a finite non-empty alphabet,* $N = \{X_1, \ldots, X_n\}$ *a finite non-empty set of non-terminal symbols and* Φ *a finite set of language symbols, that correspond to fixed languages. Let* $\phi = (\phi_1, \ldots, \phi_n)$ *be a vector of formulae over* Σ, Φ *and* N. *Then:*

$$\begin{aligned} X_1 &= \phi_1(X_1, \ldots, X_n) \\ &\cdots \\ X_n &= \phi_n(X_1, \ldots, X_n) \end{aligned}$$

is called a system of language equations over Σ, Φ and N. *The above system is usually denoted by* $X = \phi(X)$, *where* $X = (X_1, \ldots, X_n)$. *An interpretation* I *is said to be a solution of the above system, if for every* i $(1 \leq i \leq n)$, *it holds that* $I(X_i) = \hat{I}(\phi_i(X_1, \ldots, X_n))$.

Before presenting the initial semantics of boolean grammars, we need one final definition:

Definition 6. *Let* $X = (X_1, \ldots, X_n)$ *be a vector of variables and* I *be an interpretation of* $\{X_1, \ldots, X_n\}$. *Then,* I *is a solution of a system of language equations* $X = \phi(X)$ *modulo some set* M *if and only if* $I(X_i) \cap M = \hat{I}(\phi_i(X_i)) \cap M$, *for every* $1 \leq i \leq n$.

We are now in a position to define the *semantics of the unique solution in the strong sense* of a given boolean grammar [Okh04].

Let G be a given boolean grammar. Then, it is straightforward to construct from G a corresponding system of language equations: for each non-terminal of G we create a single equation whose right hand side is the disjunction of all the right hand sides of the rules for this particular non-terminal in G. The semantics of this system can now be defined as follows:

Definition 7. *A system of language equations, which corresponds to a Boolean grammar G, is said to be compliant to the* semantics of the unique solution in the strong sense *if for every finite set of strings M closed under substring, the system has a unique solution modulo M.*

There exist boolean grammars that do not posses a unique solution in the strong sense. In other words, the semantics is not defined for all boolean grammars (this is also the case for other semantic approaches that will be developed in the coming sections). Moreover, as discussed in [Okh04], given a boolean grammar one can not effectively decide whether the grammar complies to this semantics.

A second approach to the semantics of boolean grammars was also defined in [Okh04]. For convenience, given an interpretation I of N and a finite language M we denote by $I^{\cap M}$ the interpretation with $I^{\cap M}(A) = I(A) \cap M$ for every $A \in N$.

Definition 8. *Let $(X_1, \ldots, X_n) = \phi(X_1, \ldots, X_n)$ be a system of equations, which corresponds to a Boolean grammar $G = (\Sigma, N, P, S)$, with $N = \{X_1, \ldots, X_n\}$. An interpretation I is called* a naturally reachable solution *of the system if for every finite modulus M closed under substring and for every string $u \notin M$ (such that all proper substrings of u are in M) every sequence of interpretations of the form: $I^{(0)}, I^{(1)}, \ldots, I^{(i)}, \ldots$ which satisfies the properties*

- $I^{(0)} = I^{\cap M}$
- $I^{(i+1)} \neq I^{(i)}$ *and*
- *there exists some j such that $I^{(i+1)}(X_j) = \hat{I}^{(i)}(\phi_j(X_1, \ldots, X_n)) \cap (M \cup \{u\})$ and $I^{(i+1)}(X_k) = I^{(i)}(X_k)$ for all $k \neq j$*

converges to $I^{\cap(M \cup \{u\})}$ in finitely many steps.

As discussed in [Okh04], this approach also suffers from the same undecidability problems as the previous one. Certain other shortcomings of these approaches are also identified and discussed in [KNR06].

4 Stratified Boolean Grammars

In this section we present the *stratified semantics* which was developed by M. Wrona [Wro05]. As the techniques of the previous section, the stratified semantics is applicable to a subclass of boolean grammars; however, an advantage of the technique is that it is effective to decide whether a grammar is stratified (actually, in linear time). We start by defining the notion of stratification (recall that ω denotes the first infinite ordinal):

Definition 9 ([Wro05]). *A boolean grammar $G = (\Sigma, N, P, S)$ is called* stratified *if there exists a function $g : N \to \omega$ such that for every rule*

$$C \to \alpha_1 \& \cdots \& \alpha_m \& \neg\beta_1 \& \cdots \& \neg\beta_n$$

in P the following conditions hold:

- *for every i, $1 \le i \le m$ and for every $A \in N$ that appears in α_i, $g(C) \ge g(A)$*
- *for every j, $1 \le j \le n$ and for every $B \in N$ that appears in β_j, $g(C) > g(B)$.*

Intuitively, if a grammar is stratified then the set of non-terminal symbols can be partitioned into a finite set of strata, so that if C depends on D, then D cannot belong to a stratum higher than the stratum of C; furthermore if C depends on D through negation, D must belong to a stratum lower than the stratum of C. For stratified grammars, the following semantics can be defined:

Definition 10 ([Wro05]). *Let $G = (\Sigma, N, P, S)$ be a Boolean grammar stratified by function g, let $N_k = \{A \in N \mid g(A) = k\}$ denote the set of non-terminal symbols in the k-th stratum and let $X = \phi(X)$ be the system of equations that correspond to G (where X is a vector obtained by an arbitrary ordering of the non-terminal symbols in N).*

The stratified semantics of G is the interpretation S_G such that for every k, $0 \le k \le m$, and for every $A \in N_k$ it is $S_G(A) = I_k(A)$, where I_k is the least solution of the system of equations obtained from $X = \phi(X)$ as follows:

- *equations in which the left-hand side variable is not in N_k are eliminated.*
- *every occurrence in the right-hand side of the remaining equations of a non-terminal symbol $Y \in \bigcup_{0 \le i < k} N_i$ is replaced by a language symbol $\mathcal{L}_Y$ which corresponds to the language $I_{g(Y)}(Y)$.*

The stratifiability of a Boolean grammar can be efficiently decided, using well-known graph algorithms.

Theorem 1 ([Wro05]). *We can decide whether a boolean grammar G is stratified or not in time $\mathcal{O}(|G|)$, where $|G|$ denotes the size of the representation of G.*

5 Locally Stratified Boolean Grammars

In this section we present the class of locally stratified boolean grammars introduced in [NR07], which extends the class of stratified grammars. To motivate the new class, consider the following example:

Example 2. Consider the boolean grammar $G = (\{a\}, \{E, O\}, P, E)$, where P contains the following rules:

$$\begin{aligned} E &\to \epsilon \\ E &\to aO \\ O &\to \neg E \end{aligned}$$

The above boolean grammar G is obviously not stratified. However, it can easily be seen that it defines the (regular) set of strings of even length over the alphabet

$\Sigma = \{a\}$. For example, the string aa belongs to the language corresponding to E because the string a belongs to the language corresponding to O (since it does not belong to the language corresponding to E).

Grammars such as the above are locally stratified. Informally, if a grammar is locally stratified then the pairs in $(N \times \Sigma^*)$ can be partitioned into a (possibly infinite) set of strata so that if the membership of w in the language defined by nonterminal C depends on the membership of w' in the language defined by nonterminal D, then (D, w') cannot belong to a stratum higher than the stratum of (C, w); furthermore if the above dependency is obtained through negation, (D, w') must belong to a stratum lower than the stratum of (C, w). Formally:

Definition 11. *A boolean grammar $G = (\Sigma, N, P, S)$ is locally stratified if there exists a function $f : (N \times \Sigma^*) \to \omega$ such that for every rule*

$$C \to \alpha_1 \& \cdots \& \alpha_m \& \neg\beta_1 \& \cdots \& \neg\beta_n$$

in P, the following conditions hold for every i, $1 \leq i \leq m$ and for every j, $1 \leq j \leq n$:

- *Suppose that $\alpha_i = \sigma_1 A_1 \sigma_2 A_2 \ldots \sigma_k A_k \sigma_{k+1}$, for $k \geq 1, \sigma_p \in \Sigma^*, A_p \in N$. Then for every $w_1, w_2, \ldots, w_k \in \Sigma^*$ and for every p, $1 \leq p \leq k$, it holds $f(C, \sigma_1 w_1 \sigma_2 w_2 \ldots \sigma_k w_k \sigma_{k+1}) \geq f(A_p, w_p)$.*
- *Suppose that $\beta_j = \tau_1 B_1 \tau_2 B_2 \ldots \tau_\ell B_\ell \tau_{\ell+1}$, for $\ell \geq 1, \tau_q \in \Sigma^*, B_q \in N$. Then for every $w_1, w_2, \ldots, w_\ell \in \Sigma^*$ and for every q, $1 \leq q \leq \ell$, it holds $f(C, \tau_1 w_1 \tau_2 w_2 \ldots \tau_\ell w_\ell \tau_{\ell+1}) > f(B_q, w_q)$.*

Actually, as Corollary 2 demonstrates, it suffices to use a stratum-function of a special form. This is convenient in the definition of the semantics of locally stratified Boolean grammars.

Definition 12. *Let $G = (\Sigma, N, P, S)$ be a boolean grammar locally stratified by a function f. We say that f is a canonical stratum-function if*

- *for every $w, w' \in \Sigma^*$ and for every $A, B \in N$, if $|w| > |w'|$ then $f(A, w) > f(B, w')$.*
- *for every $w, w' \in \Sigma^*$ and for every $A \in N$, if $|w| = |w'|$ then $f(A, w) = f(A, w')$.*

We can now demonstrate that local stratifiability of boolean grammars is reduced (in polynomial time) to stratifiability, and therefore it is decidable in polynomial time. Before we state Theorem 2 that proves this fact, we need the following definition:

Definition 13. *Let $G = (\Sigma, N, P, S)$ be a boolean grammar. The skeleton of G is the grammar $G' = (\Sigma, N, P', S)$, where P' is obtained from P by removing from the right-hand side of each rule every literal that equals ϵ or $\neg\epsilon$, or contains terminal symbols and then removing all rules that end up with an empty right-hand side.*

The following Theorem is proved in [NR07]:

Theorem 2. *A boolean grammar $G = (\Sigma, N, P, S)$ is locally stratified if and only if its skeleton $G' = (\Sigma, N, P', S)$ is stratified.*

Corollary 1. *We can decide whether a boolean grammar G is locally stratified or not in time $\mathcal{O}(|G|)$, where $|G|$ denotes the size of the representation of G.*

Corollary 2. *A boolean grammar G is locally stratified if and only if it is locally stratified by a canonical stratum-function.*

Corollary 3. *If a boolean grammar G is stratified then it is locally stratified.*

Example 3. Consider again the Boolean grammar G of example 2. The skeleton of G contains a single rule:

$$O \rightarrow \neg E$$

and it is stratified by the function g, with $g(E) = 0$ and $g(O) = 1$. Therefore, G is locally stratified by the canonical stratum function f, with $f(E, w) = 2 \cdot |w|$ and $f(O, w) = 2 \cdot |w| + 1$.

Examples 2 and 3 show that the converse of Corollary 3 does not hold.

We next demonstrate how one can define the semantics of a Boolean grammar that is locally stratified. The languages defined by the non-terminal symbols in a locally stratified boolean grammar, can be constructed in stages. During the i-th stage, for every pair (A, w) that belongs to the i-th stratum we decide whether w belongs to the language defined by A. The following definition will be needed:

Definition 14. *Let Σ be an alphabet. We denote by Σ^n the set $\{w \in \Sigma^* \mid |w| = n\}$ and by $\Sigma^{\leq n}$ the set $\bigcup_{i=0}^{n} \Sigma^i$.*

Let $G = (\Sigma, N, P, S)$ be a boolean grammar, I an interpretation, $M \subseteq N$ be a set of non-terminal symbols, and $n \geq 0$ be an integer. We first define the conjunctive grammar $G/(I, M, n)$ that is used to decide the membership of strings of length n, in the languages corresponding to symbols in M (as defined by the rules in G), provided that some subsets of these languages are known and determined by I. Formally:

Definition 15. *Let $G = (\Sigma, N, P, S)$ be a Boolean grammar, I be an interpretation, $M \subseteq N$ be a set of non-terminal symbols, and $n \geq 0$ be an integer. Let R be the set of all literals that appear in the right hand sides of the rules in P in which the left-hand side symbol is in M. We denote by $G/(I, M, n)$ the grammar (Σ, N', P', S), such that:*

- *$N' = N \cup \{D_l \mid l \in R\}$, where the D_l's are new non-terminal symbols not belonging to N.*
- *For every rule of the form $C \rightarrow l_1 \& l_2 \& \ldots \& l_m$ in P, such that $C \in M$, P' contains the rule: $C \rightarrow D_{l_1} \& D_{l_2} \& \ldots \& D_{l_m}$.*
- *For every literal $l \in R$ and for every $w \in (\hat{I}(l) \cap \Sigma^n)$, P' contains the rule $D_l \rightarrow w$.*

– *if $n > 0$ then for every literal $l = A_1 A_2 \cdots A_k \in R \cap N^+$ and for every i, $1 \leq i \leq k$, if $\epsilon \in \bigcap_{1 \leq j \leq k, j \neq i} \hat{I}(A_j)$ then P' contains the rule $D_l \to A_i$.*
– *if $n = 0$ then for every literal $l = A_1 A_2 \cdots A_k \in R \cap N^*$, P' contains the rule $D_l \to l'$, where $l' = \alpha_1 \alpha_2 \cdots \alpha_k$, with $\alpha_i = \epsilon$ if $\epsilon \in I(A_i)$ and $\alpha_i = A_i$ otherwise.*

Based on the above definition we can now formally define the locally stratified semantics of boolean grammars. (The semantics of conjunctive grammars can be found in [Okh01]):

Definition 16. *Let $G = (\Sigma, N, P, S)$ be a boolean grammar stratified by a canonical stratum-function f. Let n_i be the (unique) length of strings in the i-th stratum and N_i be the set of nonterminal symbols of this same stratum. The locally stratified semantics of G is the interpretation L_G such that $L_G(A) = \bigcup_{i=0}^{\infty} I_i(A)$ for every $A \in N$, where $I_0 = \perp$ and $I_{i+1}(A) = I_i(A) \cup \Delta_i(A)$, for every $A \in N$, and Δ_i is the interpretation that corresponds to the semantics of the conjunctive grammar $G_i = G/(I_i, N_i, n_i)$.*

The semantics assigned to a locally stratified Boolean grammar according to the above definition, is independent of the choice of the stratum function (and actually coincides with the well-founded semantics of the grammar which happens in this case to be two-valued, see next section).

Theorem 3. *([NR07]) Let $G = (\Sigma, N, P, S)$ be a boolean grammar that is locally stratified. Then, L_G is independent of the choice of the canonical stratum function.*

6 Well-Founded Semantics

In this section we present the *well-founded semantics* of boolean grammars [KNR06]. The basic idea behind the well-founded semantics is that it is a three-valued approach: the membership of a string in the denotation of a non-terminal can not be classified as just either true or false but also as *unknown*. This helps in the cases of grammars that use negation in a problematic circular way (such as for example the grammar $S \to \neg S$). We therefore need to redefine many standard notions from ordinary formal language theory:

Definition 17. *Let Σ be a finite non-empty set of symbols. Then, a (three-valued) language over Σ is a function from Σ^* to the set $\left\{0, \frac{1}{2}, 1\right\}$.*

Intuitively, given a three-valued language L and a string w over the alphabet of L, there are three-cases: either $w \in L$ (ie., $L(w) = 1$), or $w \notin L$ (ie., $L(w) = 0$), or finally, the membership of w in L is unclear (ie., $L(w) = \frac{1}{2}$). The following definition, which generalizes the familiar notion of concatenation of languages, will be used in the following:

Definition 18. *Let Σ be a finite set of symbols and let $L_1, \ldots, L_n$ be (three-valued) languages over Σ. We define the* three-valued concatenation *of the languages $L_1, \ldots, L_n$ to be the language L such that:*

$$L(w) = \max_{\substack{(w_1, \ldots, w_n): \\ w = w_1 \cdots w_n}} \left(\min_{1 \leq i \leq n} L_i(w_i) \right)$$

The concatenation of $L_1, \ldots, L_n$ will be denoted by $L_1 \circ \cdots \circ L_n$.

We can now define the notion of *interpretation* of a given boolean grammar:

Definition 19. *An* interpretation *I of a boolean grammar $G = (\Sigma, N, P, S)$ is a function $I : N \rightarrow \left(\Sigma^* \rightarrow \left\{0, \frac{1}{2}, 1\right\}\right)$.*

An interpretation I can be recursively extended to apply to expressions that appear as the right-hand sides of boolean grammar rules:

Definition 20. *Let $G = (\Sigma, N, P, S)$ be a boolean grammar and I be an interpretation of G. Then I can be extended to become a* truth valuation *$\widehat{I}$ as follows:*

- *For the empty sequence ϵ and for all $w \in \Sigma^*$, it is $\widehat{I}(\epsilon)(w) = 1$ if $w = \epsilon$ and 0 otherwise.*
- *Let $A \in N$ and $w \in \Sigma^*$. Then, $\widehat{I}(A)(w) = I(A)(w)$.*
- *Let $a \in \Sigma$ be a terminal symbol. Then, for every $w \in \Sigma^*$, $\widehat{I}(a)(w) = 1$ if $w = a$ and 0 otherwise.*
- *Let $\alpha = \alpha_1 \cdots \alpha_n$, $n \geq 1$, be a sequence in $(\Sigma \cup N)^*$. Then, for every $w \in \Sigma^*$, it is $\widehat{I}(\alpha)(w) = (\widehat{I}(\alpha_1) \circ \cdots \circ \widehat{I}(\alpha_n))(w)$.*
- *Let $\alpha \in (\Sigma \cup N)^*$. Then, for every $w \in \Sigma^*$, $\widehat{I}(\neg\alpha)(w) = 1 - \widehat{I}(\alpha)(w)$.*
- *Let $l_1, \ldots, l_n$ be literals. Then, for every $w \in \Sigma^*$, it is $\widehat{I}(l_1 \& \cdots \& l_n)(w) = \min\{\widehat{I}(l_1)(w), \ldots, \widehat{I}(l_n)(w)\}$.*

We are now in a position to define the notion of a model of a boolean grammar:

Definition 21. *Let $G = (\Sigma, N, P, S)$ be a boolean grammar and I an interpretation of G. Then, I is a* model *of G if for every rule $A \rightarrow l_1 \& \cdots \& l_n$ in P and for every $w \in \Sigma^*$, it is $\widehat{I}(A)(w) \geq \widehat{I}(l_1 \& \cdots \& l_n)(w)$.*

In the definition of the well-founded model, two orderings on interpretations play a crucial role (see for example [Prz89,PP90]). Given two interpretations, the first ordering (usually called the *standard ordering*) compares their *degree of truth*:

Definition 22. *Let $G = (\Sigma, N, P, S)$ be a boolean grammar and I, J be two interpretations of G. Then, we say that $I \preceq J$ if for all $A \in N$ and for all $w \in \Sigma^*$, $I(A)(w) \leq J(A)(w)$.*

Among the interpretations of a given boolean grammar, there is one which is the least with respect to the $\preceq$ ordering, namely the interpretation $\perp$ which for all A and all w, $\perp(A)(w) = 0$. The second ordering (usually called the *Fitting ordering*) compares the *degree of information* of two interpretations:

Definition 23. *Let $G = (\Sigma, N, P, S)$ be a boolean grammar and I, J be two interpretations of G. Then, we say that $I \preceq_F J$ if for all $A \in N$ and for all $w \in \Sigma^*$, if $I(A)(w) = 0$ then $J(A)(w) = 0$ and if $I(A)(w) = 1$ then $J(A)(w) = 1$.*

Among the interpretations of a given boolean grammar, there is one which is the least with respect to the $\preceq_F$ ordering, namely the interpretation $\perp_F$ which for all A and all w, $\perp_F (A)(w) = \frac{1}{2}$.

Given a set U of interpretations, we will write $lub_{\preceq}U$ (respectively $lub_{\preceq_F}U$) for the least upper bound of the members of U under the standard ordering (respectively, the Fitting ordering). Formally:

$$(lub_{\preceq}U)(A)(w) = \begin{cases} 1, \text{if there exists } I \in U,\ I(A)(w) = 1 \\ 0, \text{if for all } I \in U,\ I(A)(w) = 0 \\ \frac{1}{2}, \text{otherwise} \end{cases}$$

$$(lub_{\preceq_F}U)(A)(w) = \begin{cases} 1, \text{if there exists } I \in U,\ I(A)(w) = 1 \\ 0, \text{if there exists } I \in U,\ I(A)(w) = 0 \\ \frac{1}{2}, \text{otherwise} \end{cases}$$

Notice now that $lub_{\preceq_F}U$ is not always well-defined. However, $lub_{\preceq_F}U$ is well-defined if U is a directed set of interpretations, ie., if for every $I_1, I_2 \in U$ there exists $J \in U$ such that $I_1 \preceq_F J$ and $I_2 \preceq_F J$.

We can now define the well-founded semantics of boolean grammars. The basic idea is that the intended model of the grammar is constructed in stages, ie., there is a stratification process that is related to the levels of negation used by the grammar. At each step of this process and for every nonterminal symbol, the values of certain strings are computed and fixed (as either true or false); at each new level, the values of more and more strings become fixed (and this is a monotonic procedure in the sense that values of strings that have been fixed for a given nonterminal in a previous stage, cannot be altered by the next stages). At the end of all the stages, certain strings for certain nonterminals may have not managed to get the status of either true or false (this will be due to circularities through negation in the grammar). Such strings are classified as unknown (ie., $\frac{1}{2}$).

Consider the boolean grammar G. Then, for any interpretation J of G we define the operator $[\Theta_G]_J : \mathcal{I} \to \mathcal{I}$ on the set $\mathcal{I}$ of all 3-valued interpretations of G. Notice that this operator is analogous to the ones that have been used in the logic programming domain, but has some important differences from them. More specifically, in [Prz89] two operators are used which produce two sets of atoms corresponding to true and false conclusions of the program respectively. When applied on arbitrary interpretations, these operators may produce inconsistent sets of atoms. In [PP90], one operator is defined whose definition however is not functional. These deficiencies are remedied by the following definition:

Definition 24. *Let $G = (\Sigma, N, P, S)$ be a boolean grammar, let $\mathcal{I}$ be the set of all three-valued interpretations of G and let $J \in \mathcal{I}$. The operator $[\Theta_G]_J : \mathcal{I} \to \mathcal{I}$ is defined as follows. For every $I \in \mathcal{I}$, for all $A \in N$ and for all $w \in \Sigma^*$:*

1. *$[\Theta_G]_J(I)(A)(w) = 1$ if there exists a rule $A \to l_1 \& \cdots \& l_r$ in P such that for all positive l_i it is $\hat{I}(l_i)(w) = 1$ and for all negative l_i it is $\hat{J}(l_i)(w) = 1$;*
2. *$[\Theta_G]_J(I)(A)(w) = 0$ if for every rule $A \to l_1 \& \cdots \& l_r$ in P, either there exists a positive l_i such that $\hat{I}(l_i)(w) = 0$, or there exists a negative l_i such that $\hat{J}(l_i)(w) = 0$;*
3. *$[\Theta_G]_J(I)(A)(w) = \frac{1}{2}$, otherwise.*

An important fact regarding the operator $[\Theta_G]_J$ is that it is monotonic with respect to the $\preceq$ ordering of interpretations. In addition, $[\Theta_G]_J$ has a unique least fixed point:

Theorem 4. *Let $G = (\Sigma, N, P, S)$ be a boolean grammar and let J be an interpretation of G. Define:*

$$\begin{array}{ll} [\Theta_G]_J^{\uparrow 0} & = \perp \\ [\Theta_G]_J^{\uparrow n+1} & = [\Theta_G]_J\,([\Theta_G]_J^{\uparrow n}) \\ [\Theta_G]_J^{\uparrow \omega} & = lub_{\preceq}\{[\Theta_G]_J^{\uparrow n} \mid n < \omega\} \end{array}$$

Then, the sequence $\{[\Theta_G]_J^{n}\}_{n<\omega}$ is increasing with respect to $\preceq$ and $[\Theta_G]_J^{\uparrow \omega}$ is the unique least fixed point of the operator $[\Theta_G]_J$ with respect to $\preceq$.

We will denote by $\Omega_G(J)$ the least fixed point $[\Theta_G]_J^{\uparrow \omega}$ of $[\Theta_G]_J$. Given a grammar G, we can use the Ω_G operator to construct a sequence of interpretations whose least upper bound M_G (with respect to $\preceq_F$) will prove to be a distinguished model of G.

Definition 25. *Let $G = (\Sigma, N, P, S)$ be a boolean grammar. Define:*

$$\begin{array}{ll} M_{G,0} & = \perp_F \\ M_{G,n+1} & = \Omega_G(M_{G,n}) \\ M_G & = lub_{\preceq_F}\{M_{G,n} \mid n < \omega\} \end{array}$$

From the above definition, it is not immediately obvious that M_G is well-defined since, as we have remarked earlier, $lub_{\preceq_F}$ is not always well-defined. However, we can prove that the operator Ω_G is monotonic with respect to $\preceq_F$ and this ensures that the sequence $\{M_{G,n}\}_{n<\omega}$ is increasing (which ensures that $lub_{\preceq_F}$ is well-defined).

Theorem 5. *Let $G = (\Sigma, N, P, S)$ be a boolean grammar. Then, the sequence $\{M_{G,n}\}_{n<\omega}$ is increasing with respect to the $\preceq_F$ ordering of interpretations. Moreover, M_G is the least fixed point of the operator Ω_G.*

Using the above theorem, the following follows easily:

Theorem 6. *Let $G = (\Sigma, N, P, S)$ be a boolean grammar. Then, M_G is a model of G (which will be called the* well-founded model *of G).*

Actually, it can be shown (following a similar reasoning as in [RW05]) that the model M_G is the *least* model of G according to a syntax-independent relation.

7 Conclusions and Future work

We have presented a survey of the existing approaches to the semantics of boolean grammars. It appears that boolean grammars are closely connected to logic programs and this relationship has been exploited in order to define new concepts in the grammars domain. Since boolean grammars are much simpler than logic programs, certain notions that are undecidable in the case of logic programs are decidable (and even efficiently so) in the case of boolean grammars (for example, local stratifiability).

One very interesting open problem is to establish the relationship between the classes of languages defined by the various semantics. For example, is the class of languages defined by the locally stratified semantics a superset of that defined by the stratified semantics? Questions like this one do not appear to have straightforward answers: a positive answer would imply a separation of the boolean grammars from the conjunctive ones [Okh01], which is already an open problem [Okh04]. Another interesing question is whether one can characterize (syntactically) the broadest class of boolean grammars whose members have a two-valued well-founded semantics.

Closing, we can say that boolean grammars is an extension of context-free grammars whose semantics is much more sophisticated that the usual (derivation-based) semantics of context-free grammars. In fact, the approaches we have presented in this survey follow what is usually termed a *denotational* semantics while the simple derivation approach of context-free grammars is what is often termed an *operational* semantics (see for example [Ten91]). We believe that one can define much more sophisticated grammars than the existing ones, grammars that use powerful constructs that resemble those encountered in programming languages. These new grammars can then be assigned semantics using techniques that have been developed in the programming languages theory domain. In conclusion, we believe that a further investigation of the connections between formal language theory and the theory of programming languages semantics will prove to be very rewarding.

References

[KNR06] Kountouriotis, V., Nomikos, Ch., Rondogiannis, P.: Well-Founded Semantics of Boolean Grammars. DLT (2006) 203–214.

[NR07] Nomikos, Ch., Rondogiannis, P.: Locally Stratified Boolean Grammars, LATA (2007).

[Okh01] Okhotin, A.: Conjunctive Grammars. Journal of Automata, Languages and Combinatorics **6(4)** (2001) 519–535.

[Okh04] Okhotin, A.: Boolean Grammars. Information and Computation **194(1)** (2004) 19–48.

[PP90] Przymusinska, H., Przymusinski, T.: Semantic Issues in Deductive Databases and Logic Programs. In R. Banerji, editor, Formal Techniques in Artificial Intelligence: a Source-Book, North Holland (1990) 321–367.

[Prz89] Przymusinski, T.: Every Logic Program has a Natural Stratification and an Iterated Fixed Point Model. In Proceedings of the 8th Symposium on Principles of Database Systems ACM SIGACT-SIGMOD (1989) 11–21.

[RW05] Rondogiannis, P., Wadge, W., W.: Minimum Model Semantics for Logic Programs with Negation-as-Failure. ACM Transactions on Computational Logic **6(2)** (2005) 441–467.

[Ten91] Tennent, R.: Semantics of Programming Languages. Prentice Hall (1991).

[vGRS91] van Gelder, A, Ross, K., A., Schlipf, J., S.: The Well-Founded Semantics for General Logic Programs. Journal of the ACM **38(3)** (1991) 620–650.

[Wro05] Wrona M.: Stratified Boolean Grammars. MFCS (2005) 801–812.