Boolean Grammars

Rémi Varloot

Informatique 2010

Definition of boolean grammars

Definition of boolean grammars

Definition

A Boolean grammar is a quadruple $G = (\Sigma, N, P, S)$ in which:

- Σ is a finite nonempty set of terminal symbols;
- N is a finite nonempty set of nonterminal symbols, with $N \cap \Sigma = \varnothing$;
- P is a finite set of rules of the form

$$A \to \alpha_1 \& \dots \& \alpha_k \& \neg\alpha_{k+1} \& \dots \& \neg\alpha_{k+l}$$

  where $k + l > 0$ and $\alpha_i \in (\Sigma \cup N)^*$ for all i in $\{1, \dots, k + l\}$;
- $S \in N$ is the start symbol of the grammar.

Why are these grammars interesting?

Why are these grammars interesting?

Example

We consider the grammar

$$G = \left(\{a\}, \left\{S, X, X', X'', X''', Y', Y'', Y''', Z, T\right\}, P, S\right)$$

where P is the following set of rules:

$$\begin{array}{llll} S \to X \& \neg aX & X \to aX'X' & Y \to aaY'Y' & Z \to Y \\ S \to \neg X \& aX & X' \to \neg X''X'' & Y' \to Y''Y'' \& T & Z \to aY \\ S \to Z \& \neg aZ & X'' \to \neg X'''X''' & Y'' \to \neg Y'''Y''' \& T & T \to aaT \\ S \to \neg Z \& aZ & X''' \to \neg X & Y''' \to \neg Y \& T & T \to \epsilon \end{array}$$

Why are these grammars interesting?

Example

We consider the grammar

$$G = \left(\{a\}, \left\{S, X, X', X'', X''', Y', Y'', Y''', Z, T\right\}, P, S\right)$$

where P is the following set of rules:

$$\begin{array}{llll}
S \to X \& \neg aX & X \to aX'X' & Y \to aaY'Y' & Z \to Y \\
S \to \neg X \& aX & X' \to \neg X''X'' & Y' \to Y''Y'' \& T & Z \to aY \\
S \to Z \& \neg aZ & X'' \to \neg X'''X''' & Y'' \to \neg Y'''Y''' \& T & T \to aaT \\
S \to \neg Z \& aZ & X''' \to \neg X & Y''' \to \neg Y \& T & T \to \epsilon
\end{array}$$

The language described by this grammar is the language:

$$L(G) = \left\{a^{2^n} \mid n \geq 0\right\}$$

All too easy. . .

All too easy...

What to think of the following rule?

$$S \rightarrow \neg S$$

All too easy...

What to think of the following rule?

$$S \rightarrow \neg S$$

But also...

Some grammars can accept multiple languages, and yet not recognize the union of these languages.

All too easy...

What to think of the following rule?

$$S \to \neg S$$

But also...

Some grammars can accept multiple languages, and yet not recognize the union of these languages.

In other words

Unlike usual context-free grammars, Boolean grammars ensure neither existence nor uniqueness of a solution.

Systems of language equations

Systems of language equations

Example

$$\begin{cases} S = X\&\neg aX \vee \neg X\&aX \vee Z\&\neg aZ \vee \neg Z\&aZ \\ X = a\left(\neg\left(\neg(\neg X)^2\right)^2\right)^2 \\ Y = aa\left(\neg\left(\neg(\neg Y\&T)^2\&T\right)^2\&T\right)^2 \\ Z = Y \vee aY \\ T = aaT \vee \epsilon \end{cases}$$

Systems of language equations

Example

$$\begin{cases} S &= X\&\neg aX \vee \neg X\&aX \vee Z\&\neg aZ \vee \neg Z\&aZ \\ X &= a\left(\neg\left(\neg(\neg X)^2\right)^2\right)^2 \\ Y &= aa\left(\neg\left(\neg(\neg Y\&T)^2\&T\right)^2\&T\right)^2 \\ Z &= Y \vee aY \\ T &= aaT \vee \epsilon \end{cases}$$

Recognized language

Let (s, x, y, z, t) be the *unique* solution to this system. The language recognized by the grammar is:

$$L(G) = s = \left\{a^{2^n} \mid n \geq 0\right\}$$

Characterizing convenient systems

Characterizing convenient systems

Two possibilities

Characterizing convenient systems

Two possibilities

- Finding a means of characterizing systems with a unique solution;

Characterizing convenient systems

Two possibilities

- Finding a means of characterizing systems with a unique solution;
- Defining a "most convenient" solution for systems with multiple solutions.

Characterizing convenient systems

Two possibilities

- Finding a means of characterizing systems with a unique solution;
- Defining a "most convenient" solution for systems with multiple solutions.

Naturally reachable solutions

We introduce *naturally reachable solutions*, unique peculiar solutions to some systems with multiple solutions.

Naturally reachable solutions

Naturally reachable solutions

Definition

Let $X = \phi(X)$ be a system. For a finite language M closed under substring, a string u not in M such that all proper substrings of u are in M and a pair (P, Q) of language vectors, we write $P \xrightarrow[M,u]{\phi} Q$ if their exists an integer k such that:

$$Q_i = P_i \quad (i \neq k) \qquad Q_k = \phi_k(P) \cap (M \cup \{u\})$$

Naturally reachable solutions

Definition

Let $X = \phi(X)$ be a system. For a finite language M closed under substring, a string u not in M such that all proper substrings of u are in M and a pair (P, Q) of language vectors, we write $P \xrightarrow[M,u]{\phi} Q$ if their exists an integer k such that:

$$Q_i = P_i \quad (i \neq k) \qquad Q_k = \phi_k(P) \cap (M \cup \{u\})$$

$L = (L_1, \ldots, L_n)$ is a naturally reachable solution of the system $X = \phi(X)$ if for each finite language M closed under substring and each string u not in M such that all proper substrings of u are in M, we have

$$(L_1 \cap M, \ldots, L_n \cap M) \xrightarrow[M,u]{\phi} \ldots \xrightarrow[M,u]{\phi} (L_1 \cap (M \cup \{u\}), \ldots, L_n \cap (M \cup \{u\}))$$

Conclusion

Conclusion

Theorem

If a system has a naturally reachable solution L, then:

Conclusion

Theorem

If a system has a naturally reachable solution L, then:

- L is unique;

Conclusion

Theorem

If a system has a naturally reachable solution L, then:

- L is unique;
- L is a solution of the system.

Conclusion

Theorem

If a system has a naturally reachable solution L, then:

- L is unique;
- L is a solution of the system.

At the end of the day...

The semantics of naturally reachable solutions for Boolean grammars are a powerful means of describing certain languages, including some languages out of the scope of usual context-free grammars.