The Link between r-contiguous Detectors and k-CNF Satisfiability

Thomas Stibor, Jonathan Timmis and Claudia Eckert

***Abstract*—In the context of generating detectors using the r-contiguous matching rule, questions have been raised at the efficiency of the process. We show that the problem of generating r-contiguous detectors can be transformed in a k-CNF satisfiability problem. This insight allows for the wider understanding of the problem of generating r-contiguous detectors. Moreover, we apply this result to consider questions relating to the complexity of generating detectors, and when detectors are generable.**

I. Introduction

The natural immune system is a powerful information processing network and provides a rich source of inspiration for the creation of techniques for solving information processing and computational problems. Artificial immune systems are immune-inspired techniques and algorithms which are applied on problem domains such as pattern classification, clustering and optimization — see [1], [2] for an overview. An early (and popular) immune-inspired algorithm for pattern classification was negative selection. During negative selection in the immune system, self-reactive T-Lymphocytes which carry antibodies on their surface, are eliminated by a controlled death. As a result, only self-tolerant T-Lymphocytes survive the negative selection process and are then released into the blood stream. Roughly speaking, the immune negative selection is a process in which self-tolerant lymphocytes are generated, thus allowing the immune system to discriminate self proteins from foreign proteins (termed non-self). Through the process of abstraction, antibodies can be represented as bit-strings (termed detectors), and the process of generating these detectors can be created (termed negative selection). However, a fundamental question arises : *how are detectors generated efficiently ?*. In recent years, many detector generating algorithms have been proposed, however all of these algorithms have an infeasible runtime or space complexity. An overview on the complexity of recent (r-contiguous) detector generating algorithms can be found in [3].

In this paper we begin to address the question : *is it possible to generate r-contiguous detector efficiently ?*. In previous works, Esponda et al. [4], [5] have shown the connection between the boolean satisfiability problem (SAT) and a negative database[1]. In this paper we specialize the approach presented in [4], [5]. More specifically, we show that the problem of generating r-contiguous detectors can be transformed in a k-CNF satisfiability problem. Through this, we can begin to understand in a deeper way, issues surrounding the generation of detectors using r-contiguous matching rule.

In this paper, we also report a revised result on the r-contiguous matching probability between two randomly drawn bit-strings, as noticed by Ranang [6]. The paper is organized as follows : we introduce the original negative selection approach in section II. In section III the r-contiguous matching rule is formally described and the revised result on the r-contiguous matching probability is reported. Additionally, set partitioning is presented when applying negative selection in conjunction with the r-contiguous matching rule. The k-CNF satisfiability problem is outlined in section IV and the link to r-contiguous detectors in made in section V. In section V-A we demonstrate how statements on boolean unsatisfiability can be applied in the context of no generable r-contiguous detectors. Moreover in section V-B statements on the complexity of r-contiguous detector generation are discussed, and impacts on negative selection algorithms are considered in section V-C.

II. Negative Selection Principle

The immune negative selection process is a mechanism employed to help protect the body against self-reactive lymphocytes. This process inspired Forrest et al. [7] to propose a negative selection algorithm to detect data manipulation caused by computer viruses. The basic idea is to generate a number of detectors in the complementary space, and then to apply these detectors to classify new (unseen) data as self (no data manipulation) or non-self (data manipulation). The negative selection algorithm proposed by Forrest et al. is illustrated in figure 1 and summarized in the following steps.
Given an universe U which contains all unique bit-strings of length l, self set $S \subset U$ and non-self set $N \subset U$, where

$$U = S \cup N \quad and \quad S \cap N = \emptyset.$$

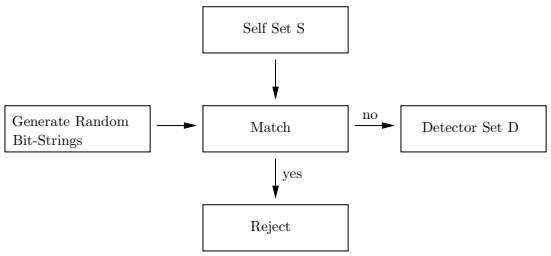
1) Define self as a set S of bit-strings of length l in U.

Thomas Stibor is with the Department of Computer Science, Darmstadt University of Technology, 64289 Darmstadt, Germany (email: stibor@sec.informatik.tu-darmstadt.de).

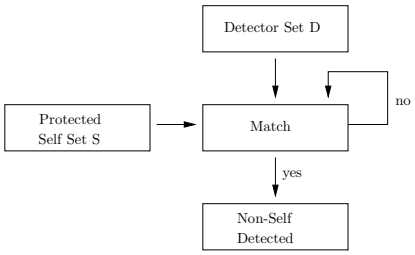
Jonathan Timmis is with the Department of Computer Science and Department of Electronics, University of York, Heslington, York, YO10 5DD, United Kingdom (email: jtimmis@cs.york.ac.uk).

Claudia Eckert is with the Department of Computer Science, Darmstadt University of Technology, 64289 Darmstadt, Germany (email: eckert@sec.informatik.tu-darmstadt.de).

[1] representing bit-strings in a compress form in the complementary Hamming space



(a) Generation of Detector Set



(b) Monitor Protected Strings for Manipulation

Figure 1. Negative selection algorithm proposed by Forrest et al.

2) Generate a set D of detectors, such that each fails to match any bit-string in S.
3) Monitor S for changes by continually matching the detectors in D against S.

By considering this negative selection algorithm, especially with regards to step (2), a fundamental question arises : how are detectors generated efficiently ?. Of course, to answer this question we require a matching rule which defines a "match" between two bit-strings from U.

III. R-Contiguous Matching Rule

The r-contiguous matching rule was proposed by Percus et. al [8]. It abstracts the process of the binding between an antibody and an antigen, and computes an affinity between the two. From a chemical point of view, an antibody and an antigen are a sequence of amino acids. An antibody recognizes an antigen, when over a certain sequence-length, a number of amino acids are identical in both substances. By coding amino acids as bit-strings, a formal match between antibody and antigen can be defined as follows :

Definition 1: An element $e \in \{0,1\}^l$ with $e = e_1 e_2 \dots e_l$ and detector $d \in \{0,1\}^l$ with $d = d_1 d_2 \dots d_l$, match with r-contiguous rule, if a position p exists where $e_i = d_i$ for $i = p, \dots, p+r-1$ and $p \leq l - r + 1$.
Informally, two elements, with the same length, match if at least r contiguous characters are identical.

Example 1:
$$\begin{array}{lll} 0\,1\,\mathbf{1}\,\mathbf{0}\,\mathbf{1}\,0\,1 & = & u_1 \\ 0\,0\,\mathbf{1}\,\mathbf{0}\,\mathbf{1}\,1\,0 & = & u_2 \end{array}$$

Example (1) illustrates a r-contiguous match for $r = 3$ between bit-string u_1 and u_2 both of length $l = 7$.

A. Revised R-Contiguous Matching Probability

The algorithm proposed first for the generation of r-contiguous detectors was a simple random search [7]. A bit-string (candidate detector) was randomly drawn from U and matched against all bit-strings in S. If no match occurred, then the detector was stored in the detector set D (see Fig. 1(a)). To estimate the number of detectors which must be drawn to obtain a number of suitable (censored) detectors, a probabilistic analysis was proposed by Forrest et al. [7]. This analysis was based on results proposed by Percus et al. [8].

Percus et al. [8] approximate the probability P_S that a random detector recognizes[2] a random antigen with

$$P_S = m^{-r}\left[(l-r)(m-1)/m + 1\right] \tag{1}$$

where m is the alphabet size and l, r the parameters from definition (1). Percus et al. mentioned that term (1) is a proper approximation when $m^{-r} \ll 1$, as the probability of long matching region in antibody-antigen matching is very small. Percus et al. argument is of course correct, when regarding antibody-antigen matching regions from a pure immunological point of view. However, applying this argument naively and unverified to artificial immune system is dangerous and can lead to incorrect results. Approximation (1) has been shown not to be correct for $r \leq l$.

The correct probability approximation[3] for $r \leq l$ and alphabet size 2 was originally derived by William Feller and is presented in his textbook [9]. To approximate the probability that a randomly drawn bit-string recognizes (when using the r-contiguous matching rule) a randomly drawn bit-string is formally defined by Feller [9] as follows :

"*A sequence of n letters S and F contains as many S-runs of length r as there are non-overlapping uninterrupted blocks containing exactly r letters S each*".

Given a Bernoulli trial with outcomes S (success) and F (failure), the probability of no success running of length r in l trials is according to Feller

$$\frac{1-px}{(r+1-rx)q} \cdot \frac{1}{x^{l+1}} \tag{2}$$

where

$$p = q = \frac{1}{2} \quad \text{and} \quad x = 1 + qp^r + (r+1)(qp^r)^2 + \dots$$

as term (2) gives the probability of no success run of length r in l trials, the correct approximation that a random detector recognizes with r-contiguous matching rule a random antigen results in

$$P_{WF} = 1 - \left(\frac{1-px}{(r+1-rx)q} \cdot \frac{1}{x^{l+1}}\right) \tag{3}$$

We would like to emphasize here that the link between r-contiguous matching rule and term (2) was first demonstrated by Ranang [6] — we have summarized his results

[2] with r-contiguous matching rule
[3] also presented (slightly different) in Ranang master thesis [6]

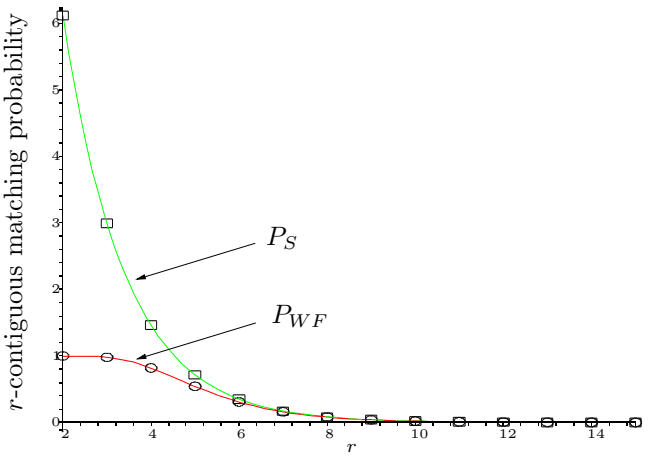
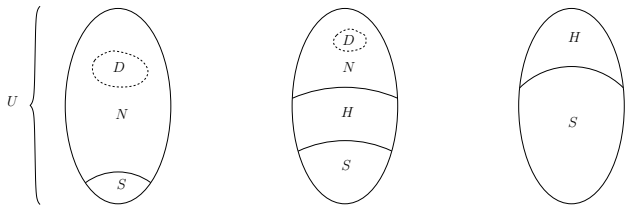


Figure 2. Difference between Percus et al. and Feller's (probability) approximation for $l = 49$ and $r = \{2, \ldots, 15\}$



(a) The universe U is partitioned in self set S and non-self set N only. The r-contiguous detectors from $D \subseteq N$ cover all non-self bit-strings from the set N.

(b) S contains a certain number of distinct bit-strings which induces holes from the undetectable set H. The remaining holes are induced by a mixture of bit-strings from U.

(c) S contains such a large number of distinct bit-strings and therefore U is only partitioned in S and H, i.e. no detectors are generable at all and all bit-strings from U are undetectable.

Figure 3. Coherences between universe U, self set S, non-self set N, detector set D and hole set H

in this section. Ranang showed that for $m = 2, l = 49$ and $r = 4$ the approximation (1) results in

$$P_S = 2^{-4}\left[\frac{(49-4)(2-1)}{2} + 1\right] = 1.46875$$

which is greater than 1 and therefore does *not* describe a probability distribution. Verifying term (3) for $l = 49$ and $r = 4$ results in $P_{WF} \approx 0.82$. The difference between term (1) and (3) for small values of r is very large (see Fig. 2). When a certain value for r is reached, both terms adjust and decrease asymptotically to 0 — this was probably the reason that nobody except Ranang noticed this incorrect approximation for $r \leq l$. For the sake of completeness, we have to mentioned that Wierzchoń [10] has noticed this, and showed that approximation (1) is only valid when $r \geq l/2$.

B. Coherence between Universe, Self, Non-Self and arisen Holes

As outlined in the introduction, the immune system discriminates between self and non-self, in part, as a result of the process of negative selection. Through the application of the r-contiguous matching rule in the negative selection algorithm, an interesting set partition occurs. Let U be an universe which contains all unique bit-strings of length l and a self set S which contains only *one* bit-string arbitrarily chosen. Let D be the set of all generable r-contiguous detectors. In this case, the following coherence holds:

$$U = N \cup S\,, \quad S \cap N = \emptyset \quad \text{and} \quad D \subseteq N$$

The detector set is a subset of the bit-strings from the set N (see Fig. 3(a)). However, if S contains a certain number of distinct bit-strings then an additional set occur — the set H of non-detectable bit-strings (see Fig. 3(b)). More specifically the following coherence holds:

$$U = N \cup S \cup H \quad \text{where}$$
$$N \cap S = \emptyset, \quad N \cap H = \emptyset, \quad H \cap S = \emptyset \quad \text{and}$$
$$D \subseteq N$$

If the cardinality of S reaches a certain number of bit-strings, then the universe U will consist only of the sets S and H (see Fig. 3(c)), i.e. no detectors are generable at all. The set H represents bit-strings which are *not* members of S and N. Bit-strings from H are termed holes and occur when a certain number of distinct bit-strings from S exist [11], [12]. More specifically, holes are induced by distinct bit-strings from S and can be constructed by the crossover-closure method [11], [12]. The idea behind the crossover-closure is presented in example 2 and figure 4. Each bit-string $s \in S$ is subdivided in $l-r+1$ substrings[4] $s[1, \ldots, r], s[2, \ldots, r+1], \ldots, s[l-r+1, \ldots, l]$ and connected with a direct edge, if the last $r-1$ bits of $s[i, \ldots, r+i-1]$ are matched[5] with the first bits of $s[i+1, \ldots, r+i]$, for $i = 1, \ldots, l-r$ and all $s \in S$. Substrings which are connected with a direct edge are merged over $r-1$ equal bits to one bit-string of length l.

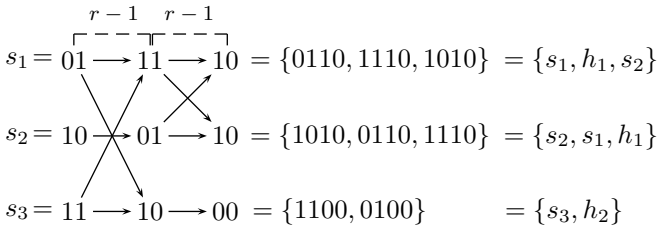
Holes are *not only* induced by bit-strings from S, but also by bit-strings from N and bit-strings which are constructed by the crossover-closure of S. These additionally holes can also be constructed with the crossover-closure method. This is demonstrated in the following example and illustrated in figures 4(b),4(c).

Example 2: Let $l = 4$, $r = 2$ and $S = \{s_1, s_2, s_3\}$, where $s_1 = \{0110\}$, $s_2 = \{1010\}$ and $s_3 = \{1100\}$. One can easily verify that only *one* r-contiguous detector is generable, namely 0001. That implies that all bit-strings

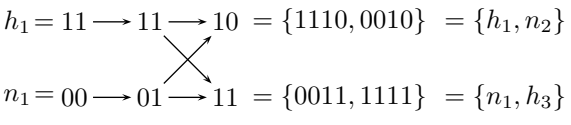
[4] $s[1, \ldots, l]$ denotes characters of s at positions $1 \ldots l$
[5] equal bits

in the set[6] $\{00**, *00*, **01\}$ are detectable. On the other hand, all bit-strings from $U \setminus \{00**, *00*, **01\} = H \cup S$ are not detectable.

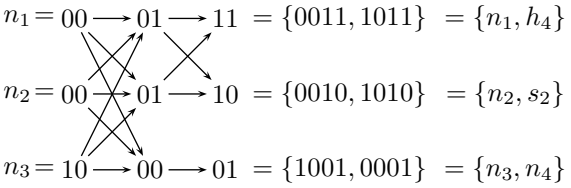
In figure 4(a) one can see, that self bit-strings s_1, s_2 and s_3 induce the holes $h_1 = 1110$ and $h_2 = 0100$. Moreover as illustrated in figure 4(b), the non-self bit-string $n_1 = 0011$ and hole $h_1 = 1110$ induce the additional hole $h_3 = 1111$. The hole $h_4 = 1011$ is induced by non-self bit-strings n_1, n_2 and n_3 (see Fig. 4(c)).



(a) Holes ($h_1 = 1110$, $h_2 = 0100$) are induced by self bit-strings $s_1 = 0110$, $s_2 = 1010$ and $s_3 = 1100$.



(b) An additional hole is induced by the constructed hole $h_1 = 1110$ and non-self bit-string $n_1 = 0011$.



(c) The further hole $h_4 = 1011$ is induced by non-self bit-strings $n_1 = 0011$, $n_2 = 0010$ and $n_3 = 1001$.

Figure 4. Holes constructed by means of the crossover-closer method for bit-strings from sets N, S and H

The latter hole ($h_5 = 0111$) can be constructed by applying the crossover-closure with bit-strings s_1 and h_3. This example illustrated that holes are *not only* induced by bit-strings from S when applying the r-contiguous matching rule, but also by bit-strings from $U \setminus S$.

IV. K-CNF Satisfiability

In this section we outline the k-CNF satisfiability problem and subsequently show how r-contiguous detectors are related to the k-CNF satisfiability problem.

The boolean satisfiability problem is a decision problem and can be formulated in terms of the language SAT [13]. An instance of SAT is a boolean formula ϕ composed of $\wedge$ (AND), $\vee$ (OR), $\bar{}$ (NOT), $\rightarrow$ (implications), $\leftrightarrow$ (if and only if), variables $x_1, x_2, \ldots$, and parentheses. In SAT problems, one has to decide if there is some assignment of *true* and *false* values to the variables that will make the boolean formula ϕ true. In the following sections, we will focus on boolean formulas in conjunctive normal form.

A boolean formula is in conjunctive normal form (CNF), if it is expressed as an AND-combination of clauses and each clause is expressed as an OR-combination of one or more literals. A literal is an occurrence of a boolean variable x or its negation $\overline{x}$.

Example 3:

$$\underbrace{(\overbrace{x_1}^{\text{literal}} \vee \overline{x}_1 \vee \overline{x}_2)}_{\text{clause}} \wedge (x_3 \vee x_2 \vee x_4) \wedge (\overline{x}_1 \vee \overline{x}_3 \vee \overline{x}_4)$$

A boolean formula is in k-CNF, if each clause has exactly k distinct literals. Example (3) shows a 3-CNF boolean formula. A k-CNF boolean formula is *satisfiable* if there exists a set of values ($0 \equiv$ false and $1 \equiv$ true) for the literals that causes it to evaluate to 1, i.e. the logical value *true*. A possible assignment set of boolean values that evaluate in example (3) to true is, $x_1 = 1, x_2 = 1, x_3 = 0, x_4 = 0$ (or expressed as a bit-string 1100). In k-CNF-SAT, we are asked whether a given boolean formula in k-CNF is satisfiable. It is known that for $k > 2$, k-CNF-SAT is $\mathcal{NP}$-complete [14], i.e. this problem is *verifiable* in polynomial time, but nobody has yet discovered an algorithm for solving[7] k-CNF-SAT in polynomial time.

We will now consider a special subset of boolean formulas in k-CNF which are defined as follows :

Definition 2: A k-CNF boolean formula ϕ_{rcb} is in l-k-CNF, when ϕ_{rcb} has $(l-k+1)$ clauses $C_1, C_2, \ldots, C_{l-k+1}$ for $1 \leq k \leq l$ and $k-1$ equal literals in C_i, C_{i+1} for $i = 1, 2, \ldots, l-k$

$$\begin{aligned} C_1 &= (x_1 \vee x_2 \vee \ldots \vee x_k) \\ C_2 &= (x_2 \vee x_3 \vee \ldots \vee x_{k+1}) \\ &\vdots \\ C_{l-k+1} &= (x_{l-k+1} \vee x_{l-k+2} \vee \ldots \vee x_l). \end{aligned}$$

Example 4: Let $l = 8$, $k = 3$ and $C_1, C_2, \ldots, C_6$

[6] the symbol * represents either a 1 or 0

[7] generating solutions in polynomial time which evaluate to 1

clauses with for instance randomly chosen literals

$$\begin{aligned} C_1 &= (\overline{x}_1 \vee x_2 \vee \overline{x}_3) \\ C_2 &= (x_2 \vee \overline{x}_3 \vee x_4) \\ C_3 &= (\overline{x}_3 \vee x_4 \vee x_5) \\ C_4 &= (x_4 \vee x_5 \vee \overline{x}_6) \\ C_5 &= (x_5 \vee \overline{x}_6 \vee x_7) \\ C_6 &= (\overline{x}_6 \vee x_7 \vee \overline{x}_8) \end{aligned}$$

A boolean formula ϕ_{rcb} in l-k-CNF has then the following form :

$$\phi_{rcb} = C_1 \wedge C_2 \wedge C_3 \wedge C_4 \wedge C_5 \wedge C_6$$

It can be seen that all possible boolean formulas in k-CNF with $(l-k+1)$ clauses, contain as a subset boolean formulas in l-k-CNF, i.e. l-k-CNF $\subset$ k-CNF.

Even though l-k-CNF $\subset$ k-CNF, it is "simple[8]" to satisfy a boolean formula ϕ_{rcb} in l-k-CNF. This can be performed, by setting in the first clause C_1 each literal to true, and then subsequent in each clause $C_2, C_3, \ldots, C_{l-k+1}$ the last literal to true. With this simple construction, it is possible to find a satisfiability in runtime of $\Theta(l)$, where l is the number of variables.

V. Transforming r-contiguous detectors into a k-CNF boolean formula

Recall r-contiguous detectors are bit-strings of length l from U which do not match any bit-strings of length l from S with the r-contiguous matching rule. In this section, we show a transformation of arbitrary bit-strings from S into l-k-CNF boolean formulas.

Let $b \in \{0,1\}$ and $\mathcal{L}(b)$ a mapping defined as :

$$\mathcal{L}(b) \rightarrow \begin{cases} x & \text{if } b = 0 \\ \overline{x} & \text{otherwise} \end{cases}$$

where $x, \overline{x}$ are literals.

Let $k, l \in \mathbb{N}$, where $k \leq l$ and $s \in \{0,1\}^l$, where $s[i]$ denotes the bit at position i of bit-string s, and $\mathcal{C}(s,k)$ a l-k-CNF transform mapping defined as :

$$\begin{aligned} \mathcal{C}(s,k) \rightarrow\ & (\mathcal{L}(s[1]) \vee \mathcal{L}(s[2]) \vee \ldots \vee \mathcal{L}(s[k])) \wedge \\ & (\mathcal{L}(s[2]) \vee \mathcal{L}(s[3]) \vee \ldots \vee \mathcal{L}(s[k+1])) \wedge \\ & \vdots \\ & (\mathcal{L}(s[l-k+1]) \vee \ldots \vee \mathcal{L}(s[l])) \end{aligned}$$

For the sake of clarity we denote a boolean formula in l-k-CNF which is obtained by $\mathcal{C}(s,k)$ for $s \in S$ as ϕ_{rcb}. Moreover we denote a boolean formula $\bigwedge_{i=1}^{|S|} \phi_{rcb}^i$ which is obtained by $\mathcal{C}(s_1,k) \wedge \mathcal{C}(s_2,k) \wedge \ldots \wedge \mathcal{C}(s_{|S|},k)$ for $|S| \geq 1$ and all $s_i \in S,\ i = 1, \ldots, |S|$ as $\widehat{\phi}_{rcb}$. If $|S| = 1$, then $\phi_{rcb} \equiv \widehat{\phi}_{rcb}$.

[8]for one (self) bit-string

Proposition 1: Given an universe U which contains all unique bit-strings of length l, a set $S \subset U$ and the set D which contains all generable r-contiguous detectors, which do not match any bit-string from S. The boolean formula $\widehat{\phi}_{rcb}$ which is obtained by $\mathcal{C}(s,r)$ for all $s \in S$ is satisfiable only with the assignment set D.

Proof: Transforming $s_1 \in S$ with $\mathcal{C}(s_1,k)$ in a l-k-CNF, where $k := r$, results due to $\mathcal{L}(\cdot)$ in a boolean formula which is only satisfiable with bit-strings from $U \setminus F_1$, where the symbol $*$ represents either a 1 or 0 and

$$\begin{aligned} F_1 = \{ & s_1[1,\ldots,r] \underbrace{*\,*\ldots*}_{l-r}, \\ & *\, s_1[2,\ldots,r+1] \underbrace{*\,*\ldots*}_{l-r-1}, \\ & \vdots \\ & \underbrace{*\,*\ldots*}_{l-r}\, s_1[l-r+1,\ldots,l] \} \end{aligned}$$

Transforming the remaining $s_i = s_2, s_3, \ldots, s_{|S|}$ with $\mathcal{C}(s_i,k)$ and constructing $\widehat{\phi}_{rcb} = \phi_{rcb}^1 \wedge \phi_{rcb}^2 \wedge \ldots \wedge \phi_{rcb}^{|S|}$ results in a boolean formula which is only satisfiable with bit-strings from $U \setminus (F_1 \cup F_2 \cup \ldots \cup F_{|S|})$. Each r-contiguous detector from D has *no* matching bits at $s_i[1,\ldots,r], s_i[2,\ldots,r+1], \ldots, s_i[l-r+1,\ldots,l]$ for $i = 1,2,\ldots,|S|$. Hence, $\widehat{\phi}_{rcb}$ is only satisfiable with assignment set $U \setminus (F_1 \cup F_2 \cup \ldots \cup F_{|S|}) = D$. ■

Example 5: Let $l = 5$, $r = 3$ and $S = \{s_1, s_2, s_3, s_4, s_5, s_6\}$ with the following bit-strings :

$$\begin{aligned} s_1 &= \{01011\},\ s_2 = \{01100\},\ s_3 = \{01110\}, \\ s_4 &= \{10010\},\ s_5 = \{10100\},\ s_6 = \{11100\} \end{aligned}$$

Generating all possible r-contiguous detectors of length $l = 5$ and $r = 3$ by given the self set S, one obtains the detector set $D = \{d_1, d_2, d_3, d_4, d_5\}$:

$$\begin{aligned} d_1 &= \{00000\},\ d_2 = \{00001\},\ d_3 = \{11000\}, \\ d_4 &= \{11001\},\ d_5 = \{00111\} \end{aligned}$$

Transforming all $s \in S$ with $\mathcal{C}(s,r)$, one obtains :

$$\begin{aligned}
\phi^1_{rcb} &= (x_1 \vee \overline{x}_2 \vee x_3) \wedge (\overline{x}_2 \vee x_3 \vee \overline{x}_4) \wedge \\
&\quad (x_3 \vee \overline{x}_4 \vee \overline{x}_5) \\
\phi^2_{rcb} &= (x_1 \vee \overline{x}_2 \vee \overline{x}_3) \wedge (\overline{x}_2 \vee \overline{x}_3 \vee x_4) \wedge \\
&\quad (\overline{x}_3 \vee x_4 \vee x_5) \\
\phi^3_{rcb} &= (x_1 \vee \overline{x}_2 \vee \overline{x}_3) \wedge (\overline{x}_2 \vee \overline{x}_3 \vee \overline{x}_4) \wedge \\
&\quad (\overline{x}_3 \vee \overline{x}_4 \vee x_5) \\
\phi^4_{rcb} &= (\overline{x}_1 \vee x_2 \vee x_3) \wedge (x_2 \vee x_3 \vee \overline{x}_4) \wedge \\
&\quad (x_3 \vee \overline{x}_4 \vee x_5) \\
\phi^5_{rcb} &= (\overline{x}_1 \vee x_2 \vee \overline{x}_3) \wedge (x_2 \vee \overline{x}_3 \vee x_4) \wedge \\
&\quad (\overline{x}_3 \vee x_4 \vee x_5) \\
\phi^6_{rcb} &= (\overline{x}_1 \vee \overline{x}_2 \vee \overline{x}_3) \wedge (\overline{x}_2 \vee \overline{x}_3 \vee x_4) \wedge \\
&\quad (\overline{x}_3 \vee x_4 \vee x_5)
\end{aligned}$$

$$\widehat{\phi}_{rcb} = \phi^1_{rcb} \wedge \phi^2_{rcb} \wedge \phi^3_{rcb} \wedge \phi^4_{rcb} \wedge \phi^5_{rcb} \wedge \phi^6_{rcb}$$

The boolean formula $\widehat{\phi}_{rcb}$ is satisfied only with the assignment set $\{00000, 00001, 11000, 11001, 00111\} = \{d_1, d_2, d_3, d_4, d_5\} = D$.

Of course it is possible to perform the reverse transformation when given $\widehat{\phi}_{rcb}$. However, a k-CNF boolean formula which is in a non l-k-CNF can not be transformed with this approach. This means that finding a satisfying set for $\widehat{\phi}_{rcb}$ is not "harder" then finding a satisfying set for a boolean formula in non l-k-CNF. However this *not* implies that finding a satisfying set for $\widehat{\phi}_{rcb}$ is a $\mathcal{NP}$-complete problem.

A. Unsatisfiable CNF Formula and No Generable Detectors

In this section, we use our obtained transformation result (proposition 1) to demonstrate involving properties on the number of generable r-contiguous detectors. An example is the question : Given S and r, is it possible to generate any detectors at all ?. By obtaining with $\mathcal{C}(s,r)$ a boolean formula $\widehat{\phi}_{rcb}$ in CNF, this question can be answered by means of the resolution method [15], [16]. The resolution is a method for demonstrating that a CNF formula is unsatisfiable, i.e. a deduction to the empty clause (symbol $\square$), or in our case that no detectors can be generated. Roughly speaking, it is based on the idea of successively adding resolvents to the formula. Resolvents are clauses which do not modify the (growing) formula.

Specifically, let C_i and C_j be clauses and let x be a literal which occurs in C_i and also occurs in C_j as $\overline{x}$, i.e. $x \in C_i$ and $\overline{x} \in C_j$. The resolvent of C_i and C_j is $C'_i \cup C'_j$, where $C'_i := C_i \setminus \{x\}$ and $C'_j := C_j \setminus \{\overline{x}\}$. For example, $(x_1 \vee x_3)$ is the resolvent of $(x_1 \vee x_2)$ and $(x_1 \vee \overline{x}_2 \vee x_3)$.

Example 6: Let S contain the following bit-strings $\{110, 000, 010, 001\}$ and let $r = 2$. The obtained boolean formula $\widehat{\phi}_{rcb}$ results in

$$\begin{aligned}
\widehat{\phi}_{rcb} &= (\overline{x}_1 \vee \overline{x}_2) \wedge (\overline{x}_2 \vee x_3) \wedge \\
&\quad (x_1 \vee x_2) \wedge (x_2 \vee x_3) \wedge \\
&\quad (x_1 \vee \overline{x}_2) \wedge (\overline{x}_2 \vee x_3) \wedge \\
&\quad (x_1 \vee x_2) \wedge (x_2 \vee \overline{x}_3)
\end{aligned}$$

By applying the resolution method (see Fig. 5), one can see that $\widehat{\phi}_{rcb}$ is reduced to the empty clause $\square$, i.e. $\widehat{\phi}_{rcb}$ is *not* satisfiable and therefore no detectors are generable.

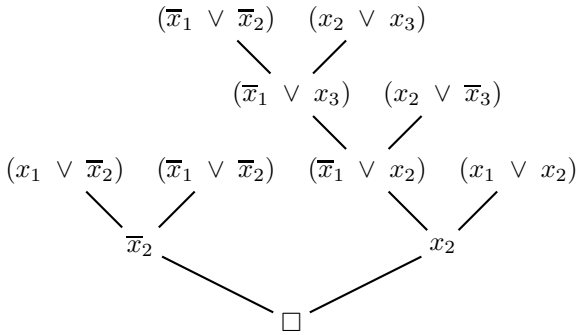


Figure 5. Resolution method results in the empty clause $\square$ and implies that $\widehat{\phi}_{rcb}$ is not satisfiable

However, we would like to emphasize here, that the resolution method for determining if detectors are generable is interesting mainly from the theoretical point of view. As unfortunely, it takes an exponential number of resolution steps until an empty clause is obtained — information on the complexity of the resolution method is provided in [16].

Another approach to answer the question : is it possible to generate any detectors at all ?, is to apply a variant of the Lovász Local Lemma [16]. More specifically we define according to [16], $vbl(C)$ as the set of variables that occur in clause C, i.e. $\{x \in V | x \in C \text{ or } \overline{x} \in C\}$, where V is a set of boolean variables. Moreover, as defined in [16], the *neighborhood* of C in ϕ_{rcb} is the set of clauses distinct from C in ϕ_{rcb} that depend on C, or more formally :

$$\Gamma_{\phi_{rcb}}(C) := \{C' \in \phi_{rcb} \,|\, C' \neq C \text{ and } vbl(C) \cap vbl(C') \neq \emptyset\}$$

Proposition 2: Let S be a set of bit-strings of length l, where all $s \in S$ are consisting of pairwise distinct substrings $s[1, \ldots, r], s[2, \ldots, r+1], \ldots, s[l-r+1, \ldots, l]$. R-contiguous detectors are generable, if

$$|S| < \frac{2^r\, e^{-1} + 1}{2r - 1}$$

Proof: For each $s \in S$ construct a boolean formula ϕ^i_{rcb} in l-k-CNF by $\mathcal{C}(s,r)$. Construct a related k-CNF boolean formula $\widehat{\phi}_{rcb} = \phi^1_{rcb} \wedge \phi^2_{rcb} \wedge \ldots \wedge \phi^{|S|}_{rcb}$. Let C^j_i be the i-th clause in ϕ^j_{rcb}, $1 \leq j \leq |S|$. C^j_i has at most $2\,(r-1)$ many neighborhood clauses

in ϕ^j_{rcb} and at most $(2\,(r-1)+1)\cdot(|S|-1)$ many neighborhood clauses in all remaining boolean formulas $\phi^1_{rcb}, \phi^2_{rcb}, \ldots, \phi^{j-1}_{rcb}, \phi^{j+1}_{rcb}, \ldots, \phi^{|S|}_{rcb}$. In total this results in $|S|\cdot(2r-1)-1$ dependent clauses (see Fig. 6 on last page).

A variant of the Lovász Local Lemma [16] implies that if $|\Gamma_F(C)| \leq 2^{k-2},\ k \in \mathbb{N}$ for all clauses C in a k-CNF formula F, then F is satisfiable.

Applying the variant of the Lovász Local Lemma results in

$$|S| \cdot (2r-1) - 1 \leq 2^{r-2} < 2^r/e$$

■

B. Complexity of l-k-CNF Satisfiability

As previously mentioned, a satisfiability for a boolean formula in l-k CNF can be obtained in $\Theta(l)$. However, in this case a boolean formula for exactly *one* bit-string from S is constructed. If there are $|S| > 1$ distinct bit-strings, then this simple method of finding a satisfiability does not work.

In this paper, we will not propose an additional r-contiguous algorithm and determine the complexity. Rather, we attempt to answer the question, if it is possible to generate r-contiguous detectors with a non-exponential complexity in r.

By transforming the problem to generate r-contiguous detectors into a k-CNF satisfiability problem, we assume that at least $\Omega(2^k)$ evaluations are required for finding a *complete* assignment set, i.e. generating all possible detectors. This assumption is justified thereby, that $\Omega(2^k)$ evaluations are required for finding a complete satisfying set for the first clause of each $s \in S$. Additionally, the remaining $(l-r)$ clauses of each $s \in S$ must be verified, which in total could be done in $\mathcal{O}(|S| \cdot 2^k)$. We would also like to emphasize here that this assumption is not theoretically verified and requires further exploration. However there is a strong evidence that at least $\Omega(2^k)$ evaluations are required for generating all generable detectors, as no efficient algorithms (for $k > 2$) are known which are able to solve the k-CNF satisfiability problem in polynomial time. More specifically, as outlined in section IV, the k-CNF satisfiability problem is a decision problem, where the input is a boolean formula f and the output is "Yes", if f is satisfiable, and "No", otherwise. The currently fastest known deterministic algorithm that decides the 3-CNF problem, runs in time $\mathcal{O}(1.473^n)$ [17], where n is the number of variables. The probabilistic algorithm variant runs in time $\mathcal{O}(1.3302^n)$ [18]. For $k = 4, 5, 6$ the deterministic and probabilistic algorithms runtimes become slightly worse [19].

We would like to emphasize here, that the (deterministic and probabilistic) k-CNF algorithms proposed in [19], [18] decides if a boolean formula is satisfiable, however the algorithms not output *all* satisfiable assignment sets — in our case, all generable detectors.

C. Impacts on Negative Selection Algorithms

The efficient generation of r-contiguous detectors is an important building block in many negative selection approaches, and has been explored in the recent years intensively [20], [21], [3]. However, all proposed r-contiguous detector generating algorithms have a runtime or space complexity which is exponential in r [3] — more specifically it is $\mathcal{O}(2^r)$. Using for instance a matching length of $r = 64$ and $|S| = 2^{(r/2)}$ many self bit-strings, one obtains an algorithm complexity which is infeasible to be computationally practical. Combining our results from this paper, with former complexity results, we believe very strongly that generating r-contiguous detectors in the negative selection can not be performed efficiently and casts doubt on the use of such an approach on its applicability in a large-scale, real-world scenario.

VI. Conclusion

In this paper we have demonstrated the link between generating r-contiguous detectors and the k-CNF satisfiability problem. Specifically, we have shown that the problem of generating r-contiguous detectors, when given self set S and matching length r can be transformed to an instance of the k-CNF satisfiability problem. The assignment set of the boolean formula in k-CNF is directly linked to the generable r-contiguous detector set. This result provides an interesting insight into better understanding the problem of generating r-contiguous detectors. Furthermore, results taken from the field of boolean satisfiability can be utilized to study more formally the problem of generating r-contiguous detectors. In this paper we have demonstrated two utilize statements in the context of unsatisfiability, i.e. no generable detectors. Moreover, we have discussed the question, are r-contiguous detectors efficiently generable. We have conclude that at least $\Omega(2^r)$ evaluations are required to generate all possible detectors. This conclusion was justified with the k-CNF satisfiability problem when considering the first clause of each $s \in S$ only.

Acknowledgment

Thomas Stibor would like to thanks Emo Welzl, for his valuable suggestions and conversations that sparked many of these ideas, especially the link to the variant of the Lovász Local Lemma.

References

[1] E. Hart and J. Timmis, "Application areas of AIS: Past, present and future," in *Proceedings of the 4th International Conference on Artificial Immune Systems (ICARIS)*, ser. Lecture Notes in Computer Science, vol. 3627. Springer-Verlag, 2005, pp. 483–497.

[2] L. N. de Castro and J. Timmis, *Artificial Immune Systems: A New Computational Intelligence Approach*. Springer Verlag, 2002.

[3] M. Ayara, J. Timmis, R. de Lemos, L. N. de Castro, and R. Duncan, "Negative selection: How to generate detectors," in *Proceedings of the 1nd International Conference on Artificial Immune Systems (ICARIS)*. Unversity of Kent at Canterbury Printing Unit, 2002, pp. 89–98.

[4] F. Esponda, E. S. Ackley, S. Forrest, and P. Helman, "On-line negative databases (with experimental results)," *International Journal of Unconventional Computing*, vol. 1, no. 3, pp. 201–220, 2005.

[5] F. Esponda, "Negative representations of information," Ph.D. dissertation, University of New Mexico, 2005.

[6] M. T. Ranang, "An artificial immune system approach to preserving security in computer networks," Master's thesis, Norges Teknisk-Naturvitenskapelige Universitet, 2002.

[7] S. Forrest, A. S. Perelson, L. Allen, and R. Cherukuri, "Self-nonself discrimination in a computer," in *Proceedings of the 1994 IEEE Symposium on Research in Security and Privacy*. IEEE Computer Society Press, 1994.

[8] J. K. Percus, O. E. Percus, and A. S. Perelson, "Predicting the size of the T-cell receptor and antibody combining region from consideration of efficient self-nonself discrimination," *Proceedings of National Academy of Sciences USA*, vol. 90, pp. 1691–1695, 1993.

[9] W. Feller, *An Introduction to Probability Theory and its Applications*, 3rd ed. John Wiley & Sons, 1968, vol. 1.

[10] S. T. Wierzchoń, "Discriminative power of the receptors activated by k-contiguous bits rule," *Journal of Computer Science and Technology*, vol. 1, no. 3, pp. 1–13, 2000.

[11] J. Balthrop, F. Esponda, S. Forrest, and M. Glickman, "Coverage and generalization in an artificial immune system," in *GECCO 2002: Proceedings of the Genetic and Evolutionary Computation Conference*. New York: Morgan Kaufmann Publishers, 9-13 July 2002, pp. 3–10.

[12] T. Stibor, J. Timmis, and C. Eckert, "On the appropriateness of negative selection defined over hamming shape-space as a network intrusion detection system," in *Congress On Evolutionary Computation – CEC 2005*. IEEE Press, 2005, pp. 995–1002.

[13] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd ed. MIT Press, 2002.

[14] K. R. Reischuk, *Einführung in die Komplexitätstheorie*. B.G. Teubner Stuttgart, 1990.

[15] J. A. Robinson, "A machine-oriented logic based on the resolution principle," *Journal of the Association for Computing Machinery (JACM)*, vol. 12, no. 1, pp. 23–41, January 1965.

[16] E. Welzl, "Boolean satisfiability — combinatorics and algorithms," 2005, lecture Notes (http://www.inf.ethz.ch/∼emo/SmallPieces/SAT.ps).

[17] T. Brueggemann and W. Kern, "An improved deterministic local search algorithm for 3-SAT," *Theoretical Computer Science*, vol. 329, no. 1–3, pp. 303–313, 2004.

[18] T. Hofmeister, U. Schöning, R. Schuler, and O. Watanabe, "A probabilistic 3-SAT algorithm further improved," in *19th Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, ser. Lecture Notes in Computer Science, vol. 2285. Springer-Verlag, 2002, pp. 192–202.

[19] U. Schöning, "A probabilistic algorithm for k-SAT and constraint satisfaction problems," in *40th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE Press, 1999, pp. 410–414.

[20] P. D'haeseleer, S. Forrest, and P. Helman, "An immunological approach to change detection: algorithms, analysis, and implications," in *Proceedings of the 1996 IEEE Symposium on Research in Security and Privacy*, IEEE Computer Society. IEEE Computer Society Press, May 1996, pp. 110–119.

[21] S. T. Wierzchoń, "Generating optimal repertoire of antibody strings in an artificial immune system," in *Intelligent Information Systems*. Springer Verlag, 2000, pp. 119–133.

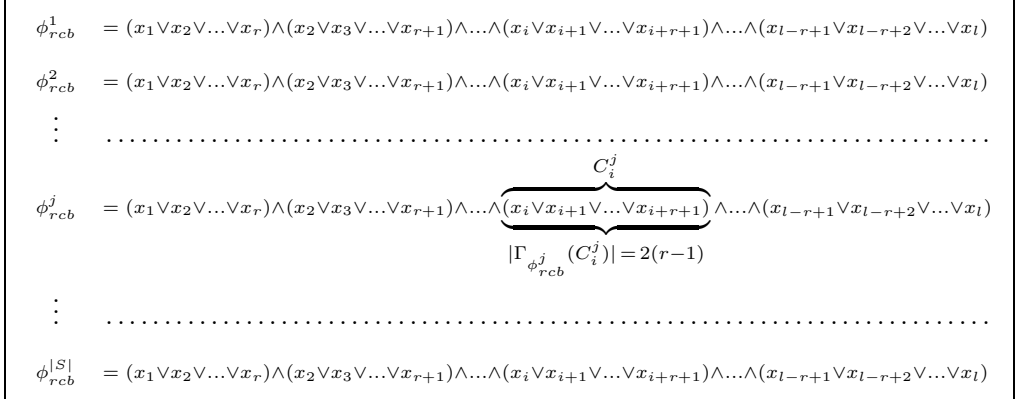


Figure 6. C^j_i has at most $2\,(r-1)$ many neighborhood clauses in ϕ^j_{rcb} ($r-1$ to left and $r-1$ to right) and at most $(2\,(r-1)+1)\cdot(|S|-1)$ many neighborhood clauses in all remaining boolean formulas $\phi^1_{rcb}, \phi^2_{rcb}, \ldots, \phi^{j-1}_{rcb}, \phi^{j+1}_{rcb}, \ldots, \phi^{|S|}_{rcb}$.