On the Code Length of TCAM Coding Schemes

Ori Rottenstreich and Isaac Keslassy
Department of Electrical Engineering
Technion, Haifa 32000, Israel
{or@tx,isaac@ee}.technion.ac.il

***Abstract*—All high-speed Internet devices need to implement *classification*, i.e. they must determine whether incoming packet headers belong to a given subset of a search space. To do it, they encode the subset using ternary arrays in special high-speed devices called TCAMs (ternary content-addressable memories). However, the optimal coding for arbitrary subsets is unknown. In particular, to encode an arbitrary range subset of the space of all W-bit values, previous works have successively reduced the upper-bound on the code length from $2W-2$ to $2W-4$, then $2W-5$, and finally W TCAM entries. In this paper, we prove that this final result is optimal for typical prefix coding and cannot be further improved, i.e. the bound of W is tight. To do so, we introduce new analytical tools based on independent sets and alternating paths.**

I. Introduction

A. Background

High-speed Internet devices distinguish between packets using *classification*, i.e. they apply different actions depending on whether the incoming packet header belongs to a given subset of a search space. For instance, they could accept all packets belonging to this subset, and drop all other packets.

Classification is a key building block in networking theory. It is the core function behind many network applications, such as routing, filtering, intrusion detection, accounting, monitoring, load-balancing, policy enforcement, differentiated services, virtual routers, and virtual private networks [1]–[4].

Today, high-speed packet classification is implemented using hardware-based *TCAMs* (Ternary Content-Addressable Memories) [5], [6]. TCAMs are special devices that encode the searched subset using a sequence of fixed-width ternary arrays composed of 0s, 1s, and $*$s (don't-care). Given an incoming packet header, they return the action associated with the first matched ternary array. The goal of a TCAM coding scheme is to *minimize the code length*, i.e. minimize the number of TCAM ternary arrays needed to encode a given subset.

Throughout this paper, given some $W \geq 1$, we will define the search space as the set of all W-bit values. This search space contains 2^W elements. We will want to encode an arbitrary *range subset*, i.e. an arbitrary contiguous set of values.

The following examples illustrate how TCAMs work.

Example 1: Assume that $W = 3$. We want to code the W-bit range $R = [1,6] \subseteq \left[0, 2^W-1\right]$, so that only packets in R are accepted, while all other packets are denied (default action). Fig. 1(a) shows R in a binary tree. R consists of all values from 001 to 110. It can be coded as the union of $\{001\}$, $\{01*\} = \{010, 011\}$, $\{10*\} = \{100, 101\}$, and $\{110\}$.

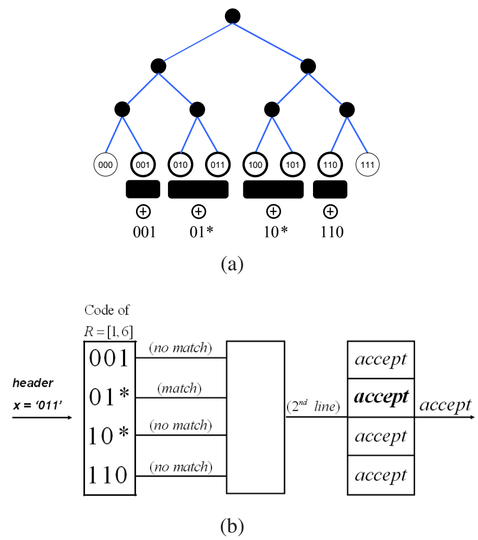

Fig. 1. Internal TCAM coding of $R = [1,6]$

Therefore, it has a code length of 4.
Fig. 1(b) illustrates how this code is implemented in a TCAM. This code for R is presented on the left side and consists of 4 consecutive entries. Then, on the right side, for each incoming header, the TCAM picks the first matched entry and outputs the associated action.
For instance, consider an incoming header $x = 011$. The header x belongs to the range R. Indeed, it matches the second TCAM entry $\{01*\}$, as shown in parentheses. Since the first (and only) matched TCAM entry is this second entry, the output is the action displayed in the second line, and the packet is accepted. Incidentally, note that if no TCAM entry is matched, for instance for header 000, then the TCAM outputs a default action, which is assumed to be *deny*.

The code in the example above is a simple code. However, it is not optimal, because it does not consider the full potential of TCAM coding, and in particular the *order of the entries*. The following example shows how we can further reduce the TCAM code length.

Example 2: Fig. 2(a) illustrates again the range $R = [1,6]$ from Example 1. It now shows how R can be coded with a code length of 3 instead of 4. The new code uses *external* coding, which exploits a different entry order by coding the complementary of R first. It starts by coding this complementary using the two entries $\{000\}$ and $\{111\}$, and then associates all remaining entries to R using $\{***\}$. Note that

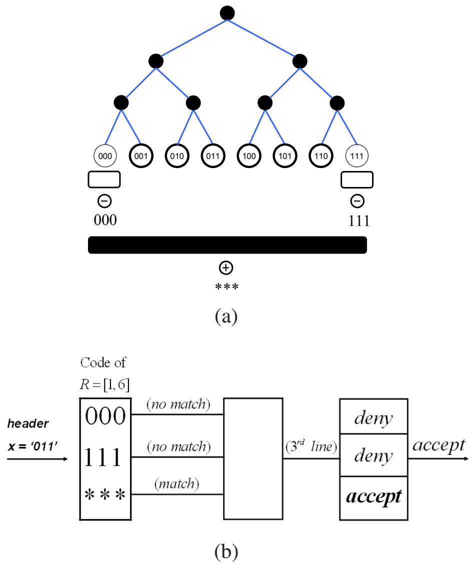

Fig. 2. External TCAM coding of $R = [1, 6]$

the code length is equal to $W = 3$; in this paper, we want to prove that encoding a range indeed needs a code length of at least W in the worst case.

Fig. 2(b) shows how the TCAM implements this code. When header $x = 011$ arrives, the third entry is the only one to match it, and the output of the third line is indeed the same as above, i.e. *accept*. Incidentally, note that if header 111 arrived, it would match both the second and the third entries; then, the second entry would be selected, because it appears first, and the output would be *deny* as expected.

B. Related Work

There have been several past results on the upper-bounds for the range code length. First, each range defined over W-bit values can be coded in at most $2W-2$ TCAM entries for $W \geq 2$ using an *internal* coding as in Example 1, i.e. a coding that only uses entries from within the range [7].

This upper-bound on the code length can be reduced to $2W-4$ (for sufficiently large W) using internal TCAM coding that is non-prefix, i.e. with an arbitrary position of the "$*$" symbols, and a connection to Boolean Disjunctive Normal Form (DNF) [8]. It can also be reduced to $2W-4$ using Gray codes instead of binary codes with prefix internal coding [9]. This result has since been improved to $2W-5$ using a more complex coding [10].

Recently, *external* coding that exploits a different entry order was used to improve the bound on the worst-case code length to W [11], [12].

Lower bounds on coding length have more rarely been considered. If coding is constrained to be *internal*, the worst-case code length is known to be at least W [9]. Also, an independent set of minterms in sum-of-products expressions is presented in [13]. However, none of these consider *external* coding, and therefore they do not fully exploit TCAM properties.

Last, in the general case, TCAMs can be used to code multidimensional rules of many range subsets with many actions [14]. We restrict this paper to the analysis of the single-dimensional case with a single range and a binary action.

II. Model and Notations

A. Terminology

We first formally define the terminology used in this paper. For simplicity we do not distinguish between a W-bit binary string (in $\{0,1\}^W$) and its value (in $[0, 2^W-1]$).

Definition 1 (Header): A packet *header* $x = (x_1, \ldots, x_d) \in \{0,1\}^W$ is defined as a W-bit string.

Definition 2 (Range Rule): The *range rule* R is defined as an integer range $[r_1, r_2]$, where r_1 and r_2 are W-bit integers and $r_1 \leq r_2$. A packet header $x \in \{0,1\}^W$ is said to match R whenever $x \in [r_1, r_2]$.

For instance, in Example 1, $x = 001$ matches $R = [1, 6]$.

Definition 3 (TCAM Entry): A *TCAM entry* $S \rightarrow a$ is composed of a TCAM line $S = (s_1, \ldots, s_W) \in \{0, 1, *\}^W$, where $0, 1$ are bit values and $*$ stands for *don't-care*, and a binary action $a \in \{0, 1\}$, with default action $a_d = 0$. A W-bit string $b = (b_1, \ldots, b_W)$ matches S, denoted as $b \in S$, if for all $i \in [1, W]$, $s_i \in \{b_i, *\}$.

Definition 4 (TCAM Coding Scheme): Consider the indicator function $\alpha : \{0,1\}^W \rightarrow \{0,1\}$ of a searched subset. Then a *TCAM coding scheme* ϕ maps α to an ordered set of $n_\phi(\alpha)$ TCAM entries $\left(S_1 \rightarrow a_1, \ldots, S_n \rightarrow a_{n_\phi(\alpha)}\right)$ if and only if for any header $x \in \{0,1\}^W$, $\alpha(x)$ is the action associated with the first TCAM entry that matches x (and $\alpha(x) = a_d = 0$ if no TCAM entry matches x).

The number $n_\phi(\alpha)$ of TCAM entries is the *code length* of coding scheme ϕ for function α. Let Φ denote the set of all TCAM coding schemes.

Each *range* R is uniquely characterized by its range indicator function α_R, which takes a value of 1 on R and 0 outside R. We will use *range* to indicate either R or its indicator function α_R.

For instance, in Example 1, the range $R = [1, 6]$ is defined using $\alpha_R(1) = \ldots = \alpha_R(6) = 1$ (i.e., *accept*) and $\alpha_R(0) = \alpha_R(7) = 0$ (i.e., *deny*). The code length is $n_\phi(\alpha_R) = 4$, and the TCAM entries are $(001 \rightarrow 1, 01* \rightarrow 1, 10* \rightarrow 1, 110 \rightarrow 1)$.

We now define a TCAM *prefix* coding scheme, in which the 0's and 1's are restricted to appear as prefixes in the TCAM entries, and the *'s as suffixes.

Definition 5 (Prefix Coding Scheme): A TCAM *prefix coding scheme* ϕ is a TCAM coding scheme such that for any TCAM entry $S \rightarrow a$ with $S = (s_1, \ldots, s_W) \in \{0, 1, *\}^W$, if $s_j = *$ for some $j \in [1, W]$, then $s_{j'} = *$ for any $j' \in [j, W]$. Let Φ_p denote the set of all prefix coding schemes, so that $\Phi_p \subset \Phi$.

B. Optimal Range Coding Problem

We want to find a TCAM prefix coding scheme $\phi \in \Phi_p$ that minimizes the worst-case TCAM prefix code length $n_\phi(\alpha_R)$ over all possible range functions α_R. We first focus on prefix coding schemes, and later consider non-prefix schemes. To do so, we will first define extremal ranges, then define the TCAM code-length minimization problem over all extremal ranges, before defining the TCAM code-length minimization problem over all possible ranges.

Definition 6 (Extremal Ranges): We define two types of *extremal ranges* over $[0, 2^W - 1]$.
(i) A *left-extremal* range R^{LE} denotes a range of the form $R^{LE} = [0, y]$ for some arbitrary value of y.
(ii) Likewise, a *right-extremal* range R^{RE} denotes a range of the form $R^{RE} = [y, 2^W - 1]$ for some arbitrary value of y.

A *non-extremal* range $R = [y_1, y_2]$ is a range such that $0 < y_1 \leq y_2 < 2^W - 1$. Therefore, a range is either left-extremal, right-extremal, or non-extremal. Let $ER(W)$ be the set of all extremal ranges.

Definition 7 (Extremal Range Code Length): Define the *extremal range code length* $g(W)$ as the best-achievable code length given all coding schemes $\phi \in \Phi$ for extremal ranges. Then,

$$g(W) = \min_{\phi} \max_{R \in ER(W)} n_\phi(\alpha_R). \quad (1)$$

Likewise, define $g_p(W)$ over all prefix coding schemes $\phi \in \Phi_p$.

Definition 8 (Range Code Length): For any positive integer W and any TCAM coding scheme $\phi \in \Phi$, the *range code length of* ϕ, denoted $f_\phi(W)$, is the worst-case TCAM expansion $n_\phi(\alpha_R)$ over all possible range functions α_R, i.e.

$$f_\phi(W) = \max_{R \subseteq [0, 2^W - 1]} n_\phi(\alpha_R), \quad (2)$$

The *range code length* $f(W)$ is defined as the best-achievable range code length for W-bit ranges given all coding schemes, i.e.

$$f(W) = \min_{\phi \in \Phi} \left(\max_{R \subseteq [0, 2^W - 1]} n_\phi(\alpha_R) \right) \quad (3)$$

Likewise, we define $f_p(W)$ as the best-achievable range code length given all *prefix* coding schemes $\phi \in \Phi_p$.

C. Past Results

The following results are known from [11].

Property 1: For all $W \geq 1$, the extremal range code length satisfies the following upper-bound:

$$g(W) \leq g_p(W) \leq \left\lceil \frac{W+1}{2} \right\rceil \quad (4)$$

Property 2: For all $W \geq 1$, the worst-case range code length satisfies the following upper-bound:

$$f(W) \leq f_p(W) \leq W. \quad (5)$$

III. Main Results

Our main results consist in proving the code-length optimality of specific coding schemes. In particular, we first prove their optimality over the set of all extremal ranges.

Theorem 1: For all $W \in \mathbb{N}^*$, the extremal range code length satisfies

$$g(W) \geq \left\lceil \frac{W+1}{2} \right\rceil. \quad (6)$$

It follows from Property 1 that the bound is tight, i.e. $g(W) = \left\lceil \frac{W+1}{2} \right\rceil$.

Then, we also prove the optimality of known prefix coding schemes over the set of all ranges. Note that this optimality is restricted to prefix coding schemes, which are by far the most common among all coding schemes.

Theorem 2: For all $W \in \mathbb{N}^*$, for prefix coding schemes, the range code length satisfies

$$f_p(W) \geq W. \quad (7)$$

It follows from Property 2 that the bound is tight, i.e. $f_p(W) = W$.

To provide these two results, we introduce novel analytical tools that are suited for TCAM analysis. We first define the hull property, and use it to define an independent set of n points. We further demonstrate that *an independent set of n points cannot be encoded in less than n TCAM entries*. This property is true given any arbitrary TCAM entries, in any order, and with any corresponding actions. To our knowledge, it is the first characterization of TCAM coding properties using independent sets.

IV. Hull, Independence, and Alternating Paths

We now want to introduce new general analytical tools that will help us analyze the minimum number of TCAM entries needed to code a classifier function. Intuitively, given any range that we need to encode, we will want to exhibit n points that are *independent* in some sense, and prove that they cannot be encoded in less than n TCAM entries.

First, we define the *hull* of a set of W-bit strings in the W-dimensional string space (this hull is also known as the *isothetic rectangle hull*, *minimum bounding rectangle*, or *minimum axis-aligned bounding box* in different contexts).

Definition 9 (Hull): Let $(n, W) \in \mathbb{N}^{*2}$, and consider n strings $a^1, \ldots, a^n$ of W bits each, with $a^i = (a^i_1, \ldots, a^i_W)$ for each $i \in [1, n]$. Then the hull of $\{a^1, \ldots, a^n\}$, denoted $H(a^1, \ldots, a^n)$, is the smallest cuboid containing $a^1, \ldots, a^n$ in the W-dimensional string space, and is defined as

$$H(a^1, \ldots, a^n) = \{x = (x_1, \ldots, x_W) \in \{0,1\}^W | \forall j \in [1, W], x_j \in \{a^1_j, \ldots, a^n_j\}\} \quad (8)$$

We can now relate the hull of a set of points to the TCAM entries that they jointly match.

Proposition 3: Let $(n, W) \in \mathbb{N}^{*2}$, and consider n strings $a^1, \ldots, a^n$ of W bits each. Then $a^1, \ldots, a^n$ match the same TCAM entry iff all the strings in the hull $H(a^1, \ldots, a^n)$ match this TCAM entry.

Proof: First, by Equation (8) defining the hull, we always have $\{a^1, \ldots, a^n\} \subseteq H(a^1, \ldots, a^n)$. Therefore, if all strings in $H(a^1, \ldots, a^n)$ match a TCAM entry, so does any a^i.

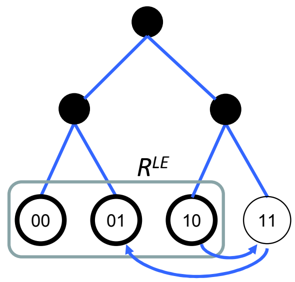

Fig. 3. Alternating path: R^{LE} requires at least two TCAM entries using any coding scheme.

On the other hand, assume that $a^1, \ldots, a^n$ match a TCAM entry $S \to a$, with $S = (s_1, \ldots, s_W) \in \{0, 1, *\}^W$. Then by Definition 3 of TCAM entry matching, for all $i \in [1, n]$ and for all $j \in [1, W]$, $s_j \in \{a^i_j, *\}$. Now consider $x = (x_1, \ldots, x_W) \in H(a^1, \ldots, a^n)$. Then by Equation (8), for all $j \in [1, W]$, $x_j \in \{a^1_j, \ldots, a^n_j\}$. Therefore, for each bit j, either all a^i_j are equal, and x_j obviously matches s_j like all a^i_j, or some of them are distinct, and then $s_j = *$, so x_j matches s_j again. ∎

Using the definition of the hull, we now define *independent sets* of points, and then show that an independent set of n points cannot be coded in less than n TCAM entries. Therefore, this result enables us to simply exhibit an appropriate independent set of points whenever we want to prove a lower bound on the code length of a classifier function.

Definition 10 (Alternating Path and Independent Set): Let n and W be positive integers, and let $\alpha : \{0,1\}^W \to \{0, 1\}$ be a classifier function. Then an *alternating path* A_n of size n is defined as an ordered set of $2n - 1$ W-bit strings $A_n = (a^1, \ldots, a^{2n-1})$ that satisfies the following two conditions:

(i) Alternation: For $i \in [1, 2n - 1]$,

$$\begin{aligned} &\alpha(a^1) = \alpha(a^3) = \cdots = \alpha(a^{2n-1}) = 1, \text{ and} \\ &\alpha(a^2) = \alpha(a^4) = \cdots = \alpha(a^{2n-2}) = 0. \end{aligned} \tag{9}$$

(ii) Hull: For any i_1, i_2, i_3 such that $1 \le i_1 < i_2 < i_3 \le 2n-1$,

$$a^{i_2} \in H(a^{i_1}, a^{i_3}). \tag{10}$$

In such an alternating path, $(a^1, a^3, a^5, \ldots, a^{2n-1})$ is an *independent set* of size n.

Example 3: As shown in Fig. 3, let $W = 2$, $n = 2$, and consider the left-extremal range $R^{LE} = [0, 2] = \{\{00\}, \{01\}, \{10\}\}$. Let $a^1 = 2 = \{10\}$, $a^2 = 3 = \{11\}$, and $a^3 = 1 = \{01\}$. Then $A_2 = (a^1, a^2, a^3)$ is an *alternating path of size* 2 and (a^1, a^3) is an independent set, because they satisfy the two needed conditions:

(i) Alternation: $a^1 \in R^{LE}$, $a^2 \notin R^{LE}$, $a^3 \in R^{LE}$.

(ii) Hull: $a^2 \in H(a^1, a^3)$, i.e. $\{11\} \in H(\{10\}, \{01\})$, because it shares its first bit with a^1 and its second bit with a^3.

Lemma 4: Let n be a positive integer, and $(a^1, \ldots, a^{2n+1})$ be an alternating path of size $n + 1$. Then removing any two successive elements in the alternating path yields an alternating path of size n.

Proof: Removing elements a^i and a^{i+1} yields $(a^1, \ldots, a^{i-1}, a^{i+2}, \ldots, a^{2n+1})$ for any $i \in [1, 2n]$. Then the two conditions defined above for the alternating path still hold. First, odd elements should still yield action 1, and even elements action 0. Second, for any three elements in the list, the middle element is still in the hull of the two others, since it was already before the removal of the two elements. ∎

Proposition 5: A classifier function with an alternating path of size n cannot be coded in less than n TCAM entries.

Proof: The proof is *by induction* on n.

Induction basis: For $n = 1$, we need to code at least one element with a non-default action of 1, therefore we need at least one TCAM entry.

Induction step: We assume that we cannot code a classifier function with an alternating path of size n in less than n TCAM entries, and want to show it for $n + 1$ as well.

Assume by contradiction that we can code a classifier function with an alternating path $A_{n+1} = (a^1, \ldots, a^{2n+1})$ of size $n + 1$ in less than $n + 1$ TCAM entries. Then consider the first TCAM entry $S \to a$ (as defined in Definition 4), and distinguish several cases.

(i) If none of the elements of A_{n+1} are in this first TCAM entry, which we denote $A_{n+1} \cap S = \emptyset$, then S does not impact A_{n+1}, and we can actually code the elements of A_{n+1} in the next (at most) $n - 1$ TCAM entries. But by Lemma 4, we can extract from A_{n+1} an alternating path of size n, e.g. $(a^1, \ldots, a^{2n-1})$, and by induction we know that it cannot be coded in $n - 1$ TCAM entries.

(ii) If a single element a^i out of A_{n+1} is in this first TCAM entry, i.e. $A_{n+1} \cap S = \{a^i\}$, then by Lemma 4, we can remove two successive elements from A_{n+1}, including a^i, and obtain an alternating path A_n of size n that does not contain a^i. But then we need to code A_n in the next $n - 1$ TCAM entries, because $A_n \cap S = \emptyset$, and by induction we know that it is impossible.

(iii) If at least two elements out of A_{n+1} are in this first TCAM entry, i.e. $|A_{n+1} \cap S| > 1$, then they all must yield the same action by definition of the TCAM entry. Without loss of generality, assume that $\{a^{i_1}, a^{i_2}\} \subseteq A_{n+1} \cap S$, with $i_1 < i_2$. Then since they yield the same action, we have $i_1 < i_1 + 1 < i_2$, and therefore $a^{i_1+1} \in H(a^{i_1}, a^{i_2})$ (Definition 10). Therefore, by Proposition 3, a^{i_1+1} also matches the same TCAM entry, even though it should yield a different action than a^{i_1} and a^{i_2}. Contradiction again. ∎

V. Range Code-Length Optimality

A. Extremal Range Code-Length Optimality

Thanks to the tools developed above, we can now prove the following theorem, which shows that the upper-bound $g(W) \le \left\lceil \frac{W+1}{2} \right\rceil$ is tight, and therefore that their iterative coding scheme reaches the optimal extremal-range code length. We are now ready to prove Theorem 1.

Proof: We have to show that $g(W) \ge \left\lceil \frac{W+1}{2} \right\rceil$.

The case of $W = 1$ is trivial. To distinguish between the two left-extremal ranges $R^{LE}_1 = [0, 0]$ and $R^{LE}_2 = [0, 1]$, it is clear that we need at least one TCAM entry.

Assume $W \geq 2$. First, notice that for each even value of $W \in \mathbb{N}^*$, the upper-bound is the same for $g(W)$ and $g(W+1)$, and is equal to $\left(\frac{W}{2}+1\right)$, i.e. $\left\lceil \frac{W+1}{2} \right\rceil = \left\lceil \frac{(W+1)+1}{2} \right\rceil = \left(\frac{W}{2}+1\right)$. Therefore, to prove the tightness of the upper-bound, it is sufficient to do it for the positive even values of W.

More specifically, for each positive even value of W, we simply need to exhibit a left-extremal range $R^{LE}(W) \subseteq [0, 2^W - 1]$ that cannot be coded in less than $\left(\frac{W}{2}+1\right)$ TCAM entries. As a consequence, this left-extremal range $R^{LE}(W)$ would also suffice to prove the tightness of the upper-bound for $W+1$, because $R^{LE}(W) \subseteq [0, 2^W-1] \subseteq [0, 2^{W+1}-1]$, and $\left\lceil \frac{(W+1)+1}{2} \right\rceil = \frac{W}{2}+1$.

Therefore, we assume that $W \geq 2$ is even. Define W-bit string $c = 1010...10 = \{10\}^{\frac{W}{2}}$. The binary value of c is

$$c = \sum_{k=0}^{\frac{W}{2}-1} 2 \cdot 2^{2k} = \frac{2}{3}\left(2^W - 1\right) \qquad (11)$$

Consider the left-extremal range $R^{LE}(W) = \left[0, \frac{2}{3}\left(2^W-1\right)\right] = \left\{\{0\}^W, \ldots, c\right\} \subseteq [0, 2^W - 1]$. Then by Proposition 5, it suffices to show that in $R^{LE}(W)$ there exists an alternating path of size $\frac{W}{2}+1$. Note that we showed this already for $W = 2$ in Example 3, and will now generalize the proof for any even $W \geq 2$.

We define $a^1 = \{01\}^{\frac{W}{2}}$, and then construct the alternating path $(a^1, \ldots, a^{W+1})$ by flipping each time the i^{th} bit of a^i to obtain a^{i+1}: by flipping the first bit of a^1, we get $a^2 = \{11\}\{01\}^{\frac{W}{2}-1}$. Then by flipping the second bit of a^2, we get $a^3 = \{10\}\{01\}^{\frac{W}{2}-1}$, and likewise until $a^{W+1} = \{10\}^{\frac{W}{2}} = c$. Therefore, for $i \in [1, W+1]$, a^i has the same first $i-1$ bits as a^{W+1} and the same last $W-(i-1)$ bits as a^1. As a consequence, by the hull definition (Definition 9), for any i_1, i_2, i_3 such that $1 \leq i_1 < i_2 < i_3 \leq W+1$, $a^{i_2} \in H(a^{i_1}, a^{i_3})$, because a^{i_2} shares its first $i_2 - 1$ bits with a^{i_3}, and its other bits with a^{i_1}.

Now we only need to prove the alternation property of $(a^1, \ldots, a^{W+1})$. As defined in the alternating path definition (Definition 10), we only need to show that the odd-indexed elements are in $R^{LE}(W) = [0, a^{W+1}]$ while the even-indexed are not, i.e. $a^i \leq a^{W+1}$ for $i = 1, 3, \ldots, W-1$, while $a^i > a^{W+1}$ for $i = 2, 4, \ldots, W$.

To compare between the two W-bit binary strings a^i and a^{W+1}, we use the lexicographic order, i.e. $a^i < a^{W+1}$ iff there exists some most significant different bit j such that their first $j-1$ bits are equal, and the j^{th} bit of a^i is 0 while the j^{th} bit of a^{W+1} is 1. In addition, we know that a^i only shares the first $i-1$ bits with a^{W+1}, and all other bits are different. Therefore, for $i \in [1, W]$, the most significant different bit between a^i and a^{W+1} is the i^{th} bit. Since the i^{th} bit of $a^{W+1} = c = \{10\}^{\frac{W}{2}}$ is 1 for odd i and 0 for even i, the result follows. ■

B. Range Code-Length Optimality

We will now prove that the upper bound on the range code length $f_p(W)$ from [11] is actually tight among all TCAM prefix coding schemes, and therefore their prefix coding scheme is optimal among all prefix coding schemes for the worst-case range code length.

Let's now provide an outline of the proof of Theorem 2. Please refer to [14] for the full proof.

Proof Outline: We first assume that W is odd, and define $R_1 = \left[\frac{1}{3}\left(2^{W-1}-1\right), 2^{W-1}-1\right]$, $R_2 = \left[2^{W-1}, 2^{W-1} + \frac{2}{3}\left(2^{W-3}-1\right)\right]$, and $R = R_1 \bigcup R_2 = \left[\frac{1}{3}\left(2^{W-1}-1\right), 2^{W-1} + \frac{1}{3}\left(2^{W-2}\right) - \frac{2}{3}\right]$. We show by iteration on odd W's that R cannot be coded in less than W TCAM prefix entries, whether the first TCAM entry codes the range R (internal coding) or its complementary (external coding). Then, we do the same for even W's, and conclude. ■

VI. Conclusion

The paper deals with the fundamental capacity region of TCAMs. This paper is unique in that it introduces fundamental analytical tools based on independent sets and alternating paths which can be used to prove the optimality of previous coding schemes. In particular, we demonstrate that a previous upper bound on the code length of extremal ranges is tight over all coding schemes. We also prove the optimality of a previous coding scheme over all prefix coding schemes for general ranges.

While we proved the optimality over all *prefix* coding schemes, the range-coding optimality over all *general* non-prefix coding schemes is left as an open question for future research.

Acknowledgment

This work was partly supported by the European Research Council Starting Grant no. 210389. We would also like to thank David Hay, Danny Hendler and Tuvi Etzion for their helpful participation and suggestions.

References

[1] D. E. Taylor, “Survey and taxonomy of packet classification techniques,” *ACM Comput. Surv.*, vol. 37, no. 3, pp. 238-275, 2005.
[2] G. Varghese, *Network Algorithmics*. Morgan Kaufmann, 2005.
[3] J. Chao and B. Liu, *High Performance Switches and Routers*. Wiley, 2007.
[4] J. Naous, D. Erickson, A. Covington, G. Appenzeller, and N. McKeown, “Implementing an OpenFlow switch on the NetFPGA platform,” in *ACM ANCS*, 2008.
[5] NetLogic Microsystems, http://www.netlogicmicro.com/.
[6] Renesas, http://www.renesas.com/.
[7] V. Srinivasan, G. Varghese, S. Suri, and M. Waldvogel, “Fast and scalable layer four switching,” 1998, pp. 191–202.
[8] B. Schieber, D. Geist, and A. Zaks, “Computing the minimum DNF representation of Boolean functions defined by intervals,” *Discrete Applied Mathematics*, vol. 149, no. 1-3, pp. 154-173, 2005.
[9] A. Bremler-Barr and D. Hendler, “Space-efficient TCAM-based classification using Gray coding,” in *IEEE Infocom*, 2007, pp. 1388–1396.
[10] R. Roth, Personal communication.
[11] O. Rottenstreich and I. Keslassy, “Worst-case TCAM rule expansion,” in *IEEE INFOCOM Mini-Conference*, 2010.
[12] R. Cohen and D. Raz, “Simple efficient TCAM based range classification,” in *IEEE INFOCOM Mini-Conference*, 2010.
[13] T. Sasao, “On the complexity of classification functions,” in *ISMVL*, 2008, pp. 57–63.
[14] O. Rottenstreich and I. Keslassy, “Worst-case TCAM rule expansion,” Comnet, Technion, Israel, Technical Report TR09-01, 2009. [Online]. Available: http://webee.technion.ac.il/~isaac/papers.html